Fuzzy Systems and Data Mining IX
A.J. Tallón-Ballesteros and R. Beltrán-Barba (Eds.)


doi:10.3233/FAIA231071

# Two Count Sketch Kaczmarz Algorithms for Linear Systems

Yikang WANG[1], Pingping ZHANG
[a] *School of Science, Chongqing University of Posts and Telecommunications, Chongqing, 400065*

**Abstract.** In this paper, combining count sketch, we construct a count sketch two greedy subspace Kaczmarz (CS-2GSK) algorithm. In addition, the block count s-ketch Kaczmarz (BCSK) algorithm is also given, which selects multiple different rows simultaneously at each iteration. We prove that our methods converge to the unique solution of the linear systems. Finally, the numerical experiments show the high efficiency and robustness of our methods.

**Keywords.** Kaczmarz, count sketch, image reconstruction

## 1. Introduction

We only consider over-determined linear systems

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $m$ is much bigger than $n$, and $x$ is what we want to gain. In particular, we also require linear systems (1) to be consistent. The Kaczmarz method [1] is a common projection method for solving (1), which has been studied for many years, and its iterative scheme is

$$x_{k+1} = x_k + \frac{b_i - a_i^T x_k}{\| a_i \|^2} a_i.$$

In 2018, Bai and Wu [2] proposed the greedy randomized Kaczmarz(GRK) algorithm and Gu and Liu [3] extended the GRK algorithm to the 2GSK algorithm in 2020, and the theoretical guarantee the convergence of the 2GSK algorithm.

Numerous random sketching matrices [4–7] have been discovered by researchers over the years. The maximal weighted residual Kaczmarz [8] algorithm was used by Li and Zhang [9] in 2021, and they provided a CS-MWRK method for solving the systems (1). Numerical experiments demonstrated that the CS-MWRK method can effectively reduce computing time, but the number of iterations has significantly increased.

[1]Corresponding Author: Yikang Wang, male, Han, Chongqing, major in numerical algebra.: wyk10071996@163.com.

In this research, we combine the count sketch and 2GSK method, and we present the count sketch two greedy subspace Kaczmarz algorithm (CS-2GSK), which is motivated by the works of Zhang and Gu. Furthermore, in order to decrease the calculation times and iteration times of the CS-2GSK method, we suggest a block count sketch Kaczmarz algorithm (BCSK), which uses a selecting criterion to choose a block.

The main content of our work is structured as follows. Section 2 begins with several lemmas and the definition of count sketch. Sections 3 and 4 detail the proposed approaches as well as the accompanying convergence. Finally, in Section 5, we show the numerical results and followed the conclusion in Section 6.

## 2. Preliminaries

The following is about the definition and the property of count sketch, respectively. They have been given in detail in [4, 5, 10].

**Definition 2.1** A count sketch matrix $\mathbf{S} \in \mathbb{R}^{d\times m}$ is constructed in terms of $\Phi D$. Here, $D \in \mathbb{R}^{m\times m}$ is a random diagonal matrix. Besides, each diagonal entry of $D$ is chosen to be positive unity or minus one, and $\Phi$ is an $d \times m$ arbitrary matrix whose entries are either zeros or ones.

**Lemma 2.2** If $\mathbf{S}$ is a count sketch matrix defined in **Definition 2.1** with $(n^2 + n)/(\delta\varepsilon^2)$ rows, where $\delta$ is a positive constant and $\varepsilon$ is a normal number less than 1, then we have

$$(1-\varepsilon)||Ax||_2 \leq ||\mathbf{S}Ax||_2 \leq (1+\varepsilon)||Ax||_2.$$

## 3. The Count Sketch two greedy subspace Kaczmarz algorithm

Given a sufficiently large positive integer $\ell$ and initial point $x_0$, then, the specific process of the CS-2GSK algorithm is in **Algorithm 2**.

