一种加速的随机 Bregman-Kaczmarz 方法用于强凸线性约束优化*

Lionel Tondji[1], Dirk A. Lorenz[1] and Ion Necoara[2,3]

***Abstract*— 在本文中，我们提出了一种用于在线性约束下强凸函数最小化的随机加速方法。该方法是 Kaczmarz 型的，即每次迭代仅使用一个线性方程。为了获得加速效果，我们利用了 Kaczmarz 方法与坐标下降法对偶的事实。我们采用最近提出的随机坐标下降加速方法，并将其转移到原始空间。该方法继承了许多加速坐标下降方法的优点，包括其最坏情况下的收敛速率。给出了所提方法的收敛性理论分析。数值实验表明，所提出的方法比现有方法在解决相同问题时更有效且更快。**

I. 介绍

我们考虑大规模线性系统求解近似解的基本问题，形式如下：

$$\boldsymbol{A}x = b \tag{1}$$

其中矩阵为 $\boldsymbol{A} \in \mathbb{R}^{m\times n}$，右端项为 $b \in \mathbb{R}^m$。我们考虑矩阵不能同时完全访问的情况，但可以一次只处理系统 (1) 的单一行。这类问题出现在工程和物理问题的多个领域中，例如传感器网络、信号处理、偏微分方程、滤波、计算机断层扫描、最优控制、反问题以及机器学习等，这只是其中的一部分例子 [28], [20], [23], [9], [10], [21] 。考虑到可能有多个解的情况 (1) ，我们设法找到由函数 f 特征化的唯一解，即

$$f^* \stackrel{\text{def}}{=} \min_{x\in\mathbb{R}^n} f(x) \quad \text{subject to} \quad \boldsymbol{A}x = b, \tag{2}$$

带有强凸函数 f。然而，我们将不假设 f 的平滑性。一个可能的示例是 $f(x) = \lambda\cdot\|x\|_1 + \frac{1}{2}\|x\|_2^2$，并且已知对于适当选择的 $\lambda > 0$，此函数倾向于稀疏解，见 [2], [29], [13], [14], [24]。我们在本文中假设 m 和 n 都很大，并且系统是一致的。我们用 a_i^T 表示 $\boldsymbol{A}$ 的行，并假设对于所有 $i \in [m] := \{1,\ldots,m\}$ 都有 $a_i \neq 0$。由于 f 是强凸的，问题 (2) 有一个唯一的解 $\hat{x}$。在应用中，通常找到一个离 $\hat{x}$ 不太远的点就足够了。特别地，选择误差容限 $\varepsilon > 0$，并旨在找到满足 $\|x-\hat{x}\|_2 \le \varepsilon$ 的点 x。由于我们的方法将是一种随机方法，因此迭代 x 将是随机向量。因此，我们的目标是计算满足 $\mathbb{E}[\|x-\hat{x}\|_2] \le \varepsilon$ 的 (2) 的近似解，其中 $\mathbb{E}[\cdot]$ 表示关于算法随机性的期望。

A. 相关工作

线性系统 (1) 可能非常大，使得全矩阵操作成本非常高昂甚至不可行。因此，使用每次迭代计算和存储量较低的迭代算法来生成 (2) 的良好近似解似乎是可取的。Kaczmarz 方法 [11] 及其随机变体 [26], [8], [7], [16] 用于计算一致线性系统的最小 ℓ_2-范数解。在每次迭代 k 中，从系统 (1) 随机选择一个行向量 $a_i^\top$ 的 $\boldsymbol{A}$，并将当前迭代 x_k 投影到该方程的解空间以获得 x_{k+1}。请注意，此更新规则要求每次迭代的成本较低且存储量为 $\mathcal{O}(n)$ 阶。最近，一种新的随机 Kaczmarz（RK）方法的变体即随机稀疏 Kaczmarz 方法（RSK）[14], [24] 显示出在逼近大型一致线性系统的稀疏解时具有良好的性能，并且其几乎拥有相同的低计算成本和存储需求。论文 [13], [24] 通过将其解释为顺序随机 Bregman 投影方法（其中的 Bregman 投影是相对于函数 f 进行的）对这一方法进行了分析，而 [22] 则通过对偶性将它与坐标下降法联系起来。RSK 的各种变体包括块/平均变体 [18], [22], [16], [4], [15] , 平均方法 [27] 以及适应于最小二乘问题的方法在 [30], [3], [25] 中给出。在这些变体中，通常需要访问矩阵 $\boldsymbol{A}$ 的多行，以增加内存为代价。坐标下降法经常被用于最小化复合凸目标函数 [17], [19], [6] 的上下文中。在每次迭代中，它们仅更新变量向量的一个坐标，因此使用的是偏导数而非整个梯度。一种加速随机坐标下降方法在 [19] 中提出，适用于光滑函数。加速方法将邻近坐标下降转化为一

[1]Institute of Analysis and Algebra, TU Braunschweig, 38092 Braunschweig, Germany, `l.ngoupeyou-tondji@tu-braunschweig.de`, `d.lorenz@tu-braunschweig.de`.

