AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(8): 22067–22090.
DOI:10.3934/math.20241073
Received: 24 November 2023
Revised: 09 April 2024
Accepted: 23 April 2024
Published: 15 July 2024

Research article

Modified nonmonotonic projection Barzilai-Borwein gradient method for nonnegative matrix factorization

Xiaoping Xu[1], Jinxuan Liu[1], Wenbo Li[1,*], Yuhan Xu[2] and Fuxiao Li[1]

[1] School of Sciences, Xi'an University of Technology, Xi'an 710054, China

[2] School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China

* **Correspondence:** Email: wbli57@hotmail.com.

Abstract: In this paper, an active set recognition technique is suggested, and then a modified nonmonotonic line search rule is presented to enhance the efficiency of the nonmonotonic line search rule, in which we introduce a new parameter formula to attempt to control the nonmonotonic degree of the line search, and thus improve the chance of discovering the global minimum. By using a modified linear search and an active set recognition technique, a global convergence gradient solution for nonnegative matrix factorization (NMF) based on an alternating nonnegative least squares framework is proposed. We used a Barzilai-Borwein step size and greater step-size tactics to speed up the convergence. Finally, a large number of numerical experiments were carried out on synthetic and image datasets, and the results showed that our presented method was effective in calculating the speed and solution quality.

Keywords: active set; projected Barzilai-Borwein method; nonmonotonic line search; alternating nonnegative least squares; greater step size
Mathematics Subject Classification: 15A23, 65F30

1. Introduction

Nonnegative matrix factorization (NMF) [13, 25, 26, 31, 32] is a representative method of linear dimensionality reduction for nonnegative data, and has been widely used as a significant tool. Generally speaking, the basic NMF issue can be expressed as: given an $m \times n$ dimensional nonnegative matrix $V = (V_{ij})$ with $V_{ij} \geq 0$, and a pre-assigned positive integer $r < \min(m, n)$, NMF wants to find two nonnegative matrices $W \in \mathbb{R}_+^{m\times r}$ and $H \in \mathbb{R}_+^{r\times n}$ such that

$$V \approx WH. \tag{1.1}$$

A common way to solve NMF (1.1) is

$$\begin{aligned} &\min_{W,H} f(W,H) \equiv \frac{1}{2}\|V - WH\|_F^2 \\ &\text{s.t.} \quad W \geq 0, H \geq 0 \end{aligned} \tag{1.2}$$

where $\|\cdot\|_F$ is the Frobenius norm.

The projected Barzilai-Borwein gradient (PBB) method is a prevailing and forceful means for solving (1.2). This method was proposed by Barzilai and Borwein [3]. Recently, the literature [8, 33, 34, 42, 43] has shown that the PBB method is very effective in solving the optimization issue [17–22]. Because of its simplicity and numercal efficiency, the PBB approach has attracted much attention in different fields. To date, the PBB approach has been triumphantly applied to NMF (see [16, 23, 24, 27, 28]). On account of W and H being completely symmetric, this thesis primarily ponders updating matrix W using the PBB method. Let H^k denote the approximate value of H after their kth update, and let

$$f(W, H^k) = \frac{1}{2}\|V - WH^k\|_F^2 \quad \forall k. \tag{1.3}$$

At each step for solving (1.3), there are three different updates:

$$W^{k+1} = \min_{W \geq 0} f(W, H^k); \tag{1.4a}$$

$$W^{k+1} = \min_{W \geq 0} f(W, H^k) + \langle \nabla f(W), W - W^k \rangle + \frac{L_W^k}{2}\|W - W^k\|_F^2; \tag{1.4b}$$

$$W^{k+1} = \min_{W \geq 0} \langle \nabla f(W^k), W - W^k \rangle + \frac{L_W^k}{2}\|W - W^k\|_F^2, \tag{1.4c}$$

where $L_W^k > 0$, $\nabla f(W) = \nabla_W f(W, H^k)$.

The original cost function (1.4a) is the most frequently used form in the PBB method for NMF and has been widely and deeply researched [4,16,27,28,46]. But the major disadvantage of (1.4a) is that it is not strongly convex, and we can merely hope that the algorithms seek out a stationary point instead of the global or even local minimizer. To overcome this drawback, a proximal modification of the cost function (1.4a) is presented in [23, 24], namely, the proximal cost function (1.4b).

At present, the proximal cost function (1.4b) has been used with the PBB method for NMF in [23, 24]. As the cost function (1.4b) is a strongly convex quadratic programming with a lower bound of zero, the subproblem (1.4b) has a unique minimizer. In [23], the authors proposed a quadratic regularization nonmonotone PBB method to solve (1.4b), and established a global convergence result under mild conditions. Recently, this method was revisited in [24], indicating that the monotone PBB method converges globally to the stationary point of (1.3), and the numerical experiments indicated that the monotone PBB method was better than the nonmonotone one under some circumstances. Nevertheless, most of the existing PBB gradient methods for (1.4a) and (1.4b) tend to converge slowly on account of the nonnegative constraints. This prompted us to exploit new and faster NMF algorithms.

In this article, first, we enhance the active set recognition technique in [6] so that it can reasonably identify the active constraints for NMF. Then, we present a modified nonmonotonic line search technique for the sake of enhancing the efficiency of the nonmonotonic line search. By using the active set identification technology and the improved nonmonotonic line search, a globally convergent gradient-based method to solve (1.4c) on the basis of the alternating nonnegative least squares framework is proposed. To accelerate the algorithm, we use the Barzilai-Borwein step size and the greater step-size tactics. Ultimately, numerical experiments are carried out on synthetic and image datasets to verify the effectiveness of the proposed method.

This article is organized as shown below. In Section 2, we put forward an effective NMF algorithm and present its global convergence. The experimental results are shown in Section 4. Ultimately, Section 5 summarizes the work.

2. An active set nonmonotone PBB algorithm

2.1. Main algorithm

In this section, we put forward an efficient algorithm for solving the NMF (1.3) and establish the global convergence of our suggested algorithm. To set up the primary consequence of this section, let us first present some known properties about the objective function $f(W, H^k)$.

Lemma 1. [15] *The two statements given below are effective.*

(i) The objective function $f(W, H^k)$ of (1.3) is convex.

(ii) The gradient

$$\nabla_W f(W, H^k) = (WH^k - V)(H^k)^T$$

is Lipschitz continuous with the constant $L_W = \|H^k(H^k)^T\|_2$.

For the sake of argument, we will be centered on (1.4c) and rewrite it as

$$\min_{W\geq 0} \varphi(U, W) := \langle \nabla f(U), W - U\rangle + \frac{L_W}{2}\|W - U\|_F^2, \tag{2.1}$$

where the fixed matrix $U \geq 0$.

Distinctly, it can be seen from (ii) of Lemma 1 that $\varphi(U, W)$ is strictly convex in W for each fixed U. In each iteration, first, we will solve the following strongly convex quadratic minimization problem to compute point Z_t:

$$\min_{W\geq 0} \varphi(W_t, W). \tag{2.2}$$

Since the objective function of problem (2.2) has a strong convex property, this issue has a unique closed-form solution:

$$Z_t = P[W_t - \frac{1}{L_W}\nabla_W f(W_t, H^k)], \tag{2.3}$$

where the operator $P[X]$ projects all of the negative entries of X to zero.

Let $W_{t+1} = Z_t + D_t$, and here D_t stands for the direction. We discover that the convergence of $\{W_{t+1}\}$ can not be ensured. Therefore, researchers have come up with a tactic for globalization based on the modified Armiji line search [45], namely, we will seek out a step size λ_t that makes

$$f(Z_t + \lambda_t D_t) \leq \max_{0\leq j\leq min\{t,M-1\}} f(Z_{t-j}) + \gamma\lambda_t\langle\nabla f(Z_t), D_t\rangle, \tag{2.4}$$

where $M > 0$. Because of the maximum function, in any iteration, it is possible to discard the generated good function values, meanwhile, numerical performances depend heavily on the choice of M under some circumstances (see [7]).

To overcome these shortcomings in each procedure, we present a modified nonmonotonic line search rule. Our line search is represented like this: for a given iterative point Z_t and a search direction D_t at Z_t, we select $\eta_t \in [\eta_{min}, \eta_{max}]$, here $0 < \eta_{min} < \eta_{max} < 1$, $\gamma_t \in [\gamma_{min}, \gamma_{max}(1 - \eta_{max})]$, where $0 < \gamma_{max} < 1$, and $0 < \gamma_{min} < (1 - \eta_{max})\gamma_{max}$, to get a λ_t satisfying the inequality shown below:

$$S_{t+1} \leq S_t - \frac{\gamma_t \lambda_t}{\alpha_t} \|D_t\|^2, \tag{2.5}$$

where S_t is defined as

$$S_t = \begin{cases} f(W_0), & \text{if} \quad t = 0, \\ f(W_t) + \eta_{t-1}(S_{t-1} - f(W_t)), & \text{if} \quad t \geq 1. \end{cases} \tag{2.6}$$

As similar with M in (2.4), the selection of η_t in (2.6) plays a key role in controlling the degree of nonmonotonicity (see [14]). So, for the sake of enhancing the efficiency of the nonmonotonic line search, Ahookhosh et al. [1] selected a varying value for the parameter η_t by using a simple formula. Later, Nosratipour et al. [30] thought that η_t should be related to an appropriate criterion for measuring the distance to the optimal solution. Hence, they defined η_t by

$$\eta_t = 1 - e^{-\|\nabla f(Z_t)\|}. \tag{2.7}$$

However, we found that if the sequence of iteration $\{Z_t\}$ is trapped in a confined crooked valley, then that can lead to $\nabla f(Z_t) = 0$, from which we can get $\eta_t = 0$, so the nonmonotonic line search is decreased to the normative Armijo line search, which is inefficient by producing very short or tortuous steps. For the sake of overcoming this shortcoming, we suggest the following η_t:

$$\eta_t = \frac{2}{\pi} arctan(|f(Z_t) - f(Z_{t-1})|). \tag{2.8}$$

It is obvious that $|f(Z_t) - f(Z_{t-1})|$ is large when the function value decreases rapidly, and then η_t will also be large, hence the nonmonotonic tactics will be stronger. However, as $f(Z_t)$ is close to the optimal solution, we can obtain $|f(Z_t) - f(Z_{t-1})|$ tending to zero, and after that η_t also tends to zero, hence, the nonmonotonic rule will weaken and tend to the monotonic rule.

