

# 基于 Count Sketch 的预处理贪婪 Kaczmarz 方法

叶雨欣, 殷俊锋

(同济大学 数学科学学院, 上海 200092)

**摘要:** 在贪婪 Kaczmarz 方法中, 通过对系数矩阵进行正交三角分解引入右预处理子能够提高贪婪 Kaczmarz 方法的收敛速率。但在系数矩阵的行数远大于列数的情况下, 正交三角分解的成本过高。为降低预处理的成本, 通过引入 Count Sketch 变换, 提出了基于 Count Sketch 的预处理贪婪 Kaczmarz 方法, 并对新方法进行了收敛性分析。理论分析说明了新方法在系数矩阵条件数较大时比已有方法具有更好的收敛速率。数值实验验证了新方法的有效性。

**关键词:** Kaczmarz 方法; 预处理; Count Sketch; 收敛性

**中图分类号:** O241.6

**文献标志码:** A

## Preconditioning Greedy Kaczmarz Method Based on Count Sketch

YE Yuxin, YIN Junfeng

(School of Mathematical Sciences, Tongji University, Shanghai 200092, China)

**Abstract:** The convergence rate of greedy Kaczmarz method can be improved by introducing right preconditioner through orthogonal triangularization of coefficient matrix. However, when the number of rows of the coefficient matrix is much larger than the number of columns, the cost of orthogonal triangularization is too high. By introducing Count Sketch transform, a preconditioning greedy Kaczmarz method based on Count Sketch is proposed to reduce the cost. Convergence analysis of the new algorithm is provided, and the theoretical analysis shows that the new method has better convergence rate than the existing method when the condition number of coefficient matrix is large. The numerical experiments verified the effectiveness.

**Keywords:** Kaczmarz method; preconditioning; Count Sketch; convergence property

考虑求解大规模超定相容线性方程组

$$Ax = b \quad (1)$$

其中, 矩阵  $A \in \mathbb{R}^{m \times n}$ ,  $m > n$ ,  $b \in \mathbb{R}^m$ ,  $x \in \mathbb{R}^n$  为未知向量。Kaczmarz 方法<sup>[1]</sup>是求解线性方程组(1)的一种经典的行作用迭代法, 在分布式计算、计算机断层扫描、数字信号处理等领域均有广泛应用<sup>[2-4]</sup>。假定初始近似值为  $x_0$ , 那么经典的 Kaczmarz 方法可以表示为

$$x_{k+1} = x_k + \frac{b_{i_k} - a_{i_k}^T x_k}{\|a_{i_k}\|_2^2} a_{i_k}, \quad k = 0, 1, 2, \dots \quad (2)$$

式中:  $\|\cdot\|_2$  表示欧几里得范数;  $a_{i_k}$  为矩阵  $A$  的第  $i_k$  行;  $b_{i_k}$  为向量  $b$  的第  $i_k$  个元素,  $i_k = (k \bmod m) + 1$ 。

2009 年, Strohmer 和 Vershynin<sup>[5]</sup>首次证明了当系数矩阵列满秩时, 随机 Kaczmarz 方法的线性收敛速率, 且该速率只取决于系数矩阵的缩放条件数  $\kappa(A) = \|A\|_F \|A^{-1}\|_2$ , 这再度使得 Kaczmarz 方法成为科学计算领域的一个热点。后来, Nutini 等人<sup>[6]</sup>提出基于最大距离规则的贪婪 Kaczmarz 方法, 它在每次迭代中选取与当前迭代解距离最大的超平面作为工作行。Bai 和 Wu<sup>[7]</sup>结合贪婪与随机的思想, 提出了贪婪随机 Kaczmarz 方法, 它在迭代过程中动态跟踪残差向量, 大大提高了随机 Kaczmarz 方法的效率。更多 Kaczmarz 类方法的研究参见文献<sup>[8-11]</sup>。

Sketching 方法是将 Sketch 矩阵  $S \in \mathbb{R}^{d \times m}$  作用于原始矩阵  $A$  的过程, 其中  $S \in \mathbb{R}^{d \times m}$  是从矩阵的某个分布中指定的一个的随机矩阵, 且  $n < d \ll m$ <sup>[8]</sup>。该方法使高成本的运算可以在较小的矩阵  $\hat{A} = SA \in \mathbb{R}^{d \times n}$  上执行, 从而加快原始问题的求解。近年来, 有很多 sketch 矩阵被提出并加以研究。例如: Gaussian Sketch<sup>[9]</sup>, 子采样随机 Hadamard 变换<sup>[10]</sup>, Count Sketch<sup>[11-12]</sup> 以及稀疏 Johnson-Lindenstrauss 变换<sup>[13]</sup>等。最近, 有学者将 Sketching

收稿日期: 2023-05-22

基金项目: 国家自然科学基金(11971354)

