Received: 21 September 2017 | Revised: 13 February 2019 | Accepted: 22 February 2019
DOI: 10.1002/nla.2237

RESEARCH ARTICLE

On greedy randomized coordinate descent methods for solving large linear least-squares problems

Zhong-Zhi Bai[1,2] | Wen-Ting Wu[1,2]

[1]State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China

[2]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing, China

Correspondence
Zhong-Zhi Bai, State Key Laboratory of Scientific/Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100190, China.
Email: bzz@lsec.cc.ac.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 11671393

Summary

For solving large scale linear least-squares problem by iteration methods, we introduce an effective probability criterion for selecting the working columns from the coefficient matrix and construct a greedy randomized coordinate descent method. It is proved that this method converges to the unique solution of the linear least-squares problem when its coefficient matrix is of full rank, with the number of rows being no less than the number of columns. Numerical results show that the greedy randomized coordinate descent method is more efficient than the randomized coordinate descent method.

KEYWORDS

convergence property, coordinate descent method, linear least-squares problem, randomized iteration

1 | INTRODUCTION

Let $A \in \mathbb{R}^{m\times n}$, with $m \geq n$, be a real $m\times n$ matrix of full column rank and $b \in \mathbb{R}^m$ be a real m-dimensional vector. Then the linear least-squares problem is to find a real n-dimensional vector $x_\star \in \mathbb{R}^n$ such that the function $\|b - Ax\|_2$ is minimized for all $x \in \mathbb{R}^n$, that is,

$$x_\star := \arg\min_{x\in\mathbb{R}^n} \|b - Ax\|_2^2.$$

Here and in the sequel, $\|\cdot\|_2$ indicates the Euclidean norm of either a vector or a matrix. Note that such a least-squares solution $x_\star$ exists and is unique, and has the analytical expression $x_\star = A^\dagger b$, with $A^\dagger = (A^TA)^{-1}A^T$ and A^T being the Moore–Penrose pseudoinverse and the transpose of matrix A, respectively (see the works of Björck,[1] and Golub and Van Loan[2]). We remark that in the sequel $(\cdot)^T$ is used to represent the transpose of either a vector or a matrix. It is well known that the linear least-squares problem is mathematically equivalent to the normal equation

$$A^TAx = A^Tb \tag{1}$$

and also equivalent to the unconstraint quadratic programming problem

$$\min_{x\in\mathbb{R}^n} f(x) := \frac{1}{2}x^TA^TAx - b^TAx. \tag{2}$$

One of the iteration methods that can be used to solve the linear least-squares problems economically and effectively is the coordinate descent method, which is actually the classical Gauss–Seidel iteration method directly applied to the normal equation in Equation (1) (see the work of Ruhe[3]). Indeed, in solving the optimization problems, the coordinate descent method enjoys a long history and has many applications in a broad spectrum of fields such as tomography,[4,5] protein structure,[6] machine learning,[7] biological feature selection,[8] and so on. Although there were some analyses about the convergence of the sequence of the gradients of the cost function evaluated at the iterates in the works of Censor and Lent,[9] and Hildreth,[10] whether the sequence of the iterates itself converges or whether it is even bounded had not been known until Luo and Tseng[11] gave a proof of linear convergence for smooth convex minimization in 1992. This approach was then generalized broadly, for example, to gradient projection methods, reduced gradient methods, and matrix splitting methods (see the works of Luo and Tseng[12,13]). However, none of them were proved to possess a convergence rate expressed explicitly in terms of a natural condition measure in linear algebra, for example, $\|A^{-1}\|_2$ or $\kappa_2(A) := \|A\|_2\|A^{-1}\|_2$, etc. Here $\|A^{-1}\|_2$ is defined to be the smallest positive constant η such that $\|Ax\|_2 \geq \frac{1}{\eta}\|x\|_2$ holds for all $x \in \mathbb{R}^n$, and $\kappa_2(A)$ is called the relative condition number of matrix A. In addition, this methodology of designing and analyzing the coordinate descent methods was also technically adopted to solve other kinds of minimization problems such as nonconvex minimization,[14] composite and linearly constrained convex minimization,[15,16] and variational inequalities,[17] etc.

Inspired by the work of Strohmer and Vershynin[18] who gave a bound for the convergence rate of the randomized Kaczmarz method in terms of a natural linear-algebraic condition measure[19] $\kappa_{F,2}(A) := \|A\|_F\|A^{-1}\|_2$ (called the scaled condition number), with $\|A\|_F$ being the Frobenius norm of matrix A, Leventhal and Lewis[20] found that by choosing a unit coordinate direction as a search direction randomly according to an appropriate probability distribution, a convergence rate for the resulting *randomized coordinate descent* (**RCD**) method can be obtained in terms of the scaled or the relative condition number $\kappa_{F,2}(A)$ or $\kappa_2(A)$. Of course, we have realized that a unified convergence theory for the RCD method was established in the work of Ma et al.[21] In addition, some other randomized versions of the coordinate descent method, which can be regarded as generalizations of the RCD method, were provided in the work of Nesterov,[22] and arguments about the randomized block versions of the coordinate descent method can be found in the works of Lu and Xiao,[23] Necoara et al.,[24] and Richtárik and Takáč.[25] For more discussions about a variety of randomized versions of the coordinate descent method, we refer to other works[26–28] and the references therein.

In the RCD method, at the kth iteration, the probability criterion

$$\Pr(\text{column} = j_k) = \frac{\|A_{(j_k)}\|_2^2}{\|A\|_F^2}, \quad j_k \in \{1, 2, \ldots, n\}, \tag{3}$$

is adopted to select the active or working column index j_k from the coefficient matrix $A \in \mathbb{R}^{m\times n}$, where $A_{(j_k)}$ represents the j_kth column of matrix A. However, if matrix A is scaled with a diagonal matrix that normalizes the Euclidean norms of all of its columns to be a same constant in advance, this criterion will become a uniform column sampling and the probabilities for selecting all columns will be the same.

In this paper, inspired by the recent work of Bai and Wu[29] which proposed a greedy randomized Kaczmarz method that converges much faster than the randomized Kaczmarz method, by adopting a different but more effective probability criterion that is capable of capturing larger entries of the residual vector with respect to the normal equation in Equation (1) at each iteration, we propose a *greedy randomized coordinate descent* (**GRCD**) method for solving the linear least-squares problem. Theoretical analysis shows that the GRCD method possesses a faster convergence rate than the RCD method in terms of natural linear-algebraic condition measures, and numerical computations exhibit that the former outperforms the latter significantly in both number of iteration steps and computing time.

The organization of this paper is as follows. After presenting necessary preliminaries and notation in Section 2, we derive and analyze the GRCD method in Section 3. In Section 4, we report and discuss the experimental results. Finally, in Section 5, we end this paper with concluding remarks.

2 | PRELIMINARIES AND NOTATION

For any vector $z \in \mathbb{R}^n$, $z^{(i)}$ represents its ith entry. We indicate by e_i the column vector with 1 at the ith position and 0 elsewhere. In addition, for a matrix $G = (g_{ij}) \in \mathbb{R}^{m\times n}$, $G_{(j)}$, $\|G\|_2 := \max_{x\neq 0}\frac{\|Gx\|_2}{\|x\|_2}$, and $\|G\|_F := \sqrt{\sum_{i=1}^m \sum_{j=1}^n |g_{ij}|^2}$ are used to denote its jth column, Euclidean norm, and Frobenius norm, respectively. Moreover, if matrix G is square, then $\mathrm{tr}(G)$ indicates its trace, and, if, in particular, G is symmetric, then $\lambda_{\min}(G)$ and $\lambda_{\max}(G)$ represent the smallest nonzero and the largest eigenvalue of G, respectively. Note that $\|G\|_F = \sqrt{\mathrm{tr}(G^TG)}$ holds for any matrix G. Given a symmetric positive-definite matrix $G \in \mathbb{R}^{n\times n}$, for any vector $x \in \mathbb{R}^n$ we define the corresponding energy norm, induced from the Euclidean norm, as $\|x\|_G := \sqrt{x^T G x}$.