[2]Automatic Control and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania, `ion.necoara@upb.ro`

[3]Gheorghe Mihoc-Caius Iacob Institute of Mathematical Statistics and Applied Mathematics of the Romanian Academy, 050711 Bucharest, Romania.

个具有最优 $\mathcal{O}(1/k^2)$ 复杂度的算法，在轻微增加计算成本的情况下 [19], [6]，其最优性差距减少为 $\mathcal{O}(1/k)$。

B. 贡献

据我们所知，加速的 Bregman-Kaczmarz 变体尚未在文献中被提出和分析。具体来说，我们做出了以下贡献：

- 除了将 Bregman-Kaczmarz 方法解释为对偶坐标下降外，我们提出了一种仅使用矩阵一行的加速变体。我们将该方法称为加速随机 Bregman-Kaczmarz 方法（ARBK），更多细节见算法 3。
- 通过利用原始更新和对偶更新之间的联系，我们获得了收敛性作为副产品，并且得到了迄今为止尚未获得的收敛速率。我们证明了我们的加速方法比其标准对应方法具有更快的收敛速度。
- 我们也通过实证验证了这一点，并在 `Python` 中提供了我们的算法实现。

C. 概述。

论文的其余部分组织如下。第 II 节提供了符号、凸性的简要概述和 Bregman 距离。在第 III 节中，我们陈述了我们的方法，并给出了它在对偶空间中的解释。第 IV 节为我们提出的方法提供了收敛性保证。在第 V 节中，数值实验展示了我们方法的有效性，并对其行为和超参数提供了见解。最后，第 VI 节得出了一些结论。

II. 符号和基本概念

对于整数 m，我们表示 $[m] \overset{\text{def}}{=} \{1,2,\ldots,m\}$。给定一个对称正定矩阵 $\boldsymbol{B}$,，我们用

$$\langle x,y\rangle_{\boldsymbol{B}} \overset{\text{def}}{=} \langle x,\boldsymbol{B}y\rangle = \sum_{i,j\in[n]} x_i\boldsymbol{B}_{ij}y_j, \quad x,y\in\mathbb{R}^n$$

表示诱导的内积，并用 $\|.\|_{\boldsymbol{B}}^2 \overset{\text{def}}{=} \langle\cdot,\cdot\rangle_{\boldsymbol{B}}$ 表示其诱导范数，并使用简写符号 $\|.\|_2$ 来表示 $\|.\|_{\boldsymbol{I}}$。对于一个 $n\times m$ 实矩阵 $\boldsymbol{A}$，我们分别用 $\mathcal{R}(\boldsymbol{A}), \|\boldsymbol{A}\|_F$ 和 $a_i^\top$ 表示其值域空间、其 Frobenius 范数和其第 i 行。用 e_i 表示单位矩阵 $\boldsymbol{I}_n\in\mathbb{R}^{n\times n}$ 的 i 列。对于依赖于随机索引 i 的随机向量 x_i（其中 i 以概率 p_i 被选择），我们表示为 $\mathbb{E}[x_i] \overset{\text{def}}{=} \sum_{i\in[q]} p_i x_i$，当概率分布从上下文中清楚时，我们将只写 $\mathbb{E}[x_i]$。给定一个向量 $x\in\mathbb{R}^n$，我们定义软阈值操作符，该操作符对一个向量 x 的每个分量进行作用。

$$\big(S_\lambda(x)\big)_j = \max\{|x_j|-\lambda,0\}\cdot\operatorname{sign}(x_j)\,. \tag{3}$$

现在我们收集一些关于凸性和 Bregman 距离的基本概念。设 $f:\mathbb{R}^n\to\mathbb{R}$ 是凸的（请注意，我们假设 f 在任何地方都是有限的，因此它也是连续的）。次微分的 f 在任何 $x\in\mathbb{R}^n$ 处由

$$\partial f(x) \overset{\text{def}}{=} \{x^*\in\mathbb{R}^n | f(y)\geq f(x)+\langle x^*,y-x\rangle, \forall\, y\in\mathbb{R}^n\},$$

定义，它是非空、紧致和凸的。函数 $f:\mathbb{R}^n\to\mathbb{R}$ 被称为 α 强凸，如果对于所有 $x,y\in\mathbb{R}^n$ 和次梯度 $x^*\in\partial f(x)$ 我们有

$$f(y)\geq f(x)+\langle x^*\,,\,y-x\rangle+\tfrac{\alpha}{2}\cdot\|y-x\|_2^2\,.$$

如果 f 是 α-强凸的，则 f 是强制性的，即

$$\lim_{\|x\|_2\to\infty} f(x)=\infty\,,$$

并且其芬切尔共轭 $f^*:\mathbb{R}^n\to\mathbb{R}$ 给出为

$$f^*(x^*) \overset{\text{def}}{=} \sup_{y\in\mathbb{R}^n}\langle x^*\,,\,y\rangle - f(y)$$