第一作者: 叶雨欣, 博士生, 主要研究方向为数值分析与科学计算。E-mail: yeyuxin@tongji.edu.cn

通信作者: 殷俊锋, 教授, 博士生导师, 理学博士, 主要研究方向为数值分析与科学计算。E-mail: yinjf@tongji.edu.cn



论文  
拓展  
介绍

方法与Kaczmarz类方法结合起来,提高Kaczmarz类方法的效率。Katrutsa等<sup>[14]</sup>通过行抽样实现sketching过程,对变换后的矩阵通过正交三角分解得到右预处理子,再用Kaczmarz方法进行求解,大大减少预处理时间,同时加速了Kaczmarz方法的收敛。Needell等<sup>[15]</sup>将Gaussian sketch与分块Kaczmarz方法结合,提出分块Gaussian Kaczmarz方法,说明了某些特定情况下使用分块Gaussian Kaczmarz方法求解线性方程组的优势。Zhang等<sup>[16]</sup>将Count Sketch与最大加权残差Kaczmarz方法结合起来,提出一种求解高度超定线性方程组的快速随机算法,大大提高了算法的计算效率。

在基于sketching预处理Kaczmarz方法<sup>[14]</sup>的基础上,结合贪婪距离的思想,本文提出了一种基于Count Sketch的预处理贪婪Kaczmarz方法。该方法先对超定线性系统进行Count Sketch变换,然后对变换后的矩阵进行正交三角分解得到右预处理子,再与贪婪Kaczmarz方法结合求解预处理后的方程组。我们分析了新方法的收敛性,理论分析说明在系数矩阵条件数较大的情形下,新方法的收缩因子比已有方法的收缩因子小。数值实验进一步验证了新方法的有效性。

## 1 基于Count Sketch的预处理贪婪Kaczmarz方法

本文采用 $R(A)$ 表示由矩阵 $A$ 列向量张成的值空间。其中, $R(A)=\{Ax|x\in R^n\}$ 。 $A^T$ ,  $A^\dagger$ ,  $\|A\|_F$ ,  $\sigma_{\min}(A)$ 和 $\sigma_{\max}(A)$ 分别表示矩阵 $A$ 的转置, Moore-Penrose伪逆, Frobenius范数, 最小非零奇异值和最大奇异值;  $x_\star=A^\dagger b$ 是线性代数方程组(1)的最小范数解。

Kaczmarz方法的收敛速率与行指标的选择策略密切相关,选取与当前迭代点距离最大超平面的贪婪距离准则便是一种有效策略, Nutini等<sup>[6]</sup>证明了在某些应用中贪婪选择策略和随机选择策略的计算成本是相当的,并且在许多情况下贪婪策略比随机策略具有更快的收敛速度。

结合贪婪距离策略得到的贪婪Kaczmarz方法(见算法1)在文献<sup>[17]</sup>中以最大加权残差Kaczmarz的名称最早出现,但其给出的收敛估计并不容易计算。后来,文献<sup>[18]</sup>在贪婪随机Kaczmarz方法<sup>[7]</sup>收敛性证明的基础上更新了贪婪Kaczmarz方法收敛速度的理论估计,如引理1所示。

**算法1** 贪婪Kaczmarz方法<sup>[6]</sup>。①置 $k:=0$ 。

②计算 $i_k=\arg\max_{i\in\{1,\dots,m\}}\frac{|b_i-a_i^T x_k|^2}{\|a_i\|_2^2}$ 。③计算 $x_{k+1}=$

$x_k+\frac{b_{i_k}-a_{i_k}^T x_k}{\|a_{i_k}\|_2^2}a_{i_k}$ 。④置 $k=k+1$ ,转步骤②。

**引理1** <sup>[18]</sup>若线性方程组 $Ax=b$ 相容,初始估计 $x_0\in R(A^T)$ 。则通过贪婪Kaczmarz方法生成的迭代序列 $\{x_k\}_{k=0}^\infty$ 收敛到其最小范数解 $x_\star=A^\dagger b$ ,且迭代序列 $\{x_k\}_{k=0}^\infty$ 的误差满足

$$\|x_1-x_\star\|_2^2\leq\left(1-\frac{\sigma_{\min}^2(A)}{\|A\|_F^2}\right)\|x_0-x_\star\|_2^2$$

对于 $k=1,2,\dots$ ,

$$\|x_{k+1}-x_\star\|_2^2\leq\left(1-\frac{\sigma_{\min}^2(A)}{\max_{1\leq j\leq m}\sum_{i=1,i\neq j}^m\|a_i\|_2^2}\right)\|x_k-x_\star\|_2^2$$

贪婪Kaczmarz方法的收敛速度很大程度上依赖于系数矩阵的条件数 $k(A)$ ,当 $k(A)$ 较大时,贪婪Kaczmarz方法的收敛速度会变慢。解决此问题的方法之一是使用预处理矩阵。由于本问题中 $m>n$ ,因此考虑如下形式的右预处理方法:

$$APy=b, \quad x=Py$$

其中, $P\in R^{n\times n}$ ,且 $k(AP)<k(A)$ 。

因此,最理想的预处理子 $P$ 应满足 $k(AP)=1$ 。受到文献<sup>[14]</sup>的启发,满足此条件的预处理子可通过矩阵的正交三角分解获取:计算 $A$ 的正交三角分解 $A=QR$ , $Q\in R^{m\times n}$ , $R\in R^{n\times n}$ ,然后取 $P=R^\dagger$ 。使用这样的 $P$ 进行预处理,再用贪婪Kaczmarz方法求解,可得到正交三角分解预处理的贪婪Kaczmarz方法,该方法在迭代步数方面能够大大加快收敛速率。