Let $\mathbb{E}_k$ denote the expected value conditional on the first k iterations, that is,

$$\mathbb{E}_k[\cdot] = \mathbb{E}[\cdot \,|\, j_0, j_1, \dots, j_{k-1}],$$

where $j_l (l = 0, 1, \dots, k-1)$ is the column chosen at the lth iteration. Then, from the law of iterated expectations, we have $\mathbb{E}[\mathbb{E}_k[\cdot]] = \mathbb{E}[\cdot]$.

With respect to the optimization problem in Equation (2), at the kth iteration, by performing an exact linesearch in a given nonzero direction d_k, we can obtain the next iterate x_{k+1} by the formula

$$x_{k+1} = x_k + \alpha_k d_k,$$

where the step size α_k is chosen to be

$$\alpha_k = \arg\min_{\alpha\in\mathbb{R}} f(x_k + \alpha d_k),$$

so that it is given explicitly by

$$\alpha_k = \frac{d_k^T A^T (b - A x_k)}{d_k^T A^T A d_k}.$$

One natural choice of a set of easily computable search directions is to select d_k from the set of standard unit coordinate vectors, say, $\{e_1, e_2, \dots, e_n\}$. For example, if we let $d_k = e_{j_k}, j_k \in \{1, 2, \dots, n\}$, then the corresponding iteration scheme can be written as

$$x_{k+1} = x_k + \frac{A_{(j_k)}^T (b - A x_k)}{\|A_{(j_k)}\|_2^2} e_{j_k}, \quad k = 0, 1, 2, \dots .$$

When $j_k = (k \bmod n) + 1$, it is just the classical Gauss–Seidel method directly applied to the normal equation in Equation (1), which is known to be linearly convergent but with a rate not easily expressible in terms of typical matrix quantities (see, e.g., the works of Ruhe[3] and Quarteroni et al.[30]). Moreover, when j_k is determined according to the probability criterion in Equation (3), it gives the RCD method proposed by Leventhal and Lewis,[20] which is demonstrated to achieve a convergence rate in terms of the scaled or the relative condition number $\kappa_{F,2}(A)$ or $\kappa_2(A)$. This convergence result is precisely described in the following theorem, which can be found in Theorem 3.2 in the work of Leventhal and Lewis[20] and the estimate (3.10) in the work of Ma et al.[21]

Theorem 1 (See Leventhal and Lewis,[20] and Ma et al.[21]).
Consider the linear least-squares problem $\min_{x\in\mathbb{R}^n} \|b - Ax\|_2$, *where* $A \in \mathbb{R}^{m\times n}$ *is a rectangular matrix of full column rank,* $m \geq n$, *and* $b \in \mathbb{R}^m$ *is a given vector. Then, the following statements hold true.*

(i) The iteration sequence $\{x_k\}_{k=0}^{\infty}$, *generated by the RCD method starting from any initial guess* $x_0 \in \mathbb{R}^n$, *converges to the unique least-squares solution* $x_\star$ *in expectation.*
(ii) The solution error in expectation for the iteration sequence $\{x_k\}_{k=0}^{\infty}$ *satisfies*

$$\mathbb{E}\|x_k - x_\star\|_{A^TA}^2 \leq \left(1 - \frac{\lambda_{\min}(A^TA)}{\mathrm{tr}(A^TA)}\right)^k \|x_0 - x_\star\|_{A^TA}^2, \quad k = 1, 2, \dots .$$

This theorem implies that the RCD method may exhibit a fast convergence rate if matrix A is reasonably well conditioned with all of its singular values being far away from the origin. Besides, we remark that the quantity $\frac{\lambda_{\min}(A^TA)}{\mathrm{tr}(A^TA)}$ can be equivalently rewritten as $\frac{1}{[\kappa_{F,2}(A)]^2}$, which is inversely proportional to the square of the scaled condition number $\kappa_{F,2}(A)$ with respect to matrix A.

3 | THE GRCD ITERATION METHOD

In this section, we are going to construct the GRCD method and analyze its convergence property.

It is easily known that the least-squares solution $x_\star$ satisfying $A^TAx_\star = A^Tb$ is on the intersection of the hyperplanes $\{x \,|\, A_{(j)}^T A x = A_{(j)}^T b\}, j = 1, 2, \dots, n$, and $b - Ax_\star$ is on the intersection of the vertical spaces of $A_{(j)}, j = 1, 2, \dots, n$.

Note that at the kth iteration of the RCD method, we actually project the current iterate x_k to the hyperplane $\{x \,|\, A^T_{(j_k)} Ax = A^T_{(j_k)} b\}$ and the residual vector $r_k = b - Ax_k$ to the vertical space of $A_{(j_k)}$. In general, to compute the next iterate x_{k+1}, we may want that the angle between the residual vector r_k and the vertical space of the working column $A_{(j_k)}$ is relatively big. Besides, for any $i, j \in \{1, 2, \ldots, n\}$, when $|A^T_{(i)} r_k| > |A^T_{(j)} r_k|$, we may also want that the probability for selecting the ith column is larger than that for selecting the jth column. In this manner, we expect that the resulting randomized iteration method will possess a fast convergence rate. This is the main idea of the GRCD method. Here, we remark that in accordance with the above arguments it is reasonable for us to use the word "greedy" in the name of the GRCD method, as in computing the next iterate x_{k+1} we choose the working column locally based on the current kth iterate x_k with a certain probability criterion from only some of the columns that lead to the relatively largest entries in absolute value of the current residual vector $A^T r_k$.

In order to construct the GRCD method, we need to introduce the index set

$$\mathscr{V}_k = \left\{ j \,\middle|\, \left|A^T_{(j)} r_k\right|^2 \geq \delta_k \|A^T r_k\|_2^2 \|A_{(j)}\|_2^2 \right\}$$

associated with the current iterate x_k, where

$$\delta_k = \frac{\max\limits_{1 \leq j \leq n} \left(\frac{\left|A^T_{(j)} r_k\right|^2}{\|A_{(j)}\|_2^2} \right)}{2\|A^T r_k\|_2^2} + \frac{1}{2\|A\|_F^2}.$$

We claim that this set is not empty, that is, $\mathscr{V}_k \neq \emptyset$. As a matter of fact, if

$$\frac{\left|A^T_{(\ell)} r_k\right|^2}{\|A_{(\ell)}\|_2^2} = \max_{1 \leq j \leq n} \left(\frac{\left|A^T_{(j)} r_k\right|^2}{\|A_{(j)}\|_2^2} \right), \quad \text{for some} \quad \ell \in \{1, 2, \ldots, n\},$$

since

$$\frac{\max\limits_{1 \leq j \leq n} \left(\frac{\left|A^T_{(j)} r_k\right|^2}{\|A_{(j)}\|_2^2} \right)}{\sum\limits_{j=1}^{n} \frac{\|A_{(j)}\|_2^2}{\|A\|_F^2} \frac{\left|A^T_{(j)} r_k\right|^2}{\|A_{(j)}\|_2^2}} \geq 1,$$

or equivalently,

$$\frac{\max\limits_{1 \leq j \leq n} \left(\frac{\left|A^T_{(j)} r_k\right|^2}{\|A_{(j)}\|_2^2} \right)}{\|A^T r_k\|_2^2} \geq \frac{1}{\|A\|_F^2},$$

it holds that

$$\frac{\left(\frac{\left|A^T_{(\ell)} r_k\right|^2}{\|A_{(\ell)}\|_2^2} \right)}{\|A^T r_k\|_2^2} = \frac{\max\limits_{1 \leq j \leq n} \left(\frac{\left|A^T_{(j)} r_k\right|^2}{\|A_{(j)}\|_2^2} \right)}{\|A^T r_k\|_2^2} \geq \frac{\max\limits_{1 \leq j \leq n} \left(\frac{\left|A^T_{(j)} r_k\right|^2}{\|A_{(j)}\|_2^2} \right)}{2\|A^T r_k\|_2^2} + \frac{1}{2\|A\|_F^2} = \delta_k,$$

or in other words,

$$\frac{\left|A^T_{(\ell)} r_k\right|^2}{\|A_{(\ell)}\|_2^2} \geq \delta_k \|A^T r_k\|_2^2.$$

This implies that $\ell \in \mathscr{V}_k$, so that $\mathscr{V}_k \neq \emptyset$.

