Kaczmarz Algorithm, row action methods, and statistical learning algorithms

Xuemei Chen

Abstract. The Kaczmarz algorithm is an iterative row action method that typically solves an overdetermined linear system. The randomized Kaczmarz algorithm, which was introduced a decade ago, revived this simple algorithm and raised a lot of interest in this community. It has come to people's attention that there is overlap between many other iterative methods, statistical learning algorithms and the (randomized) Kaczmarz algorithm. This note brings these methods together and discuss connections and theories, with a focus on the convergence rate of the deterministic Kaczmarz algorithm. Moreover, the convergence under noise for the deterministic case is discussed in Theorem 3.2.

1. Introduction

The original Kaczmarz algorithm [**23**] is an iterative algorithm that solves a consistent and overdetermined system $Ax = b$, where the size of A is $m \times d$. Let a_i be the ith row of A, and b_i be the ith coordinate of b, then the algorithm is as follows:

$$\begin{aligned} &\text{Initialize } x^{(0)} \\ &x^{(k+1)} = x^{(k)} + \alpha_k \frac{b_{i(k)} - \langle a_{i(k)}, x^{(k)} \rangle}{\|a_{i(k)}\|^2} a_{i(k)}, k \geq 0 \\ &\text{with } i(k) = (k \mod m) + 1. \end{aligned} \tag{1.1}$$

We define hyperplane $H_i = \{x : \langle a_i, x \rangle = b_i\}$. At each iteration, $x^{(k+1)}$ is the projection of $x^{(k)}$ onto the convex set $H_{i(k)}$ (if the relaxation parameter $\alpha_k = 1$). The choice of $i(k)$ here allows the algorithm iterate through all the rows sequentially and then start over. Figure 1 shows 8 iterations of the Kaczmarz algorithm for solving a 4×2 system.

The Kaczmarz algorithm is a *row action method* where only one row is used in each iteration [**6**]. Due to its simplicity, the Kaczmarz method has found numerous applications including image reconstruction, distributed computation, signal processing, etc. [**6, 13, 18**]. The Kaczmarz algorithm has also been rediscovered in

2010 *Mathematics Subject Classification.* Primary: 65K15, 75S60; Secondary: 15A60 .

Key words and phrases. Kaczmarz, randomized Kaczmarz, stochastic gradient descent, row action method, relaxation method.

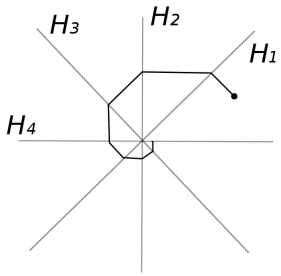


FIGURE 1

the field of image reconstruction and called ART (Algebraic Reconstruction Technique) [**16**]. More work on the (deterministic) Kaczmarz algorithm can be found in [**38**], [**6**], and [**11**].

It is well known that the Kaczmarz algorithm produces monotonically improving approximations as the iteration number increases. However, it can be difficult to quantify the associated rates of convergence, and in some cases, the convergence can be quite slow. To tackle this issue, Strohmer and Vershynon [**37**] introduced a randomized version. For the *randomized Kaczmarz algorithm*, rather than processing each row cyclically, at each iteration, a row of A is randomly selected to perform the projection, as

$$\begin{aligned} &\text{Initialize } x^{(0)} \\ &x^{(k+1)} = x^{(k)} + \frac{b_{i(k)} - \langle a_{i(k)}, x^{(k)} \rangle}{\|a_{i(k)}\|^2} a_{i(k)}, k \geq 0 \\ &\text{with } i(k) \text{ picking from } \{1, 2, \cdots, m\} \text{ with probability } \|a_i\|^2 / \|A\|_F^2. \end{aligned} \tag{1.2}$$

Moreover, the relaxation parameter has been set to 1. Throughout this paper, $\|\cdot\|$ indicates the Euclidean norm and $\|\cdot\|_F$ the Frobenius norm of a matrix. Moreover, cond$(\cdot)$ is the condition number of a matrix and $\sigma(\cdot)$ denotes the smallest singular value of a matrix. In contrast to the randomized Kaczmarz algorithm (1.2), we will call algorithm (1.1) the *cyclic Kaczmarz algorithm*.

Strohmer and Vershynin proved that this randomized approach achieves mean squared error with a rate that is quantifiable as

$$\mathbb{E}\|x^* - x^{(k)}\|^2 \leq \left(1 - \frac{\sigma^2(A)}{\|A\|_F^2}\right)^k \|x^* - x^{(0)}\|^2, \tag{1.3}$$

where x^* is the solution of the system.

