Int J Data Sci Anal (2017) 3:23–34
DOI 10.1007/s41060-016-0037-7

REGULAR PAPER

Accelerated anti-lopsided algorithm for nonnegative least squares

Duy Khuong Nguyen[1,2] · **Tu Bao Ho**[1,3]

Received: 23 October 2015 / Accepted: 19 November 2016 / Published online: 1 December 2016

Abstract Nonnegative least squares (NNLS) problem has been widely used in scientific computation and data modeling, especially for low-rank representation such as nonnegative matrix and tensor factorization. When applied to large-scale datasets, first-order methods are preferred to provide fast flexible computation for regularized NNLS variants, but they still have the limitations of performance and convergence as key challenges. In this paper, we propose an accelerated anti-lopsided algorithm for NNLS with linear over-bounded convergence rate $\left[\left(1-\frac{\mu}{L}\right)\left(1-\frac{\mu}{nL}\right)^{2n}\right]^k$ in the subspace of passive variables where μ and L are always bounded as $\frac{1}{2} \leq \mu \leq L \leq n$, and n is the dimension size of solutions, which is highly competitive with current advanced methods such as accelerated gradient methods having sub-linear convergence $\frac{L}{k^2}$, and greedy coordinate descent methods having convergence $\left(1-\frac{\mu}{nL}\right)^k$, where μ and L are unbounded. The proposed algorithm transforms the variable x into the new space satisfying the second derivative equals constant $\frac{\partial^2 f}{\partial x_i^2} = 1$ for all variables x_i to implicitly exploit the second-order derivative, and to guarantee that μ and L are always bounded in order to achieve over-bounded convergence of the algorithm, and to enhance the performance of internal processes based on exact line search, greedy coordinate descent methods, and accelerated search. The experiments on large matrices and real applications of nonnegative matrix factorization clearly show the higher performance of the proposed algorithm in comparison with the state-of-the-art algorithms.

Keywords Nonnegative least squares · Accelerated anti-lopsided algorithm · First-order methods

✉ Duy Khuong Nguyen
khuongnd@gmail.com; khuongnd@vnu.edu.vn;
khuongnd@jasit.ac.jp

[1] Japan Advanced Institute of Science and Technology, Nomi, Japan

[2] University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

[3] John von Neumann Institute, Vietnam National University, Ho Chi Minh City, Vietnam

1 Introduction

Minimizing the sum of squares of the errors is one of the most fundamental problems in numeric analysis as known as the nonnegative least squares (NNLS) problem. It has been widely used in scientific computation and data mining to approximate observations [5]. Specially, in many fields such as image processing, computer vision, text mining, environmetrics, chemometrics, or speech recognition, observations $b \in \mathbb{R}^d$ are often approximated by a set of measurements or basis factors $\{A_i\}$ contained in a matrix $A \in \mathbb{R}^{d\times n}$ via minimizing $\frac{1}{2}\|Ax-b\|_2^2$. Moreover, in comparison with least squares (LS), NNLS has more concisely interpretable solutions, of which nonnegative coefficients $x \in \mathbb{R}_+^n$ can be interpreted as contributions of the measurements over the observations. In contrast, mixed-sign coefficients of LS solutions are uninterpretable because they lead to overlapping and mutual elimination of the measurements.

Because of no generic formula of solutions unlike least squares (LS) problem, although NNLS is a convex optimization problem, multiple iterative algorithms and gradient methods are widely employed to solve NNLS. The performance of NNLS algorithms mainly depends on selecting appropriate directions to optimize the objective function. To improve the performance, most effective algorithms remove redundant variables based on the concept of active sets

[2,5] in each iteration with different strategies [5]. These algorithms are fundamentally based on the observation that several variables can be ignored if they are negative when the problem is unconstrained [2,15,21]. In other words, NNLS can be considered an unconstrained problem in a subspace of several variables [13] that are positive in the optimal solution. In addition, algorithms using the second derivative [2,15,21] discover effective directions to more effectively reduce the objective function value. However, these approaches have two main drawbacks: invertibility of $A^T A$ and its heavy computation, especially for the methods recomputing $(A^T A)^{-1}$ several times for different passive sets. Hence, first-order methods [7,13,19] can be more effective for large-scale least squares problems.

Since 1990s, the methods of nonnegative matrix or tensor factorizations have widely used NNLS to achieve low-rank representation of nonnegative data [14,22]. Specially, the low-rank representation transfers data instances into a lower-dimensional space of latent components to obtain increased speed and accuracy, and more concise interpretability of data processing that is essential in applications of signal and image processing, machine learning, and data mining [5]. However, the low-rank representation is usually a non-convex problem, and it often employs iterative multiplicative update algorithms. In addition, exact algorithms often lack flexibility for low-rank regularized variants and also have high complexity and slow convergence. Hence, fast approximate algorithms based on the first-order methods are more preferred to naturally provide a flexible framework for low-rank models [4,9–11].

In our view, to discover more appropriate gradient directions is to critically enhance the performance of NNLS algorithms based on the first-order methods for the low-rank representation. In this paper, we propose a fast and robust iterative algorithm called accelerated anti-lopsided algorithm, which combines several algorithms and ideas with different advantages to implicitly exploit the second-order derivative and reduce negative effects of variable scaling problems to obtain fast convergence. The proposed algorithm has the following advantages:

- **Convergence**: the accelerated anti-lopsided algorithm for NNLS attains linear convergence rate of $[(1 - \frac{\mu}{L})(1 - \frac{\mu}{nL})^{2n}]^k$ in the subspace of passive variables where n is the dimension size of solutions, and μ and L are always bounded as $\frac{1}{2} \leq \mu \leq L \leq n$ to guarantee over-bounded convergence rate. Meanwhile, current advanced first-order methods are accelerated gradient methods having sub-linear convergence $\mathcal{O}(\frac{L}{k^2})$ and greedy coordinate descent algorithm having convergence $(1 - \frac{\mu}{nL})^k$, where μ and L are unbounded.
- **Robustness**: the algorithm can stably work in ill-conditioned cases for NNLS regularizations since it is totally based on the first derivative and it does not require computing the inverse of matrices $(A^T A)$ like Newton methods. In addition, it can exploit the second derivative by guaranteeing $\frac{\partial^2 f}{\partial x_i^2} = 1, \forall i$ to void the worst cases and discover more effective gradient directions, while keeping the low complexity of each iteration $\mathcal{O}(n^2)$. Moreover, μ and L are always bounded as $\frac{1}{2} \leq \mu \leq L \leq n$, which increase the effectiveness of greedy coordinate descent and exact line search algorithms that depend on these parameters μ and L.
- **Effectiveness**: the experimental results for NNLS are highly competitive with the state-of-the-art methods. These results additionally show that the algorithm is the fastest first-order method for NNLS in both practice and theory.

The rest of the paper is organized as follows. Section 2 discusses the background and related work on nonnegative least square problems. Section 3 presents the accelerated anti-lopsided algorithm for NNLS. The theoretical analysis is discussed in Sect. 4. Section 5 shows the experimental results, and Sect. 6 summarizes the main contributions of this paper.

2 Background and related works

This section introduces the nonnegative least square (NNLS) problem, its equivalent nonnegative quadratic problem (NQP), and significant milestones in the algorithmic development for NNLS.