Finally, let

$$W_{t+1} = Z_t + \lambda_t D_t, \tag{2.9}$$

where λ_t is the step size you get by employing nonmonotonic line search (2.5).

As everyone knows from [12] that the larger step size technology can significantly accelerate the convergence rate of the algorithm, by adding a relaxation factor s to the renewal rule of W_{t+1} (2.9), we modify the update rule (2.9) as

$$W_{t+1} = Z_t + s\lambda_t D_t \tag{2.10}$$

for relaxation factor $s > 1$. We show that the optimal parameter s in (2.10) is $s = 1.7$ by number experiments in Section 4.4.

As was observed in [6,43], the active set method can enhance the efficiency of the local convergence algorithm and reduce the computing cost. Therefore, we will recommend an active set recognition technology to approximate the right sustain of the solution points. In our context, the active set is considered as the subset of zero components of Z^*. Now, similar to the idea proposed in [6], we define the active set $\bar{A}$ as the index set corresponding to the zero component, meanwhile, the inactive set $\bar{F}$ will be the support of Z^*.

Definition 1. *Let $\Omega = \{ij : Z_{ij} \geq 0$ and $Z \in \mathbb{R}^{m\times r}\}$ and Z^* be a stationary point of (1.3). We define the active set as follows:*

$$\bar{A} = \{ij : Z^*_{ij} = 0\}. \tag{2.11}$$

We further define an inactive set $\bar{F}$ which is a complementary set of $\bar{A}$,

$$\bar{F} = I\backslash\bar{A}, \tag{2.12}$$

where $I = \{11, 12, \cdots, 1r, 21, 22, \cdots, 2r, \cdots, m1, m2, \cdots, mr\}$.

Then, for any $Z \in \Omega$, we give approximations as shown below for $A(Z_t)$ and $F(Z_t)$ as $\bar{A}$ and $\bar{F}$, respectively,

$$A(Z_t) = \{ij : (Z_t)_{ij} \leq \alpha_t \nabla f(Z_t)_{ij}\}, \tag{2.13}$$
$$F(Z_t) = I\backslash A(Z_t), \tag{2.14}$$

where α_t is the BB step size. Similar to Proposition 3.1 in [6], we have that if the strict complementarity is satisfied at Z_t, then $A(Z_t)$ overlaps with the active set if Z_t is close enough to Z^*, namely $A(Z_t) = \bar{A}, F(Z_t) = \bar{A}$.

In order to get a good estimate of the active set, the active set is further subdivided into two sets

$$A_1(Z_t) = \{ij \in A(Z_t) : (D_t)_{ij} \geq c\}, \tag{2.15}$$

and

$$A_2(Z_t) = \{ij \in A(Z_t) : (D_t)_{ij} < c\}, \tag{2.16}$$

where $c > 0$ is a very small constant. It is clear that $A_2(Z_t)$ is a variable index set that approximately meets the first-order necessary conditions. Thus, it is reasonable for us to use the scaled projected gradient as the descent direction in the corresponding subspace. Furthermore, realizing that $A_1(Z_t)$ is a variable index set that goes against the first-order necessary conditions, to define a reasonable search direction, we further subdivided $A_1(Z_t)$ into two subsets

$$\bar{A}_1(Z_t) = \{ij : ij \in A_1(Z_t) \text{ and } (Z_t)_{ij} = 0\}, \tag{2.17}$$

and

$$\tilde{A}_1(Z_t) = \{ij : ij \in A_1(Z_t) \text{ and } (Z_t)_{ij} \neq 0\}. \tag{2.18}$$

We consider the direction of the form 0 for variables with indexes in $\bar{A}_1(Z_t)$ and $-Z_t$ for variables with indexes $\tilde{A}_1(Z_t)$ to increase the corresponding components. Thus, we define the previously stated direction as a compact form as shown below:

$$(D_t)_{ij} = \begin{cases} 0, & \text{if } ij \in \bar{A}_1(Z_t), \\ -(Z_t)_{ij}, & \text{if } ij \in \tilde{A}_1(Z_t), \\ (P[Z_t - \alpha_t \nabla f(Z_t)] - Z_t)_{ij}, & \text{if } ij \in A_2(Z_t) \cup F(Z_t), \end{cases} \tag{2.19}$$

where α_t is the BB step size.

Based on the above discussion, we present an active set strategy-based nonmonotonic projection Barzilai-Borwein gradient method and outline the proposed algorithm in Algorithm 1. We can follow a similar procedure for updating H.

Algorithm 1 Active set nonmonotone projected Barzilai-Borwein algorithm (ANMPBB).

1. Initialize $\alpha_0 = 1$, $\eta_t \in (0, 1)$, choose parameters $\eta_t \in [\eta_{min}, \eta_{max}]$, $\gamma_t \in [\gamma_{min}, \gamma_{max}(1 - \eta_{max})]$, $\alpha_{max} > \alpha_{min} > 0, c > 0, \rho \in (0, 1)$, $s > 1$, $L_W = \|H^k(H^k)^T\|_2$ and $W_0 = W^k$. Set $t = 0$.
2. If $\|P[W_t - \nabla f(W_t)] - W_t\| = 0$, stop.
3. Compute $Z_t = P[W_t - \frac{1}{L_W} \nabla f(W_t, H^k)]$.
4. Compute S_t by (2.6) and compute D_t by (2.19).
5. Perform the nonmonotonic line search. Provide an integer m_t that is a minimum nonnegative and satisfies

$$S_{t+1} \leq S_t - \frac{\gamma_t \rho^m}{\alpha_t} \|D_t\|^2, \tag{2.20}$$

   where $D_t = P[Z_t - \alpha_t \nabla f(Z_t)] - Z_t$. Set $\lambda_t = \rho^{m_t}$, calculate $W_{t+1} = Z_t + s\lambda_t D_t$.
6. Calculate $X_t = W_{t+1} - Z_t$ and $Y_t = \nabla f(W_{t+1}) - \nabla f(Z_t)$. If $\langle X_t, X_t \rangle / \langle X_t, Y_t \rangle \leq 0$, set $\alpha_{t+1} = \alpha_{max}$; otherwise, set $\alpha_{t+1} = \min\{\alpha_{max}, \max\{\alpha_{min}, \langle X_t, X_t \rangle / \langle X_t, Y_t \rangle\}\}$.
7. Press $t = t + 1$ to proceed to step 2.

To keep things simple, define the direction of the scaling projection gradient as

$$D_\alpha(W) = P[W - \alpha \nabla f(W)] - W \tag{2.21}$$

for each $\alpha > 0$ and $W \geq 0$. The next Lemma 2 is very important in our proof.

Lemma 2. [2] *For each $\alpha \in (0, \alpha_{max}]$, $W \geq 0$,*

(i) $\langle \nabla f(W), D_\alpha(W) \rangle \leq -\frac{1}{\alpha} \|D_\alpha(W)\|^2 \leq -\frac{1}{\alpha_{max}} \|D_\alpha(W)\|^2$,
(ii) The stationary point of (1.3) is at W if and only if $D_\alpha(W) = 0$.

The lemma that follows states that $D_t = 0$ is true if and only if the stationary point of problem (1.3) is the iteration point $\{Z_t\}$.

Lemma 3. *Let D_t be calculated by (2.19), then $D_t = 0$ if and only if Z_t is a stationary point of problem (1.3).*

Proof. Let $(D_t)_{ij} = 0$. Clearly, $(Z_t)_{ij}$ is a stationary point of problem (1.3) when $ij \in \bar{A}_1(Z_t)$. If $ij \in \tilde{A}_1(Z_t)$, we have

$$0 = (D_t)_{ij} = -(Z_t)_{ij} \geq -\alpha_t(\nabla f(Z_t))_{ij}.$$

The above inequality implies that $(\nabla f(Z_t))_{ij} \geq 0$. By the Karush-Kuhn-Tucker (KKT) condition, we can get that $(Z_t)_{ij}$ is a stationary point of problem (1.3). In the case of $(D_t)_{ij} = 0, ij \in A_2(Z_t) \cup F(Z_t)$, by (ii) of Lemma 2, we know that $(Z_t)_{ij}$ is a stationary point of problem (1.3).