Following [**37**], there has been a great amount of work that highlights other favorable properties of the Kaczmarz algorithm. We will only name a few here. The work [**12**] accelerates the convergence of the Kaczmarz algorithm in high dimensions with help of the Johnson-Lindenstrauss Lemma. The work [**9**] discusses the almost sure convergence when the measurement or data is drawn from a more general random distribution. The work in [**33**] and [**10**] extends to processing multiple rows in one iteration. The work in [**41**] deals with inconsistent systems and develops a variation of the randomized Kaczmarz algorithm so that the approximates will converge to the least square solution with a better rate. The paper [**28**] further deals with the underdetermined (A does not have full rank) case (cf. [**39**]). There is

also work to apply the Kaczmarz algorithm to compressed sensing [**27**] and to phase retrieval [**40**]. However, there has been relatively less interest in the convergence rate of the cyclic (deterministic) method, which will be a focus of this note.

With this increased interest in the Kaczmarc algorithm, researchers have found many connections with other row action methods, which will be elaborated in Section 2. Moreover, the ability of reading measurements as it becomes available [**9**] is very reminiscent of online machine learning. As it turns out, the Kaczmarz algorithm (both the cyclic and randomized version) can be viewed as an instance of some gradient methods. This will be discussed in Section 4. This note will first provide a survey of all these connections.

The second contribution of this paper is to analyze the convergence rate of the cyclic Kaczmarz algorithm under noise in Section 3. See Theorem 3.2. Comparison of the current noise-free deterministic Kaczmarz convergence rates are also given, and Theorem 3.2 will recover one of the those results.

2. Connection with projection methods and row action methods

Row action methods are iterative methods such that only one row of the matrix A is utilized in each iteration. Such methods are often applied to large scale and especially sparse systems due to its little computational work per iteration. But this is also a liability as the rate of convergence could be dismally low [**14, 29**]. This liability is the motivation for the randomized Kaczmarz algorithm. In this section we will discuss other row action methods and how they can be reduced to the Kaczmarz algorithm. The randomized versions of these row action methods, to speed up the convergence, are also discussed.

2.1. Projection onto Convex Sets. The method of alternating projection (MAP) is a broader concept than the row action methods. It is an iterative scheme for finding the best approximation to any given point in a Hilbert space from the intersection of a finite collection of closed subspaces. We let $M_1, M_2, \cdots, M_k$ be k closed subspaces in the Hilbert space X, and let P_{M_i} denote the orthogonal projection onto M_i. Both Von Neuman [**35**] in 1950 and Halperin [**20**] in 1962 proved the following theorem:

$$\lim_{n\to\infty} \| \left(P_{M_k} P_{M_{k-1}} \cdots P_{M_1}\right)^n (x) - P_{\cap_i^k M_i}(x)\| = 0, \tag{2.1}$$

Although no specific convergence rate is given. The cyclic Kaczmarz algorithm (1.1) is a special case here where $M_i = H_i$.

A more general setting was considered later. The projection onto convex sets (POCS) algorithm aims to find a point in the intersection of convex sets Q_i, which is a very common problem in diverse area of mathematics and physical sciences. The POCS algorithm was first introduced by Bregman [**5**] in 1966 with a very general projection function where they discussed application to convex programming. Around the same time, Gubin, Polyak and Raik [**17**] introduced their successive projection algorithm. The algorithm starts with a random guess $x^{(0)}$, and is iterative as

$$x^{(k+1)} = x^{(k)} + \alpha_k (P_{i(k)} x^{(k)} - x^{(k)}),$$

where P_j is the projection onto Q_j, and $i(k)$ indicates the order of which convex sets are selected. For example, we can choose the common cyclic control as $i(k) = (k$

mod m) + 1. It is obvious that the Kaczmarz algorithm is a particular instance of POCS algorithm where each convex set is the affine hyperplane.

Gubin et al. [**17**] showed that the sequence $x^{(k)}$ converges to some point in the intersection when the relaxation parameter $\alpha_k \in (\epsilon_1, 2-\epsilon_2)$. They also noticed that, however, the rate of convergence can be slow in certain cases. The authors talked about the special case for solving a system of linear inequalities, which becomes the relaxation method (See Section 2.2).