2.1 Background

Nonnegative least square (NNLS) can be considered one of the most central problems in data modeling, of which solution can estimate the parameters of models for describing the data [5]. It comes from scientific applications where we need to estimate a large number of vector observations $b \in \mathbb{R}^d$ using a set of measures or basis factors $\{A_i\}$ contained in a matrix $A \in \mathbb{R}^{d \times n}$ via minimizing $\frac{1}{2}\|Ax - b\|_2^2$. Hence, we can define NNLS as follows:

Definition 1 Given n measurement vectors $A = [A_1, A_2, \ldots, A_n] \in \mathbb{R}^{d \times n}$ and an observed vector $b \in \mathbb{R}^d$, nonnegative least squares (NNLS) problem finds an optimal solution x of the optimization problem:

$$\begin{array}{ll} \underset{x}{minimize} & \frac{1}{2}\|Ax - b\|_2^2 \\ \text{subject to} & x \succeq 0 \\ \text{where} & A \in \mathbb{R}^{d \times n}, b \in \mathbb{R}^d \end{array} \tag{1}$$

For low-rank representation models such as nonnegative matrix and tensor factorization, L_2 and L_1 regularizations are usually added to control the smoothness and sparsity of these models:

$$\begin{array}{ll} \underset{x}{minimize} & \frac{1}{2}\|Ax-b\|_2^2+\frac{\alpha}{2}\|x\|_2^2+\beta\|x\|_1 \\ \text{subject to} & x \succeq 0 \\ \text{where} & A \in \mathbb{R}^{d\times n}, b \in \mathbb{R}^d, \alpha \geq 0, \beta \geq 0 \end{array} \tag{2}$$

Usually, $d \gg n$ in the low-rank representation; hence, it should be equivalently turned into a nonnegative quadratic programming (NQP) problem:

$$\begin{array}{ll} \underset{x}{minimize} & f(x)=\frac{1}{2}x^THx+h^Tx \\ \text{subject to} & x \succeq 0 \\ \text{where} & H=A^TA+\alpha I, h=-A^Tb+\beta\mathbf{1}^n, \\ & \text{and } \mathbf{1}^n=[1,\ldots,1]^T \end{array} \tag{3}$$

From NNLS and its NQP formulation, these problems are convex since Q is positive semi-definite and the nonnegativity constraints form a convex feasible set. In this paper, we will solve Problem 3 instead of Problem 2 within thousands of variables, which often occurs in the low-rank representation.

2.2 Related works

In the last s everal decades of development, different approaches have been proposed to tackle the NNLS problem, which can be divided into two main groups: active set methods and iterative methods [5].

Active set methods are based on the observation that variables can be divided into subsets of active and passive variables [8]. Particularly, the active set contains variables which are zero or negative when solving the least square problem without concerning nonnegative constraints; otherwise, the remaining variables belong to the passive set. The active set algorithms are based on the fact that if the passive set is identified, the passive variables' values in NNLS are the unconstrained least squares solution when the active variables are set to zero. However, these sets are unknown in advance. Hence, a number of iterations are employed to find out the passive set, each of which is to solve a unconstrained least squares problem on the passive set to update the passive set and the active set.

Concerning the significant milestones of the active set methods, Lawson and Hanson [15] proposed a standard algorithm for active set methods. Subsequently, Bro and De Jong [2] avoided unnecessary re-computations on multiple right-hand sides to speed up the basic algorithm [15]. Finally, Dax [6] proposed selecting a good starting point by Gauss–Seidel iterations and moving away from a "dead point" to reduce the number of iterations. Furthermore, the iterative methods use the first-order gradient on the active set to handle multiple active constraints in each iteration, while the active set methods only handle one active constraint [5]. Hence, the iterative methods can deal with larger-scale problems [12,13] than the active set methods. However, they do not guarantee the convergence rate.

More recently, Franc et al. [7] proposed a cycle block coordinate descent method having fast convergence in practice with low complexity of each iteration, but it still has been not theoretically guaranteed. Subsequently, Vamsi [19] suggested three modifications of random permutations [17], shrinking, and random projections to speed up NNLS for the case that the matrix A is not thin ($d \leq n$). Furthermore, accelerated methods [16] and proximal methods [18] having a fast convergence $O(1/k^2)$ [10] only require the first-order derivative. However, one major disadvantage of accelerated methods is that they require a large number of iterations to reach high accuracy because the step size is limited by $\frac{1}{L}$, which is usually small for large-scale NNLS problems with big matrices, where L is the Lipschitz constant. The comparison summary of NNLS solvers are presented in Table 1.

Table 1 Comparison summary of NNLS solvers

Criteria	ELS	Coord	Accer	Fast	Nm	Frugal	Antilop
Iteration complexity	n^2	n^2	n^2	n^3	$\#(nd)$	$\#(nd)$	n^2
Convergence rate	$\left(1-\frac{\mu}{L}\right)^k$	?	$\frac{L}{k^2}$	?	?	?	$\left[\left(1-\frac{\mu}{L}\right)\left(1-\frac{\mu}{nL}\right)^{2n}\right]^k$
Over-bounded convergence	✗	✗	✗	✗	✗	✗	✓
Memory size	$n(n+d)$	$n(n+d)$	$n(n+d)$	$n(n+d)$	$\#(nd)$	$\#(nd)$	$n(n+d)$
Not compute A^TA	✗	✗	✗	✗	✓	✓	✗
Not compute $(A^TA)^{-1}$	✓	✓	✓	✗	✓	✓	✓

$\#(nd)$: nonzero number of matrix having size nd

d dimension of data, n number of variables, μ convex parameter, L Lipschitz constant, *ELS* exact line search, *Coord* greedy block coordinate descent [7], *Accer* accelerated method [10], *fast* active set methods according to Bro R., de [2] *Nm* non-monotonic fast method [13], *frugal* frugal coordinate descent [19], *Antilop*: the proposed method

✓ satisfied (positive), ✗ unsatisfied, ? unknown

In summary, active set methods and iterative methods are two major approaches in solving NNLS. Active set methods accurately solve nonnegative least squares problems, but require huge computation for solving unconstrained least squares problems and are unstable when A^TA is ill-conditioned. Iterative methods are more potential for solving large-scale NNLS because they can handle multiple active constraints per each iteration. In our view, iterative methods are still ineffective due to the scaling variable problem, which seriously affects the finding of appropriate gradient directions. Therefore, we propose an accelerated anti-lopsided algorithm combining several algorithms and ideas having different advantages to reduce negative effects of the scaling problem, obtain appropriate gradient directions, and achieve over-bounded linear convergence in the subspace of passive variables.

3 Accelerated anti-lopsided algorithm

This section discusses the main ideas in the proposed algorithm Algorithm 1 to increase the speed of the NNLS solver. According to the literature, first-order methods have slow convergence because of zigzagging problems or variable scaling problems [1]. In other words, the imbalanced role of variables over the objective function causes the first derivative achieving the inappropriate direction to optimize the objective function. Hence, we propose Algorithm 1, represented in the flowchart Fig. 1, including four important parts to attain fast convergence:

- Part 1. Anti-lopsided transformation from Line 3 to Line 5: the variable vector x is transformed into a new space by $x = \varphi(y)$ as an inverse function. In the new space, the new equivalent objective function $g(y) = f(\varphi(y))$ has $\frac{\partial^2 g}{\partial y_i^2} = 1,\ \forall i$, or the acceleration of each variable equals 1. As a result, the roles of variables become more balanced because the level curve of the function becomes more spherical because $\frac{\partial^2 g}{\partial y_i^2} = 1,\ \forall i$, and $g(y)$ is convex. This part aims to make the post-processing parts more effective because it can implicitly exploit the second derivative information $\frac{\partial^2 g}{\partial y_i^2} = 1,\ \forall i$ to guarantee that μ and L are always bounded as $\frac{1}{2} \leq \mu \leq L \leq n$.
- Part 2. Exact line search from Line 12 to Line 16: this part optimizes the objective function with a guarantee of over-bounded convergence rate $(1 - \frac{\mu}{L})^k$ where $\frac{1}{2} \leq \mu \leq L \leq n$ over the space of passive variables, which has a complexity $\mathcal{O}(n^2)$. The part aims to reduce the objective functions exponentially and precisely, although it suffers from variable scaling problems and nonnegative constraints.
- Part 3. Greedy coordinate descent algorithm from Line 18 to Line 22 and repeated in Line 29: this part employs greedy coordinate descent using Gauss–Southwell rule with exact optimization to rapidly reduce the objective function with fast convergence $\mathcal{O}(1-\frac{\mu}{nL})$ for each update [17,20], which has a complexity of $\mathcal{O}(n^2)$. The part aims to reduce negative effects of variable scaling problems and nonnegative constraints, although it has zigzagging problems because of optimizing the objective function over each single variable. Due to having fast convergence in practice and reducing negative effects of variable scaling problems and nonnegative constraints, this part is repeated one more time after Part 4.
- Part 4. Accelerated search from Line 24 to Line 28: this step performs a descent momentum search based on pre-