In fact, the definitions of $\mathscr{V}_k$ and δ_k show that if $i \in \mathscr{V}_k$, then we have

$$\begin{aligned}\frac{\left|A_{(i)}^T r_k\right|^2}{\|A_{(i)}\|_2^2} &\geq \frac{1}{2}\max_{1\leq j\leq n}\left(\frac{\left|A_{(j)}^T r_k\right|^2}{\|A_{(j)}\|_2^2}\right)+\frac{1}{2}\frac{\|A^T r_k\|_2^2}{\|A\|_F^2}\\ &= \frac{1}{2}\frac{\left|A_{(\ell)}^T r_k\right|^2}{\|A_{(\ell)}\|_2^2}+\frac{1}{2}\sum_{j=1}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\frac{\left|A_{(j)}^T r_k\right|^2}{\|A_{(j)}\|_2^2}\\ &= \frac{1}{2}\left(1+\frac{\|A_{(\ell)}\|_2^2}{\|A\|_F^2}\right)\frac{\left|A_{(\ell)}^T r_k\right|^2}{\|A_{(\ell)}\|_2^2}+\sum_{\substack{j=1\\ j\neq \ell}}^{n}\frac{1}{2}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\frac{\left|A_{(j)}^T r_k\right|^2}{\|A_{(j)}\|_2^2}.\end{aligned} \tag{4}$$

Furthermore, let θ_j be the angle between the vectors $A_{(j)}$ and r_k so that

$$\left|A_{(j)}^T r_k\right| = \|A_{(j)}\|_2\|r_k\|_2\cos\theta_j, \quad \text{with} \quad \theta_j \in \left[0, \frac{\pi}{2}\right], \qquad j = 1, 2, \dots, n.$$

Then, it holds that

$$\|r_k\|_2^2\cos^2\theta_i \geq \frac{1}{2}\left(1+\frac{\|A_{(\ell)}\|_2^2}{\|A\|_F^2}\right)\|r_k\|_2^2\cos^2\theta_\ell+\sum_{\substack{j=1\\ j\neq \ell}}^{n}\frac{1}{2}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\|r_k\|_2^2\cos^2\theta_j,$$

which implies that

$$\cos^2\theta_i \geq \frac{1}{2}\left(1+\frac{\|A_{(\ell)}\|_2^2}{\|A\|_F^2}\right)\cos^2\theta_\ell+\sum_{\substack{j=1\\ j\neq \ell}}^{n}\frac{1}{2}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\cos^2\theta_j.$$

It follows that $\cos^2\theta_i$ should be larger than a weighted combination of $\cos^2\theta_j, j = 1, 2, \dots, n$, with the weighting factors being $\frac{1}{2}(1+\frac{\|A_{(\ell)}\|_2^2}{\|A\|_F^2})$ for $\cos^2\theta_\ell$ and $\frac{1}{2}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}$ for $\cos^2\theta_j, j \neq \ell$. Consequently, the angle θ_i between r_k and $A_{(i)}$ should be relatively small or, in other words, the angle between r_k and the vertical space of $A_{(i)}$, which is equal to $\frac{\pi}{2}-\theta_i$, should be relatively big. Besides, if $\|A_{(j)}\|_2, j = 1, 2, \dots, n$, are all equal to a same constant, from the inequality in Equation (4) we observe that

$$\left|A_{(i)}^T r_k\right|^2 \geq \frac{1}{2}\left(1+\frac{1}{n}\right)\left|A_{(\ell)}^T r_k\right|^2+\frac{1}{2n}\sum_{\substack{j=1\\ j\neq \ell}}^{n}\left|A_{(j)}^T r_k\right|^2,$$

which means that $|A_{(i)}^T r_k|$ is required to be relatively large.

We let $s_k = A^T r_k$ and define $\tilde{s}_k$ as follows:

$$\tilde{s}_k^{(j)} = \begin{cases} s_k^{(j)}, & \text{if} \quad j \in \mathscr{V}_k,\\ 0, & \text{if} \quad j \notin \mathscr{V}_k.\end{cases}$$

Based on the above preparation, we now give the precise description of the GRCD method.

Method 1 (The GRCD method)

```
Input:  $A, b, \ell$ and $x_0$
Output: $x_\ell$
1: for $k = 0, 1, \dots, \ell - 1$ do
2:     Select $j_k \in \mathscr{V}_k$ with probability $\Pr(\text{column} = j_k) = \frac{|\tilde{s}_k^{(j_k)}|^2}{\|\tilde{s}_k\|_2^2}$
3:     Set $x_{k+1} = x_k + \frac{s_k^{(j_k)}}{\|A_{(j_k)}\|_2^2} e_{j_k}$
4: end for
```

In the GRCD method, the probability criterion is changing with respect to the iteration step. Moreover, as

$$\begin{aligned} A^T r_{k+1} &= A^T(b - Ax_{k+1}) \\ &= A^T \left[b - A\left(x_k + \frac{s_k^{(j_k)}}{\|A_{(j_k)}\|_2^2} e_{j_k} \right) \right] \\ &= A^T r_k - \frac{s_k^{(j_k)}}{\|A_{(j_k)}\|_2^2} (A^T A)_{(j_k)} \\ &:= A^T r_k - \frac{s_k^{(j_k)}}{\|A_{(j_k)}\|_2^2} B_{(j_k)}, \end{aligned}$$

with $B = A^T A$, it holds that

$$s_{k+1} = s_k - \frac{s_k^{(j_k)}}{\|A_{(j_k)}\|_2^2} B_{(j_k)}. \tag{5}$$

This shows that if matrix $B = A^T A$ can be computed economically at the beginning, we could make the GRCD method faster.

For the convergence property of the GRCD method, we can establish the following theorem.