也是凸的、处处有限且强制性的。此外，f^* 是可微的，并具有常数 $L_{f^*}=\frac{1}{\alpha}$ 的利普希茨连续梯度，即对于所有 $x^*,y^*\in\mathbb{R}^n$，我们有

$$\|\nabla f^*(x^*)-\nabla f^*(y^*)\|_2\leq L_{f^*}\cdot\|x^*-y^*\|_2\,,$$

这蕴含了估计值

$$f^*(y^*)\leq f^*(x^*)-\langle\nabla f^*(x^*)\,,\,y^*-x^*\rangle+\tfrac{L_{f^*}}{2}\|x^*-y^*\|_2^2. \tag{4}$$

函数 f^* 被认为具有逐分量 Lipschitz 连续梯度如果

$$|\nabla_i f^*(x^*+he_i)-\nabla_i f^*(x^*)|\leq L_{f^*,i}\cdot|h|,$$

对于所有 $x^*\in\mathbb{R}^n, h\in\mathbb{R}, i\in[n]$.

定义 2.1: 布雷格曼距离 $D_f^{x^*}(x,y)$ 关于 $x,y\in\mathbb{R}^n$ 和 f 的子梯度 $x^*\in\partial f(x)$ 定义为

$$D_f^{x^*}(x,y) \overset{\text{def}}{=} f(y)-f(x)-\langle x^*\,,\,y-x\rangle\,.$$

Fenchel 的等式表明 $f(x)+f^*(x^*)=\langle x\,,\,x^*\rangle$ 如果 $x^*\in\partial f(x)$，并且意味着 Bregman 距离可以写为

$$D_f^{x^*}(x,y)=f^*(x^*)-\langle x^*\,,\,y\rangle+f(y)\,.$$

示例 *2.1:* [24] 目标函数

$$f(x) \overset{\text{def}}{=} \lambda \cdot \|x\|_1 + \tfrac{1}{2} \cdot \|x\|_2^2 \tag{5}$$

是强凸的，其常数为 $\alpha = 1$，它的共轭函数可以通过等式 (3) 中的软阈值算子计算得到

$$f^*(x^*) = \tfrac{1}{2} \cdot \|S_\lambda(x^*)\|_2^2, \quad \text{with} \quad \nabla f^*(x^*) = S_\lambda(x^*)\,.$$

它 Fenchel 共轭 f^* 具有逐分量 Lipschitz 梯度，常数为 $L_{f^*,i} = 1$，对于任意的 $x^* = x + \lambda \cdot s \in \partial f(x)$ 我们有

$$D_f^{x^*}(x, y) = \frac{1}{2} \cdot \|x - y\|_2^2 + \lambda \cdot (\|y\|_1 - \langle s\,,\, y\rangle)\,.$$

这给我们提供了 $D_f^{x^*}(x, y) = \frac{1}{2}\|x - y\|_2^2 \lambda = 0.$

以下不等式对于随机算法的收敛性分析至关重要。它们直接由 Bregman 距离的定义和 f 的强凸性假设得出，见 [12]。对于 $x, y \in \mathbb{R}^n$ 和 $x^* \in \partial f(x)$，$y^* \in \partial f(y)$ 我们有

$$\frac{\alpha}{2}\|x - y\|_2^2 \le D_f^{x^*}(x, y) \le \langle x^* - y^*\,,\, x - y\rangle, \tag{6}$$

III. 坐标下降和 Bregman-Kaczmarz 方法

注意通过适当的缩放 f，我们可以假设 $\alpha = 1$。因此，在后续部分中，我们考虑 1-强凸函数 f。特别是，使用 f 的共轭，我们可以将问题 (2) 的对偶函数写为:

$$\begin{aligned}\Psi(y) &= \inf_x \mathcal{L}(x, y) \\ &= \inf_x (f(x) - y^\top(\boldsymbol{A}x - b)) \\ &= b^\top y - f^*(\boldsymbol{A}^\top y),\end{aligned}$$

其中 $\mathcal{L}(x, y)$ 表示拉格朗日函数。(2) 的对偶问题是:

$$\Psi^* \overset{\text{def}}{=} \min_y \left[\Psi(y) := f^*(\boldsymbol{A}^\top y) - b^\top y\right] \tag{7}$$

原始最优解和对偶最优解 $\hat{x}$ 和 $\hat{y}$ 是通过

$$\boldsymbol{A}\hat{x} = b, \quad \hat{x} = \nabla f^*(\boldsymbol{A}^\top \hat{y}) \tag{8}$$