Algorithm 1: Anti-lopsided algorithm for NNLS

Input: $A \in \mathbb{R}^{d\times n}$; $b \in \mathbb{R}^d$, $\alpha > 0$, and $\beta > 0$
Output: x minimizing $f(x) = \|Ax - b\|_2^2 + \frac{\alpha}{2}\|x\|_2^2 + \beta\|x\|_1$ subject to: $x \succeq 0$

1 **begin**
2 /*Transfer $f(x)$ into $f(x')$ satisfying : $\frac{\partial^2 f}{\partial^2 x_i'} = 1,\ \forall i$*/;
3 $H = A^TA + \alpha I$;
4 $Q = \frac{H}{\sqrt{diag(H)diag(H)^T}}$;
5 $q = \frac{-A^Tb+\beta 1^n}{\sqrt{diag(H)}}$;
6 /*Minimize $f(x) = \frac{1}{2}x^TQx + q^Tx$;
7 $x^0 = \mathbf{0}^n$;
8 $\nabla f = q$;
9 **repeat**
10 $x_s = x_k$ and $\nabla f_s = \nabla f$;
11 /*Exact line search algorithm over passive variables*/;
12 $\nabla \bar{f} = \nabla f$; and $\nabla \bar{f}[x = 0$ and $\nabla f > 0] = 0$;
13 $\alpha = \underset{\alpha}{\text{argmin}}\ f(x_k - \alpha\nabla\bar{f}) = \frac{\|\nabla\bar{f}\|_2^2}{\nabla\bar{f}^TQ\nabla\bar{f}}$;
14 $x_{k+1} = x_k - \alpha\nabla\bar{f}$;
15 $\nabla f = \nabla f - \alpha Q\nabla\bar{f} - Q[x_{k+1}]_-$;
16 $x_{k+1} = [x_{k+1}]_+$;
17 /*Greedy coordinate descent algorithm*/;
18 **for** *t=1 to n* **do**
19 $p = \underset{i,i\in P(x)}{\text{argmax}}\ |\nabla_i f(x_k)|$;
20 $\triangle x_p = max(0, [x_{k+1}]_p - \frac{\nabla_p f}{Q_{pp}}) - [x_{k+1}]_p$;
21 $\nabla f = \nabla f + Q_p\triangle x_p$;
22 $[x_{k+1}]_p = [x_k]_p + \triangle x_p$;
23 /*Accelerated search carries a "momentum" based on the changes in variables in exact line search and greedy coordinate descent part*/;
24 $\triangle x = x_s - x_{k+1}$ /*x_s and ∇f_s are assigned in Line 1*/;
25 $\alpha = \underset{\alpha}{\text{argmin}}\ f(x_{k+1} - \alpha\triangle x) = \frac{\nabla f^T\triangle x}{\triangle x^TQ\triangle x} = \frac{\nabla f^T\triangle x}{\triangle x^T(\nabla f_s - \nabla f)}$;
26 $x_{k+1} = x_{k+1} - \alpha\triangle x$;
27 $\nabla f = \nabla f - \alpha Q\triangle x - Q[x_{k+1}]_-$;
28 $x_{k+1} = [x_{k+1}]_+$;
29 Repeat steps in the part of greedy coordinate descent algorithm ;
30 **until** $\|\nabla\bar{f}\|^2 < \epsilon$;
31 /*Inverse x back to the original space */;
32 $x_{k+1} = \frac{x_{k+1}}{\sqrt{diag(H)}}$;
33 **return** x_{k+1}
34 **end**

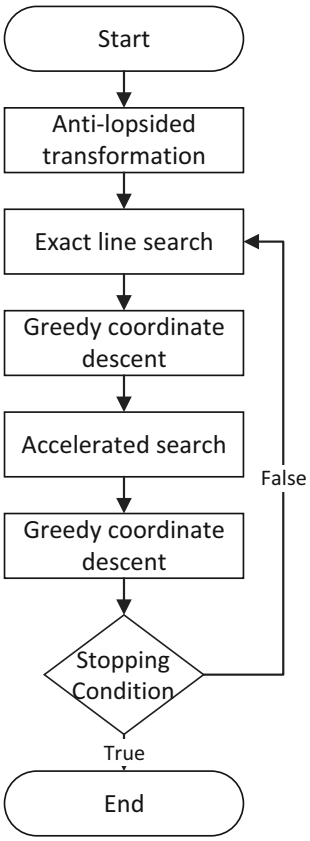

Fig. 1 Flowchart of accelerated anti-lopsided algorithm

vious changes in variables in Part 3 and Part 4, which has a low complexity of $\mathcal{O}(n{\cdot}\mathrm{nn}(n))$ where $\mathrm{nn}(n)$ is the number of negative elements in $(x_{k+1} - \alpha \triangle x)$, see Line 27 in Algorithm 1. This part relies on the global information of two distinct points to escape the local optimal information issues of the first derivative raised by the function complexity. This part originates from the idea that if the function is optimized from x_s to x_k by the exact line search and the coordinate descent algorithm, it is highly possible that the function value will be reduced along the vector $(x_k - x_s)$ because the NNLS objective function is convex and has (super) eclipse sharp.

In summary, the proposed algorithm has various advantages because it combines several algorithms. Hence, it can achieve these various advantages to significantly reduce the iteration number, and negative effects of variable scaling problems and nonnegative constraints in order to attain fast over-bounded convergence rate, although its iteration complexity increases several times.

Let us consider the effectiveness of the proposed algorithms by comparing with the exact gradient line search for the original NNLS problem, which is severely influenced by variable scaling problems. For example, if we employ the

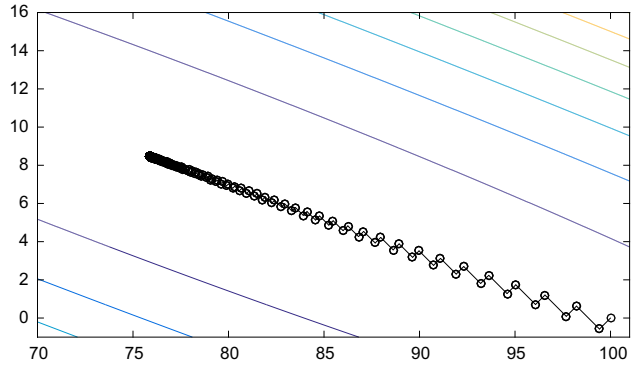

Fig. 2 A total of 273 optimizing steps by exact line search method using the first-order derivative for the Function 4 starting at $x_0 = [1000]^T$

iterative exact line search method using the first-order derivative to optimize Function 4, we need 273 iterations to reach the optimal solution (see Fig. 2):

$$f(x) = \frac{1}{2}\left\| \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 9 \end{bmatrix} x - \begin{bmatrix} 50 \\ 200 \\ 300 \end{bmatrix} \right\|_2^2 \tag{4}$$

To reduce negative effects of these scaling problems, we rescale variables into a new space, in which new variables have more balanced roles and for that reason, we name the proposed algorithm as accelerated anti-lopsided algorithm. Specially, we rescale:

$$x = \frac{y}{\sqrt{diag(H)}} \quad \text{or} \quad x_i = \frac{y_i}{\sqrt{H_{ii}}},\ \forall i \tag{5}$$

After rescaling variables, the original Problem 3 is equivalently transformed into NQP Problem 6:

$$\begin{array}{ll} \underset{y}{minimize} & f(y) = \frac{1}{2} y^T Q y + q^T y \\ \text{subject to} & y \succeq 0 \\ \text{where} & Q_{ij} = \frac{H_{ij}}{\sqrt{H_{ii} H_{jj}}};\ q_i = \frac{h_i}{\sqrt{H_{ii}}} \end{array} \tag{6}$$

Remark 1 Consider the values of matrix Q, we have:

- $\frac{\partial^2 f}{\partial^2 y_i} = Q_{ii} = \frac{H_{ii}}{\sqrt{H_{ii}^2}} = 1,\ \forall\, i = 1, \ldots, n$
- $\frac{\partial^2 f}{\partial y_i \partial y_j} = Q_{ij} = \frac{H_{ij}}{\sqrt{H_{ii} H_{jj}}},\ \forall\, 1 \le i \ne j \le n$
  $\Rightarrow |Q_{ij}| = \frac{|<A_i, A_j>|}{|A_i||A_j|+\alpha} \le |\cos(A_i, A_j)| \le 1 \,\text{since}\, \alpha > 0.$