Suppose that Z_t is a stationary point of (1.3). From the KKT condition, (2.13), and (2.14), we have

$$\bar{A} = \{ij : (Z_t)_{ij} = 0\},\ \bar{F} = \{ij : (Z_t)_{ij} > 0\}.$$

By the definition of $(D_t)_{ij}$, we have $(D_t)_{ij} = 0$ for all $ij \in A_1(Z_t)$. Then from (ii) of Lemma 2, we have $(D_t)_{ij} = 0$ for all $ij \in A_2(Z_t)$. Therefore, we have $(D_t)_{ij} = 0$ for all $ij \in \bar{A}(Z_t)$. For another case, since $\nabla f(Z_t)_{ij} = 0$, for $ij \in \bar{F}_t$, and $\{Z_t\}_{ij}$ is a feasible point, from the definition of $(D_t)_{ij}$, we have $(D_t)_{ij} = 0, \forall ij \in \bar{F}_t$. □

The lemma shown below states that when Z_t is not a stationary point of problem (1.3), D_t is the descent direction of f at Z_t.

Lemma 4. *Given sequence $\{Z_t\}$ produced by Algorithm 1, we have*

$$\langle \nabla f(Z_t), D_t(Z_t)\rangle \leq -\frac{1}{\alpha_t}\|D_t(Z_t)\|^2. \tag{2.22}$$

Proof. By (2.19), we know

$$D_{ij} = \begin{cases} 0, & \text{if } ij \in \bar{A}_1(Z_t), \\ -(Z_t)_{ij}, & \text{if } ij \in \tilde{A}_1(Z_t), \\ (P[Z_t - \alpha_t \nabla f(Z_t)] - Z_t)_{ij}, & \text{if } ij \in A_2(Z_t) \cup F(Z_t). \end{cases}$$

If $ij \in \bar{A}_1(Z_t)$, it is obvious that

$$\langle \nabla f(Z_t)_{ij}, (D_t(Z_t))_{ij}\rangle \leq -\frac{1}{\alpha_t}\|(D_t(Z_t))_{ij}\|^2 \tag{2.23}$$

holds.

If $ij \in A_2(Z_t) \cup F(Z_t)$, from (i) of Lemma 2, we have

$$\langle \nabla f(Z_t)_{ij}, (D_t(Z_t))_{ij}\rangle \leq -\frac{1}{\alpha_t}\|(D_t(Z_t))_{ij}\|^2. \tag{2.24}$$

Thus, we now only need to prove that

$$\langle \nabla f(Z_t)_{ij}, (D_t(Z_t))_{ij}\rangle \leq -\frac{1}{\alpha_t}\|(D_t(Z_t))_{ij}\|^2, \quad \forall ij \in \tilde{A}_1(Z_t). \tag{2.25}$$

If $(D_t(Z_t))_{ij} = 0$, the inequality (2.25) holds. If $(D_t(Z_t))_{ij} \neq 0$, for all $ij \in \tilde{A}_1(Z_t)$, from (2.17), we have

$$(D_t(Z_t))_{ij} = -(Z_t)_{ij} \quad \text{and} \quad (Z_t)_{ij} \leq \alpha_t \nabla f(Z_t)_{ij},$$

which leads to

$$\langle \nabla f(Z_t)_{ij}, (D_t(Z_t))_{ij} \rangle \leq -\frac{1}{\alpha_t} \|(D_t(Z_t))_{ij}\|^2, \quad \forall ij \in \tilde{A}_1(Z_t). \tag{2.26}$$

The above deduction implies that the inequality (2.22) holds for $ij \in \bar{A}_1(Z_t)$. Combining (2.23), (2.24), and (2.26), we obtain that (2.22) holds. □

The lemma shown below is borrowed from Lemma 3 [23].

Lemma 5. [23] *Suppose Algorithm 1 generates* $\{Z_t\}$ *and* $\{W_t\}$*, then there is*

$$f(Z_t) \leq f(W_t) - \frac{L_W}{2} \|Z_t - W_t\|^2. \tag{2.27}$$

Now, we are going to show the nice property of our line search.

Lemma 6. *Suppose Algorithm 1 generates sequences* $\{Z_t\}$ *and* $\{W_t\}$*, then there is*

$$f(W_t) \leq S_t. \tag{2.28}$$

Proof. Based on the definition of S_t and (2.20), we have

$$\begin{aligned} S_t - S_{t-1} &= f(W_t) + \eta_{t-1}(S_{t-1} - f(W_t)) - S_{t-1} \\ &= (1 - \eta_{t-1})(f(W_t) - S_{t-1}) \leq 0. \end{aligned} \tag{2.29}$$

From $1 - \eta_{t-1} > 0$, it concludes that $f(W_t) - S_{t-1} \leq 0$, i.e., $f(W_t) \leq S_{t-1}$.

Therefore, if $\eta_{t-1} \neq 0$, from (2.6), we have

$$\begin{aligned} S_t - f(W_t) &= f(W_t) + \eta_{t-1}(S_{t-1} - f(W_t)) - f(W_t) \\ &= \eta_{t-1}(S_{t-1} - f(W_t)) \\ &\geq 0 \end{aligned} \tag{2.30}$$

where the last inequality follows from (2.29). Thus, (2.30) indicates

$$f(W_t) \leq S_t. \tag{2.31}$$

In addition, if $\eta_{t-1} = 0$, we have $f(W_t) = S_t$. □

It follows from Lemma 6 that

$$f(W_t) \leq S_t \leq S_0 = f(W_0).$$

In addition, for any initial iterate $W_0 \geq 0$, Algorithm 1 generates sequences $\{Z_t\}$ and $\{W_t\}$ that are both included in the level set.

$$\mathcal{L}(W_0) = \{W | f(W) \leq f(W_0), W \geq 0\}.$$

Again, from Lemma 6, the theorem shown below can be easily obtained.

Theorem 1. *Assume that the level set $\mathcal{L}(W_0)$ is bounded, so the sequence $\{S_t\}$ is convergent.*

Proof. See Corollary 2.2 in [1]. □

Next, we will exhibit that the line search (2.5) is well-defined.

Theorem 2. *Assume Algorithm 1 generates sequences $\{Z_t\}$ and $\{W_t\}$, so step 5 of the Algorithm 1 is well-defined.*

Proof. For this purpose, we prove that the line search stops at a limited value of steps. To establish a contradiction, we suppose that λ_t such as in (2.20) does not exist, then for all adequately large positive integers m, according to Lemmas 5 and 6, we have

$$f(Z_t + s\rho^m D_t) > f(Z_t) - \frac{\gamma_t \rho^m}{\alpha_t(1-\eta_t)}\|D_t\|^2.$$

From Lemma 4, we have,

$$f(Z_t + s\rho^m D_t) - f(Z_t) > \frac{1}{(1-\eta_t)}\gamma_t \rho^m \langle \nabla f(Z_t), D_t\rangle.$$

According to the mean-theorem, there is a $\theta_t \in (0, 1)$ such that

$$s\rho^m \langle \nabla f(Z_t + \theta_t s\rho^m D_t), D_t\rangle > \frac{1}{(1-\eta_t)}\gamma_t \rho^m \langle \nabla f(Z_t), D_t\rangle,$$

namely,

$$\langle \nabla f(Z_t + \theta_t \rho^m D_t) - \nabla f(Z_t), D_t\rangle > (\frac{\gamma_t}{s(1-\eta_t)} - 1)\langle \nabla f(Z_t), D_t\rangle.$$

When $m \to \infty$, we get that

$$(\frac{\gamma_t}{s(1-\eta_t)} - 1)\langle \nabla f(Z_t), D_t\rangle \le 0.$$

Since $0 < \frac{\gamma_t}{1-\eta_t} < 1 < s$, $\langle \nabla f(Z_t), D_t\rangle \ge 0$ is correct. This is not consistent with the fact that $\langle \nabla f(Z_t), D_t\rangle \le 0$. Therefore, step 5 of Algorithm 1 is well-defined. □

2.2. Convergence analysis

In this part, we prove the global convergence of ANMPBB. The following result implies that there exists a minimum step size λ_t that must be satisfied, and this lower bound is indispensable to ensure the global convergence of the suggested algorithm.