**Algorithm 1** The CS-2GSK algorithm

1. Input: $A$, $b$, $\ell$ $x_0$.
2. Output: $x_\ell$.
3. Initialize: Create $\mathbf{S}$, compute $\widetilde{A} = \mathbf{S}A$ and $\widetilde{b} = \mathbf{S}b$.
4. **for** $k = 0, 1, ..., \ell - 1$ **do**
   (a) Compute $r_k = \widetilde{b} - \widetilde{A}x_k$
   (b) Select rows $i_k$ and $j_k$ that satisfy

$$i_k = arg \max_{1 \le i \le d} \{ \frac{|\widetilde{b}^{(i)} - \widetilde{A}^{(i)} x_k|}{\|\widetilde{A}^{(i)}\|_2^2} \}, \ j_k = arg \max_{i \in [m] \setminus i_k} \{ \frac{|\widetilde{b}^{(i)} - \widetilde{A}^{(i)} x_k|}{\|\widetilde{A}^{(i)}\|_2^2} \}.$$

   (c) Set

$$x_{k+1} = x_k + \frac{r_k^{(i_k)} (\widetilde{A}^{(i_k)})^T}{\| \widetilde{A}^{(i_k)} \|_2^2} + \frac{r_k^{(j_k)} (\widetilde{A}^{(j_k)})^T}{\| \widetilde{A}^{(j_k)} \|_2^2}.$$

5. **end for**.

Regarding the convergence of the CS-2GSK method, the **Theorem 3.1** is given.

**Theorem 3.1** Let each row of $A \in \mathbb{R}^{m \times n}$ not be orthogonal to each other. Solving linear system (1) by CS-2GSK method can produce a series of sequence $\{x_k\}_{k=0}^{\infty}$. Then the sequence $\{x_k\}_{k=0}^{\infty}$ converges to the unique solution of $Ax = b$. Moreover, the solution error obeys

$$\| x_{k+1} - x_\star \|_2^2 \le \prod_{q=0}^{k} (1 - \lambda_{\min}(P_q P_q^T)) \| x_0 - x_\star \|_2^2, \ k = 0, 1, 2, \dots,$$

where $P_k = P_{i_k} - P_{j_k}$ with

$$P_{i_k} = \frac{(\widetilde{A}^{(i_k)})^T}{\| \widetilde{A}^{(i_k)} \|_2} \cdot \frac{\widetilde{A}^{(i_k)}}{\| \widetilde{A}^{(i_k)} \|_2} \ and \ P_{j_k} = \frac{(\widetilde{A}^{(j_k)})^T}{\| \widetilde{A}^{(j_k)} \|_2} \cdot \frac{\widetilde{A}^{(j_k)}}{\| \widetilde{A}^{(j_k)} \|_2},$$

where $\lambda_{\min}(P_q P_q^T)$ denotes the minimum eigenvalue of $P_q P_q^T$.

**Proof.** In fact, we only let

$$P_{i_k} = \frac{(\widetilde{A}^{(i_k)})^T}{\| \widetilde{A}^{(i_k)} \|_2} \cdot \frac{\widetilde{A}^{(i_k)}}{\| \widetilde{A}^{(i_k)} \|_2} \ and \ P_{j_k} = \frac{(\widetilde{A}^{(j_k)})^T}{\| \widetilde{A}^{(j_k)} \|_2} \cdot \frac{\widetilde{A}^{(j_k)}}{\| \widetilde{A}^{(j_k)} \|_2}.$$

The rest of the proof can be completed along the same lines as in [3, Theorem 1], we omit the details. □

## 4. The block count sketch Kaczmarz algorithm

The basic goal of the block count sketch Kaczmarz (BCSK) algorithm is to gather all of the indices that are closest to the greatest entry as the $i_k$ and $j_k$ in the CS-2GSK method. Specifically, we select the index set by

$$|r_k^{(j_k)}|^2 \geq \alpha \underset{1\leq j\leq n}{max} |r_k^{(j)}|^2,\ j_k \in \{1,2,...,n\}.\ \alpha \in [0,1)$$

It's clear that BCSK algorithm is retrieving more rows as $\alpha$ increases.