The scaling variable problem is significantly reduced because the acceleration of the function over variables equals 1, and the roles of variables in the function become more balanced. For example, Fig. 3 has the change in function

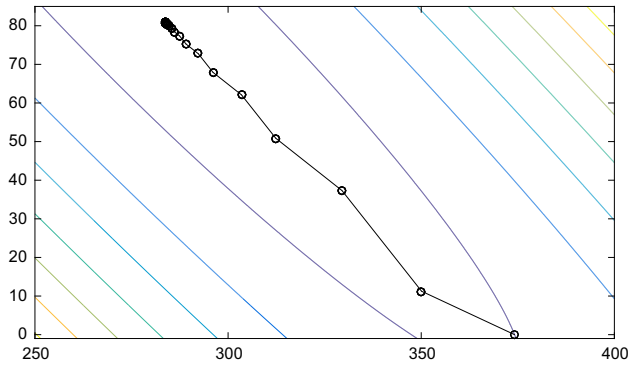

Fig. 3 A total of 21 optimizing steps by the proposed algorithm starting at $y_0 = x_0\sqrt{diag(H)}$ or $y_{0i} = x_{0i}H_{ii}, \forall i$

value over variables more balanced than in Fig. 2. In addition, the level curve of Function 4 is transformed from long-ellipse shaped (see Fig. 2) to short-ellipse shaped (see Fig. 3) in Function 6. Furthermore, by combining fast convergence algorithms such as exact line search, greedy coordinate descent, and accelerated search, the proposed algorithm can work much more effectively. For example, we need only 21 iterations instead of 271 iterations to reach the optimal solution for the Function 4 with the same initial point y_0, where $y_{0i} = x_{0i}.H_{ii}, \ \forall i$ (see Fig. 3).

For each iteration of Algorithm 1 for solving the NQP problem with input Q and q, the objective function is optimized on the passive set:

$$P(x) = \{x_i | x_i > 0 \text{ or } \nabla f_i(x) < 0\}$$

Hence, the first-order gradient will be projected into the subspace of the passive set $\nabla\bar{f} = [\nabla f]_{P(x)}$ ($\nabla\bar{f}_i = \nabla f_i$ for $i \in P(x)$, otherwise $\nabla\bar{f}_i = 0$). Noticeably, the passive set can change through iterations, and Algorithm 1 is converged when $P(x) = \emptyset$ or $\|\nabla\bar{f}\|^2 < \epsilon$ based on KKT conditions. In addition, the orthogonal projection on the subspace of passive variables $x = [x_k + \alpha\nabla\bar{f}]_+$ is trivial [13] since the NQP Problem 6 is a strongly convex problem on a convex set.

Concerning computing a of the exact line search in Algorithm 1, we have:

$$\begin{aligned} f\left(x_k - \alpha\nabla\bar{f}\right) &= -\alpha\nabla\bar{f}^T\left[Qx_k + q\right] + \frac{\alpha^2}{2}\nabla\bar{f}^T Q\nabla\bar{f} \\ &\quad + \mathrm{C} \text{ where } C \text{ is constant} \\ &\Rightarrow \frac{\partial f}{\partial\alpha} = 0 \Leftrightarrow \alpha = \frac{\nabla\bar{f}^T\left(Qx_k + q\right)}{\nabla\bar{f}^T Q\nabla\bar{f}} \\ &= \frac{\nabla\bar{f}^T\nabla f}{\nabla\bar{f}^T Q\nabla\bar{f}} = \frac{\|\nabla\bar{f}\|_2^2}{\nabla\bar{f}^T Q\nabla\bar{f}} \end{aligned}$$

Similarly, regarding computing a of the accelerated search, we have:

$$\alpha = \frac{\triangle x^T\left(Qx_{k+1} + q\right)}{\triangle x^T Q\triangle x} = \frac{\triangle x^T\nabla f}{\triangle x^T Q\triangle x} = \frac{\nabla f^T\triangle x}{\triangle x^T\left(\nabla f_s - \nabla f\right)}$$

Furthermore, to reduce the complexity of Algorithm 1, we save $Q\nabla\bar{f}$ to avoid recomputing, see Line 15, and compute $(Q\triangle x = \nabla f_s - \nabla f)$, see from Line 25 to Line 27.

4 Theoretical analysis

This section analyzes the convergence and complexity of the proposed algorithm.

4.1 Convergence

Concerning the convergence rate, our method argues Barzilai and Borwein's note that NNLS is an unconstrained optimization on the passive set of variables [2]. Moreover, the orthogonal projection on the subspace of passive variables $x = [x_k + \alpha\nabla\bar{f}]_+$ is trivial [13] since NNLS and its equivalent problem (NQP) are strongly convex on a convex set. In addition, the greedy coordinate descent using Gauss–Southwell rule with exact optimization has fast convergence rate $(1 - \frac{\mu}{nL})$ for each update [17,20]. Hence, in this section, we analyze the convergence rate of the exact line search in Algorithm 1 and determine the over-bounds of μ and L in the subspace of passive variables, which significantly influence the convergence rate of the proposed algorithm. Furthermore, we only consider convergence of NLLS solver without regularizations because it is assumed that L_1 and L_2 coefficients α, β slightly affect the convergence of algorithms for the following reasons: first, the L_1 regularized coefficient β do not change the Hessian matrix; second, the L_2 regularized coefficient α is often small, and they slightly influence $\frac{\mu}{L}$ because they change both the convex parameter μ and the Lipschitz constant L by adding the same positive value α.

Consider the convergence rate of the exact line search for $f(x) = \frac{1}{2}x^T Qx + q^T x$. Since $f(x)$ is strongly convex, we have:

- $\exists\mu, L > 0$ satisfy $\mu I \preceq \nabla^2 f \preceq LI$
- $\forall x, y : f(y) \geq f(x) + \langle\nabla f(x), (y - x)\rangle + \frac{\mu}{2}\|y - x\|^2$
- $\forall x, y : f(y) \leq f(x) + \langle\nabla f(x), (y - x)\rangle + \frac{L}{2}\|y - x\|^2$

Based on Theorem 4.5 and Section 4.1.4 in Lecture 4 of [3], we have:

Theorem 1 *After* $(k+1)$ *iterations, the convergence rate of the exact line search is* $f(x^{k+1}) - f^* \leq (1 - \frac{\mu}{L})^k(f(x^0) - f^*)$, *where* f^* *is the minimum value of* $f(x)$.

Proof Since $f(y) \leq f(x) + \langle\nabla f, y - x\rangle + \frac{L}{2}\|y - x\|_2^2, \ \forall x, y$ selecting $y = x - \frac{1}{L}\nabla f$ and x^+ is the updated value of x

after an iteration by the first-order gradient using exact line search, we have:

$$\begin{aligned} f(x^+) &\le f\left(x - \frac{1}{L}\nabla f\right) \le f(x) - \frac{1}{L}\|\nabla f\|_2^2 \\ &\quad + \frac{L}{2}\left(\frac{1}{L}\right)^2 \|\nabla f\|_2^2 \\ &\le f(x) - \frac{1}{2L}\|\nabla f\|_2^2 \end{aligned} \tag{7}$$

Hence, for the minimum value f^* of the objective function, we have:

$$f(x_{k+1}) - f^* \le \left(f(x_k) - f^*\right) - \frac{1}{2L}\|\nabla f\|_2^2 \tag{8}$$

Consider $f(y) = f(x) + \langle \nabla f, y - x\rangle + \frac{\mu}{2}\|y - x\|_2^2$ (fixing x) is a convex quadratic function of y. Hence, $f(y)$ minimizes when $\nabla f(y) = 0 \Leftrightarrow y = \tilde{y} = x - \frac{1}{\mu}\nabla f$. In addition, since $f(y) \ge f(x) + \langle \nabla f, y - x\rangle + \frac{\mu}{2}\|y - x\|_2^2,\ \forall x, y$, we have:

$$\begin{aligned} f(y) &\ge f(x) + \langle \nabla f, y - x\rangle + \frac{\mu}{2}\|y - x\| \\ &\ge f(x) + \langle \nabla f, \tilde{y} - x\rangle + \frac{\mu}{2}\|\tilde{y} - x\| \\ &= f(x) - \frac{1}{2\mu}\|\nabla f\|_2^2,\ \forall x, y \end{aligned} \tag{9}$$