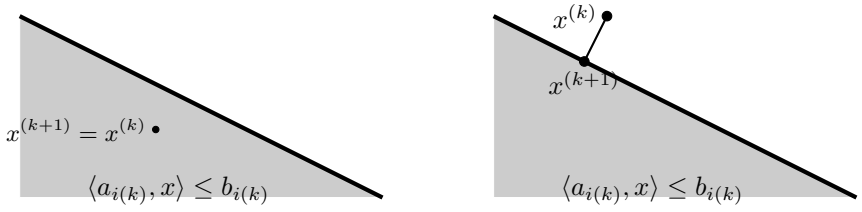
Bauschke and Borwein [**2**] wrote a general overview of projection algorithms for the consistent case. Censor and Tom discussed a variation of POCS in [**8**], especially for the inconsistent case. Numerous applications can be found in [**6**], and throughout this paper.

2.2. The Relaxation Method. The *relaxation method* for linear inequalities, as introduced in 1954 by Agmon [**1**], Motzkin and Schoenberg [**31**], is to find solutions of linear inequalities by orthogonally projecting the current iterate onto chosen halfspace, and therefore is a particular instance of POCS.

Given a linear system of inequalities $H_i = \{x : \langle a_i, x\rangle \leq b_i\}_{i=1}^m$, a solution of this system can be found by

$$
\begin{aligned}
&\text{Initialize } x^{(0)} \\
&c_k = \min\{0, \alpha_k \frac{b_{i(k)} - \langle a_{i(k)}, x^{(k)}\rangle}{\|a_{i(k)}\|^2}\}, \\
&x^{(k+1)} = x^{(k)} + c_k a_{i(k)}, \\
&\text{with } i(k) = (k \mod m) + 1.
\end{aligned} \tag{2.2}
$$

If we let the relaxation parameter $\alpha_k = 1$, the next iterate is simply projecting the current iterate onto the half space $H_{i(k)} = \{x : \langle a_{i(k)}, x\rangle \leq b_{i(k)}\}$: if $x^{(k)}$ has already satisfied the linear constraint, then simply do nothing; otherwise, an orthogonal projection is performed (see Figure 2). The algorithm (2.2) also chooses rows cyclically. Other common implementations include almost cyclic (slightly more general than cyclic) [**24**, Definition 2.3], maximal distance and maximal residual control [**1**].



(A) (B)

FIGURE 2. The Relaxation Method

In 1984, Mandel [**29**] proved that $\{x^{(k)}\}$ of (2.2) with $\alpha_k = \alpha \in (0, 2)$ converges to a point $\hat{x}$ on the boundary of $P = \{x : Ax \leq b\}$. This can be viewed as a particular instance of the convergence theorem by Gubin et al. [**17**, Theorem 1] as mentioned in the previous section. The more significant contribution of Mandel is to prove a quantifiable linear convergence rate:

THEOREM 2.1 ([**29**, Theorem 3.1]). *If $P \neq \emptyset$, then the iterates of* (2.2) *satisfy*

$$d^2(x^{(k+m)}, P) \leq \left(1 - \frac{\alpha(2-\alpha)\mu^2}{1+(m-1)\alpha^2}\right) d^2(x^{(k)}, P),$$

where $d(x, P)$ is the distance of x to P and

$$\mu = \inf_{x \notin P} \frac{\max_i d(x, H_i)}{d(x, P)}. \tag{2.3}$$

The relaxation method is a generalization of the Kaczmarz method in dealing with inequalities. As a result, Mandel applied the convergence result to $Ax = b$ and got the first convergence rate result of the Kaczmarz algorithm [**29**, Corollary 4.3].

Same as the Kaczmarz algorithm, $i(k)$ can be chosen cyclically or randomly. When chosen randomly, Leventhal et al [**25**] proved that the mean square error also converges with a linear rate, and therefore generalized the result of (1.3).

The Perceptron convergence theorem from the theory of machine learning is a rediscovery of the relaxation method; see Minsky and Papert [**30**, p. 248] and Nilsson [**36**]. More connections with machine learning is discussed in Section 4.2.