Lemma 7. *Suppose that Algorithm 1 generates a step size λ_t, if the stationary point of (1.3) is not W_{t+1}, such that there is a constant $\tilde{\lambda}$ that will cause $\lambda_t \ge \tilde{\lambda}$.*

Proof. For the resulting step size λ_t, if λ_t does not satisfy (2.20), namely,

$$f(Z_t + s\lambda_t D_t) \;>\; S_t - \frac{1}{\alpha_t(1-\eta_t)}\gamma_t \lambda_t \|D_t\|^2$$

$$
\begin{aligned}
&\geq S_t + \frac{1}{1-\eta_t}\gamma_t\lambda_t\langle \nabla f(Z_t), D_t\rangle \\
&\geq f(Z_t) + \frac{1}{1-\eta_t}\gamma_t\lambda_t\langle \nabla f(Z_t), D_t\rangle,
\end{aligned}
$$

where Lemma 4 leads to the second inequality, and similarly, Lemmas 5 and 6 lead to the final inequality. Thus,

$$
f(Z_t + s\lambda_t D_t) - f(Z_t) \geq \frac{1}{1-\eta_t}\gamma_t\lambda_t\langle \nabla f(Z_t), D_t\rangle. \tag{2.32}
$$

By the mean-value theorem, we can find an $\theta \in (0, 1)$ that makes

$$
\begin{aligned}
f(Z_t + s\lambda_t D_t) - f(Z_t) &= s\lambda_t\langle \nabla f(Z_t + \theta s\lambda_t D_t), D_t\rangle \\
&= s\lambda_t\langle \nabla f(Z_t), D_t\rangle + s\lambda_t\langle \nabla f(Z_t + \theta_t s\lambda_t D_t) \\
&\quad - \nabla f(Z_t), D_t\rangle \\
&\leq s\lambda_t\langle \nabla f(Z_t), D_t\rangle + s^2 L_W \lambda_t^2 \|D_t\|^2.
\end{aligned} \tag{2.33}
$$

Substitute the last inequality we obtained from (2.33) into (2.32), and we get

$$
\lambda_t \geq \frac{s(1-\eta_t) - \gamma_t}{L_W s^2 \alpha_{max}(1-\eta_t)}. \tag{2.34}
$$

From $\eta_{t-1} \in [\eta_{min}, \eta_{max}]$ and $\gamma_t \in [\gamma_{min}, \gamma_{max}(1-\eta_{max})]$, we have

$$
\lambda_t \geq \frac{s(1-\eta_{max}) - \gamma_{max}}{L_W s^2 \alpha_{max}(1-\eta_{min})} := \tilde{\lambda}. \tag{2.35}
$$

So, there is going to be a $\tilde{\lambda}$ that makes $\lambda_t \geq \tilde{\lambda}$. □

Lemma 8. *Assume that Algorithm 1 generates the sequence $\{W_t\}$, for the given level set $\mathcal{L}(W_0)$, if it is considered bounded, so there is*

(i)

$$
\lim_{t\to\infty} S_t = \lim_{t\to\infty} f(W_t). \tag{2.36}
$$

(ii) there is a positive constant δ that makes

$$
S_t - f(W_{t+1}) \geq \delta \|D_{t+1}\|^2. \tag{2.37}
$$

Proof. (i) By the definition of S_{t+1}, for $t \geq 1$, we have

$$
S_{t+1} - S_t = (1-\eta_t)(f(W_{t+1}) - S_t).
$$

Since $\eta_{max} \in [0, 1]$, and $\eta_t \in [\eta_{min}, \eta_{max}]$ for all t,

$$
1 - \eta_{min} \geq 1 - \eta_t \geq 1 - \eta_{max} > 0.
$$

According to Theorem 1, as $t \to \infty$,

$$\lim_{t\to\infty} \frac{1}{1-\eta_{max}}(S_{t+1} - S_t) = \lim_{t\to\infty} \frac{1}{1-\eta_{min}}(S_{t+1} - S_t) = 0. \tag{2.38}$$

which implies that

$$\lim_{t\to\infty}(f(W_{t+1}) - S_t) = 0. \tag{2.39}$$

(ii) From (2.5), we have

$$\begin{aligned} S_t - f(W_{t+1}) &\geq \frac{1}{\alpha_t(1-\eta_t)}\gamma_t\lambda_t\|D_t\|^2 \\ &\geq \frac{\gamma_{min}\tilde{\lambda}}{(1-\eta_{min})\alpha_{max}}\|D_t\|^2 \\ &= \delta\|D_t\|^2, \end{aligned} \tag{2.40}$$

where $\delta = \frac{\gamma_{min}\tilde{\lambda}}{(1-\eta_{min})\alpha_{max}}$. □

The global convergence of Algorithm 1 is proved by the theorem shown below.

Theorem 3. *Suppose that Algorithm 1 generates sequences* $\{Z_t\}$ *and* $\{W_t\}$*, so we get*

$$\lim_{t\to\infty}\|D_t\| = 0. \tag{2.41}$$

Proof. According to (ii) of Lemma 8, we have

$$S_t - f(W_{t+1}) \geq \delta\|D_t\|^2 \geq 0, \ \forall t \in \mathbf{N}.$$

Based on (i) of Lemma 8, as $t \to \infty$, we can obtain

$$\lim_{t\to\infty}\|D_t\| = 0.$$ □

According to Theorem 3, Lemma 3, and (2.10), we will exhibit the main convergence results we get as follows.

Theorem 4. *For a given level set* $\mathcal{L}(W_0)$*, assume that it is bounded, hence Algorithm 1 computes the generated sequence* $\{W_t\}$*, and any accumulation point obtained is a stationary point of (1.3).*

2.3. Complexity analysis

It is obvious that, at each iteration, the major cost of ANMPBB is to check the condition (2.20) and calculate the gradient. Therefore, the time complexity of it is $O(mnr)$+ #sub-iterations × $O(tmr^2 + tnr^2)$ in one iteration, where t is the number of trials of the nonmonotone line search procedure.

3. Numerical experiments

In the following content, by using synthetic datasets and real-world datasets (ORL image dataset and Yale image dataset *), we exhibit main numerical experiments to compare the performance of ANMPBB with that of five other efficient methods including the NeNMF [15], the projected BB method (APBB2 [16]), QRPBB [23], hierarchical alternating least squares (HALS) [5], and block coordinate descent (BCD) method [44]. All of the reported numerical results are performed using MATLAB v8.1 (R2013a) on a Lenovo laptop.

3.1. Stopping criterion

According to the Karush-Kuhn-Tucker (KKT) conditions optimized by the existing constraints, we know that (W^k, H^k) is a stationary point of NMF (1.2) if and only if $\nabla_W^P f(W,H) = 0$, and $\nabla_H^P f(W,H) = 0$ are simultaneously satisfied. Here

$$[\nabla_W^P f(W,H)]_{ij} = \begin{cases} [\nabla_W f(W,H)]_{ij}, & \text{if } W_{ij} > 0, \\ \min\{0, [\nabla_W f(W,H)]_{ij}\}, & \text{if } W_{ij} = 0, \end{cases}$$

and $\nabla_H^P f(W^{(k)}, H^{(k)})$ is also written as shown above. Hence, we employ the stopping criteria shown below, which is also used in [29] in numerical experiments:

$$\|[\nabla_W^P f(W^{(k)}, H^{(k)}), \nabla_H^P f(W^{(k)}, H^{(k)})^T]\| \tag{3.1}$$

$$\leq \epsilon \|[\nabla_W^P f(W^{(1)}, H^{(1)}), \nabla_H^P f(W^{(1)}, H^{(1)})^T]\|, \tag{3.2}$$

where $\epsilon > 0$ is a tolerance. When employing the stop criterion (3.1), we need to pay attention to the scale degrees of freedom of the NMF solution, as discussed in [11].

3.2. Synthetic data

In this section, first, the ANMPBB method and the other three ANLS-based methods are tested on synthetic datasets. Since the matrix V in this test happens to be a low-rank matrix, it will be rewritten as $V = LR$, and we generate the L and R by using the MATLAB commands $max(0, randn(m,r))$ and $max(0, randn(r,n))$, respectively.

For ANMPBB, in a later experiment, we adopt the parameters shown below:

$$\alpha_{max} = 10^{20}, \alpha_{min} = 10^{-20}, \rho = 0.25, \gamma = 10^{-8}, c = 10^{-3}.$$

The settings are identical with those of APBB2 and QRPBB. Take $s = 1.7$ for ANMPBB, and the reason for selecting the relaxation factor $s = 1.7$ is given in Section 4.4. Take $tol = 10^{-8}$ for all comparison algorithms. In addition, for ANMPBB, update η_t by the formula (2.8). We unify the maximum number of iterations of all algorithms to 50,000. All of the other parameters of APBB2, NeNMF, and QRPBB are unified as default values.