Selecting $y = x^*$ and $x = x_k$ where x^* is the optimal solution, we have:

$$-\|\nabla f\|_2^2 \le 2\mu\left(f^* - f(x_k)\right) \tag{10}$$

From (8) and (10), we have the necessary result:

$$\begin{aligned} f_{k+1} - f^* &\le \left(1 - \frac{\mu}{L}\right)\left(f(x_k) - f^*\right) \\ &\le \left(1 - \frac{\mu}{L}\right)^k \left(f(x_0) - f^*\right) \end{aligned}$$

□

Lemma 1 *Consider* $\nabla^2 f$ *of* $f(x) = \frac{1}{2}x^T Qx + q^T x$, $\frac{1}{2}I \preceq \nabla^2 f \preceq \|Q\|_2 I \le nI$, *where* $\|Q\|_2 = \sqrt{\sum_{i=1}^n \sum_{j=1}^n Q_{ij}^2}$.

Proof We have $\nabla^2 f = Q$, and $\mathbf{a_i} = \frac{\mathbf{A_i}}{\|\mathbf{A_j}\|_2}$ $\frac{1}{2}x^T Ix \le \frac{1}{2}\left(\sum_{i=1}^n x_i^2\right) + \frac{1}{2}\left\|\sum_{i=1}^n x_i \mathbf{a_i}\right\|_2^2 = \sum_{i=1}^n \sum_{j=1}^n Q_{ij}x_i x_j = x^T Qx$ for $\forall x \Rightarrow \frac{1}{2}I \preceq \nabla^2 f$ since $Q_{ij} = \mathbf{a_i}\mathbf{a_j}$ and $Q_{ii} = \mathbf{a_i}\mathbf{a_i} = 1, \forall i, j$.

Moreover, based on Cauchy–Schwarz inequality, we have:

$$\begin{aligned} \left(\sum_{i=1}^n \sum_{j=1}^n Q_{ij}x_i x_j\right)^2 &\le \left(\sum_{i=1}^n \sum_{j=1}^n Q_{ij}^2\right)\left(\sum_{i=1}^n \sum_{j=1}^n (x_i x_j)^2\right) \\ \Rightarrow \sum_{i=1}^n \sum_{j=1}^n Q_{ij}x_i x_j &\le \sqrt{\|Q\|_2^2 \left(\sum_{i=1}^n {x_i}^2\right)^2} \\ \Leftrightarrow x^T Qx &\le \|Q\|_2 x^T Ix\ \forall x \\ \Leftrightarrow \nabla f = Q &\preceq \|Q\|_2 I \end{aligned}$$

Finally, $\|Q\|_2 = \sqrt{\sum_{i=1}^n \sum_{j=1}^n Q_{ij}^2} \le \sqrt{n^2} = n$ since $|Q_{ij}| \le 1,\ \forall i, j \Rightarrow \|Q\|_2 \le n$.

Therefore: $\frac{1}{2}I \preceq \nabla^2 f \preceq \|Q\|_2 I \le nI$. □

From Theorem 1 and Lemma 1 and by setting $\mu = \frac{1}{2}$ and $L = \|Q\|_2$, we have:

Lemma 2 *After* $k + 1$ *iterations,* $f(x^{k+1}) - f(x^*) \le (1 - \frac{\mu}{L})^k (f(x^0) - f(x^*))$, *and* μ, L *are always bounded as* $\frac{1}{2} \le \mu \le L \le n$, *where* n *is the dimension of* x. *Hence, the convergence rate of exact line search in Algorithm 1 is over-bounded as* $(1 - \frac{\mu}{L})^k \le (1 - \frac{1}{2n})^k$.

Moreover, because the greedy coordinate descent using Gauss–Southwell rule with exact optimization has convergence rate $(1 - \frac{\mu}{nL})$ for each update [17,20] and these updates is conducted $2n$ times, we have:

Theorem 2 *The convergence rate of Algorithm 1 is* $[(1 - \frac{\mu}{L})(1 - \frac{\mu}{nL})^{2n}]^k$ *in the subspace of passive variables, where* μ *and* L *are always bounded as* $\frac{1}{2} \le \mu \le L \le n$. *Algorithm 1 converges at over-bounded rate* $[(1 - \frac{\mu}{L})(1 - \frac{\mu}{nL})^{2n}]^k \le [(1 - \frac{1}{2n})(1 - \frac{1}{2n^2})^{2n}]^k$.

Proof Based on Section 3.2 in [20], the greedy coordinate descent using Gauss–Southwell rule, we have: $f(x^k - \frac{1}{L}\nabla_{i_k} f(x^k)) - f(x^*) \le (1 - \frac{1}{nL})(f(x^k) - f(x^*))$.

For using exact optimization, $f(x^{k+1}) - f(x^*) \le f(x^k - \frac{1}{L}\nabla_{i_k} f(x^k)) - f(x^*)$.

Hence, $f(x^{k+1}) - f(x^*) \le (1 - \frac{1}{nL})(f(x^k) - f(x^*))$. In other words, the convergence rate of each update in the greedy coordinate descent using Gauss–Southwell rule with exact optimization is $(1 - \frac{1}{nL})$.

Overall, Algorithm 1 including one exact line search and $2n$ updates of the greedy coordinate descent has convergence rate of $[(1 - \frac{\mu}{L})(1 - \frac{\mu}{nL})^{2n}]^k$. □

4.2 Complexity

Concerning the average complexity of Algorithm 1, we consider the important parts in each iteration:

Table 2 Summery of test cases

Dataset	d	n	Type
Synthetic 1	8000	10,000	$d < n$
Synthetic 2	15,000	10,000	$d > n$
Synthetic 3	15,000	10,000	Sparse 20% $\neq 0$
ILSVRC2013	61,188	10,000	$d > n$
CIFAR	3072	10,000	$d < n$
20-NEWS	61,185	10,000	Sparse

- The complexity of the exact line search is $\mathcal{O}(n^2 + n \cdot \text{nn}(n))$,
- The complexity of the greedy coordinate descent is $\mathcal{O}(n^2)$,
- The complexity of the accelerated search is $\mathcal{O}(n \cdot \text{nn}(n))$.

where $\text{nn}(n)$ is the number of negative elements in $(x + \alpha \triangle x)$ that is sparse. Therefore, if we consider computing $A^T A$ and $A^T b$ in $\mathcal{O}(dn + dn^2)$, we have:

Theorem 3 *The average complexity of Algorithm 1 is $\mathcal{O}(dn + dn^2 + \bar{k}n^2)$, where $\bar{k}$ is the number of iterations.*

5 Experimental evaluation

This section investigates the convergence of the gradient square over passive variables $\|\bar{f}\|_2^2$ and the objective value $\|Ax - b\|_2^2/2$ in comparison with state-of-the-art algorithms belonging to different research directions: block coordinate descent, accelerated methods, active set methods, and iterative methods. Furthermore, we employ the proposed accelerated anti-lopsided algorithm for a low-rank problem as nonnegative matrix factorization to investigate the effectiveness of the proposed algorithm for real applications.

Datasets: To investigate the effectiveness of the compared algorithms, 6 datasets are used and shown in Table 2:

For 3 synthetic datasets:

- the matrix A is randomly generated by the function $rand(d, n) \times 100$ for dense matrices, and $sprand(d, n, 0.1) \times 100$ for space matrices.

For 3 real datasets:

- 10,000 first images of ILSVRC2013[1] are extracted to form the matrix A, and the images in ILSVRC2013 are resized into the size [128 × 128] before converted into vectors of 61,188 dimensions.
- 10,000 first instances of CIFAR[2] are extracted to establish the matrix A,
- 10,000 first documents of 20-NEWS[3] are extracted to form the matrix A.