Theorem 2. *Consider the linear least-squares problem* $\min_{x \in \mathbb{R}^n} \|b - Ax\|_2$, *where* $A \in \mathbb{R}^{m \times n}$ *is a rectangular matrix of full column rank,* $m \geq n$, *and* $b \in \mathbb{R}^m$ *is a given vector. Then, the following statements hold true.*

1. The iteration sequence $\{x_k\}_{k=0}^{\infty}$, *generated by the GRCD method starting from any initial guess* $x_0 \in \mathbb{R}^n$, *converges to the unique least-squares solution* $x_\star$ *in expectation.*

2. The solution error in expectation for the iteration sequence $\{x_k\}_{k=0}^{\infty}$ *satisfies*

$$\mathbb{E}\|x_1 - x_\star\|_{A^T A}^2 \leq \left(1 - \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2} \right) \|x_0 - x_\star\|_{A^T A}^2 \tag{6}$$

and

$$\mathbb{E}_k\|x_{k+1} - x_\star\|_{A^T A}^2 \leq \left(1 - \frac{\beta\,\lambda_{\min}(A^T A)}{\|A\|_F^2} \right) \|x_k - x_\star\|_{A^T A}^2, \quad k = 1, 2, \ldots, \tag{7}$$

where

$$\beta = \frac{1}{2}\left(\frac{1}{\gamma}\|A\|_F^2 + 1 \right) \quad \text{and} \quad \gamma = \|A\|_F^2 - \min_{1 \leq j \leq n} \|A_{(j)}\|_2^2.$$

As a result, it holds that

$$\mathbb{E}\,\|x_k - x_\star\|_{A^T A}^2 \leq \left(1 - \frac{\beta\,\lambda_{\min}(A^T A)}{\|A\|_F^2} \right)^{k-1} \left(1 - \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2} \right) \|x_0 - x_\star\|_{A^T A}^2. \tag{8}$$

Proof. From the equality in Equation (5), we see that

$$s_{k+1}^{(j_k)} = 0, \quad k = 0, 1, 2, \ldots.$$

Hence, for $k = 1, 2, \dots$, it holds that

$$\begin{aligned}
\delta_k \|A\|_F^2 &= \frac{\max\limits_{1\le j\le n}\left(\frac{\left|s_k^{(j)}\right|^2}{\|A_{(j)}\|_2^2}\right)}{2\sum\limits_{j=1}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\frac{|s_k^{(j)}|^2}{\|A_{(j)}\|_2^2}} + \frac{1}{2} \\
&= \frac{\max\limits_{1\le j\le n}\left(\frac{\left|s_k^{(j)}\right|^2}{\|A_{(j)}\|_2^2}\right)}{2\sum\limits_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\frac{\left|s_k^{(j)}\right|^2}{\|A_{(j)}\|_2^2}} + \frac{1}{2} \\
&\ge \frac{1}{2}\frac{\|A\|_F^2}{\sum\limits_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\|A_{(j)}\|_2^2} + \frac{1}{2} \\
&\ge \frac{1}{2}\left(\frac{\|A\|_F^2}{\gamma} + 1\right) = \beta.
\end{aligned}$$

Here, in deriving the second equality, we have used the fact that $s_k^{(j_{k-1})} = 0$. Besides, the first inequality is valid because from

$$\max_{1\le i\le n}\left(\frac{\left|s_k^{(i)}\right|^2}{\|A_{(i)}\|_2^2}\right) \ge \frac{\left|s_k^{(j)}\right|^2}{\|A_{(j)}\|_2^2}, \quad j = 1, 2, \dots, n,$$

we have

$$\begin{aligned}
\left(\sum_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\right)\max_{1\le i\le n}\left(\frac{\left|s_k^{(i)}\right|^2}{\|A_{(i)}\|_2^2}\right) &= \sum_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\left[\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\max_{1\le i\le n}\left(\frac{\left|s_k^{(i)}\right|^2}{\|A_{(i)}\|_2^2}\right)\right] \\
&\ge \sum_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\frac{\left|s_k^{(j)}\right|^2}{\|A_{(j)}\|_2^2},
\end{aligned}$$

which straightforwardly leads to the estimate

$$\frac{\max\limits_{1\le j\le n}\left(\frac{\left|s_k^{(j)}\right|^2}{\|A_{(j)}\|_2^2}\right)}{\sum\limits_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\frac{\left|s_k^{(j)}\right|^2}{\|A_{(j)}\|_2^2}} \ge \frac{1}{\sum\limits_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}} = \frac{\|A\|_F^2}{\sum\limits_{\substack{j=1\\ j\ne j_{k-1}}}^{n}\|A_{(j)}\|_2^2}.$$

Analogously, for $k = 0$, we can get

$$\delta_0 \|A\|_F^2 = \frac{\max\limits_{1\le j\le n}\left(\frac{\left|s_0^{(j)}\right|^2}{\|A_{(j)}\|_2^2}\right)}{2\sum\limits_{j=1}^{n}\frac{\|A_{(j)}\|_2^2}{\|A\|_F^2}\frac{\left|s_0^{(j)}\right|^2}{\|A_{(j)}\|_2^2}} + \frac{1}{2} \ge \frac{1}{2} + \frac{1}{2} = 1.$$

We further assert that, for any $k \ge 0$, the vector $A(x_{k+1} - x_\star)$ is perpendicular to the vector $A(x_{k+1} - x_k)$, so that the equality

$$\|A(x_{k+1} - x_\star)\|_2^2 = \|A(x_k - x_\star)\|_2^2 - \|A(x_{k+1} - x_k)\|_2^2 \tag{9}$$

holds true. In fact, from the update formula in the GRCD method, we see that

$$A(x_{k+1} - x_k) = \frac{s_k^{(j_k)}}{\|A_{(j_k)}\|_2^2} A_{(j_k)},$$

which implies that $A(x_{k+1} - x_k)$ is parallel to $A_{(j_k)}$. Because

$$\begin{aligned} A(x_{k+1} - x_\star) &= A\left(x_k - x_\star + \frac{A_{(j_k)}^T(b - Ax_k)}{\|A_{(j_k)}\|_2^2} e_{j_k} \right) \\ &= A(x_k - x_\star) + \frac{A_{(j_k)}^T A(x_\star - x_k)}{\|A_{(j_k)}\|_2^2} A_{(j_k)} \\ &= \left(I - \frac{A_{(j_k)} A_{(j_k)}^T}{\|A_{(j_k)}\|_2^2} \right) A(x_k - x_\star), \end{aligned}$$

it holds that

$$A_{(j_k)}^T [A(x_{k+1} - x_\star)] = A_{(j_k)}^T \left(I - \frac{A_{(j_k)} A_{(j_k)}^T}{\|A_{(j_k)}\|_2^2} \right) A(x_k - x_\star) = 0,$$

which implies that the vector $A(x_{k+1} - x_\star)$ is orthogonal to the vector $A_{(j_k)}$. It then follows straightforwardly that $A(x_{k+1} - x_\star)$ is perpendicular to $A(x_{k+1} - x_k)$.

Based on the equality in Equation (9), by taking the conditional expectation on both sides, we have

$$\begin{aligned} \mathbb{E}_k \|A(x_{k+1} - x_\star)\|_2^2 &= \|A(x_k - x_\star)\|_2^2 - \mathbb{E}_k \|A(x_{k+1} - x_k)\|_2^2 \\ &= \|A(x_k - x_\star)\|_2^2 - \sum_{j_k \in \mathscr{V}_k} \frac{\left|s_k^{(j_k)}\right|^2}{\sum\limits_{j \in \mathscr{V}_k} \left|s_k^{(j)}\right|^2} \frac{\left|s_k^{(j_k)}\right|^2}{\|A_{(j_k)}\|_2^2} \\ &\leq \|A(x_k - x_\star)\|_2^2 - \delta_k \|s_k\|_2^2. \end{aligned} \tag{10}$$