注意到对矩阵 $A$ 进行正交三角分解的成本为 $O(mn^2)$ ,这对于高度超定的系统( $m\gg n$ )来说过高。因此文献<sup>[14]</sup>使用了基于行抽样的sketching方法以较低的成本获得上述预处理子 $P$ 的近似,这里的行抽样sketch矩阵 $S$ 作用于系数矩阵 $A$ 后,相当于随机选择了 $A$ 的某 $d$ 行。但抽样的行数 $d$ 是一个未知的超参数,且文献<sup>[14]</sup>没有给出使用此类sketching预处理的Kaczmarz方法的收敛性证明。

Sketching方法中,一种关键技术是Johnson-Lindenstrauss型嵌入算法<sup>[19]</sup>,Count Sketch就是其中一种,并且在某些情况下它比抽样(sampling)更有优势<sup>[11]</sup>。

**定义1** <sup>[11-12]</sup>定义Count Sketch变换为 $S=$

$\Phi D \in \mathbb{R}^{d \times m}$ 。这里,  $D$  是一个  $m \times m$  的随机对角矩阵, 每个对角项以相同的概率独立被选为-1或1,  $\Phi \in \{0, 1\}^{d \times m}$  是一个  $d \times m$  的二元矩阵, 其中  $\Phi_{h(i), i} = 1$ , 其余元素为0, 这里  $h: [m] \rightarrow [d]$  是一个随机映射, 对于每个  $i \in [m]$ ,  $j \in [d]$ ,  $h(i) = j$  的概率为  $1/d$ 。

通过定义1可以发现, Count Sketch矩阵由每列有且仅有一个1的矩阵  $\Phi$  和对角元为1或-1的对角矩阵  $D$  相乘得到, 它与系数矩阵  $A$  相乘的成本仅为  $O(\text{nnz}(A))$ , 且  $\Phi$  和  $D$  均可以低成本的构造, 其中  $\text{nnz}(A)$  为矩阵  $A$  的非零项个数。

Count sketch变换还满足如下引理:

**引理2** <sup>[20]</sup> 若  $S \in \mathbb{R}^{d \times m}$  是一个 Count Sketch 变换,  $d = O((n^2 + n)/(\delta\epsilon^2))$ , 其中  $0 < \delta, \epsilon < 1$ , 则在  $1 - \delta$  的概率下, 有

$$(1 - \epsilon) \|Ax\|_2 \leq \|SAx\|_2 \leq (1 + \epsilon) \|Ax\|_2, \text{ 对所有 } x \in \mathbb{R}^n \quad (3)$$

$$(1 - \epsilon) \sigma_i(A) \leq \sigma_i(SA) \leq (1 + \epsilon) \sigma_i(A), \text{ 对所有 } 1 \leq i \leq n \quad (4)$$

Count sketch矩阵的这些性质也引起了学者的关注, 例如, 文献[16]和文献[21]分别将 Count

Sketch与最大加权残差Kaczmarz和斜投影最大加权残差Kaczmarz方法相结合, 在计算时间方面大大提高了原方法的效率。

本文提出基于Count Sketch的预处理贪婪Kaczmarz方法, 如算法2所示。

**算法2** 基于Count Sketch的预处理贪婪Kaczmarz方法。①创建Count Sketch矩阵  $S \in \mathbb{R}^{d \times m}$ ,  $n < d < m$ , 并计算  $\hat{A} = SA \in \mathbb{R}^{d \times n}$ 。②对  $\hat{A}$  进行正交三角分解,  $\hat{A} = QR$ ,  $Q \in \mathbb{R}^{d \times n}$ ,  $R \in \mathbb{R}^{n \times n}$  并将  $P = R^+$  作为右预处理矩阵。③计算  $\tilde{A} = AP$ , 并记  $\tilde{A}$  的第  $i$  行为  $\tilde{a}_i$ ,  $i = 1, 2, \dots, m$ 。④

置  $k := 0$ 。⑤计算  $i_k = \arg \max_{i \in \{1, \dots, m\}} \frac{|b_i - \tilde{a}_i^T y_k|^2}{\|\tilde{a}_i\|_2^2}$ 。⑥计

算  $y_{k+1} = y_k + \frac{b_{i_k} - \tilde{a}_{i_k}^T y_k}{\|\tilde{a}_{i_k}\|_2^2} \tilde{a}_{i_k}$ 。⑦置  $k = k + 1$ , 转步

骤⑤。⑧计算  $x = Py$ 。

表1列出基于Count Sketch的预处理贪婪Kaczmarz(PCSGK)方法、贪婪Kaczmarz(GK)方法以及正交三角分解预处理的贪婪Kaczmarz(PGK)方法的计算成本。

表1 PCSGK、GK和PGK方法的计算成本

Tab.1 Computation costs of PCSGK, GK and PGK method

参数	PCSGK	GK	PGK
计算S与A的乘积	$O(\text{nnz}(A))$	--	--
系数矩阵的正交三角分解	$O(dn^2)$	--	$O(nm^2)$
矩阵R求逆	$O(n^3)$	--	$O(n^3)$
对系数矩阵作用右预处理子	$O(mn^2)$	--	$O(mn^2)$
每步选取行指标	$O(mn)$	$O(mn)$	$O(mn)$
近似解更新	$O(n)$	$O(n)$	$O(n)$
迭代步数	$O(\kappa(AP)^2 \log(1/\epsilon))$	$O(\kappa(A)^2 \log(1/\epsilon))$	$O(\kappa(AP)^2 \log(1/\epsilon))$