The number of variables is set by 10,000 because of the usually limited requirements of low-rank algorithms and the limitation of our algorithm in computing and storing $A^T A$. In addition, the observed vectors b are randomly generated and added noisy to guarantee that NNLS will have non-trivial optimal solutions. The 6 datasets, the generated dataset code and our source codes are published for in the link.[4]

Environment settings: we develop Algorithm 1 in MATLAB with embedded code C++ to compare them with other algorithms. For NNLS, we set system parameters to use only 1 CPU for MATLAB and the IO time is excluded in the machine 8-Core Intel Xeon E5 3 GHz. In addition, for evaluating on nonnegative matrix factorization, we set the parameter to use 8 CPUs.

5.1 Comparison with state-of-the-art algorithms

In this section, we investigate the convergence of the gradient square $\|\bar{f}\|_2^2$ over the passive set (see Fig. 4) and the objective values $\|Ax - b\|_2^2/2$ (See Fig. 5) during the running time (see Fig. 5). Specially, we compare our algorithm **Antilop** with state-of-the-art algorithms as follows:

- **Coord**: this is a cycle block coordinate descent algorithm [7] with fast convergence in practice [17].
- **Accer**: this is a Nesterov accelerated method with convergence rate $\mathcal{O}(\frac{L}{k^2})$ [10]. The source code is extracted from a module in the paper [10]. The source code is downloaded from.[5]
- **Fast**: this is a modified effective version of active set methods according to Bro R., de Jong S., Journal of Chemometrics, 1997 [2], which is developed by S. Gunn.[6] This algorithm can be considered as one of the fastest algorithms of active set methods.
- **Nm**: this is a non-monotonic fast method for large-scale nonnegative least squares based on iterative methods [13]. The source code is downloaded from.[7]
- **Nm**: this is frugal coordinate descent for large-scale NNLS. This code is provided by the author [19].

[1] http://image-net.org/challenges/LSVRC/2013/.

[2] http://www.cs.toronto.edu/~kriz/cifar.html.

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

[4] http://khuongnd.appspot.com.

[5] https://sites.google.com/site/nmfsolvers/.

[6] http://web.mit.edu/~mkgray/matlab/svm/fnnls.m.

[7] http://suvrit.de/work/progs/nnls.html.

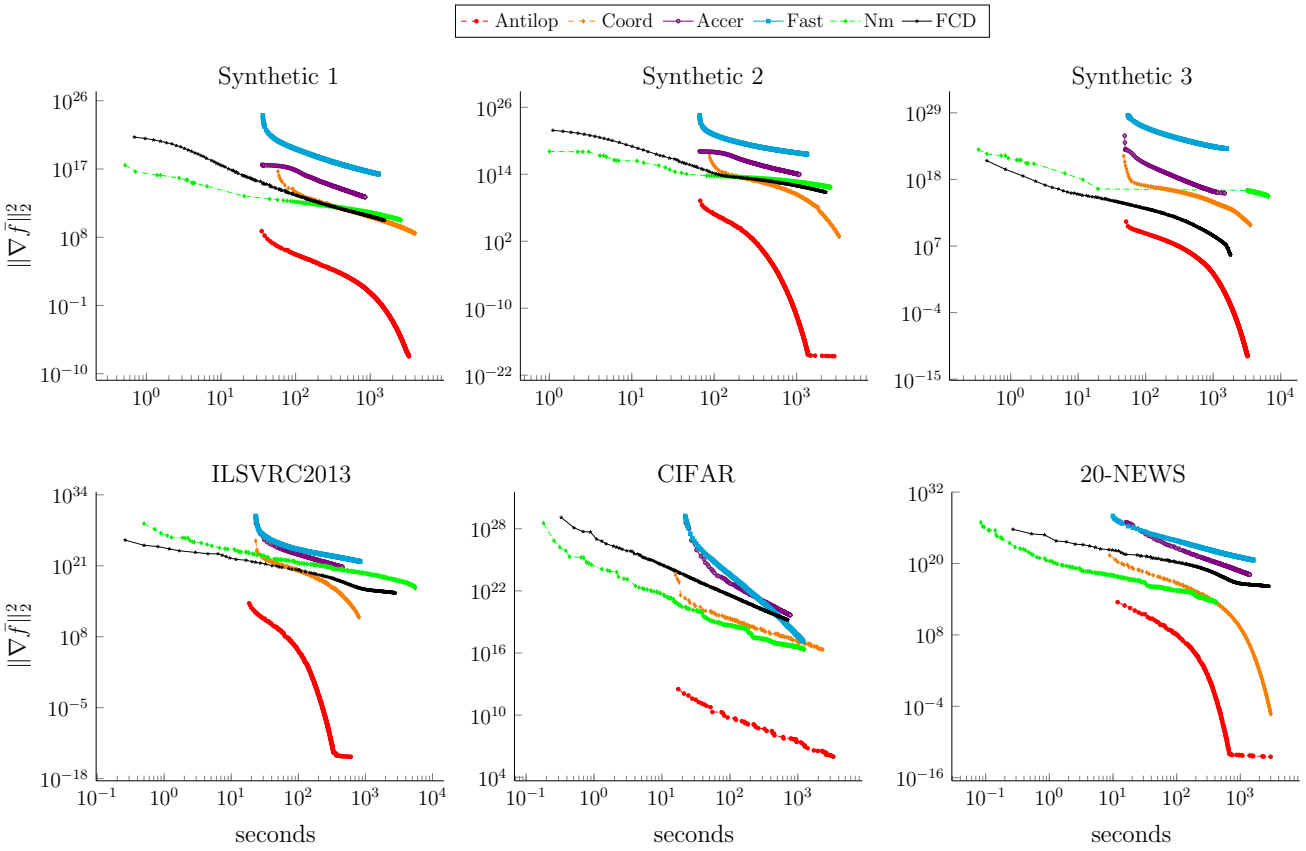

Fig. 4 Convergence of the gradient square $\|\bar{f}\|_2^2$ versus running time

Figures 4 and 5 show that the proposed algorithm with red color lines has the fastest convergence of the gradient square and the objective value:

– Concerning the convergence of the gradient square $\|\bar{f}\|_2^2$ over the passive variable set versus time in Fig. 4, the proposed algorithm has the fastest convergence over 6 datasets. At the beginning, the frugal block coordinate descent algorithm FCD [19] and the non-monotonic method Nm [13] have the fast approximation because they do not compute $A^T A$. However, for a long time, the faster convergence algorithms such as Antilop and Coord [7] will dominate, although they spend a long time on computing Hessian matrix $A^T A$. In comparison with Coord, the proposed algorithm Antilop converges much faster because Coord has zigzagging problems in optimization of multiple variable functions. For the accelerated algorithm Accer, its gradient square gradually reduces because the step size is limited in $\frac{1}{L}$. The active set method fast converges slowly because it has a high complexity at each iteration approximated to $\mathcal{O}(n^3)$ and handles a single active set simultaneously.

– Similarly, regarding the convergence of the objective value $\|Ax - b\|_2^2/2$ versus, in Fig. 5, the proposed algorithm has the fastest convergence over 6 datasets. At the beginning, the fast approximate algorithms FCD and Nm have faster convergence than the algorithms Antilop and Coord. However, Antilop and Coord more rapidly converge because they can detect the more appropriate direction to optimize the function. In comparison with Coord, the proposed algorithm Antilop converges much faster. For the accelerated algorithm Accer having convergence of $1/k^2$, the objective value gradually reduces because of its limited step size $\frac{1}{L}$ and negative effects of nonnegative constraints. In addition, the active set method fast converges slowly due to its high complexity.

These experimental results clearly indicate that the proposed algorithm has the fastest convergence of both two significant measures as the gradient square and the objective value, which are reasonable because the proposed algorithm combines several algorithms with different advantages to significantly reduce the negative effects of variable scaling problems, detect the more appreciate optimization directions, and attain the better theoretical guarantee.