联系起来的，且 (7) 的最优解集 $\mathcal{Y}^*$ 非空。此外，对偶函数 Ψ 是无约束的、可微分的，并且其梯度由以下表达式给出

$$\nabla \Psi(y) = \boldsymbol{A}\nabla f^*(\boldsymbol{A}^\top y) - b.$$

另外，对偶函数的梯度 $\nabla\Psi$ 关于欧几里得范数 $\|\cdot\|_2$ 是 Lipschitz 连续和逐点 Lipschitz 连续的，常数分别为 $L_\Psi = \|A\|_2^2$ 和 $L_{\Psi,i} = \|a_i\|_2^2$。应用于 Ψ 的坐标下降更新从 (7) 读取 [17], [19]:

$$\begin{aligned}y_{k+1} &= y_k - \frac{1}{L_{\Psi,i}} e_i \nabla_i \Psi(y_k) \\ &= y_k - \frac{\langle a_i, \nabla f^*(\boldsymbol{A}^\top y_k)\rangle - b_i}{\|a_i\|_2^2} \cdot e_i\end{aligned}$$

通过以下关系:

$$x_k^* = \boldsymbol{A}^\top y_k, \quad x_k = \nabla f^*(x_k^*)$$

我们刚刚展示了对偶坐标下降迭代转化为 Bregman-Kaczmarz 迭代，如算法 1 所示。

Algorithm 1 Bregman-Kaczmarz 方法（BK）

choose $x_0 \in \mathbb{R}^n$ and set $x_0^* = x_0$.
Output: (approximate) solution of $\min_{\boldsymbol{A}x=b} f(x)$
初始化 $k = 0$
repeat
  选择一个行索引 $i_k = i \in [m]$（循环或随机）
  更新 $x_{k+1}^* = x_k^* - \frac{\langle a_i, x_k\rangle - b_i}{\|a_i\|_2^2} \cdot a_i$
  更新 $x_{k+1} = \nabla f^*(x_{k+1}^*)$
  增量 $k = k + 1$
until a stopping criterion is satisfied
return x_{k+1}

一种更通用的随机坐标下降版本，即 APPROX，在 [6] 中被提出以加速复合函数最小化的近似坐标下降方法，我们将其表述为算法 2。我们集中于通过关系 $c_k = A^\top v_k$ 将对偶迭代转换到原空间来构建我们的加速方案，并且这导致了算法 3。

备注 *3.1:* 我们已经在一个统一的框架中编写了这些算法以强调它们的相似性。实际实现只考虑两个变量: (c_k, t_k) 用于 ARKB 和 (v_k, z_k) 用于 ACD。尽管它们相似，这两种方法被用于两种不同的目的。算法 2 输出 y_k，这是对偶问题 (7) 的一个近似解，而算法 3 返回 x_k，这是我们原始问题 (2) 的一个近似解。

我们还使用了以下序列的关系，即算法 3 和 2 的迭代对于所有 $k \ge 1$ 满足，

$$x_k^* = \boldsymbol{A}^\top y_k, \quad x_k = \nabla f^*(x_k^*) \tag{9}$$

此外，通过设置对于所有 k 都有 $\theta_k = \theta_0$，可以从算法 3 恢复算法 1。在算法 1 中，设置 $f(x) = \frac{1}{2}\|x\|_2^2$ 给我们 RK 方法而 $f(x) = \lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$ 给我们 RSK 方法。

Algorithm 2 双重加速坐标下降法（ACD）

Input: Choose y_0 and set $z_0 = y_0$ and $\theta_0 = \frac{1}{m}$, $b \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$.
Output: (approximate) solution of $\min_y \Psi(y)$
初始化 $k = 0$
repeat
  更新 $v_k = (1-\theta_k)y_k + \theta_k z_k$
  随概率 $p_i = \|a_i\|_2^2/\|\boldsymbol{A}\|_F^2$ 选择行索引 $i_k = i \in [m]$
  更新 $z_{k+1} = z_k - \frac{a_{i_k}^\top \nabla f^*(A^\top v_k) - b_{i_k}}{m\theta_k \|a_{i_k}\|_2^2} e_{i_k}$
  更新 $y_{k+1} = v_k + m\theta_k(z_{k+1} - z_k)$
  更新 $\theta_{k+1} = \dfrac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
  增量 $k = k+1$
until a stopping criterion is satisfied
return y_{k+1}

IV. 加速随机 Bregman-Kaczmarz 方法的收敛结果

在本节中，我们首先回顾 APPROX（ACD）的基本收敛结果，这将在后面用于构建我们方法的收敛结果。我们首先回忆序列 $\{\theta_k\}$ 的以下性质。

引理 4.1: [5] 由 $\theta_0 \le 1$ 和 $\theta_{k+1} = \frac{\sqrt{\theta_k^4+4\theta_k^2}-\theta_k^2}{2}$ 定义的序列 (θ_k) 满足

$$\begin{gathered}\frac{(2-\theta_0)}{k+(2-\theta_0)/\theta_0} \le \theta_k \le \frac{2}{k+2/\theta_0},\\ \frac{1-\theta_{k+1}}{\theta_{k+1}^2} = \frac{1}{\theta_k^2}, \quad \forall k = 0,1,\dots \\ \theta_{k+1} \le \theta_k, \quad \forall k = 0,1,\dots\end{gathered} \tag{10}$$