从表1可以观察到, 使用sketching方法后, 正交三角分解的复杂度从  $O(mn^2)$  降低到了  $O(dn^2)$ , 如果  $d = \gamma n$ , 则可以重写为  $O(n^3)$ , 其中  $\gamma \geq 1$  是某个给定的常数。这样, 计算预处理子  $P$  的总复杂度从  $O(mn^2) + O(n^3)$  降低为了  $O(n^3)$ , 与矩阵  $A$  的行数  $m$  无关。

同时, 注意到当系数矩阵缩放条件数  $\kappa(A)$  较大时, 经典的贪婪Kaczmarz方法可能需要很多次迭代才能收敛, 如果此时用  $P$  进行预处理, 则有  $\kappa(AP) \sim \sqrt{n}$ , 则算法可以在  $O(n)$  步内收敛。此时预处理的贪婪Kaczmarz方法在迭代步数上的改进可以远远大于预处理的成本。此外, 在很多问题上, 预处理只需要进行一次, 因此对于条件数较大的系统, 基于Count Sketch的预处理贪婪Kaczmarz

方法是一个较为高效的选择。

## 2 收敛性分析

本节说明基于Count Sketch的预处理贪婪Kaczmarz方法的收敛性。定理1是本文的关键定理:

**定理1** 若  $S \in \mathbb{R}^{d \times m}$  是一个 Count Sketch 变换,  $d = O((n^2 + n)/(\delta\epsilon^2))$ ,  $0 < \delta, \epsilon < 1$ , 初始估计  $y_0 \in R(P^T A^T)$ , 记基于Count Sketch的预处理贪婪Kaczmarz方法生成的迭代序列为  $\{y_k\}_{k=0}^\infty$ 。则在  $1 - \delta$  的概率下, 序列  $\{y_k\}_{k=0}^\infty$  收敛到  $\tilde{A}y = b$  的最小范数解  $y_\star = \tilde{A}^\dagger b$ , 且满足

$$\|y_1 - y_\star\|_2^2 \leq \left(1 - \frac{(1 - \epsilon_1)^2}{(1 + \epsilon_2)^2} \frac{1}{r}\right) \|y_0 - y_\star\|_2^2$$



对于  $k=1, 2, \dots$ ,

$$\|y_{k+1} - y_\star\|_2^2 \leq \left(1 - (1 - \epsilon_1)^2 \frac{1}{\max_{1 \leq j \leq m} \sum_{i=1, i \neq j}^m \|\tilde{a}_i\|_2^2}\right) \|y_k - y_\star\|_2^2 \quad (5)$$

其中,  $\epsilon_1 = \frac{\epsilon}{1+\epsilon}$ ,  $\epsilon_2 = \frac{\epsilon}{1-\epsilon}$ ,  $r$  为矩阵  $A$  的秩,  $\tilde{A} = AP$ 。

**证明** 由于基于 Count Sketch 的预处理贪婪 Kaczmarz 方法本质上为用贪婪 Kaczmarz 方法求解线性方程组  $APy = b$ , 即  $\tilde{A}y = b$ , 则由引理 1, 有

$$\|y_1 - y_\star\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(\tilde{A})}{\|\tilde{A}\|_F^2}\right) \|y_0 - y_\star\|_2^2 \quad (6)$$

对于  $k=1, 2, \dots$ ,

$$\|y_{k+1} - y_\star\|_2^2 \leq \left(1 - \frac{\sigma_{\min}^2(\tilde{A})}{\max_{1 \leq j \leq m} \sum_{i=1, i \neq j}^m \|\tilde{a}_i\|_2^2}\right) \|y_k - y_\star\|_2^2 \quad (7)$$

接下来对矩阵  $\tilde{A}$  的性质进行研究。根据引理 2 中式(4), 对所有  $1 \leq i \leq n$ , 有

$$(1 - \epsilon)\sigma_i(\tilde{A}) \leq \sigma_i(S\tilde{A}) \leq (1 + \epsilon)\sigma_i(\tilde{A}) \quad (8)$$

通过记  $\epsilon_1 = \frac{\epsilon}{1+\epsilon}$ ,  $\epsilon_2 = \frac{\epsilon}{1-\epsilon}$ , 可将式(8)化为

$$(1 - \epsilon_1)\sigma_i(S\tilde{A}) \leq \sigma_i(\tilde{A}) \leq (1 + \epsilon_2)\sigma_i(S\tilde{A}) \quad (9)$$

而  $S\tilde{A} = QRR^\top$ ,

$$n < d < m, Q \in \mathbb{R}^{d \times n}, R \in \mathbb{R}^{n \times n},$$

故对所有  $1 \leq i \leq r$ ,  $\sigma_i(S\tilde{A}) = \sigma_i(Q) = 1$ 。

所以

$$\sigma_i(\tilde{A}) \geq (1 - \epsilon_1)\sigma_i(S\tilde{A}) = 1 - \epsilon_1 \quad (10)$$

又因为

$$\|S\tilde{A}\|_F^2 = \sum_{i=1}^r \sigma_i^2(S\tilde{A}) = r$$

所以

$$\|\tilde{A}\|_F^2 \leq (1 + \epsilon_2)\|S\tilde{A}\|_F^2 = (1 + \epsilon_2)r \quad (11)$$