Below, we give the specific process of the BCSK algorithm in **Algorithm 2**.

---

**Algorithm 2** The BCSK algorithm

---

1. Input: $A$, $b$, $\alpha$, $\ell$, $x_0$.
2. Output: $x_\ell$.
3. Initialize: Create $\mathbf{S}$, compute $\widetilde{A} = \mathbf{S}A$ and $\widetilde{b} = \mathbf{S}b$.
4. For $k = 0,1,...,\ell-1$ **do**
   (a) Compute $r_k = \widetilde{b} - \widetilde{A}x_k$
   (b) Compute target block $\tau_k = \{j_k|\ |r_k^{(j_k)}|^2 \geq \alpha \underset{1\leq j\leq n}{max} |r_k^{(j)}|^2\}$
   (c) Set
   $$x_{k+1} = x_k + \widetilde{A}_{\tau_k}^{\dagger} r_k.$$
5. **end for**.

---

**Remark 4.1** In practice, the **CGLS** [11–13] can be used to calculate the approximation of $\widetilde{A}_{\tau_k}^{\dagger}$ in the fourth step of **Algorithm 2**.

Next, we will show the convergence of the BCSK method by **Theorem 4.2**.

**Theorem 4.2** Let the condition in **Theorem 3.1** be satisfied. Solving linear system (1) by BCSK method can produce a series of sequence $\{x_k\}_{k=0}^{\infty}$. Then the sequence $\{x_k\}_{k=0}^{\infty}$ will converge to the unique solution of (1). Moreover, the solution error obeys

$$\| x_{k+1} - x_\star \|_2^2 \leq \prod_{q=0}^{k} (1 - \frac{\alpha|\tau_q|}{m - |\tau_{q-1}|} \cdot \frac{(1-\varepsilon)^2 \lambda_{\min}(A^T A)}{(1+\varepsilon)^2 \sigma_{\max}^2(A)}) \| x_0 - x_\star \|_2^2 . \tag{2}$$

where $|\tau_q|$ represents the cardinality of $\tau_q$ and $\alpha \in (0,1]$.

**Proof.** From **Algorithm 2**, we have

$$\begin{aligned} x_{k+1} - x_\star &= x_k - x_\star + \widetilde{A}_{\tau_k}^{\dagger}(\widetilde{b}_{\tau_k} - \widetilde{A}_{\tau_k} x_k) \\ &= x_k - x_\star - \widetilde{A}_{\tau_k}^{\dagger}\widetilde{A}_{\tau_k}(x_k - x_\star) \\ &= (I - \widetilde{A}_{\tau_k}^{\dagger}\widetilde{A}_{\tau_k})(x_k - x_\star). \end{aligned}$$

Since $\widetilde{A}_{\tau_k}^{\dagger}\widetilde{A}_{\tau_k}$ is an orthogonal projector, we get

$$\begin{aligned} \| x_{k+1} - x_\star \|_2^2 &= \| (I - \widetilde{A}_{\tau_k}^{\dagger}\widetilde{A}_{\tau_k})(x_k - x_\star) \|_2^2 \\ &= \| x_k - x_\star \|_2^2 - \| \widetilde{A}_{\tau_k}^{\dagger}\widetilde{A}_{\tau_k}(x_k - x_\star) \|_2^2 \end{aligned} \tag{3}$$