*Both ORL and Yale image datasets in MATLAB format are available at http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

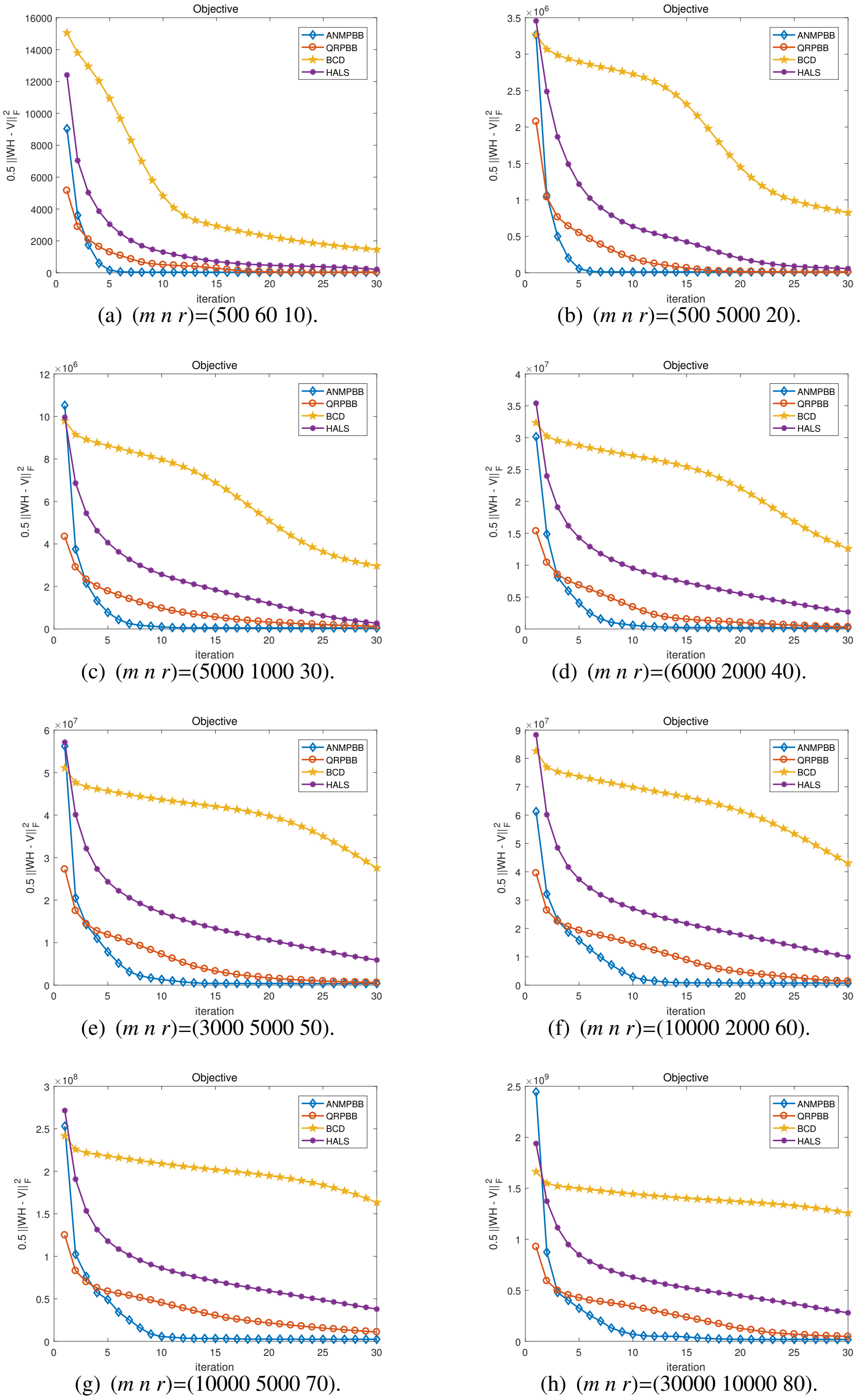
Table 1. Experimental results on the synthetic datasets.

(m n r)	Alg	Iter	Niter	Pgn	Time	Residual
(200 100 10)	NeNMF	153.3	6073.7	3.44E-5	0.25	0.4596
	APBB2	171.9	2442.8	2.76E-5	0.26	0.4596
	QRPBB	158.0	1476.4	2.66E-5	0.19	0.4596
	ANMPBB	57.2	567.0	3.05E-5	0.10	0.4596
(100 500 20)	NeNMF	1946.7	83561.7	1.62E-4	14.46	0.4257
	APBB2	2798.7	48444.2	1.31E-4	15.77	0.4257
	QRPBB	2365.7	26052.7	1.32E-4	8.49	0.4258
	ANMPBB	526.0	5794.0	1.34E-4	1.94	0.4257
(400 200 20)	NeNMF	370.7	13579.9	1.80E-4	2.28	0.4481
	APBB2	320.5	5743.2	1.40E-4	1.83	0.4481
	QRPBB	355.3	4059.7	1.55E-4	1.52	0.4481
	ANMPBB	130.3	1573.2	1.55E-4	0.55	0.4481
(700 700 30)	NeNMF	183.4	6638.0	1.04E-3	3.45	0.4588
	APBB2	161.5	3438.7	8.83E-4	4.56	0.4588
	QRPBB	153.0	2191.9	9.11E-4	2.78	0.4588
	ANMPBB	60.8	953.9	8.44E-4	1.10	0.4588
(1000 500 30)	NeNMF	221.0	7685.5	1.05E-3	4.22	0.4578
	APBB2	180.4	3513.8	8.62E-4	4.52	0.4578
	QRPBB	162.8	2195.5	9.41E-4	2.63	0.4578
	ANMPBB	62.7	995.1	8.87E-4	1.19	0.4578
(1000 600 40)	NeNMF	644.5	25379.5	1.68E-3	20.00	0.4518
	APBB2	723.3	12948.1	1.41E-3	26.16	0.4518
	QRPBB	536.5	7686.2	1.31E-3	12.55	0.4518
	ANMPBB	141.4	2419.7	1.30E-3	3.93	0.4518
(1000 2000 50)	NeNMF	330.8	12081.3	4.98E-3	25.35	0.4574
	APBB2	240.3	4783.6	4.29E-3	23.41	0.4574
	QRPBB	252.8	4264.2	3.84E-3	18.29	0.4574
	ANMPBB	76.5	1511.0	4.63E-3	6.63	0.4574
(2000 2000 50)	NeNMF	172.3	6796.9	8.25E-3	18.96	0.4629
	APBB2	147.6	3734.1	7.30E-3	24.92	0.4629
	QRPBB	149.0	2524.7	5.83E-3	16.43	0.4629
	ANMPBB	56.1	1044.5	6.62E-3	6.90	0.4629
(3000 1000 60)	NeNMF	485.7	17642.4	8.79E-3	63.10	0.4555
	APBB2	396.3	7386.3	7.29E-3	64.50	0.4555
	QRPBB	380.3	6049.4	6.77E-3	48.12	0.4555
	ANMPBB	156.3	2929.9	7.78E-3	24.19	0.4555

For all of the problems we are considering, we casually generated 10 diverse starting value, and the average outcomes obtained from using these starting points are presented in Table 1. The item *iter* represents the number of iterations required to satisfy the termination condition (3.1) is met. The item *niter* represents the total number of sub-iterations for solving W and H. $\|V - W^k H^k\|_F / \|V\|_F$ is relative error, $\|[\nabla_H^P f(W^k, H^k), \nabla_W^P f(W^k, H^k)]\|_F$ is the final value of the projected gradient norm, and CPU time (in seconds) measures performance.

Table 1 clearly indicates that all methods met the condition of convergence within a reasonable number of iterations. Table 1 also clearly indicates that our ANMPBB needs the least execution time and the least number of sub-iterations among all methods, particularly in the case of large-scale problems.

The ANMPBB method is closely related to the QRPBB method, and we all know that the hierarchical ALS (HALS) algorithm for NMF is the most effective upon most occasions, which uses the coordinate descent method to solve subproblems in NMF. We further examine algorithms of ANMPBB, QRPBB, HALS, and BCD. We show how these four methods compare on eight randomly generated independent Gaussian noises measured when the signal-to-noise ratio is 30dB in Figures 1–3. All of the methods are terminated when the stopping criterion said by the inequality in (3.1) satisfies $\epsilon = 10^{-8}$ or the maximum number of iterations is more than 30. Figure 1 shows the value of the objective function compared to the number of iterations. From Figure 1, for most of the test problems, we will draw a conclusion that ANMPBB decreases the objective function much quicker than the other three methods in 30 iterations. This may be because our ANMPBB exploits an efficient modified nonmonotone line search, uses a well active set prediction strategy of solution, and adds a relaxing factor s to the update rules of W_{t+1} and H_{t+1}. Hence our ANMPBB significantly outperformed the other three methods. Figure 2 shows the relationship between the relative residual errors and the number of iterations. Figure 3 exhibits the relative residual errors versus CPU time. The results shown in Figures 2 and 3 are consistent with those shown in Figure 1.



(a) $(m\ n\ r)$=(500 60 10).

(b) $(m\ n\ r)$=(500 5000 20).

(c) $(m\ n\ r)$=(5000 1000 30).

(d) $(m\ n\ r)$=(6000 2000 40).

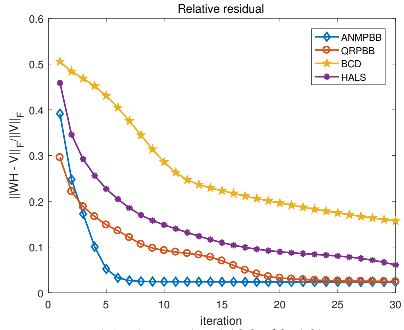
(e) $(m\ n\ r)$=(3000 5000 50).

(f) $(m\ n\ r)$=(10000 2000 60).

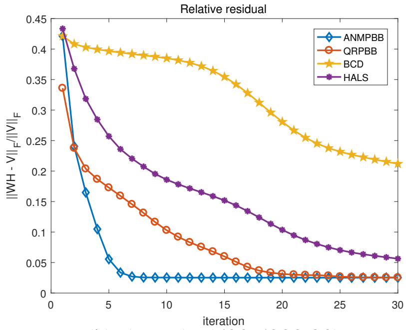
(g) $(m\ n\ r)$=(10000 5000 70).

(h) $(m\ n\ r)$=(30000 10000 80).