引理 4.2: [6] 令 y_k, z_k 是由 ACD 生成的序列，$\theta_0 = \frac{1}{m}$ 和任意的 $\hat{y} \in \mathcal{Y}^*$。然后它满足

$$\begin{aligned}&\frac{1}{\theta_{k-1}^2}\mathbb{E}[\Psi(y_k) - \Psi^*] + \frac{1}{2\theta_0^2}\mathbb{E}[\|z_k - \hat{y}\|_{\boldsymbol{B}}^2]\\ &\le \frac{1-\theta_0}{\theta_0^2}(\Psi(y_0) - \Psi^*) + \frac{1}{2\theta_0^2}\|y_0 - \hat{y}\|_{\boldsymbol{B}}^2\end{aligned} \tag{11}$$

其中 $\boldsymbol{B} = \mathbf{Diag}(\|a_1\|_2^2, \|a_2\|_2^2, \dots, \|a_m\|_2^2)$，$\mathbf{Diag}(d_1, d_2, \dots, d_m)$ 表示对角线上有 $d_1, d_2, \dots, d_m$ 的对角矩阵。

以下引理给出了原始函数 f 与对偶函数 Ψ 之间的关系。

引理 4.3: 令 $b \in \mathcal{R}(\boldsymbol{A})$ 和 (x_k^*, x_k) 是一个序列，使得 $x_k = \nabla f^*(x_k^*)$。那么，对于任何 $y_k \in \mathbb{R}^m$ 满足

Algorithm 3 加速随机 Bregman Kaczmarz 方法（ARBK）

Input: Choose x_0^* and set $t_0 = x_0^*$ and $\theta_0 = \frac{1}{m}$, $b \in \mathbb{R}^m$, $\boldsymbol{A} \in \mathbb{R}^{m\times n}$.
Output: (approximate) solution of $\min_x f(x)$ s.t. $\boldsymbol{A}x = b$
初始化 $k = 0$
repeat
  更新 $c_k = (1-\theta_k)x_k^* + \theta_k t_k$
  按概率 $p_i = \|a_i\|_2^2/\|\boldsymbol{A}\|_F^2$ 选择行索引 $i_k = i \in [m]$
  更新 $t_{k+1} = t_k - \frac{1}{m\theta_k\|a_{i_k}\|_2^2}(a_{i_k}^\top \nabla f^*(c_k) - b_{i_k})a_{i_k}$
  更新 $x_{k+1}^* = c_k + m\theta_k(t_{k+1} - t_k)$
  更新 $\theta_{k+1} = \dfrac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
  增量 $k = k+1$
until a stopping criterion is satisfied
return $x_{k+1} = \nabla f^*(x_{k+1}^*)$

$x_k^* = \boldsymbol{A}^\top y_k$，它成立。

$$D_f^{x_k^*}(x_k, \hat{x}) = \Psi(y_k) - \Psi^*. \tag{12}$$

Proof: 根据定义 2.1 我们有：

$$\begin{aligned} D_f^{x_k^*}(x_k, \hat{x}) &= f^*(x_k^*) + f(\hat{x}) - \langle x_k^*, \hat{x}\rangle \\ &= f^*(A^\top y_k) - \langle b, y_k\rangle + f(\hat{x}) \\ &= \Psi(y_k) + f^*, \end{aligned}$$

并且由于 $\Psi^* = -\max -\Psi = -f^*$（通过强对偶性），该断言得证。 ■

回忆一下，对于我们提出的方法，我们感兴趣的是展示迭代 x_k 的收敛结果，这些迭代在方程 (9) 中给出，并不是对于迭代 y_k。通过证明 ACD 和 ARBK 方法是等价的，并使用最近的理论结果 [17], [19], [6] 和引理 4.3，我们作为副产品获得了以下 ARBK 的收敛结果。

定理 4.4: 令 x_k 为由 ARBK 生成的序列，并设集合 $\theta_k = \theta_0, \forall k$。则有：

$$\frac{1}{2}\mathbb{E}[\|x_k - \hat{x}\|_2^2] \le \mathbb{E}[D_f^{x_k^*}(x_k, \hat{x})] \le \frac{2\|\boldsymbol{A}\|_F^2}{k+4}R_0^2(y_0). \tag{13}$$

其中 $R_0(y_0) = \max_y\{\min_{\hat{y}\in\mathcal{Y}^*}\|y-\hat{y}\|_2 : \Psi(y) \le \Psi(y_0)\}$.