As explained in [2], we have

$$\| A(x_k - x_\star) \|_2^2 \geq \sigma_r^2(A) \| x_k - x_\star \|_2^2,$$

which leads to

$$\begin{aligned} \| \widetilde{A}_{\tau_k}^\dagger \widetilde{A}_{\tau_k}(x_k - x_\star) \|_2^2 &\geq \frac{1}{\sigma_{\max}^2(\widetilde{A}_{\tau_k})} \| \widetilde{A}_{\tau_k}(x_k - x_\star) \|_2^2 \\ &\geq \frac{1}{(1+\varepsilon)^2 \sigma_{\max}^2(A)} \| \widetilde{A}_{\tau_k}(x_k - x_\star) \|_2^2 \\ &= \frac{1}{(1+\varepsilon)^2 \sigma_{\max}^2(A)} \sum_{i_k \in \tau_k} |\widetilde{A}^{(i_k)}(x_k - x_\star)|^2 \\ &\geq \frac{1}{(1+\varepsilon)^2 \sigma_{\max}^2(A)} \sum_{i_k \in \tau_k} \alpha \max_{1 \leq i \leq m} |\widetilde{A}^{(i)}(x_k - x_\star)|^2 \\ &\geq \frac{|\tau_k|}{(1+\varepsilon)^2 \sigma_{\max}^2(A)} \alpha \max_{1 \leq i \leq m} |\widetilde{A}^{(i)}(x_k - x_\star)|^2. \end{aligned} \tag{4}$$

From the iteration of the BCSK, we have

$$\begin{aligned} (\widetilde{b} - \widetilde{A}x_{k+1})_{\tau_k} &= \widetilde{b}_{\tau_k} - \widetilde{A}_{\tau_k} x_k - \widetilde{A}_{\tau_k} \widetilde{A}_{\tau_k}^\dagger (\widetilde{b}_{\tau_k} - \widetilde{A}_{\tau_k} x_k) \\ &= \widetilde{b}_{\tau_k} - \widetilde{A}_{\tau_k} x_k - \widetilde{A}_{\tau_k} \widetilde{A}_{\tau_k}^\dagger \widetilde{b}_{\tau_k} + \widetilde{A}_{\tau_k} \widetilde{A}_{\tau_k}^\dagger \widetilde{A}_{\tau_k} x_k \\ &= \widetilde{b}_{\tau_k} - \widetilde{A}_{\tau_k} x_k - \widetilde{A}_{\tau_k} \widetilde{A}_{\tau_k}^\dagger \widetilde{b}_{\tau_k} + \widetilde{A}_{\tau_k} x_k \\ &= \widetilde{b}_{\tau_k} - \widetilde{A}_{\tau_k} \widetilde{A}_{\tau_k}^\dagger \widetilde{b}_{\tau_k} \\ &= \widetilde{A}_{\tau_k} x_\star - \widetilde{A}_{\tau_k} \widetilde{A}_{\tau_k}^\dagger \widetilde{A}_{\tau_k} x_\star \\ &= \widetilde{A}_{\tau_k} x_\star - \widetilde{A}_{\tau_k} x_\star \\ &= 0, \end{aligned}$$

which implies the inequality

$$\| \widetilde{b} - \widetilde{A}x_k \|_2^2 = \sum_{i \in [m] \setminus \tau_{k-1}} |\widetilde{b}^{(i)} - \widetilde{A}^{(i)} x_k|^2 \leq (m - |\tau_{k-1}|) \max_{1 \leq i \leq m} |\widetilde{b}^{(i)} - \widetilde{A}^{(i)} x_k|^2.$$

Hence, for any $\alpha \in (0,1]$, combining **Lemma 2.2**, we have

$$\max_{1 \leq i \leq m} |\widetilde{b}^{(i)} - \widetilde{A}^{(i)} x_k|^2 \geq \frac{\alpha}{m - |\tau_{k-1}|} \| \widetilde{A}(x_k - x_\star) \|_2^2 \geq \frac{(1-\varepsilon)^2 \alpha}{m - |\tau_{k-1}|} \| A(x_k - x_\star) \|_2^2 .$$

As a result, we see that

$$\begin{aligned} \| \widetilde{A}_{\tau_k}^\dagger \widetilde{A}_{\tau_k}(x_k - x_\star) \|_2^2 &\geq \frac{|\tau_k|}{m - |\tau_{k-1}|} \cdot \frac{(1-\varepsilon)^2 \alpha}{(1+\varepsilon)^2 \sigma_{\max}^2(A)} \| A(x_k - x_\star) \|_2^2 \\ &\geq \frac{\alpha |\tau_k|}{m - |\tau_{k-1}|} \cdot \frac{(1-\varepsilon)^2 \lambda_{\min}(A^T A)}{(1+\varepsilon)^2 \sigma_{\max}^2(A)} \| x_k - x_\star \|_2^2 . \end{aligned} \tag{5}$$