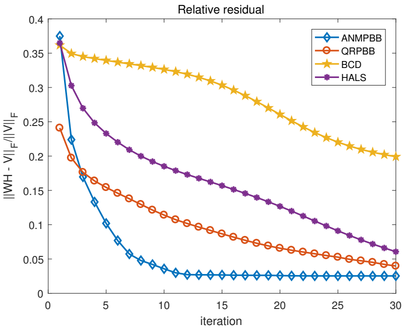
Figure 1. The relationship between the objective value and the iteration of random problem $\min_{W,H\geq 0} \frac{1}{2}\|V - WH\|_F^2$.



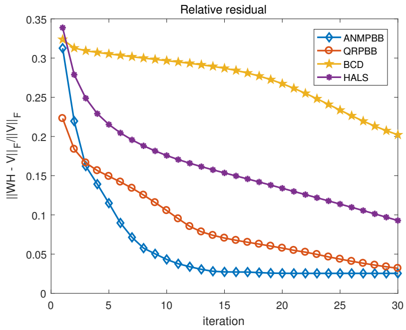
(a) $(m\ n\ r)$=(500 60 10).



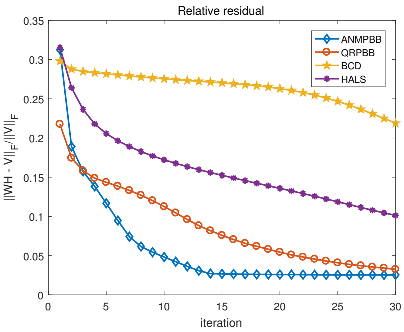
(b) $(m\ n\ r)$=(500 5000 20).



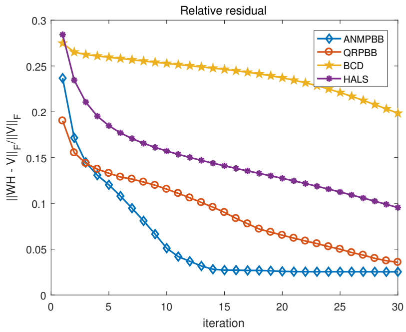
(c) $(m\ n\ r)$=(5000 1000 30).



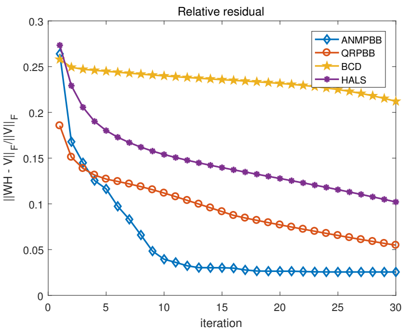
(d) $(m\ n\ r)$=(6000 2000 40).



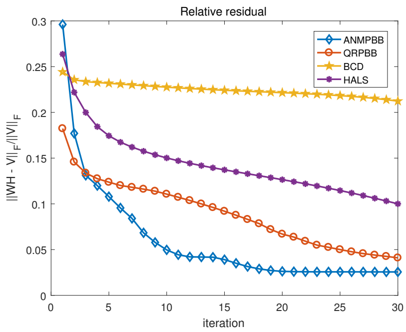
(e) $(m\ n\ r)$=(3000 5000 50).



(f) $(m\ n\ r)$=(10000 2000 60).

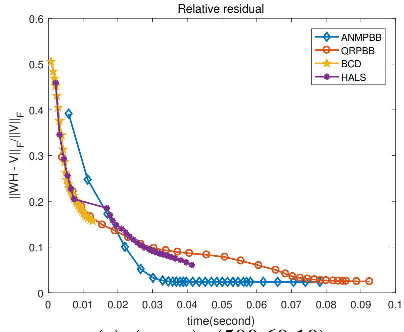


(g) $(m\ n\ r)$=(10000 5000 70).

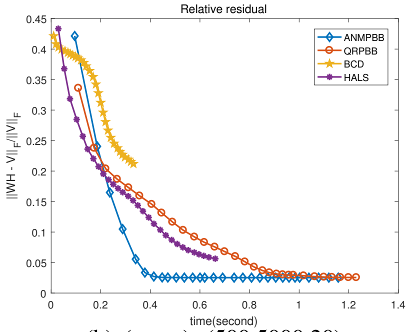


(h) $(m\ n\ r)$=(30000 10000 80).

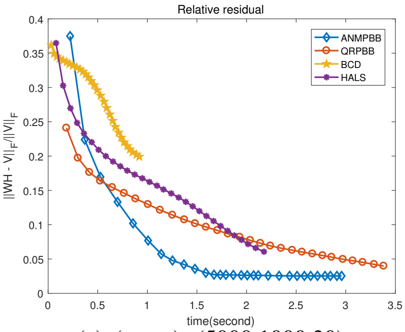
Figure 2. The relationship between the residual value and the iteration of random problem $\min_{W,H\geq 0} \frac{1}{2}\|V - WH\|_F^2$.



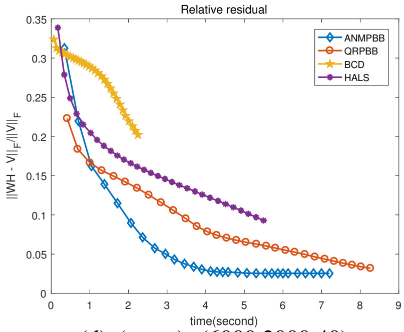
(a) $(m\ n\ r)$=(500 60 10).



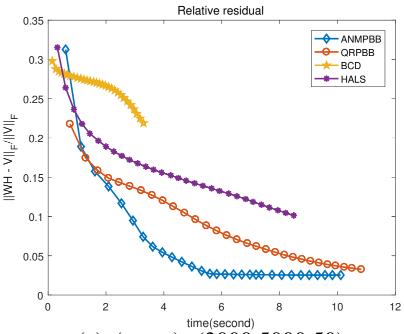
(b) $(m\ n\ r)$=(500 5000 20).



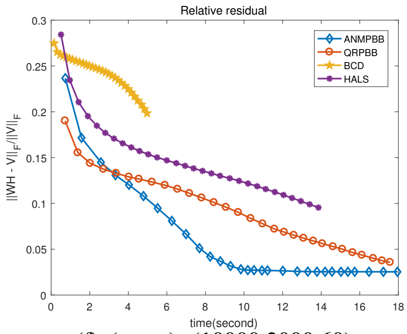
(c) $(m\ n\ r)$=(5000 1000 30).



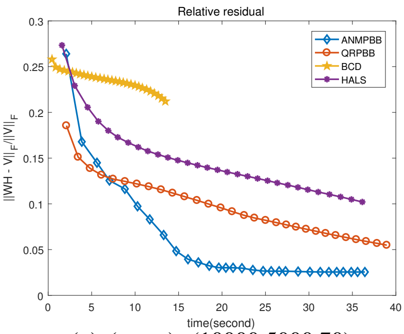
(d) $(m\ n\ r)$=(6000 2000 40).



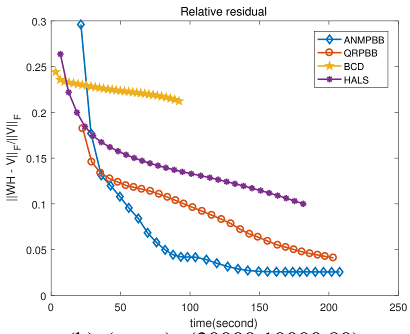
(e) $(m\ n\ r)$=(3000 5000 50).



(f) $(m\ n\ r)$=(10000 2000 60).



(g) $(m\ n\ r)$=(10000 5000 70).



(h) $(m\ n\ r)$=(30000 10000 80).

Figure 3. The relationship between the residual value and CPU time of random problem $\min_{W,H\geq 0} \frac{1}{2}\|V - WH\|_F^2$.

3.3. Image data

The ORL image dataset is a collection of 400 images of people's faces belonging to 40 individuals, 10 each. The dataset includes variations in lighting conditions, facial expressions (including whether they open their eyes and whether they smile), and facial details including whether they wear glasses. Some subjects have multiple photos taken at different times. The images were captured with the subject positioned upright and facing forward (allowing for slight movement to the sides). The background used was uniformly dark and even. All of the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The pictures used are represented by the columns of the matrix V, and V has 400 rows and 1024 columns.

The Yale face dataset was created at the Yale Center for Computational Vision and Control. It consists of 165 gray-scale images, with each person in the dataset having 11 images associated with them. In total, there are 15 people. The facial images in question were captured under different lighting conditions (left-light, center-light, right-light), with various facial expressions (calm, cheerful, sorrowful, amazed, and blinking), and with or without glasses. The pictures used are represented by the rows of the matrix V, and V has 165 rows and 1024 columns.