Here, the first equality uses the linearity of the conditional expectation and

$$\mathbb{E}_k \|A(x_k - x_\star)\|_2^2 = \|A(x_k - x_\star)\|_2^2$$

obtained from the definition of $\mathbb{E}_k[\cdot]$; the second equality is valid because when the first k iterates are fixed, at the kth iteration the probability of selecting the j_kth column of matrix A is equal to $\frac{|\tilde{s}_k^{(j_k)}|^2}{\|\tilde{s}_k\|_2^2}$, so from the definition of $\tilde{s}_k$ we have

$$\begin{aligned} \mathbb{E}_k \|A(x_{k+1} - x_k)\|_2^2 &= \mathbb{E}_k \left\| \frac{s_k^{(j_k)}}{\|A_{(j_k)}\|_2^2} A_{(j_k)} \right\|_2^2 \\ &= \sum_{j_k=1}^{n} \frac{\left|\tilde{s}_k^{(j_k)}\right|^2}{\|\tilde{s}_k\|_2^2} \frac{\left|s_k^{(j_k)}\right|^2}{\|A_{(j_k)}\|_2^2} \\ &= \sum_{j_k \in \mathscr{V}_k} \frac{\left|s_k^{(j_k)}\right|^2}{\sum\limits_{j \in \mathscr{V}_k} |s_k^{(j)}|^2} \frac{\left|s_k^{(j_k)}\right|^2}{\|A_{(j_k)}\|_2^2}; \end{aligned}$$

and the last inequality is achieved due to the definition of the index set $\mathscr{V}_k$, which shows that

$$\left|s_k^{(j_k)}\right|^2 \geq \delta_k \|s_k\|_2^2 \|A_{(j_k)}\|_2^2, \quad \text{if} \quad j_k \in \mathscr{V}_k,$$

and

$$\sum_{j_k \in \mathscr{V}_k} \frac{\left|s_k^{(j_k)}\right|^2}{\sum_{j \in \mathscr{V}_k} \left|s_k^{(j)}\right|^2} \frac{\left|s_k^{(j_k)}\right|^2}{\|A_{(j_k)}\|_2^2} \geq \sum_{j_k \in \mathscr{V}_k} \frac{\left|s_k^{(j_k)}\right|^2}{\sum_{j \in \mathscr{V}_k} \left|s_k^{(j)}\right|^2} \delta_k \|s_k\|_2^2 = \delta_k \|s_k\|_2^2.$$

Note that

$$\|A^T u\|_2^2 \geq \lambda_{\min}(A^T A)\, \|u\|_2^2$$

is valid for any vector u in the column space of A (see, e.g., the work of Bai and Wu[29]). Therefore, in accordance with Equation (10) we can further obtain

$$\begin{aligned} \mathbb{E}_k \|A(x_{k+1} - x_\star)\|_2^2 &\leq \|A(x_k - x_\star)\|_2^2 - \delta_k \, \|A^T A(x_k - x_\star)\|_2^2 \\ &\leq (1 - \delta_k \, \lambda_{\min}(A^T A))\, \|A(x_k - x_\star)\|_2^2. \end{aligned}$$

By making use of

$$\delta_0 \geq \frac{1}{\|A\|_F^2}$$

and

$$\delta_k \geq \frac{\beta}{\|A\|_F^2}, \quad k = 1, 2, \ldots,$$

we know that the GRCD method converges to the unique least-squares solution $x_\star$ in expectation, with the solution error in expectation having the reduction rates given in Equations (6) and (7).

In addition, by taking full expectation on both sides of Equation (7), from $\mathbb{E}[\mathbb{E}_k[\cdot]] = \mathbb{E}[\cdot]$ we see that

$$\mathbb{E}\|x_{k+1} - x_\star\|_{A^T A}^2 \leq \left(1 - \frac{\beta\, \lambda_{\min}(A^T A)}{\|A\|_F^2}\right) \mathbb{E}\|x_k - x_\star\|_{A^T A}^2.$$

Then, by induction on the iteration index k, we immediately get the estimate in Equation (8). □

Let the upper bounds for the convergence rates of the RCD and GRCD methods given in Theorems 1 and 2 be represented as

$$\rho_{\mathrm{RCD}} = 1 - \frac{\lambda_{\min}(A^T A)}{\|A\|_F^2}$$

and

$$\rho_{\mathrm{GRCD}} = 1 - \frac{\beta\, \lambda_{\min}(A^T A)}{\|A\|_F^2},$$

respectively. Then, based on an estimate for the constant β defined in Theorem 2, we can give a rigorous description about the relationship between the two quantities ρ_{RCD} and ρ_{GRCD}.

Corollary 1. *The following statements hold true:*

(i) $1 < \beta \leq \frac{2n-1}{2(n-1)}$;
(ii) $(1 + \frac{1}{2(n-1)})\rho_{\mathrm{RCD}} - \frac{1}{2(n-1)} \leq \rho_{\mathrm{GRCD}} < \rho_{\mathrm{RCD}}$.

Proof. We first verify the validity of (i). As

$$\gamma = \|A\|_F^2 - \min_{1 \leq j \leq n} \|A_{(j)}\|_2^2 < \|A\|_F^2,$$

we immediately see that

$$\beta = \frac{1}{2}\left(\frac{1}{\gamma}\|A\|_F^2 + 1\right) > 1.$$

On the other hand, as

$$\|A\|_F^2 = \sum_{j=1}^{n} \|A_{(j)}\|_2^2 \geq n \min_{1 \leq j \leq n} \|A_{(j)}\|_2^2,$$

we know that

$$\gamma \geq \left(1 - \frac{1}{n}\right) \|A\|_F^2.$$

Hence, it holds that

$$\beta \leq \frac{1}{2}\left(\frac{1}{\left(1 - \frac{1}{n}\right)\|A\|_F^2}\|A\|_F^2 + 1\right) = \frac{2n-1}{2(n-1)}.$$

Now, we turn to demonstrate statement (ii). Since

$$\begin{aligned}\rho_{\text{RCD}} - \rho_{\text{GRCD}} &= \left(1 - \frac{\lambda_{\min}(A^TA)}{\|A\|_F^2}\right) - \left(1 - \frac{\beta\,\lambda_{\min}(A^TA)}{\|A\|_F^2}\right)\\ &= (\beta - 1)\frac{\lambda_{\min}(A^TA)}{\|A\|_F^2},\end{aligned}$$

by making use of (i) we can straightforwardly obtain the estimate

$$0 < \rho_{\text{RCD}} - \rho_{\text{GRCD}} \leq \frac{1}{2(n-1)}\frac{\lambda_{\min}(A^TA)}{\|A\|_F^2},$$

which is exactly equivalent to the inequality in (ii). □

Remark 1. Corollary 1 shows that the upper bound derived in Theorem 2 for the convergence factor of the GRCD method is smaller, uniformly with respect to the iteration index, than that of the RCD method derived in the works of Leventhal and Lewis,[20] and Ma et al.[21] (see Theorem 1).

Remark 2. From Corollary 1, we see that the constant factor β is very close to 1, especially when the number n of the columns of the matrix $A \in \mathbb{R}^{m\times n}$ becomes large. As a result, the difference between the upper bounds of the RCD and GRCD methods given in Theorems 1 and 2 is marginal, particularly when n is large enough. However, these bounds are just the worst-case estimates, and there could be big gaps from them to the exact convergence factors (see Figures 1 and 2). In concrete applications, the GRCD method often shows much higher computational efficiency than the RCD method, implying that the actual convergence rate of the former is much faster than that of the latter. Admittedly, the numerical behavior of these methods is problem dependent, and is affected by many factors such as the sparsity structure, the condition number, and the algebraic properties of the matrix $A \in \mathbb{R}^{m\times n}$ in the linear least-squares problem $\min_{x\in\mathbb{R}^n}\|b - Ax\|_2^2$.