将式(10)和式(11)代入式(6)和式(7)即可完成定理证明。

**备注** 根据定义 1, 矩阵  $S$  的一行或几行元素可能为 0, 这意味着  $S\tilde{A}$  可能存在零行, 则有

$$\max_{1 \leq j \leq m} \sum_{i=1, i \neq j}^m \|\tilde{a}_i\|_2^2 = \|\tilde{A}\|_F^2 - \min_{1 \leq j \leq m} \|\tilde{a}_j\|_2^2$$

$$\leq (1 + \epsilon_2)\|S\tilde{A}\|_F^2 + 0 = (1 + \epsilon_2)r$$

因此, 将上式代入式(5), 在  $1 - \delta$  的概率下, 有

$$\|y_{k+1} - y_\star\|_2^2 \leq \left(1 - \frac{(1 - \epsilon_1)^2}{(1 + \epsilon_2)^2} \frac{1}{r}\right) \|y_k - y_\star\|_2^2$$

注意到, 对于引理 1, 由于

$$\frac{\sigma_{\min}^2(A)}{\|A\|_F^2} = \frac{\sigma_{\min}^2(A)}{\sum_{i=1}^r \sigma_i^2(A)} \leq \frac{\sigma_{\min}^2(A)}{r\sigma_{\min}^2(A)} = \frac{1}{r}$$

当系数矩阵病态时,  $\frac{\sigma_{\min}^2(A)}{\|A\|_F^2}$  与  $\frac{1}{r}$  相差较大, 若

$\frac{1 - \epsilon_1}{1 + \epsilon_2} \rightarrow 1$ , 则  $\frac{\sigma_{\min}^2(A)}{\|A\|_F^2}$  小于  $\frac{1 - \epsilon_1}{1 + \epsilon_2} \frac{1}{r}$ , 基于 Count Sketch 的预处理贪婪 Kaczmarz 方法从而具有更快的收敛速度。

### 3 数值实验

本节通过数值实验比较贪婪 Kaczmarz(GK)方法、Count Sketch 最大加权残差 Kaczmarz(CS-MWRK)方法、正交三角分解预处理的贪婪 Kaczmarz(PGK)方法和基于 Count Sketch 的预处理贪婪 Kaczmarz(PCSGK)方法, 数值结果显示新方法在迭代步数(简记为 IT)与计算时间(简记为 CPU)方面均具有较好的性能。

在实验中, 初始估计  $x_0$  取为零向量, 构造一个精确解  $x$ , 并且右端项  $b = Ax$ 。考虑到随机性, 本节所有数值结果取 20 次独立实验的平均值。停机准则为相对残差(简记为 RES)

$$\frac{\|b - \tilde{A}y_k\|_2}{\|b\|_2} < 10^{-3}$$

或者迭代步数超过 10 万步。实验表格中, 若在 10 万步内未达到指定精度, 即用“-”表示。对于正交三角分解预处理的贪婪 Kaczmarz 方法和基于 Count Sketch 的预处理贪婪 Kaczmarz 方法, 计算时间为预处理时间与迭代时间之和。

**例 1** 生成一个  $m \times n$  的随机高斯矩阵并进行奇异值分解, 然后将其奇异值替换为  $j^\alpha$ ,  $j = 1, 2, \dots, n$ , 以获取一个新的系数矩阵, 该矩阵的条件数即为  $n^\alpha$ 。通过把  $n$  取为 50,  $\alpha$  分别取 2 和 2.5 将系数矩阵条件数分别控制为 2 500.00 和 17 677.67; 通过把  $\gamma$  分别取为 5, 10 和 15 来改变 Sketching 的行数  $d = \gamma n$ 。

表 2 和表 3 列出了贪婪 Kaczmarz 方法、Count

Sketch 最大加权残差 Kaczmarz 方法、正交三角分解 的预处理贪婪 Kaczmarz 方法的迭代步数与计算  
预处理的贪婪 Kaczmarz 方法以及基于 Count Sketch 时间。

表 2  $\alpha=2$  时的数值结果  
Tab.2 Numerical results for  $\alpha=2$

矩阵规模		5 000×50			10 000×50			50 000×50		
算法		IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
GK		26 075.00	1.927 1	$9.99\times10^{-4}$	14 282.00	3.910 1	$9.97\times10^{-4}$	9 010.00	14.641 7	$9.78\times10^{-4}$
CS-MWRK	$\gamma=5$	75 153.80	0.584 5	$9.90\times10^{-4}$	42019.45	0.341 5	$9.88\times10^{-4}$	35 374.60	0.347 7	$9.87\times10^{-4}$
	$\gamma=10$	46 532.65	0.435 0	$9.91\times10^{-4}$	33059.60	0.362 4	$9.87\times10^{-4}$	24 240.15	0.237 7	$9.82\times10^{-4}$
	$\gamma=15$	43 037.60	0.510 7	$9.89\times10^{-4}$	29658.40	0.393 6	$9.90\times10^{-4}$	22 273.45	0.280 5	$9.84\times10^{-4}$
PGK		50.00	0.009 3	$8.94\times10^{-4}$	44.00	0.025 6	$9.16\times10^{-4}$	37.00	0.141 8	$9.86\times10^{-4}$
PCSGK	$\gamma=5$	62.60	0.007 9	$9.59\times10^{-4}$	53.80	0.020 7	$9.39\times10^{-4}$	45.00	0.093 3	$9.16\times10^{-4}$
	$\gamma=10$	55.15	0.008 1	$9.41\times10^{-4}$	49.05	0.018 1	$8.97\times10^{-4}$	40.05	0.080 3	$9.08\times10^{-4}$
	$\gamma=15$	51.40	0.006 3	$9.39\times10^{-4}$	47.60	0.017 5	$9.39\times10^{-4}$	38.40	0.079 2	$9.43\times10^{-4}$