Thus, substituting Eqs.(5) and (4) into Eq.(3), we obtain

$$\| x_{k+1} - x_\star \|_2^2 \leq (1 - \frac{\alpha|\tau_k|}{m - |\tau_{k-1}|} \cdot \frac{(1-\varepsilon)^2 \lambda_{\min}(A^T A)}{(1+\varepsilon)^2 \sigma_{\max}^2(A)}) \| x_k - x_\star \|_2^2 .$$

Then,

$$\| x_{k+1} - x_\star \|_2^2 \leq \prod_{q=0}^{k} (1 - \frac{\alpha|\tau_q|}{m - |\tau_{q-1}|} \cdot \frac{(1-\varepsilon)^2 \lambda_{\min}(A^T A)}{(1+\varepsilon)^2 \sigma_{\max}^2(A)}) \| x_0 - x_\star \|_2^2 .$$

Thus, we complete the proof. □

## 5. Numerical experiments

In this part, we will use BCSK and CS-2GSK methods to solve (1) and compare with the 2GSK [3] and CS-MWRK [9] methods in terms of the number of iteration steps (denoted as IT) and the computing time (denoted as CPU). All experiments were on a personal computer with 2.00 GHz central processing unit (Intel(R) Core(TM) i5-1038NG7 CPU), 16.00GB memory, and Windows 10 operating system. In addition, the version of MATLAB we use is R2018b.

**Example 5.1** We consider the case of stochastic systems in this experiment. The matrix $A \in \mathbb{R}^{m\times n}$ and the exact solution $x_\star$ are gained randomly by using the MATLAB function $randn(m,n)$. All the experiments start from zero vector and we repeat the experiment fifty times and average it. Otherwise, the experiment stops if

$$RES = \frac{||x_k - x_\star||_2^2}{||x_\star||_2^2} \leq 10^{-6}$$

or IT exceeds 200,000.

The numerical results are listed in **Table 1** illustrates that the CS-2GSK method has its pros and cons. More particular, CS-2GSK requires fewer IT than CS-MWRK and significantly less CPU time than 2GSK when $m \gg n$. Additionally, CS-2GSK needs more IT than the 2GSK method. This is due to the fact that the CS-2GSK approach converges a little more slowly due to its bigger convergence factor. In addition, we observe that IT of CS-2GSK approach lowers as $d$ increases, suggesting that the CS-2GSK method converges more quickly. The computation of $i_k = arg \max\limits_{1\leq i\leq d}\{\frac{|\widetilde{b}^{(i)} - \widetilde{A}^{(i)} x_k|}{\|\widetilde{A}^{(i)}\|_2^2}\}$ and $j_k = arg \max\limits_{i\in[m]\setminus i_k}\{\frac{|\widetilde{b}^{(i)} - \widetilde{A}^{(i)} x_k|}{\|\widetilde{A}^{(i)}\|_2^2}\}$ in each iteration will take longer as $d$ rises. As a result, the CS-2GSK method's overall processing time grows. Then, we can see that, for all test values of $d$, BCSK method uses substantially less CPU and has smaller IT than the other three algorithms.

**Example 5.2** In this example, we use the function $paralleltomo(N, \theta, p)$ in the *MATLAB* package *AIR TOOLS* [14]. We set $N = 30,\ \theta = 0:2:178^\circ$ *and* $p = 120$, then $A$ is a $10800 \times 1600$ matrix. What's more, we run these methods a thousand times and then observe how clear the image recovered.