Remark 3. When $m < n$, the solution of the least-squares problem $\min_{x\in\mathbb{R}^n}\|b-Ax\|_2^2$ is not unique and can be written as $\tilde{x} = A^\dagger b + (I - P_{A^T})z$, $\forall z \in \mathbb{R}^n$, where P_{A^T} is the orthogonal projection onto the range space of A^T. We are often interested in the one that has the least norm, that is, $x_\star = A^\dagger b$. From an analogous demonstration, we can prove that $\{Ax_k\}_{k=0}^\infty$ converges to $Ax_\star$ for the iteration sequence $\{x_k\}_{k=0}^\infty$ generated by the GRCD method, whereas $\{x_k\}_{k=0}^\infty$

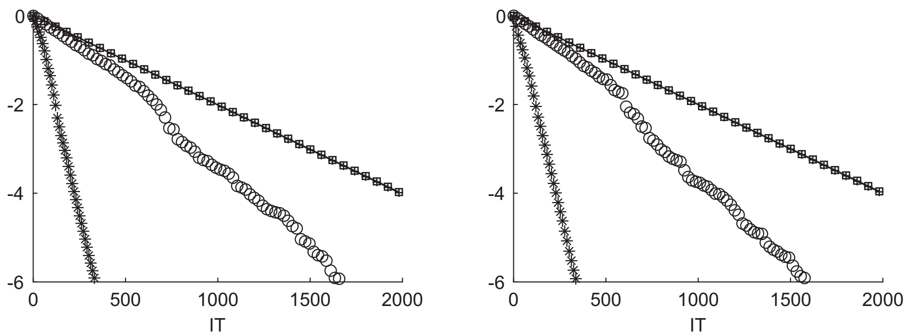

FIGURE 1 Pictures of $\log_{10}(\text{RSE})$ for RCD and GRCD, $\log_{10}(\rho_{\text{RCD}}) \times \text{IT}$ and $\log_{10}(\rho_{\text{GRCD}}) \times \text{IT}$ versus IT when $m = 5000, n = 150$, and the linear system is consistent (left) or inconsistent (right). $\log_{10}(\text{RSE})$ for RCD: "∘∘∘"; $\log_{10}(\text{RSE})$ for GRCD: "∗∗∗"; $\log_{10}(\rho_{\text{RCD}}) \times \text{IT}$: "− ⋄ −"; $\log_{10}(\rho_{\text{GRCD}}) \times \text{IT}$: "-+-"

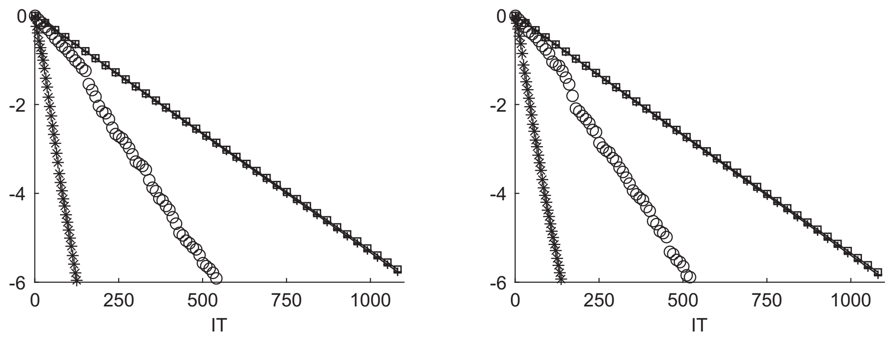

FIGURE 2 Pictures of $\log_{10}(\text{RSE})$ for RCD and GRCD, $\log_{10}(\rho_{\text{RCD}}) \times \text{IT}$ and $\log_{10}(\rho_{\text{GRCD}}) \times \text{IT}$ versus IT when $m = 1000, n = 50$, and the linear system is consistent (left) or inconsistent (right). $\log_{10}(\text{RSE})$ for RCD: "∘∘∘"; $\log_{10}(\text{RSE})$ for GRCD: "∗∗∗"; $\log_{10}(\rho_{\text{RCD}}) \times \text{IT}$: "− ⋄ −"; $\log_{10}(\rho_{\text{GRCD}}) \times \text{IT}$: "-+-"

itself does not converge to $x_\star$. In fact, the iterates of the GRCD method add the component, which is orthogonal to the column space of A^T, to the estimated vector, and do not eliminate it in minimizing the residual. The same argument is equally valid for the RCD method.

4 | EXPERIMENTAL RESULTS

We implement the GRCD and the RCD method for the matrices $A \in \mathbb{R}^{m\times n}$, which are either generated randomly by the MATLAB function `randn` subject to the standard normal distribution `N(0, 1)`, or taken directly from the work of Davis and Hu[31] demanding them to be sparse and of full rank originated in different applications such as combinatorial problem, DNA electrophoresis model, divorce laws in the 50 US states, and Pajek or world–city network, with the properties such as $m > n$ (e.g., abtaha1, Cities, divorce, and WorldCities) or $m = n$ (e.g., Trefethen_300 and cage5), as well as symmetric (e.g., Trefethen_300) or nonsymmetric (e.g., cage5).

We show that the GRCD method is more efficient than the RCD method in terms of the number of iteration steps (denoted as "IT") and the computing time in seconds (denoted as "CPU"). Here, IT and CPU are the medians of the required iteration steps and the elapsed CPU times with respect to 50 times repeated runs of the corresponding method. To give an intuitive demonstration of the advantage, we define the speed-up of the GRCD method against the RCD method as

$$\text{speed-up} = \frac{\text{CPU of RCD}}{\text{CPU of GRCD}}.$$

For the sparse matrices from the work of Davis and Hu,[31] we also present the density, defined as

$$\text{density} = \frac{\text{number of nonzeros of an } m \times n \text{ matrix}}{mn},$$

and the Euclidean condition number, denoted as $\text{cond}(A)$, in the numerical tables.

Note that if $m = n$, the matrix $A \in \mathbb{R}^{n\times n}$ is nonsingular, so that the corresponding linear least-squares problem $\min_{x\in\mathbb{R}^n} \|b - Ax\|_2^2$ and the linear system $Ax = b$ have the unique solution $A^{-1}b$. Moreover, if $m > n$, the corresponding linear system could be either consistent or inconsistent. The solution vector $x_\star$ is generated randomly by using the MATLAB function `randn` such that its entries obey the independent standard normal distribution. According to the right-hand side b, we take $b = Ax_\star$ when the linear system is consistent, and $b = Ax_\star + r_o$ when the linear system is inconsistent, where r_o is a nonzero vector belonging to the null space of A^T, say, $\text{null}(A^T)$, and $\text{null}(A^T)$ is generated by making use of the MATLAB function `null`.

In our implementations, the GRCD method is executed by exactly following the procedure defined in Method 1 and using the definitions of s_k and $\tilde{s}_k$ without explicitly forming the matrix $B = A^TA$. All computations are started from the initial vector $x_0 = 0$. The methods are terminated once the *relative solution error* (**RSE**), defined by

$$\text{RSE} = \frac{\|x_k - x_\star\|_2^2}{\|x_\star\|_2^2}$$

TABLE 1 IT and CPU of RCD and GRCD for $m \times n$ matrices A with $n = 50$ and different m when the linear system is consistent

$m \times n$		1000×50	2000×50	3000×50	4000×50	5000×50
RCD	IT	545.0	487.0	474.0	421.5	405.0
	CPU	0.0780	0.0936	0.1170	0.1404	0.1560
GRCD	IT	126.0	111.0	101.0	99.5	98.0
	CPU	0.0312	0.0312	0.0312	0.0468	0.0546
speed-up		2.50	3.00	3.75	3.00	2.86

TABLE 2 IT and CPU of RCD and GRCD for $m \times n$ matrices A with $n = 100$ and different m when the linear system is consistent