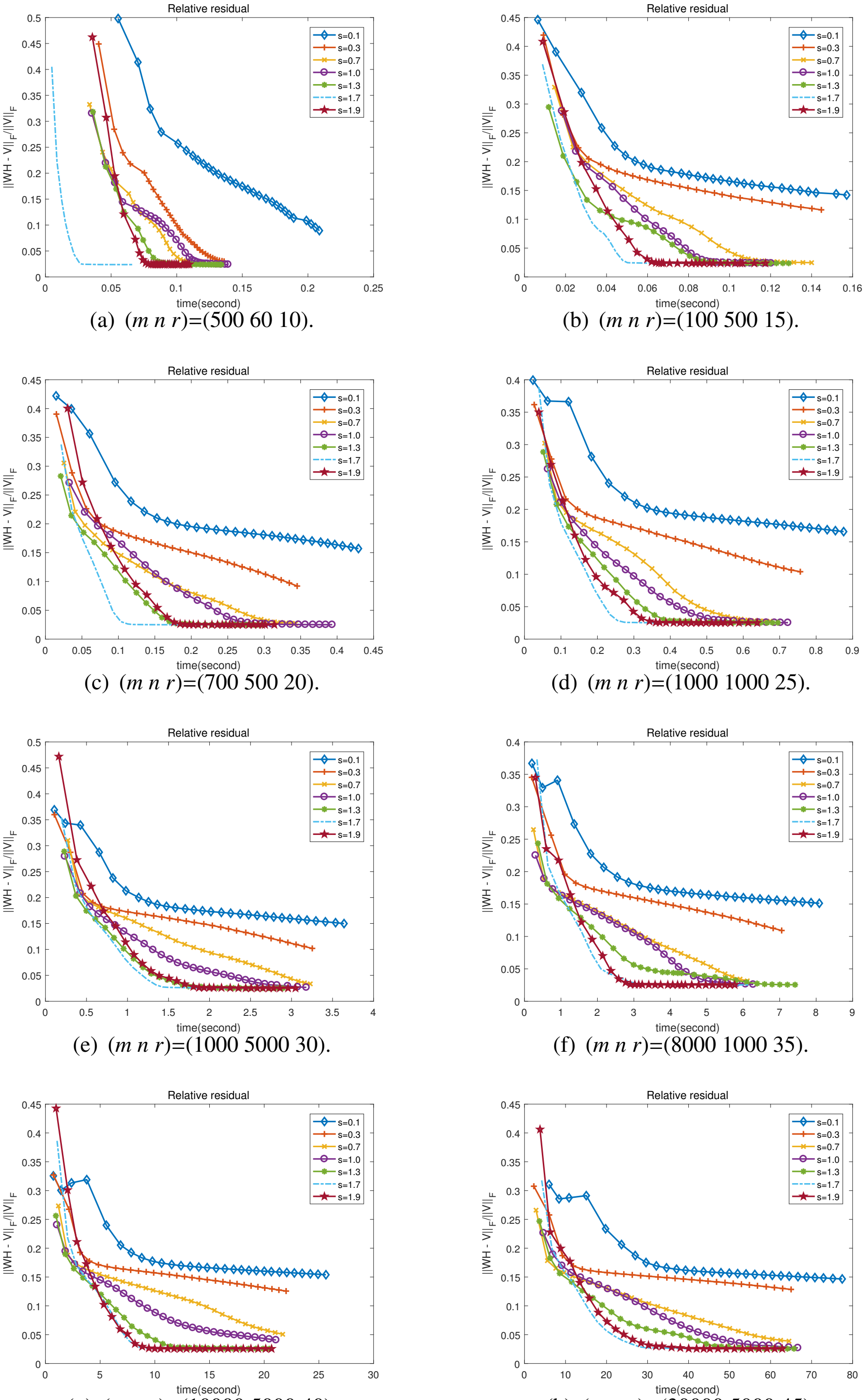
For all of the datasets we used, in (3.1), we performed diverse casually generated starting iterations with $\epsilon = 10^{-8}$, the maximum number of iterations (maxit) for all algorithms was set to 50,000, and the average results are presented in Table 2. From Table 2, we can conclude that the QRPBB method converged in fewer iterations and CPU times than APBB2 and NeNMF, and in contrast to QRPBB, our ANMPBB method required 1/3 the CPU time to satisfy the set tolerance. Although the residuals by ANMPBB were not the smallest among all of the algorithms that appeared for all of the databases we used, the results of *pgn* showed that solutions by ANMPBB were closer to the stationary point.

Table 2. Experimental results on Yale and ORL datasets.

$(m\ n\ r)$	Alg	Iter	Niter	Pgn	Time	Residual
(165 1,024 25)	NeNMF	3735.1	178254.1	4.41E-1	65.78	0.1930
	APBB2	3079.6	97375.7	6.42E-2	78.75	0.1930
	QRPBB	2711.1	54215.7	6.16E-2	42.25	0.1931
	ANMPBB	1284.8	30464.0	2.61E-2	18.78	0.1931
(400 1,024 25)	NeNMF	13613.4	836034.3	7.71E-2	349.62	0.1117
	APBB2	9430.6	446361.6	6.88E-2	474.26	0.1117
	QRPBB	7593.5	213178.5	7.05E-2	205.26	0.1117
	ANMPBB	3292.8	80530.4	6.55E-2	60.86	0.1117

3.4. The importance of relaxation factor s

In the following content, the clear experimental results indicate that relaxation factor s is used for updating the rules of W_{t+1} and H_{t+1}. We implement ANMPBB using diverse s given: $s = 0.1,\ 0.3,\ 0.7,\ 1.0,\ 1.3,\ 1.7,\ 1.9$ on synthetic datas which are the same as those in Section 4.2.



(a) $(m\ n\ r)$=(500 60 10).
(b) $(m\ n\ r)$=(100 500 15).
(c) $(m\ n\ r)$=(700 500 20).
(d) $(m\ n\ r)$=(1000 1000 25).
(e) $(m\ n\ r)$=(1000 5000 30).
(f) $(m\ n\ r)$=(8000 1000 35).
(g) $(m\ n\ r)$=(10000 5000 40).
(h) $(m\ n\ r)$=(30000 5000 45).

Figure 4. The relationship between the residual value and CPU time of random problem $\min_{W,H\geq 0} \frac{1}{2}\|V - WH\|_F^2$.

We set the required maximum number of iterations to 30, and the other parameters required in the experiment will have the same values as those in Section 4.2. Figure 4 shows the relationship between the relative residuals error and the run-time results. In Figure 4, we can see that the relaxation factor s fails to accelerate the convergence when $s < 1$ and increasing constant s significantly accelerates the convergence when $1 < s < 2$. As for ANMPBB, it is seem that $s = 1.7$ is the best compared with other experimental values in terms of the speed of convergence, hence, $s = 1.7$ was used as our ANMPBB in all experiments.

4. Conclusions

In this paper, an active set recognition technology was suggested, and then an improved non-monotonic line search rule was proposed to enhance the efficiency of the nonmonotonic line rule, in which we introduce a new parameter formula to attempt to command the nonmonotonic degree of the line search, and thus increase the likelihood of looking for the global minimum. On the basis of the alternating nonnegative least squares framework, a global convergence gradient-based NMF method was proposed by using the modified line search and the active set recognition technology. In addition, the Barzilai-Borwein step-size and greater step size technique was utilized to make convergence faster. In the end, numerical results showed that our algorithm is an NMF tool with great promise.

NMF is an important linear dimensionality reduction technique for nonnegative data, which has found numerous applications in data analysis such as various clustering tasks [9, 10, 35–37]. Therefore, in future work, we plan to extend the application of our algorithm to multi-class clustering problems. In addition, another direction for future research would be to extend the proposed algorithm to solve real-world problems [38–41].

Author contributions

Xiaoping Xu: Software, formal analysis, resources, visualization; Jinxuan Liu: Methodology, validation, data curation, project administration; Wenbo Li: Conceptualization, validation, writing original draft preparation, supervision, funding acquisition; Yuhan Xu: Conceptualization, validation, investigation; Fuxiao Li: writing original draft preparation, writing review and editing, project administration. All authors have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 12201492).

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. M. Ahookhosh, K. Amini, S. Bahrami, A class of nonmonotone Armijo-type line search method for unconstrained optimization, *Optimization*, **61** (2012), 387–404. https://doi.org/10.1080/02331934.2011.641126
2. E. G. Birgin, J. M. MartÍnez, M. Raydan, Nonmonotone spectral projected gradient methods on convex sets, *SIAM J. Optimiz.*, **10** (2000), 1196–1211. https://doi.org/10.1137/S1052623497330963
3. J. Barzilai, J. M. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.*, **8** (1988), 141–148. https://doi.org/10.1093/imanum/8.1.141
4. S. Bonettini, Inexact block coordinate descent methods with application to non-negative matrix factorization, *IMA J. Numer. Anal.*, **31** (2011), 1431–1452. https://doi.org/10.1093/imanum/drq024
5. A. Cichocki, R. Zdunek, S. Amari, Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization, In: *Independent Component Analysis and Signal Separation*, Heidelberg: Springer, 2007, 169–176. https://doi.org/10.1007/978-3-540-74494-8_22
6. A. Cristofari, M. D. Santis, S. Lucidi, F. Rinaldi, A two-stage active-set algorithm for bound-constrained optimization, *J. Optim. Theory Appl.*, **172** (2017), 369–401. https://doi.org/10.1007/s10957-016-1024-9
7. Y. H. Dai, On the nonmonotone line search, *J. Optim. Theory Appl.*, **112** (2002), 315–330. https://doi.org/10.1023/A:1013653923062
8. Y. H. Dai, L. Z. Liao, R-Linear convergence of the Barzilai-Borwein gradient method, *IMA J. Numer. Anal.*, **22** (2002), 1–10. https://doi.org/10.1093/imanum/22.1.1
9. P. Deng, T. R. Li, H. J. Wang, D. X. Wang, S. J. Horng, R. Liu, Graph regularized sparse non-negative matrix factorization for clustering, *IEEE Transactions on Computational Social Systems*, **10** (2023), 910–921. https://doi.org/10.1109/TCSS.2022.3154030
10. P. Deng, F. Zhang, T. R. Li, H. J. Wang, S. J. Horng, Biased unconstrained non-negative matrix factorization for clustering, *Knowl.-Based Syst.*, **239** (2022), 108040. https://doi.org/10.1016/j.knosys.2021.108040
11. N. Gillis, The why and how of nonnegative matrix factorization, 2014, arXiv:1401.5226. https://doi.org/10.48550/arXiv.1401.5226
12. R. Glowinski, *Numerical methods for nonlinear variational problems*, Heidelberg: Springer, 1984. https://doi.org/10.1007/978-3-662-12613-4
13. P. H. Gong, C. S. Zhang, Efficient nonnegative matrix factorization via projected Newton method, *Pattern Recogn.*, **45** (2012), 3557–3565. https://doi.org/10.1016/j.patcog.2012.02.037
14. N. Z. Gu, J. T. Mo Incorporating nonmonotone strategies into the trust region method for unconstrained optimization, *Comput. Math. Appl.*, **55** (2008), 2158–2172. https://doi.org/10.1016/j.camwa.2007.08.038
15. N. Y. Guan, D. C. Tao, Z. G. Luo, B. Yuan NeNMF: An optimal gradient method for nonnegative matrix factorization, *IEEE T. Signal Proces.*, **60** (2012), 2882–2898. https://doi.org/10.1109/TSP.2012.2190406