Proof: 此速率来源于坐标下降方法的经典结果 [19]，并应用了引理 4.3 和方程 (6)。 ■

定理 4.5: 令 x_k 由 ARBK 生成的序列，$\theta_0 = \frac{1}{m}$ 和任何 $\hat{y} \in \mathcal{Y}^*$。然后，有：

$$\mathbb{E}[D_f^{x_k^*}(x_k, \hat{x})] \leq \frac{4m^2}{(k-1+2m)^2} C_0, \tag{14}$$

$$\mathbb{E}[\|x_k - \hat{x}\|_2^2] \leq \frac{8m^2}{(k-1+2m)^2} C_0, \tag{15}$$

其中

$$C_0 = \left(1 - \frac{1}{m}\right) D_f^{x_0^*}(x_0, \hat{x}) + \frac{1}{2}\|y_0 - \hat{y}\|_{\boldsymbol{B}}^2$$

Proof: 此结果来自于引理 4.2，引理 4.3，方程 (6) 以及引理 4.1 中的第一个不等式。 ■

定理 4.5 表明算法 3 的迭代 x_k 以速率 $\mathcal{O}(1/k^2)$ 收敛，从而加速其标准对应算法 1，后者具有 $\mathcal{O}(1/k)$ 的收敛速度（参见。定理 4.4）. 据我们所知，加速 Kaczmarz 变体尚未被提出用于问题 (2)，且 ARBK 算法的收敛性保证来自定理 4.5 是新的。

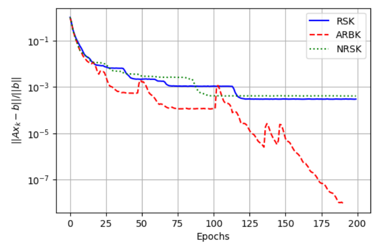
V. 实验

我们展示了几项实验以证明算法 3 在各种条件下的有效性。特别是，我们研究了稀疏参数 λ 和矩阵 $\boldsymbol{A}$ 的条件数 κ 的影响。模拟是在 Intel Core i7 计算机上进行的，该计算机拥有 16GB RAM，在 `Python` 上运行。对于所有实验我们考虑 $f(x) = \lambda \cdot \|x\|_1 + \frac{1}{2}\|x\|_2^2$，其中 λ 是稀疏参数，并且我们比较了 ARBK、算法 1（在这种情况下是随机稀疏 Kaczmarz 方法(RSK)）和 RSK 的 Nesterov 加速 (NRSK)[19, Method ACDM in Section 5]。实验的合成数据生成如下：数据矩阵 $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ 中的所有元素都是从标准正态分布 $\mathcal{N}(0,1)$ 独立同等地选取。我们构建了超定、方阵和欠定线性系统。为了构造稀疏解 $\hat{x} \in \mathbb{R}^n$，我们从标准正态分布 $\mathcal{N}(0,1)$ 中生成一个随机向量 y，并设定 $\hat{x} = S_\lambda(\boldsymbol{A}^\top y)$，它来自于方程 (8)，相应的右侧是 $b = \boldsymbol{A}\hat{x} \in \mathbb{R}^m$。对于每个实验，我们运行独立的试验，每次试验都从初始迭代 $x_0 = 0$ 开始。我们通过绘制平均相对残差误差 $\|\boldsymbol{A}x - b\|_2 / \|b\|_2$ 和平均误差 $\|x_k - \hat{x}\|_2 / \|\hat{x}\|_2$ 随 epoch 数的变化来衡量性能。

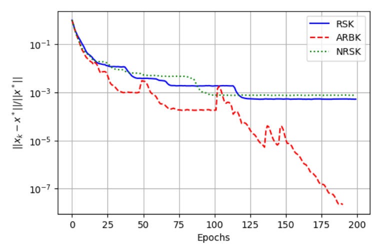
图 1 和图 2 显示了对于一个超定且一致的系统的结果，其中使用了值 $\lambda = 30$。请注意，通常的 RSK 和 NRSK 变体表现一直良好。此外，我们通过实验观察到 ARBK 方法给我们带来了更快的收敛速度。

图 3 和图 4 分别显示了一个适定和不适定的欠定且一致系统的结果，其中使用了值 $\lambda = 30$。所有方法都利用了向量 $\hat{x}$ 是稀疏的事实。此外，在图 2 中，RSK 和 NRSK 方法没有像 ARBK 方法那样快速减小残差。

图 5 报告了 RSK、NRSK 和 ARBK 在 Tensorflow 框架中可用的 Mnist 数据集上的性能 [1]。我们随机选择一个数据点并将其视为我们的 $\hat{x}$。我们使用一个欠定矩阵 A，并展示了相对残差、相对误差以及 4 个图像，这些图像对应于原始图像和使用不同方法进行重建的图像。



(a) 相对残差



(b) 错误

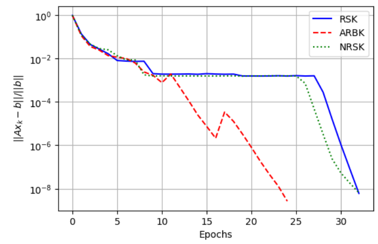
Fig. 1: 随机稀疏 Kaczmarz 方法（蓝色）、Nesterov 加速方案（绿色）和 ARBK 方法（红色）的比较，$m = 700, n = 700$，稀疏度 $s = 182$，$\lambda = 30$，$\kappa(A) = 1150$。