2.3. The Hildreth's Algorithm. The Hildreth's algorithm also solves a system of linear inequalities, but with one more benefit: finding the closest point in the solution set to a given point, i.e., it solves the following problem

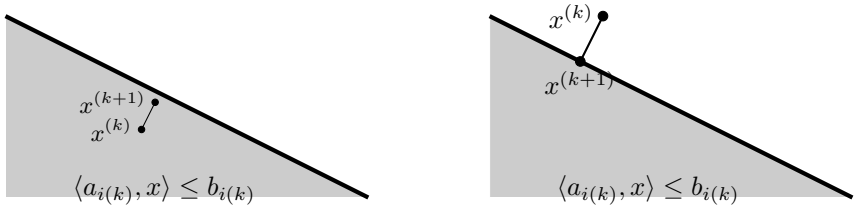
$$\begin{aligned} &x^* = \arg\min \|x - x^{(0)}\| \\ &\text{subject to } Ax \leq b. \end{aligned} \tag{2.4}$$

The algorithm is slightly more complicated than the relaxation method. It is defined as

$$\begin{aligned} &\text{Initialize } x^{(0)}, z^{(0)} = 0 \\ &c_k = \min\{z_{i(k)}^{(k)}, \alpha_k \frac{b_{i(k)} - \langle a_{i(k)}, x^{(k)} \rangle}{\|a_{i(k)}\|^2}\}, \\ &x^{(k+1)} = x^{(k)} + c_k a_{i(k)} \\ &z_i^{(k+1)} = \begin{cases} z_i^{(k)}, & i \neq i(k) \\ z_{i(k)}^{(k)} - c_k, & i = i(k) \end{cases} \\ &\text{with } i(k) = (k \mod m) + 1. \end{aligned} \tag{2.5}$$

Figure 3 shows the nice geometric interpretation when $\alpha_k = 1$. If the constraint $\langle a_{i(k)}, x \rangle \leq b_{i(k)}$ is satisfied, we move the last approximate closer to the hyperplane $\{x : \langle a_{i(k)}, x \rangle = b_{i(k)}\}$ (Figure 3 (A)). The definition of c_k guarantees no over projection (Staying on the halfspace). This is what's different from the relaxation method. If the constraint is violated, we project the last approximate to the halfspace, just like in the relaxation method (Figure 3 (B)). The vector $z^{(k)}$ only gets updated at ith component, where i is the index of the row/constraint in that iteration. It can be shown that all entries of $z^{(k)}$ are never negative [**24**]. Hildreth's method is a primal-dual optimization method, where $z^{(k)}$ is the sequence of dual iterates.

The algorithm was first presented by Hildreth [**19**] in 1957. Lent and Censor [**24**] studied it extensively in 1980 and supplied a proof of convergence of the



(A) (B)

FIGURE 3. Hildreth's Method

Hildreth's algorithm under almost cyclic control. Iusem and De Pierro [**21**] further provided a convergence rate under almost cyclic control in 1990.

THEOREM 2.2 ([**21**, Theorem 1]). *Let x^* and $x^{(k)}$ be given in* (2.4) *and* (2.5)*, and the relaxation parameter $\alpha_k = \alpha \in (0, 2)$, then after one cycle,*

$$\|x^{(k+m)} - x^*\|^2 \leq \frac{1}{1 + \frac{(2-\alpha)\alpha\mu^2}{1+\alpha^2(m-1)}} \|x^{(k)} - x^*\|^2,$$

where the quantity μ is very similar to the one in (2.3)*, but defined slightly differently.*

The Hildreth's algorithm can be once again reduced to the Kaczmarz algorithm since $\langle a_i, x\rangle = b_i$ is equivalent to $\langle a_i, x\rangle \leq b_i$ and $\langle -a_i, x\rangle \leq -b_i$.

If i is chosen at random from $\{1, 2, \cdots, m\}$ with the distribution $\Pr(i = k) = \|a_i\|^2/\|A\|_F^2$, then the resulted method is called *the randomized Hildreth's algorithm* as proposed in [**22**]. Jamil et al. show that the randomized version also enjoys a linear convergence rate similar to the randomized Kaczmarz algorithm (1.3) (see [**22**, Theorem 4.5]).