$m \times n$		1000×100	2000×100	3000×100	4000×100	5000×100
RCD	IT	1329.5	1135.0	993.0	983.0	987.5
	CPU	0.2496	0.2496	0.3588	0.3978	0.5148
GRCD	IT	344.5	268.5	232.0	226.0	216.5
	CPU	0.1092	0.1248	0.1482	0.1716	0.1950
speed-up		2.29	2.00	2.42	2.32	2.64

TABLE 3 IT and CPU of RCD and GRCD for $m \times n$ matrices A with $n = 150$ and different m when the linear system is consistent

$m \times n$		1000×150	2000×150	3000×150	4000×150	5000×150
RCD	IT	2405.5	1966.5	1795.0	1698.5	1676.0
	CPU	0.5460	0.4992	0.7332	0.9438	1.2558
GRCD	IT	583.5	433.5	395.5	368.5	336.0
	CPU	0.2184	0.2262	0.3120	0.3588	0.4758
speed-up		2.50	2.21	2.35	2.63	2.64

TABLE 4 IT and CPU of RCD and GRCD for $m \times n$ matrices A with $n = 50$ and different m when the linear system is inconsistent

$m \times n$		1000×50	2000×50	3000×50	4000×50	5000×50
RCD	IT	527.5	467.5	451.5	422.5	403.0
	CPU	0.0936	0.0936	0.1248	0.1248	0.1404
GRCD	IT	139.0	106.0	99.0	100.0	97.0
	CPU	0.0468	0.0468	0.0468	0.0624	0.0780
speed-up		2.00	2.00	2.67	2.00	1.80

at the current iterate x_k, satisfies RSE $< 10^{-6}$, or the number of iteration steps exceeds 200,000. The latter is given a label "−−" in the numerical tables. In addition, all experiments are carried out using MATLAB (version R2013a) on a personal computer with 2.83 GHz central processing unit (Intel^{R} Core$^{(TM)}$2 Quad CPU Q9550), 8.00 GB memory, and Windows operating system (Windows 7).

For the randomly generated matrices, we list the numbers of iteration steps and the computing times for both RCD and GRCD methods in Tables 1–3 when the linear system is consistent, and in Tables 4–6 when the linear system is inconsistent. From these tables, we see that the GRCD method vastly outperforms the RCD method in terms of both iteration counts and CPU times with significant speed-ups, regardless of whether the corresponding linear system is consistent or inconsistent. We observe that when the linear system is consistent, the speed-up is at least 2.00, and the biggest reaches 3.75; and when the linear system is inconsistent, the speed-up is at least 1.80, and the biggest reaches 2.82.

We can see the data intuitively in Figures 1–4. In Figures 1 and 2, we depict the curves of the RSE in base-10 logarithm, and the curves for the estimated upper bounds $\log_{10}(\rho_{\mathrm{RCD}}) \times \mathrm{IT}$ and $\log_{10}(\rho_{\mathrm{GRCD}}) \times \mathrm{IT}$ of the RSE in base-10 logarithm, versus the iteration step. We observe that the RSE of GRCD is decaying much more quickly than that of RCD when the iteration step is increasing, and the upper bounds about the convergence rates of the RCD and GRCD methods are both not tight, with that of the GRCD method being a little smaller than that of the RCD method. In Figures 3 and 4, we plot the curves of the iteration step and CPU time versus the number m of rows of the matrix $A \in \mathbb{R}^{m \times n}$ when the linear system is either consistent or inconsistent. We observe that the number of iteration steps for the GRCD method decreases, whereas the CPU time increases versus m for a fixed n.

TABLE 5 IT and CPU of RCD and GRCD for $m \times n$ matrices A with $n = 100$ and different m when the linear system is inconsistent

$m \times n$		1000 × 100	2000 × 100	3000 × 100	4000 × 100	5000 × 100
RCD	IT	1386.5	1176.0	1020.0	1030.5	1002.5
	CPU	0.2652	0.2808	0.3588	0.4368	0.5382
GRCD	IT	327.5	257.0	225.0	218.5	210.0
	CPU	0.1092	0.1092	0.1404	0.1638	0.1950
speed-up		2.43	2.57	2.56	2.67	2.76

TABLE 6 IT and CPU of RCD and GRCD for $m \times n$ matrices A with $n = 150$ and different m when the linear system is inconsistent

$m \times n$		1000 × 150	2000 × 150	3000 × 150	4000 × 150	5000 × 150
RCD	IT	2329.0	1901.5	1787.0	1659.5	1599.5
	CPU	0.4914	0.5148	0.7722	0.9672	1.1622
GRCD	IT	535.0	414.5	362.5	353.5	341.5
	CPU	0.1872	0.2028	0.2964	0.3432	0.4524
speed-up		2.63	2.54	2.61	2.82	2.57

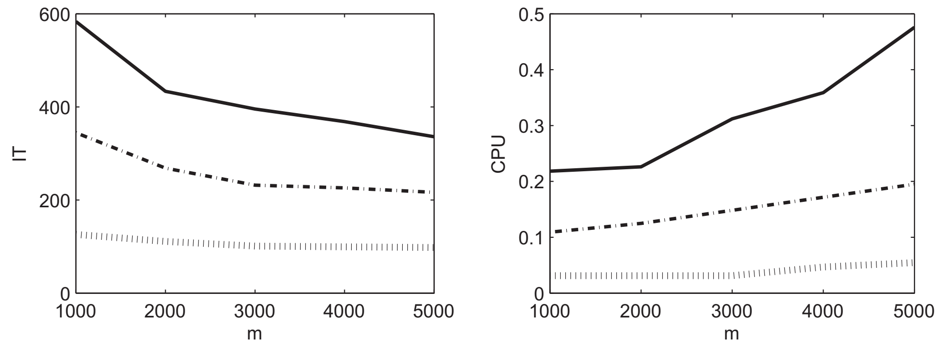

FIGURE 3 Pictures of IT (left) and CPU (right) versus m for GRCD when the linear system is consistent. $n = 50$: the dotted line "$\cdots$"; $n = 100$: the dash-dotted line "$- \cdot - \cdot -$"; $n = 150$: the solid line "—"

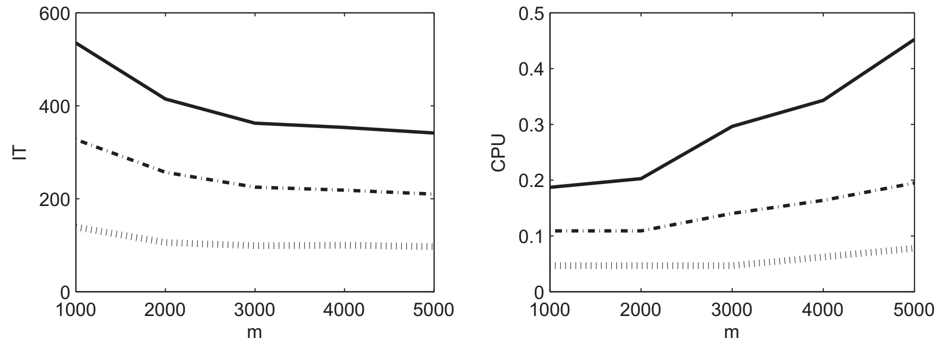

FIGURE 4 Pictures of IT (left) and CPU (right) versus m for GRCD when the linear system is inconsistent. $n = 50$: the dotted line "$\cdots$"; $n = 100$: the dash-dotted line "$- \cdot - \cdot -$"; $n = 150$: the solid line "—"

For the sparse full-rank matrices from the work of Davis and Hu,[31] we list the numbers of iteration steps and the computing times for both RCD and GRCD methods in Table 7 when the linear system is consistent, and in Table 8 when the linear system is inconsistent. We observe from these tables that for the matrices Cities and Trefethen_300 the RCD method cannot successfully compute an approximate solution to the linear least-squares problems, whereas the GRCD method can do so. In Table 7, the speed-up is at least 3.40, and the biggest reaches 7.38; and in Table 8, the speed-up is at least 3.67, and the biggest reaches 6.43. Hence, we see again that the GRCD method significantly outperforms the RCD method for both iteration counts and CPU times.