VI. 结论

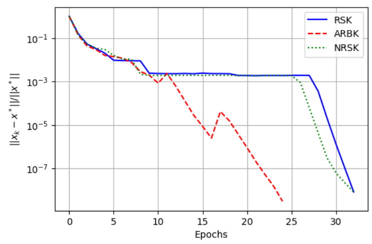
在本文中，我们证明了加速随机 Bregman-Kaczmarz 方法（算法 3）的迭代对于一致线性系统以速率 $\mathcal{O}(1/k^2)$ 在期望下收敛。数值实验表明该方法（算法 3）在 λ 的一系列值范围内表现一致良好，随着 λ 的增加提供了非常好的重建质量，并展示了使用此方法恢复线性系统稀疏解的好处。

致谢

导致这些结果的研究得到了以下资金支持：由欧盟的地平线 2020 研究与创新计划玛丽 · 斯克沃多夫斯卡-居里奖学金协议编号 861137 资助的 ITN-ETN 项目 TraDE-OPT；NO Grants 2014-2021，RO-NO-2019-0184，在 ELO-Hyp 项目下，合同编号 24/2020；

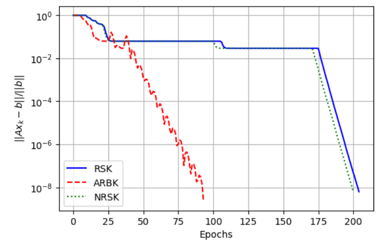


(a) 相对残差

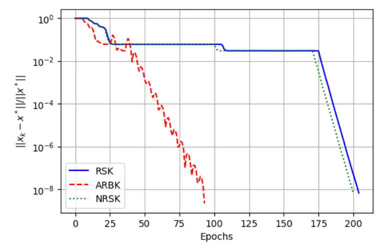


(b) 错误

Fig. 2: 随机稀疏 Kaczmarz 方法（蓝色）、Nesterov 加速方案（绿色）和 AR BK 方法（红色）的比较，$m = 900, n = 200$, 稀疏性 $s = 65$，$\lambda = 30$，$\kappa(A) = 2.70$。



(a) 相对残差



(b) 错误

Fig. 3: 随机稀疏 Kaczmarz（蓝色）、Nesterov 加速方案（绿色）和 ARbk 方法（红色）的比较，$m = 500, n = 784$, 稀疏性 $s = 7$，$\lambda = 60$，$\kappa(A) = 8.98$。

UEFISCDI PN-III-P4-PCE-2021-0720，在 L2O-MOC 项目下，编号 70/2022。

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[2] Jian-Feng Cai, Stanley Osher, and Zuowei Shen. Linearized Bregman iterations for compressed sensing. *Mathematics of Computation*, 78(267):1515–1536, 2009.

[3] Kui Du. Tight upper bounds for the convergence of the randomized extended Kaczmarz and Gauss–Seidel algorithms. *Numerical Linear Algebra with Applications*, 26(3):e2233, 2019.

[4] Kui Du, Wu-Tao Si, and Xiao-Hui Sun. Randomized extended average block Kaczmarz for solving least squares. *SIAM Journal on Scientific Computing*, 42(6):A3541–A3559, 2020.

[5] Olivier Fercoq and Zheng Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358*, 2016.

[6] Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[7] Robert M. Gower, Denali Molitor, Jacob Moorman, and Deanna Needell. On adaptive sketch-and-project for solving linear systems. *SIAM Journal on Matrix Analysis and Applications*, 42(2):954–989, 2021.

[8] Robert M Gower and Peter Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.

[9] Godfrey N Hounsfield. Computerized transverse axial scanning (tomography): Part 1. description of system. *The British journal of radiology*, 46(552):1016–1022, 1973.

[10] Yuling Jiao, Bangti Jin, and Xiliang Lu. Preasymptotic convergence of randomized Kaczmarz method. *Inverse Problems*, 33(12):125012, 21, 2017.

[11] S. Kaczmarz. Angenäherte Auflösung von Systemen linearer Gleichungen. *Bull. Internat. Acad. Polon. Sci. Lettres A*, pages 355–357, 1937.

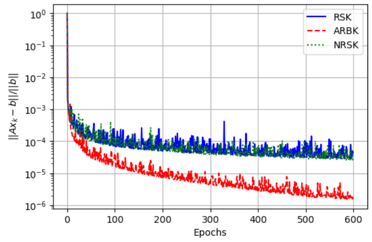
[12] D. A. Lorenz, F. Schöpfer, and S. Wenger. The linearized Bregman method via split feasibility problems: Analysis and generalizations. *SIAM J. Imaging Sciences*, 7(2):1237–1262, 2014.

[13] Dirk A Lorenz, Frank Schopfer, and Stephan Wenger. The linearized Bregman method via split feasibility problems: analysis and generalizations. *SIAM Journal on Imaging Sciences*, 7(2):1237–1262, 2014.