16. L. X. Han, M. Neumann, U. Prasad, Alternating projected Barzilai-Borwein methods for nonnegative matrix factorization, *Electronic Transactions on Numerical Analysis*, **36** (2009), 54–82. https://doi.org/10.1007/978-0-8176-4751-3_16

17. G. Hu, B. Du, X. F. Wang, G. Wei, An enhanced black widow optimization algorithm for feature selection, *Knowl.-Based Syst.*, **235** (2022), 107638. https://doi.org/10.1016/j.knosys.2021.107638

18. G. Hu, J. Y. Zhong, G. Wei, SaCHBA_PDN: Modified honey badger algorithm with multi-strategy for UAV path planning, *Expert Syst. Appl.*, **223** (2023), 119941. https://doi.org/10.1016/j.eswa.2023.119941

19. G. Hu, J. Y. Zhong, G. Wei, C. T. Chang, DTCSMO: An efficient hybrid starling murmuration optimizer for engineering applications, *Comput. Method. Appl. M.*, **405** (2023), 115878. https://doi.org/10.1016/j.cma.2023.115878

20. G. Hu, J. Wang, M. Li, A. G. Hussien, M. Abbas, EJS: Multi-strategy enhanced jellyfish search algorithm for engineering applications, *Mathematics*, **11** (2023), 851. https://doi.org/10.3390/math11040851

21. G. Hu, R. Yang, X. Q. Qin, G. Wei, MCSA: Multi-strategy boosted chameleon-inspired optimization algorithm for engineering applications, *Comput. Method. Appl. M.*, **403** (2022), 115676. https://doi.org/10.1016/j.cma.2022.115676

22. G. Hu, X. N. Zhu, G. Wei, C. Chang, An marine predators algorithm for shape optimization of developable Ball surfaces, *Eng. Appl. Artif. Intel.*, **105** (2021), 104417. https://doi.org/10.1016/j.engappai.2021.104417

23. Y. K. Huang, H. W. Liu, S. S. Zhou, Quadratic regularization projected alternating Barzilai-Borwein method for nonnegative matrix factorization, *Data Min. Knowl. Disc.*, **29** (2015), 1665–1684. https://doi.org/10.1007/s10618-014-0390-x

24. Y. K. Huang, H. W. Liu, S. Zhou, An efficint monotone projected Barzilai-Borwein method for nonnegative matrix factorization, *Appl. Math. Lett.*, **45** (2015), 12–17. https://doi.org/10.1016/j.aml.2015.01.003

25. D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, **401** (1999), 788–791. https://doi.org/10.1038/44565

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

27. X. L. Li, H. W. Liu, X. Y. Zheng, Non-monotone projection gradient method for non-negative matrix factorization, *Comput. Optim. Appl.*, **51** (2012), 1163–1171. https://doi.org/10.1007/s10589-010-9387-6

28. H. W. Liu, X. L. Li, Modified subspace Barzilai-Borwein gradient method for non-negative matrix factorization, *Comput. Optim. Appl.*, **55** (2013), 173–196. https://doi.org/10.1007/s10589-012-9507-6

29. C. J. Lin, Projected gradient methods for non-negative matrix factorization, *Neural Comput.*, **19** (2007), 2756–2779. https://doi.org/10.1162/neco.2007.19.10.2756

30. H. Nosratipour, A. H. Borzabadi, O. S. Fard, On the nonmonotonicity degree of nonmonotone line searches, *Calcolo*, **54** (2017), 1217–1242. https://doi.org/10.1007/s10092-017-0226-3

31. D. Kim, S. Sra, I. S. Dhillon, Fast Newton-type methods for the least squares nonnegative matrix approximation problem, *SIAM International Conference on Data Mining,* **1** (2007), 343–354. https://doi.org/10.1137/1.9781611972771.31

32. P. Paatero, U. Tapper, Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics*, **5** (1994), 111–126. https://doi.org/10.1002/env.3170050203

33. M. Raydan, On the Barzilai-Borwein choice of steplength for the gradient method, *IMA J. Numer. Anal.*, **13** (1993), 321–326. https://doi.org/10.1093/imanum/13.3.321

34. M. Raydan, The Barzilai and Borwein gradient method for the large-scale unconstrained minimization problem, *SIAM J. Optimiz.*, **7** (1997), 26–33. https://doi.org/10.1137/S1052623494266365

35. D. X. Wang, T. R. Li, P. Deng, J. Liu, W. Huang, F. Zhang, A generalized deep learning algorithm based on NMF for multi-view clustering, *IEEE T. Big Data*, **9** (2023), 328–340. https://doi.org/10.1109/TBDATA.2022.3163584

36. D. X. Wang, T. R. Li, P. Deng, F. Zhang, W. Huang, P. F. Zhang, et al., A generalized deep learning clustering algorithm based on non-negative matrix factorization, *ACM T. Knowl. Discov. D.*, **17** (2023), 1–20. https://doi.org/10.1145/3584862

37. D. X. Wang, T. R. Li, W. Huang, Z. P. Luo, P. Deng, P. F. Zhang, et al., A multi-view clustering algorithm based on deep semi-NMF, *Inform. Fusion*, **99** (2023), 101884. https://doi.org/10.1016/j.inffus.2023.101884

38. Z. J. Wang, Z. S. Chen, L. Xiao, Q. Su, K. Govindan, M. J. Skibniewski, Blockchain adoption in sustainable supply chains for Industry 5.0: A multistakeholder perspective, *J. Innov. Knowl.*, **8** (2023), 100425. https://doi.org/10.1016/j.jik.2023.100425

39. Z. J. Wang, Z. S. Chen, S. Qin, K. S. Chin, P. Witold, M. J. Skibniewski, Enhancing the sustainability and robustness of critical material supply in electrical vehicle market: An AI-powered supplier selection approach, *Ann. Oper. Res.*, **2023** (2023), 102690. https://doi.org/10.1007/s10479-023-05698-4

40. Z. J. Wang, Y. Y. Sun, Z. S. Chen, G. Z. Feng, Q. Su, Optimal versioning strategy of enterprise software considering the customer cost-acceptance level, *Kybernetes*, **52** (2023), 997–1026. https://doi.org/10.1108/K-04-2021-0339

41. Z. J. Wang, Y. Y. Sun, Q. Su, M. Deveci, K. Govindan, M. J. Skibniewski, et al., Smart contract application in resisting extreme weather risks for the prefabricated construction supply chain: prototype exploration and assessment, *Group Decis. Negot.*, (2024). https://doi.org/10.1007/s10726-024-09877-x

42. Y. H. Xiao, Q. J. Hu, Subspace Barzilai-Borwein gradient method for large-scale bound constrained optimization, *Appl. Math. Optim.*, **58** (2008), 275–290. https://doi.org/10.1007/s00245-008-9038-9

43. Y. H. Xiao, Q. J. Hu, Z. X. Wei, Modified active set projected spectral gradient method for bound constrained optimization, *Appl. Math. Model.*, **35** (2011), 3117–3127. https://doi.org/10.1016/j.apm.2010.09.011

44. Y. Y. Xu, W. T. Yin, A block coordinate descent method for regularized multi-convex optimization with applications to nonnegative tensor factorization and completion, *SIAM J. Imaging Sci.*, **6** (2013), 1758–1789. https://doi.org/10.1137/120887795

45. H. C. Zhang, W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM J. Optimiz.*, **14** (2004), 1043–1056. https://doi.org/10.1137/S1052623403428208

46. R. Zdunek, A. Cichocki, Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems, *Comput. Intel. Neurosc.*, **2008** (2008), 939567. https://doi.org/10.1155/2008/939567

AIMS Press