表 3  $\alpha=2.5$  时的数值结果  
Tab.3 Numerical results for  $\alpha=2.5$

矩阵规模		5 000×50			10 000×50			50 000×50		
算法		IT	CPU	RES	IT	CPU	RES	IT	CPU	RES
GK		16 940.00	1.152 7	$9.97\times10^{-4}$	11 319.00	2.949 0	$9.75\times10^{-4}$	8 396.00	16.146 9	$9.72\times10^{-4}$
CS-MWRK	$\gamma=5$	40 299.80	0.313 8	$9.87\times10^{-4}$	35 991.45	0.262 3	$9.86\times10^{-4}$	34 652.20	0.324 8	$9.87\times10^{-4}$
	$\gamma=10$	28 866.85	0.274 8	$9.83\times10^{-4}$	26 393.20	0.250 9	$9.89\times10^{-4}$	26 023.40	0.317 9	$9.87\times10^{-4}$
	$\gamma=15$	26 065.40	0.332 2	$9.73\times10^{-4}$	23 087.00	0.268 1	$9.89\times10^{-4}$	22 005.00	0.276 7	$9.80\times10^{-4}$
PGK		48.00	0.010 7	$8.78\times10^{-4}$	44.00	0.026 3	$9.62\times10^{-4}$	40.00	0.169 3	$7.96\times10^{-4}$
PCSGK	$\gamma=5$	61.75	0.007 0	$9.52\times10^{-4}$	58.40	0.019 9	$9.13\times10^{-4}$	44.05	0.096 4	$8.52\times10^{-4}$
	$\gamma=10$	54.60	0.006 8	$9.43\times10^{-4}$	48.15	0.018 3	$8.94\times10^{-4}$	40.40	0.085 3	$9.19\times10^{-4}$
	$\gamma=15$	51.60	0.006 0	$9.21\times10^{-4}$	47.00	0.016 3	$9.17\times10^{-4}$	40.80	0.080 7	$9.03\times10^{-4}$

从表 2 和表 3 可以看出, PGK 方法在迭代步数方面表现的最好, 这是因为 PGK 方法的预处理子在理论上可以将系统的条件数改善到最优值 1, 而 PCSGK 方法中的 Count Sketch 过程在加速正交三角分解的过程中牺牲了一定的精度, 所以在迭代步数方面, PCSGK 方法相对于 PGK 方法稍显逊色。

从表 2 和表 3 的数据中同样可以观察到, 通过选取适当的 sketching 行数, PCSGK 方法的迭代步数可以十分接近 PGK 方法的迭代步数, 同时 Count Sketch 过程有效减少了预处理时间, 这使得 PCSGK 方法在计算时间方面明显优于 PGK 方法。综合计算时间和迭代步数两方面的因素, PCSGK 在这 4 种方法中效率是最高的。此外, 在生成的随机矩阵的条件数不变, 即  $\alpha$  不变时, 随着 sketching 行数的增加, PCSGK 方法在迭代步数的加速效果越明显, 这可以解释为随着  $\gamma$  的增大, 定理 1 中  $\frac{1-\epsilon_1}{1+\epsilon_2}$  这一项越来越接近于 1, 从而加快了 PCSGK 方法的收敛。

图 1 绘制了矩阵规模为  $10\,000\times50$ ,  $\alpha=2.5$ ,  $\gamma=15$  时, 4 种方法的  $\log(\text{RES})$  分别随迭代步数和计算时间变化的曲线。从图 1 中可看出, 4 种方

法的收敛曲线进一步验证了 PCSGK 方法的高效性。

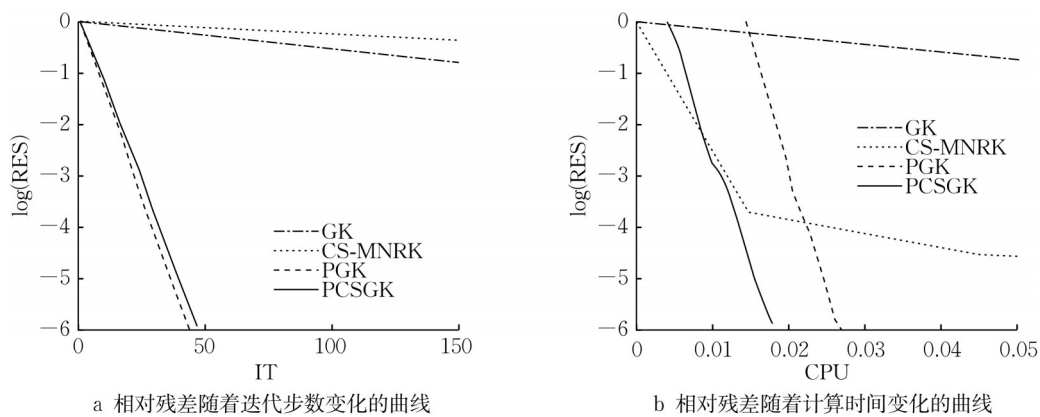
**例 2** 选取了佛罗里达稀疏矩阵库中的一些矩阵进行实验, 其中欠定矩阵取其转置。这里包含矩阵 lp\_fit2d、lp\_d6cube、relat5 和 relat6。