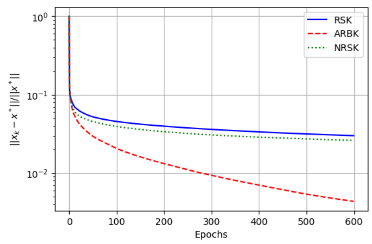
[14] Dirk A Lorenz, Stephan Wenger, Frank Schöpfer, and Marcus Magnor. A sparse Kaczmarz solver and a linearized Bregman method for online compressed sensing. In *2014 IEEE international conference on image processing (ICIP)*, pages 1347–1351. IEEE, 2014.

[15] Cun-Qiang Miao and Wen-Ting Wu. On greedy randomized average block Kaczmarz method for solving large linear systems. *Journal of Computational and Applied Mathematics*, 413:114372, 2022.

[16] Ion Necoara. Faster randomized block Kaczmarz algorithms. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1425–1452, 2019.

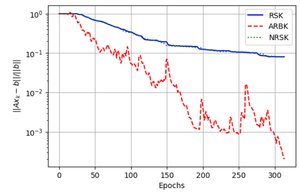


(a) 相对残差

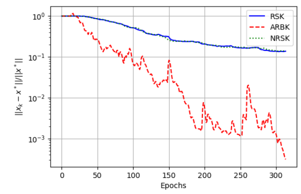


(b) 错误

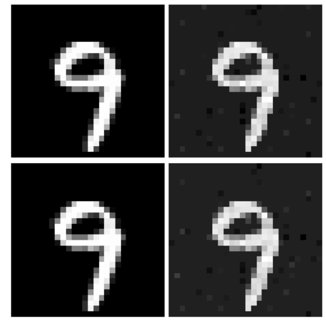
Fig. 4: 随机稀疏 Kaczmarz 方法（蓝色）、Nesterov 加速方案（绿色）和 ARbk 方法（红色）的比较，$m = 300, n = 900$, 稀疏度 $s = 231$，$\lambda = 15$，$\kappa(A) = 2990$。



(a) 相对残差



(b) 错误



(c) 重建

Fig. 5: 随机稀疏 Kaczmarz（RSK）（蓝色）、Nesterov 加速方案（NRSK）（绿色）和 ARBK 方法（红色）的比较。顶部从左到右，原始图片，RSK 的重建结果。底部从左到右，ARBK 的重建结果和 NRSK 的重建结果。$m = 500, n = 784$, 稀疏度 $s = 135$, $\lambda = 30$, $\kappa(A) = 8.98$.

[17] Ion Necoara and Dragos Clipici. Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds. *SIAM Journal on Optimization*, 26(1):197–226, 2016.

[18] Deanna Needell and Joel A Tropp. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra and its Applications*, 441:199–221, 2014.

[19] Yu Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[20] Maxim A Olshanskii and Eugene E Tyrtyshnikov. *Iterative methods for linear systems: theory and applications*. SIAM, 2014.

[21] Andrei Patrascu and Ion Necoara. Nonasymptotic convergence of stochastic proximal point methods for constrained convex optimization. *The Journal of Machine Learning Research*, 18(1):7204–7245, 2017.

[22] Stefania Petra. Randomized sparse block Kaczmarz as randomized dual block-coordinate descent. *Analele Stiintifice Ale Universitatii Ovidius Constanta-Seria Matematica*, 23(3):129–149, 2015.

[23] J. C. Rabelo, Y. F. Saporito, and A. Leitão. On stochastic Kaczmarz type methods for solving large scale systems of ill-posed equations. *Inverse Problems*, 38(2):Paper No. 025003, 23, 2022.

[24] Frank Schöpfer and Dirk A Lorenz. Linear convergence of the randomized sparse Kaczmarz method. *Mathematical Programming*, 173(1):509–536, 2019.

[25] Frank Schöpfer, Dirk A Lorenz, Lionel Tondji, and Maximilian Winkler. Extended randomized Kaczmarz method for sparse least squares and impulsive noise problems. *Lineare Algebra and Applications*, 652:132–154, 2022.

[26] Thomas Strohmer and Roman Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009.

[27] Lionel Tondji and Dirk A Lorenz. Faster randomized block sparse Kaczmarz by averaging. *Numerical Algorithms*, pages 1–35, 2022.

[28] Joel A Tropp. Improved analysis of the subsampled randomized Hadamard transform. *Advances in Adaptive Data Analysis*, 3(01n02):115–126, 2011.

[29] Wotao Yin. Analysis and generalizations of the linearized Bregman method. *SIAM Journal on Imaging Sciences*, 3(4):856–877, 2010.

[30] Anastasios Zouzias and Nikolaos M Freris. Randomized extended Kaczmarz for solving least squares. *SIAM Journal on Matrix Analysis and Applications*, 34(2):773–793, 2013.