**Table 1. Numerical results for the 2GSK, BCSK (with $\alpha = 0.16$), CS-MWRK and CS-2GSK methods with random matrix.**

| | | *IT* | | | | *CPU* | | | |
|---|---|---|---|---|---|---|---|---|---|
| $m \times n$ | d | 2GSK | CS-MWRK | BCSK | CS-2GSK | 2GSK | CS-MWRK | BCSK | CS-2GSK |
| $5000 \times 50$ | $10n$ | 34.0000 | 85.2000 | 1.1200 | 56.0000 | 0.0069 | 0.0109 | 0.0013 | 0.0100 |
| | $20n$ | 34.0000 | 67.4800 | 1.0400 | 49.4400 | 0.0075 | 0.0100 | 0.0044 | 0.0091 |
| | $30n$ | 32.0000 | 61.5400 | 1.0600 | 48.9800 | 0.0084 | 0.0128 | 0.0034 | 0.0106 |
| | $40n$ | 35.0000 | 58.5000 | 1.0000 | 47.6000 | 0.0075 | 0.0138 | 0.0028 | 0.0103 |
| | $50n$ | 31.0000 | 56.5600 | 1.0200 | 47.2800 | 0.0075 | 0.0125 | 0.0044 | 0.0134 |
| $5000 \times 100$ | $10n$ | 60.0000 | 172.2600 | 1.3800 | 119.7000 | 0.0219 | 0.0159 | 0.0075 | 0.0150 |
| | $20n$ | 64.0000 | 137.6000 | 1.1200 | 103.1400 | 0.0222 | 0.0175 | 0.0166 | 0.0166 |
| | $30n$ | 60.0000 | 126.6200 | 1.1000 | 99.0800 | 0.0200 | 0.0297 | 0.0097 | 0.0306 |
| | $40n$ | 60.0000 | 122.0600 | 1.0800 | 97.2200 | 0.0213 | 0.0437 | 0.0091 | 0.0362 |
| | $50n$ | 61.0000 | 118.6600 | 1.0800 | 95.5800 | 0.0213 | 0.0750 | 0.0088 | 0.0316 |
| $5000 \times 150$ | $10n$ | 92.0000 | 260.8600 | 2.1200 | 189.7400 | 0.0547 | 0.0431 | 0.0128 | 0.0316 |
| | $20n$ | 95.0000 | 213.4600 | 1.4400 | 164.1200 | 0.0603 | 0.0587 | 0.0219 | 0.0547 |
| | $30n$ | 97.0000 | 198.6200 | 1.4000 | 157.5800 | 0.0597 | 0.1744 | 0.0184 | 0.0881 |
| | $40n$ | 90.0000 | 192.2600 | 1.3200 | 154.0800 | 0.0512 | 0.2334 | 0.0244 | 0.1441 |
| | $50n$ | 95.0000 | 191.0000 | 1.1400 | 151.1000 | 0.0591 | 0.2972 | 0.0262 | 0.1844 |
| $5000 \times 200$ | $10n$ | 131.0000 | 354.3600 | 4.1600 | 267.0400 | 0.1244 | 0.0847 | 0.0338 | 0.0666 |
| | $20n$ | 131.0000 | 295.5000 | 2.0800 | 233.1800 | 0.1247 | 0.3050 | 0.0259 | 0.1672 |
| | $30n$ | 131.0000 | 281.5600 | 1.9000 | 222.7800 | 0.1259 | 0.4469 | 0.0344 | 0.2822 |
| | $40n$ | 135.0000 | 271.9200 | 1.6800 | 212.9000 | 0.1256 | 0.5791 | 0.0384 | 0.3781 |
| | $50n$ | 132.0000 | 265.7800 | 1.3000 | 204.1200 | 0.1278 | 0.7116 | 0.0397 | 0.5525 |