表 4 列出了矩阵信息, 以及 Sketching 行数  $d$  选取一些适当的值时, 4 种方法的数值结果。

从表 4 可以观察到, PCSGK 方法相对 GK 方法和 CS-MWRK 方法具有明显较少的迭代步数。

图 2 绘制了对于表 4 中的矩阵算例 relat5, 4 种方法的  $\log(\text{RES})$  分别随迭代步数和计算时间变化的曲线。

从图 2 可以观察到, 对于算例 relat5 这样的系统, PGK 方法所需的预处理时间较长, 甚至出现了 PCSGK 方法已经收敛但 PGK 方法还未完成预处理的情形, PCSGK 方法虽然牺牲了预处理子的精确度, 在计算时间方面大大提高了算法效率。因此综合迭代步数与计算时间这两方面的因素, 基于 Count Sketch 的预处理贪婪 Kaczmarz 方法是一个较好的选择。



a 相对残差随着迭代步数变化的曲线

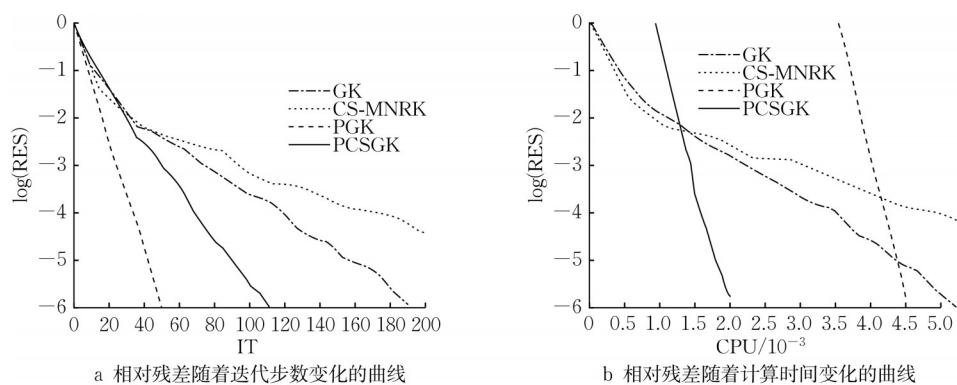
b 相对残差随着计算时间变化的曲线

图 1 矩阵规模为  $10\,000 \times 50$ ,  $\alpha = 2.5$ ,  $\gamma = 15$  时的收敛曲线Fig.1 Convergence curves for matrix of  $10\,000 \times 50$ , with  $\alpha = 2.5$ ,  $\gamma = 15$ 

表 4 例 2 的矩阵信息和数值结果

Tab.4 Matrix information and numerical results for Example 2

矩阵名称		lp_fit2d <sup>T</sup>	lp_d6cube <sup>T</sup>	relat5	relat6
$m \times n$		10 524 × 25	6 184 × 415	340 × 35	2 340 × 157
稠密度/%		49.05	1.47	8.89	2.21
条件数		$1.74 \times 10^3$	$4.93 \times 10^{19}$	Inf	Inf
$d$		300	800	70	300
GK	IT	192.00	7 130.00	196.00	1 359.00
	CPU	0.057 2	0.769 4	0.005 1	0.079 4
	RES	$7.76 \times 10^{-4}$	$9.56 \times 10^{-4}$	$9.82 \times 10^{-4}$	$9.82 \times 10^{-4}$
CS-MWRK	IT	26 723.15	53 185.70	334.65	3 544.70
	CPU	0.503 4	3.426 3	0.012 1	0.154 9
	RES	$9.85 \times 10^{-4}$	$9.78 \times 10^{-4}$	$9.53 \times 10^{-4}$	$9.93 \times 10^{-4}$
PGK	IT	31.00	521.00	56.00	283.00
	CPU	0.038 4	12.527 1	0.004 5	0.411 3
	RES	$8.09 \times 10^{-4}$	$9.89 \times 10^{-4}$	$8.50 \times 10^{-4}$	$9.91 \times 10^{-4}$
PCSGK	IT	35.45	1 631.30	113.20	763.10
	CPU	0.012 7	2.827 1	0.002 1	0.082 1
	RES	$9.85 \times 10^{-4}$	$9.96 \times 10^{-4}$	$9.68 \times 10^{-4}$	$9.90 \times 10^{-4}$



a 相对残差随着迭代步数变化的曲线

b 相对残差随着计算时间变化的曲线

图 2 矩阵 relat5 的收敛曲线

Fig.2 Convergence curves for matrix relat5

## 4 结语

提出了一种基于 Count Sketch 的预处理贪婪 Kaczmarz 方法,理论分析证明了新方法的收敛性,且在系数矩阵条件数较大的情况下,基于 Count

Sketch 的预处理贪婪 Kaczmarz 方法的收敛因子小于经典的收敛因子,即新方法具有较快的收敛速度。数值实验的结果表明,提出的新方法在计算时间与迭代步数方面均有较大的优势。