TABLE 7 IT and CPU of RCD and GRCD for $m \times n$ matrices A with different m and n when the linear system is consistent

name		abtaha1	Cities	divorce	WorldCities	Trefethen_300	cage5
$m \times n$		14596 × 209	55 × 46	50 × 9	315 × 100	300 × 300	37 × 37
density		1.68%	53.04%	50.00%	23.87%	5.20%	17.02%
cond(A)		12.23	207.15	19.39	66.00	1772.69	15.42
RCD	IT	97810.0	−−	2788.0	40316.0	−−	16784.0
	CPU	47.4399	−−	0.2652	4.4928	−−	0.7188
GRCD	IT	13658.0	49650.0	576.0	3607.0	1173.0	2205.0
	CPU	8.3695	7.1370	0.0780	0.6084	0.3281	0.1484
speed-up		5.67	−−	3.40	7.38	−−	4.84

TABLE 8 IT and CPU of RCD and GRCD for $m \times n$ matrices A with different m and n when the linear system is inconsistent

name		abtaha1	Cities	divorce	WorldCities
$m \times n$		14596 × 209	55 × 46	50 × 9	315 × 100
density		1.68%	53.04%	50.00%	23.87%
cond(A)		12.23	207.15	19.39	66.00
RCD	IT	103460.0	−−	3564.0	40429.0
	CPU	50.4273	−−	0.3432	4.4928
GRCD	IT	12957.0	46920.0	654.5	4452.5
	CPU	7.8469	6.7158	0.0936	0.7410
speed-up		6.43	−−	3.67	6.06

5 | CONCLUDING REMARKS

We have proposed a probability criterion for choosing a column from the coefficient matrix at each iteration step, and constructed a randomized iteration method, called the GRCD method, for solving the linear least-squares problems of full column ranks. We have also established the convergence theory for the GRCD method, and derived an estimate about its convergence rate. This probability criterion is much different from that adopted in the RCD method, and it leads to a faster convergence rate of the GRCD method in both theory and computations.

ACKNOWLEDGEMENTS

The authors are very much indebted to the referees for their constructive comments and valuable suggestions, which greatly improved the original manuscript of this paper. This paper was supported by the National Natural Science Foundation of China under Grant 11671393. There are no conflicts of interest to this work.

ORCID

Zhong-Zhi Bai https://orcid.org/0000-0001-8353-3803

REFERENCES

1. Björck Å. Numerical Methods for Least Squares Problems. Philadelphia, PA: SIAM; 1996.
2. Golub GH, Van Loan CF. Matrix Computations. 3rd ed. Baltimore, MD: Johns Hopkins University Press; 1996.
3. Ruhe A. Numerical aspects of Gram-Schmidt orthogonalization of vectors. Linear Algebra Appl. 1983;52–53:591–601.
4. Bouman CA, Sauer K. A unified approach to statistical tomography using coordinate descent optimization. IEEE Trans Image Process. 1996;5(3):480–492.
5. Ye JC, Webb KJ, Bouman CA, Millane RP. Optical diffusion tomography by iterative-coordinate-descent optimization in a Bayesian framework. J Opt Soc Am A. 1999;16(10):2400–2412.
6. Canutescu AA, Dunbrack Jr. RL. Cyclic coordinate descent: A robotics algorithm for protein loop closure. Protein Sci. 2003;12(5):963–972.
7. Chang K-W, Hsieh C-J, Lin C-J. Coordinate descent method for large-scale L2-loss linear support vector machines. J Mach Learn Res. 2008;9:1369–1398.
8. Breheny P, Huang J. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Ann Appl Stat. 2011;5(1):232–253.
9. Censor Y, Lent A. Optimization of "log x" entropy over linear equality constraints. SIAM J Control Optim. 1987;25(4):921–933.
10. Hildreth C. A quadratic programming procedure. Naval Res Logist Quart. 1957;4(1):79–85.

11. Luo Z-Q, Tseng P. On the convergence of the coordinate descent method for convex differentiable minimization. J Optim Theory Appl. 1992;72(1):7–35.
12. Luo Z-Q, Tseng P. Error bound and convergence analysis of matrix splitting algorithms for the affine variational inequality problem. SIAM J Optim. 1992;2(1):43–54.
13. Luo Z-Q, Tseng P. Error bound and reduced-gradient projection algorithms for convex minimization over a polyhedral set. SIAM J Optim. 1993;3(1):43–59.
14. Luo Z-Q, Tseng P. Error bounds and convergence analysis of feasible descent methods: A general approach. Ann Oper Res. 1993;46-47: 157–178.
15. Luo Z-Q, Tseng P. On the linear convergence of descent methods for convex essentially smooth minimization. SIAM J Control Optim. 1992;30(2):408–425.
16. Luo Z-Q, Tseng P. On the convergence rate of dual ascent methods for linearly constrained convex minimization. Math Oper Res. 1993;18(4):846–867.
17. Tseng P. On linear convergence of iterative methods for the variational inequality problem. J Comput Appl Math. 1995;60(1–2):237–252.
18. Strohmer T, Vershynin R. A randomized Kaczmarz algorithm with exponential convergence. J Fourier Anal Appl. 2009;15:262–278.
19. Demmel JW. The probability that a numerical analysis problem is difficult. Math Comput. 1988;50:449–480.
20. Leventhal D, Lewis AS. Randomized methods for linear constraints: Convergence rates and conditioning. Math Oper Res. 2010;35(3):641–654.
21. Ma A, Needell D, Ramdas A. Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods. SIAM J Matrix Anal Appl. 2015;36(4):1590–1604.
22. Nesterov Y. Efficiency of coordinate descent methods on huge-scale optimization problems. SIAM J Optim. 2012;22(2):341–362.
23. Lu Z, Xiao L. On the complexity analysis of randomized block-coordinate descent methods. Math Program. 2015;152(1–2):615–642.
24. Necoara I, Nesterov Y, Glineur F. Random block coordinate descent methods for linearly constrained optimization over networks. J Optim Theory Appl. 2017;173(1):227–254.
25. Richtárik P, Takáč M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math Program. 2014;144(1–2):1–38.
26. Nesterov Y, Stich SU. Efficiency of the accelerated coordinate descent method on structured optimization problems. SIAM J Optim. 2017;27(1):110–123.
27. Shalev-Shwartz S, Tewari A. Stochastic methods for ℓ_1-regularized loss minimization. J Mach Learn Res. 2011;12:1865–1892.
28. Wright SJ. Coordinate descent algorithms. Math Program. 2015;151(1):3–34.
29. Bai Z-Z, Wu W-T. On greedy randomized Kaczmarz method for solving large sparse linear systems. SIAM J Sci Comput. 2018;40(1):A592–A606.
30. Quarteroni A, Sacco R, Saleri F. Numerical Mathematics. 2nd ed. Berlin: Springer-Verlag; 2007.
31. Davis TA, Hu Y. The University of Florida Sparse Matrix Collection. ACM Trans Math Softw. 2011;38(1). Article 1, 25 pp.

How to cite this article: Bai Z-Z, Wu W-T. On greedy randomized coordinate descent methods for solving large linear least-squares problems. *Numer Linear Algebra Appl*. 2019;e2237. https://doi.org/10.1002/nla.2237