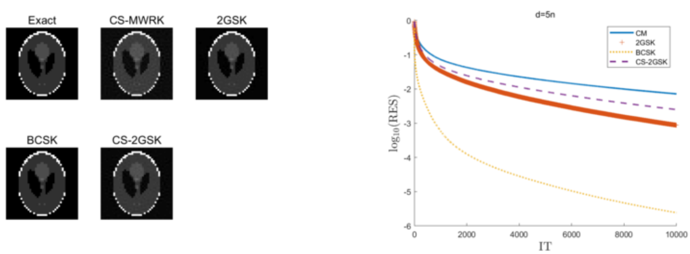


**Figure 1. Numerical results for the 2GSK, BCSK (with $\alpha = 0.16$), CS-MWRK and CS-2GSK methods with paralleltomo test problem.**

From the left of **Figure 1**, we see again that, under the same IT, BCSK algorithm can achieve better reconstruction results than CS-2GSK, CS-MWRK and 2GSK algorithms with generally smaller working blocks. The right side of **Figure 1** shows that, as IT increases, the RES of the BCSK method decays considerably more fast than those of CS-2GSK, CS-MWRK, and 2GSK algorithms. In fact, we can see that at the thousandth iteration, the RES of BCSK is less than $10^{-6}$.

## 6. Conclusion

In order to solve a sizable consistent linear system (1), we modified the GRK algorithm in this study to the count sketch two greedy subspace Kaczmarz (CS-2GSK) method and the block count sketch Kaczmarz (BCSK) method. We offer theoretical assurances that the two algorithms would converge. We demonstrate certain situations in the experiments section, where the BCSK algorithm performs better than the 2GSK algorithm and the CS-MWRK algorithm in terms of IT and CPU times. In light of this, the BCSK algorithm could be a helpful tool when compared with the CS-2GSK algorithm and the CS-MWRK algorithm.

## References


[1] Karczmarz S. Angenäherte auflösung von systemen linearer gleichungen[J]. Bull. Int. Acad. Pol. Sic. Let., Cl. Sci. Math. Nat., 1937: 355-357.

[2] Bai Z Z, Wu W T. On greedy randomized Kaczmarz method for solving large sparse linear systems[J]. SIAM Journal on Scientific Computing, 2018, 40(1): A592-A606.

[3] Liu Y, Gu C. Two Greedy Subspace Kaczmarz Algorithm for Image Reconstruction[J]. IAENG International Journal of Applied Mathematics, 2020, 50(4).

[4] Charikar M, Chen K, Farach-Colton M. Finding frequent items in data streams[C]. Springer Berlin Heidelberg, 2002: 693-703.

[5] Thorup M, Zhang Y. Tabulation-based 5-independent hashing with applications to linear probing and second moment estimation[J]. SIAM Journal on Computing, 2012, 41(2): 293-331.

[6] Woodruff D P. Sketching as a tool for numerical linear algebra[J]. Foundations and Trends in Theoretical Computer Science, 2014, 10(1-2): 1-157.

[7] Sarlos T. Improved approximation algorithms for large matrices via random projections[C]. IEEE, 2006: 143-152.

[8] Du K, Gao H. A new theoretical estimate for the convergence rate of the maximal weighted residual Kaczmarz algorithm[J]. Numer. Math. Theor. Meth. Appl, 2019, 12(2): 627-639.

[9] Zhang Y, Li H. A count sketch maximal weighted residual Kaczmarz method for solving highly overdetermined linear systems[J]. Applied Mathematics and Computation, 2021, 410: 126486.

[10] Meng X, Mahoney M W. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression[C]. Proceedings of the forty-fifth annual ACM symposium on Theory of computing. 2013: 91-100.

[11] Björck Å. Numerical methods for least squares problems[M]. Society for Industrial and Applied Mathematics, 1996.

[12] Drineas P, Mahoney M W, Muthukrishnan S, et al. Faster least squares approximation[J]. Numerische mathematik, 2011, 117(2): 219-249.

[13] Clarkson K L, Woodruff D P. Low-rank approximation and regression in input sparsity time[J]. Journal of the ACM (JACM), 2017, 63(6): 1-45.

[14] Hansen P C, Saxild-Hansen M. AIR tools-a MATLAB package of algebraic iterative reconstruction methods[J]. Journal of Computational and Applied Mathematics, 2012, 236(8): 2167-2178.