基于 Count Sketch 的预处理贪婪 Kaczmarz 方法能

够快速求解超定线性方程组(1)的最小范数解,主要原因在于采用了正交三角分解进行预处理,并采用贪婪策略选择工作行。因此如何选用成本更低的预处理以及更高效选取工作行的策略,仍然是值得研究的问题。

### 作者贡献声明:

叶雨欣:算法设计和算法研究执行,收敛性证明,数值实验和数据分析、论文初稿写作;

殷俊锋:研究构思,实验设计指导,数据分析,论文写作与修改。

### 参考文献:

- [1] KACZMARZ S. Angenäherte auflösung von systemen linearer gleichungen [J]. Bulletin International de l'Academie Polonaise des Sciences et des Lettres, 1937, 35: 355.
- [2] CENSOR Y. Parallel application of block-iterative methods in medical imaging and radiation therapy [J]. Mathematical programming, 1988, 42(1): 307.
- [3] ELBLE J M, SAHINIDIS N V, VOUZIS P. GPU computing with Kaczmarz's and other iterative algorithms for linear systems [J]. Parallel computing, 2010, 36(5/6): 215.
- [4] BYRNE C. A unified treatment of some iterative algorithms in signal processing and image reconstruction[J]. Inverse problems, 2003, 20(1): 103.
- [5] STROHMER T, VERSHYNIN R. A randomized Kaczmarz algorithm with exponential convergence [J]. Journal of Fourier Analysis and Applications, 2009, 15(2): 262.
- [6] NUTINI J, SEPEHRY B, LARADJI I, *et al.* Convergence rates for greedy Kaczmarz algorithms, and faster randomized Kaczmarz rules using the orthogonality graph [EB/OL]. [2023-04-22 ] <https://doi.org/10.48550/arXiv.1612.07838>.
- [7] 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.
- [8] BAI Z Z, WU W T. On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems [J]. Applied Mathematics Letters, 2018, 83: 21.
- [9] 杜亦疏, 殷俊锋, 张科. 求解大型稀疏线性方程组的贪婪距离随机 Kaczmarz 方法[J]. 同济大学学报(自然科学版), 2020, 48(8): 1224.  
DU Yishu, YIN Junfeng, ZHANG Ke. Greedy randomized-distance Kaczmarz method for solving large sparse linear systems [J]. Journal of Tongji University (Natural Science), 2020, 48(8): 1224.
- [10] 王泽, 殷俊锋. 求解线性方程组稀疏解的稀疏贪婪随机 Kaczmarz 算法[J]. 同济大学学报(自然科学版), 2021, 49(11): 1505.  
WANG Ze, YIN Junfeng. Sparse greedy randomized Kaczmarz method for sparse solutions to linear equations [J]. Journal of Tongji University (Natural Science), 2021, 49(11): 1505.
- [11] XIAO A Q, YIN J F, ZHENG N. On fast greedy block Kaczmarz methods for solving large consistent linear systems [J]. Computational and Applied Mathematics, 2023, 42(3): 119.
- [12] WOODRUFF D P. Sketching as a tool for numerical linear algebra [J]. Foundations and Trends® in Theoretical Computer Science, 2014, 10(1/2): 1.
- [13] BOUTSIDIS C, DRINEAS P, MAGDON-ISMAIL M. Near-optimal column-based matrix reconstruction [J]. SIAM Journal on Computing, 2014, 43(2): 687.
- [14] AILON N, CHAZELLE B. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform [C]//Proceedings of the thirty-eighth annual ACM symposium on Theory of computing. 2006: 557-563.
- [15] CHARIKAR M, CHEN K, FARACH-COLTON M. Finding frequent items in data streams [J]. Theoretical Computer Science, 2004, 312(1): 3.
- [16] 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.
- [17] KANE D M, Nelson J. Sparser Johnson-Lindenstrauss transforms [J]. Journal of the ACM (JACM), 2014, 61(1): 1.
- [18] KATRUTSA A, OSELEDETS I. Preconditioning Kaczmarz method by sketching [EB/OL]. [2023-04-22 ] <https://doi.org/10.48550/arXiv.1903.01806>.
- [19] REBROVA E, NEEDELL D. On block Gaussian sketching for the Kaczmarz method [J]. Numerical Algorithms, 2021, 86(1): 443.
- [20] 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.
- [21] MCCORMICK S F. The methods of Kaczmarz and row orthogonalization for solving linear equations and least squares problems in Hilbert space [J]. Indiana University Mathematics Journal, 1977, 26(6): 1137.
- [22] DU K, GAO H. A new theoretical estimate for the convergence rate of the maximal weighted residual Kaczmarz algorithm [J]. Numerical Mathematics: Theory, Methods and Applications, 2019, 12(2): 627.
- [23] JOHNSON W B, LINDENSTRAUSS J. Extensions of Lipschitz mappings into a Hilbert space [J]. Contemporary Mathematics, 1984, 26: 189.
- [24] 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.[S.l.]: ACM, 2013: 91-100.
- [25] ZHANG P, LI L, ZHANG P. A count sketch maximal weighted residual kaczmarz method with oblique projection for highly overdetermined linear systems [J]. Advances in Pure Mathematics, 2022, 12(4): 260.