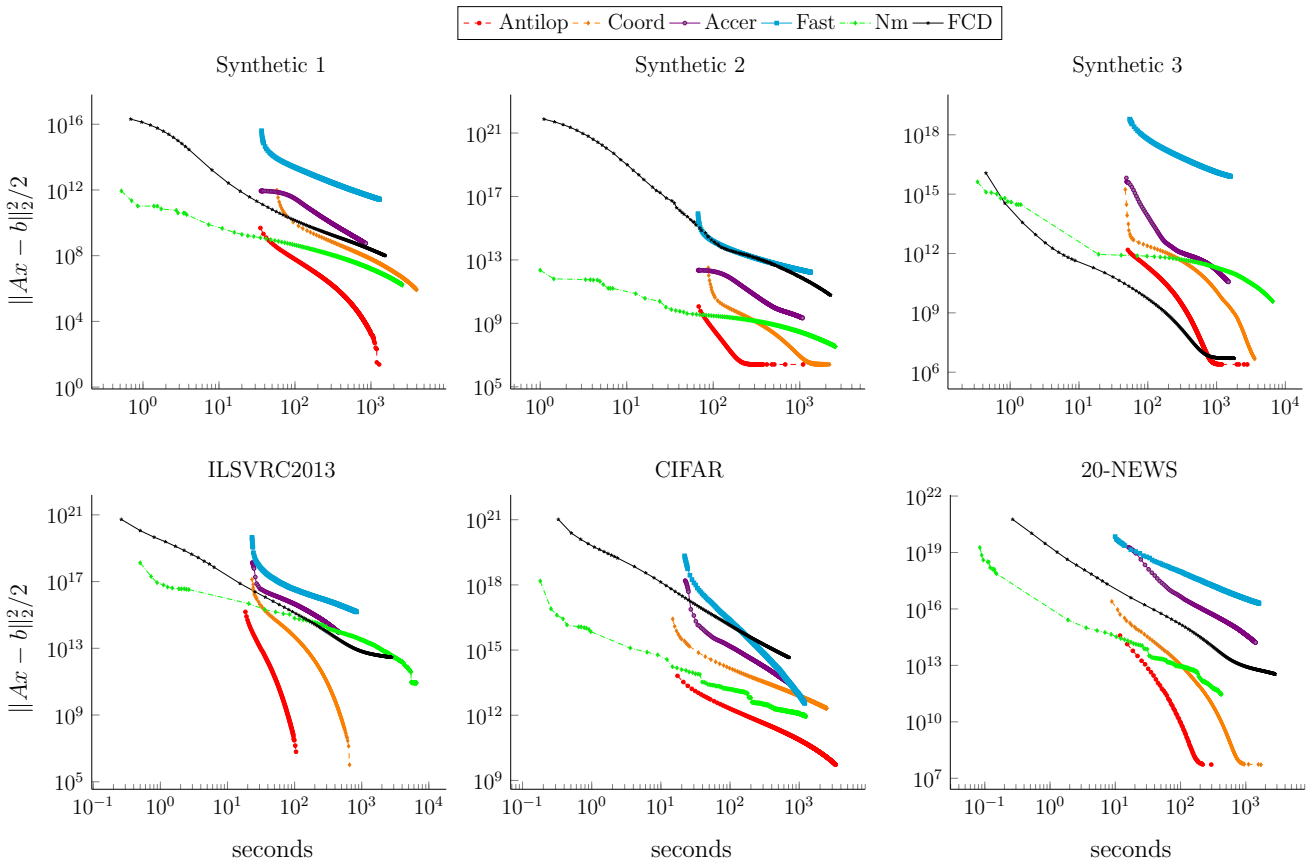

Fig. 5 Convergence of the objective value $\|Ax - b\|_2^2/2$ versus running time

Table 3 Summery of test cases for NMF

Dataset	n	m
CIFAR	3072	60,000
ILSVRC2013	61,188	60,000

5.2 Application for low-rank representation

NLSS is widely used as the core algorithm in low-rank representation, especially for nonnegative matrix and tensor factorization [23]. In these applications, fast convergence and high accuracy both are required. In this section, we investigate the effectiveness of the proposed algorithm for low-rank representation as nonnegative matrix factorization (NMF) on the large datasets CIFAR and ILSVRC2013 with different ranks $r = \{150, 200, 250\}$. For the datasets CIFAR and ILSVRC2013, 60,000 instances are employed as in Table 3.

In the NMF problem, a given nonnegative matrix V is factorized into the product of two matrix $V \approx WF$. For Frobenius norm, multiple iterative algorithm like EM algorithm is usually employed, which contains two main steps. In each step, one of the two matrices W or F is fixed to find the other optimal matrix. For example, when the matrix W is fixed, the new matrix is determined by $F \approx \operatorname{argmin}_{F\geq 0} \|V - WF\|_2^2 = \operatorname{argmin}_{F_i\geq 0} \sum_{i=1}^{r} \|V_i - WF_i\|_2^2$. Hence, a large number of NNLS problems must be approximately solved in NMF, and employing the proposed algorithm in NMF is a reasonable way to test its effectiveness.

We compare the algorithm NMF_Antilop using the proposed algorithm Antilop with state-of-the-art methods:

- NMF_Coord [11] using a greedy block coordinate descent method,
- NMF_HALS [9] using the cycle block coordinate descent method,
- NMF_Accer [10] using an accelerated method.

In comparison with state-of-the-art methods, the algorithm NMF_Antilop using the proposed NNLS algorithm converges much faster and has higher accuracy than the other algorithms in almost all test cases:

- Figure 6 shows the convergence of the objective value $\|V - WF\|_2^2/2$ versus running time. For the dataset CIFAR, the algorithm NMF_Antilop always converges

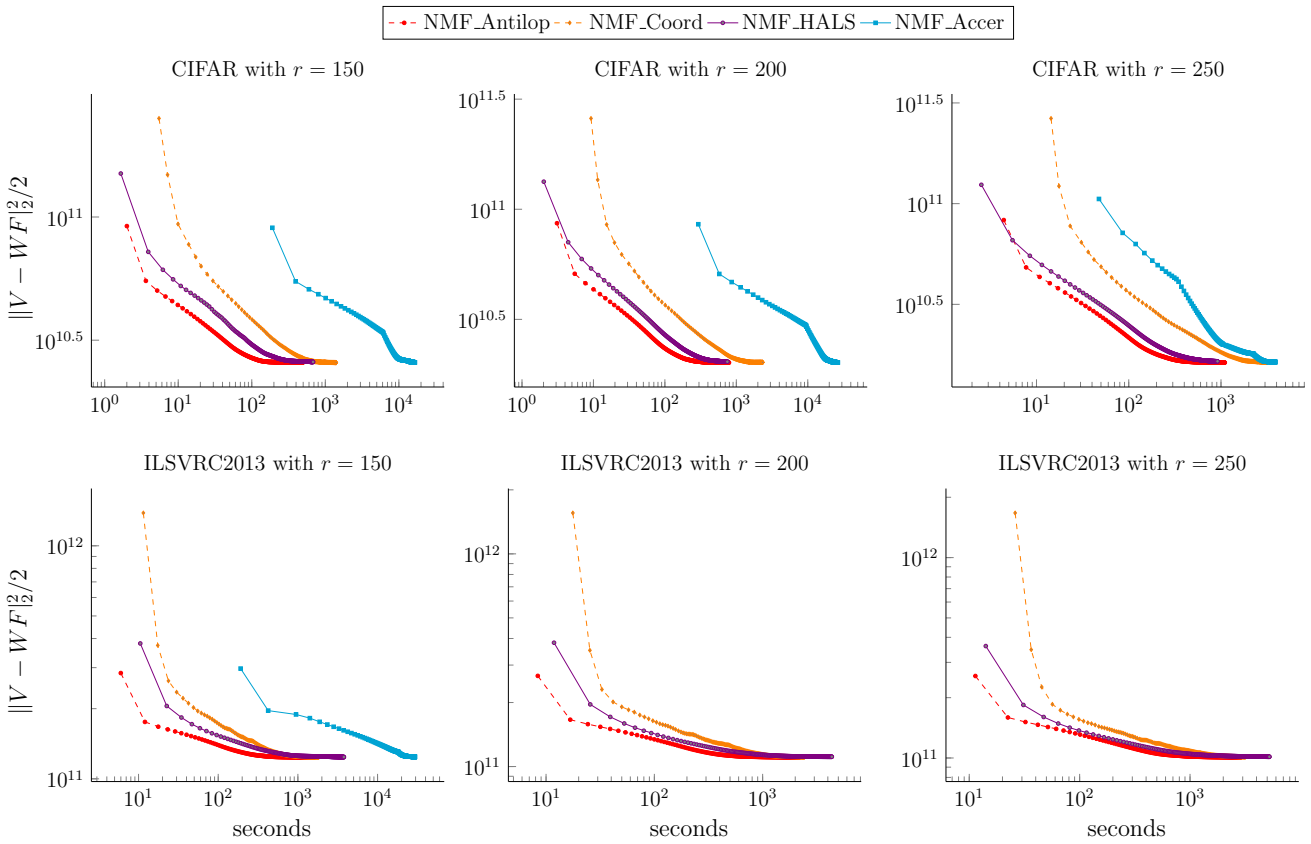

Fig. 6 Convergence of the objective value $\|Ax - b\|_2^2/2$ versus running time

faster than the other compared algorithms. For the dataset ILSVRC2013, NMF_Antilop converges slowly at the beginning. However, the algorithm NMF_Antilop has faster convergence rate than the other algorithms to obtain the lower objective values $\|V - WF\|_2^2/2$ at the ending time. In addition, the algorithm NMF_Accer has the slowest convergence and its results has been not reported for the dataset ILSVRC2013 with $r = \{200, 250\}$ because of its long running time.

– Moreover, Table 4 shows the objective values after 300 iterations of the multiple iterative algorithm. Based on the results, the algorithm NMF_Antilop has the highest accuracy in all the test cases. The results indicate that the proposed NNLS algorithm obtains higher accuracy than the other algorithms employed in NMF for the following reasons: first, NMF is a hard optimization problem within a large number of variables. It is difficult to reduce the objective value when the variables converge to the optimal solution, which is represented in Fig. 6. Second, algorithm Antilop has fast convergence with high accuracy to obtain the better objective values.

Hence, the results in Fig. 6 and Table 4 have shown that the proposed NNLS algorithm has both the fastest convergence and highest accuracy, which can be potentially applied to low-rank representation.

6 Conclusion and discussion

In the paper, we proposed an accelerated anti-lopsided algorithm to solve the nonnegative least squares problem as one of the most fundamental problems for low-rank representation. The proposed algorithm combines several algorithms and ideas, namely anti-lopsided variable transformation, exact line search, greedy block coordinate descent, and accelerated search to reduce the number of iterations and to increase the speed of the NNLS solver. These techniques aim to deal with variable scaling problems and nonnegative constraints of NNLS, although the combinational algorithm's iteration complexity increases several times. In addition, the proposed algorithm has over-bounded linear convergence rate $[(1 - \frac{\mu}{L})(1 - \frac{\mu}{nL})^{2n}]^k$ in the subspace of passive variables, where n is the dimension of solutions, and μ and L are always bounded as $\frac{1}{2} \leq \mu \leq L \leq n$.

In addition, we carefully compare the proposed algorithm with state-of-the-art algorithms in different research directions for both synthetic and real datasets. The results clearly

Table 4 $\|V - WH\|_2^2/2$ of NMF solvers after 300 iterations (unit: $\times 10^{10}$)

Method	NMF_Antilop	NMF_Coord	NMF_HALS	NMF_Accer
CIFAR $+ r = 150$	**2.565**	**2.565**	2.581	2.575
CIFAR $+ r = 200$	**2.014**	2.017	2.031	2.016
CIFAR $+ r = 250$	**1.625**	1.630	1.649	1.636
ILSVRC2013 $+ r = 150$	**12.390**	12.409	12.400	12.433
ILSVRC2013 $+ r = 200$	**11.070**	11.089	11.116	
ILSVRC2013 $+ r = 250$	**10.097**	10.127	10.141	

The most optimal values are shown in bold

shows that the proposed algorithm achieves the fastest convergence of the gradient square over passive variables and the objective value. Moreover, we investigate the effectiveness of the proposed algorithm in a real application of nonnegative matrix factorization, in which numerous NNLS problems must be approximately solved. The results also indicate that the NMF solver employing the proposed algorithm converges fastest and has the best accuracy in almost all the test cases.

Besides these advantages, our proposed algorithm still has several drawbacks such as computing and storing the Hessian matrix ($A^T A$). Fortunately, in low-rank representation, the Hessian matrix is computed only once time, and parallel threads can use the same shared memory. Hence, the proposed algorithm can potentially be applied for low-rank representation models with Frobenius norm. In the future researches, we will apply the proposed algorithm to low-rank representation problems, especially for nonnegative matrix and tensor factorization.

Acknowledgements This work was supported by 322 Scholarship from Vietnam Ministry of Education and Training, and Asian Office of Aerospace R&D under Agreement Number FA2386-15-1-4006.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
2. Bro, R., De Jong, S.: A fast non-negativity-constrained least squares algorithm. J. Chemom. **11**(5), 393–401 (1997)
3. Caramanis, L., Jo, S.J.: Ee 381v: large scale optimization fall 2012. http://users.ece.utexas.edu/~cmcaram/EE381V_2012F/Lecture_4_Scribe_Notes.final.pdf (2012)
4. Cevher, V., Becker, S., Schmidt, M.: Convex optimization for big data: scalable, randomized, and parallel algorithms for big data analytics. Sig. Process. Mag. IEEE **31**(5), 32–43 (2014)
5. Chen, D., Plemmons, R.J.: Nonnegativity constraints in numerical analysis. In: Symposium on the Birth of Numerical Analysis, pp. 109–140 (2009)
6. Dax, A.: On computational aspects of bounded linear least squares problems. ACM Trans. Math. Softw. (TOMS) **17**(1), 64–73 (1991)
7. Franc, V., Hlaváč, V., Navara, M.: Sequential coordinate-wise algorithm for the non-negative least squares problem. In: Gagalowicz, A., Philips, W (eds.) Proceedings of the 11th International Conference, CAIP 2005, Versailles, France, September 5-8, 2005. Computer Analysis of Images and Patterns, pp. 407–414. Springer, Berlin (2005)
8. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. 1981. Academic, London (1987)
9. Gillis, N., Glineur, F.: Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization. Neural Comput. **24**(4), 1085–1105 (2012)
10. Guan, N., Tao, D., Luo, Z., Yuan, B.: NeNMF: an optimal gradient method for nonnegative matrix factorization. IEEE Trans. Sig. Process. **60**(6), 2882–2898 (2012)
11. 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, pp. 1064–1072. ACM (2011)
12. Kim, D., Sra, S., Dhillon, I.S.: A new projected quasi-Newton approach for the nonnegative least squares problem. Computer Science Department, University of Texas at Austin (2006)
13. Kim, D., Sra, S., Dhillon, I.S.: A non-monotonic method for large-scale non-negative least squares. Optim. Methods Softw. **28**(5), 1012–1039 (2013)
14. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
15. Lawson, C.L., Hanson, R.J.: Solving Least Squares Problems, vol. 161. SIAM, Philadelphia (1974)
16. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $o(1/k^2)$. Sov. Math. Dokl. **27**, 372–376 (1983)
17. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. Core Discussion Papers 2010002, Université Catholique de Louvain. Center for Operations Research and Econometrics (CORE) (2010)
18. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. **1**(3), 123–231 (2013)
19. Potluru, V.K.: Frugal coordinate descent for large-scale NNLS. In: AAAI (2012)
20. Schmidt, M., Friedlander, M.: Coordinate descent converges faster with the Gauss–Southwell rule than random selection. In: NIPS OPT-ML Workshop (2014)
21. Van Benthem, M.H., Keenan, M.R.: Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. J. Chemom. **18**(10), 441–450 (2004)
22. Zhang, Z.-Y.: Nonnegative matrix factorization: models, algorithms and applications. In: Holmes, D.E., Jain, L.C. (eds.) Data Mining: Foundations and Intelligent Paradigms, vol. 2. Springer, Berlin (2012)
23. Zhou, G., Cichocki, A., Zhao, Q., Xie, S.: Nonnegative matrix and tensor factorizations: an algorithmic perspective. Sig. Process. Mag. IEEE **31**(3), 54–65 (2014)