2.4. Schwarz Iterative Method. The *Schwarz iterative method* (or subspace correction method) is for solving symmetric positive semi-definite linear systems [**39**]. Consider a separable Hilbert space V, let $a(\cdot, \cdot)$ be a continuous symmetric positive sesqui-linear form on V, and let F be a bounded linear functional on V. The method aims to solve the variational problem: Find $u \in V$ such that $a(u, v) = F(v)$. The concept of stable space splittings is used here. Let V_a be the Hilbert space with the scalar product given by the form $a(\cdot, \cdot)$. We will represent V_a by a finite number of Hilbert spaces V_{a_i} with associated scalar products a_i and corresponding linear bounded operators $R_i : V_{a_i} \to V_a$. We define the linear operators $T_i : V_a \to V_{a_i}$ via $a_i(T_i v, v_i) = a(v, R_i v_i)$. The additive Schwarz iteration is given by

$$x^{(k+1)} = x^{(k)} + w_k \sum_{i=1}^{N} R_i T_i (u - x^{(k)}), \tag{2.6}$$

and the sequential Schwarz iteration is

$$x^{(k+1)} = x^{(k)} + w_l R_{i(k)} T_{i(k)} (u - x^{(k)}). \tag{2.7}$$

The interested reader can find details on how to compute $u - x^{(k)}$ in [**39**]. The additive Schwarz iteration could be thought of the generalization of the gradient method

(4.3) and the sequential Schwarz iteration should be compared to the Stochastic gradient descent (4.4) with minibatch size 1.

For a system $Ax = b$, set $V = \text{range}(A^T)$ with $a(x, x) = x^T x$. Let $V_{a_i} = \mathbb{C}, a_i(y_i, y_i) = \|a_i\|^2 |y_i|^2, R_i y_i = y_i a_i$, then (2.7) becomes the Kaczmarz algorithm or the randomized version depending on the ordering of the rows. Osward and Zhou [**39**] analuze the convergence rate of the Schwarz iterative methods, and in return achieves a convergence rate of both versions of the Kaczmarz methods. We will list the cyclic version here in comparison to Theorem 2.1 and Theorem 2.2.

THEOREM 2.3. *if x^* is the only solution of $Ax = b$, and the relaxation parameter α_k is chose properly, then the iterates of* (1.1) *satisfy*

$$\|x^* - x^{(k+m)}\|^2 \leq \left(1 - \frac{1}{\log(2m)C^2(B)}\right) \|x^* - x^{(k)}\|^2, \tag{2.8}$$

where B is the matrix one obtains after normalizing each row of A, and $C(B)$ is the condition number of B.

3. Convergence rate of Kaczmarz algorithm under noise

3.1. Summary of deterministic convergence results. Dai and Schön [**11**] also studied the cyclic Kaczmarz convergence result with $\alpha_k = \alpha$:

THEOREM 3.1 ([**11**, Theorem 1]). *if x^* is the only solution of $Ax = b$, then the iterates of* (1.1) *satisfy*

$$\|x^* - x^{(k+m)}\|^2 \leq \left(1 - \frac{\alpha(2-\alpha)}{(2+\alpha^2 m)\sigma(B)^2}\right) \|x^* - x^{(k)}\|^2, \tag{3.1}$$

where B is the matrix one obtains after normalizing each row of A, and $\sigma(B)$ is the smallest singular value of B.

Proofs of both Theorem 2.1 and Theorem 2.2 can be applied to the cyclic Kaczmarz algorithm, with the setup

$$\begin{aligned} &x^* \text{ is the only solution to } Ax = b \\ &H_i = \{x : \langle a_i, x\rangle = b_i\} \\ &\mu = \inf_{x \neq x^*} \frac{\max_i d(x, H_i)}{d(x, x^*)}. \end{aligned} \tag{3.2}$$

We summarize the results in Table 1 where we assume the rows of A are normalized, i.e. $A = B$. The first column is the linear rates presented in the theorems and the second column is the linear rates with the relaxation parameter optimized. The rates (I) and (II) are quite similar. The reader can refer to [**11**] for a comparison of rates (I), (III) and (IV).

3.2. Convergence rate of deterministic Kaczmarz under noise. In this section, we consider the case where the measurements b_i are slightly perturbed. The stability of randomized Kaczmarz algorithm was studied in [**32**], but there is no result on noisy case for the deterministic (cyclic) Kaczmarz algorithm. The following theorem uses proof techniques from [**21**].

THEOREM 3.2. *Let x^* be the solution of $Ax = b$. If the system $Ax = b$ is perturbed as $\langle a_i, x\rangle = b_i + \epsilon_i, i \in [m]$, then the iterates of* (1.1) *with $\alpha_k = 1$ follows*

	linear rates	Optimal α
Mandel, Theorem 2.1	$1-\frac{\alpha(2-\alpha)\mu^2}{1+(m-1)\alpha^2}$	$1-\frac{2\mu^2}{1+\sqrt{4m-3}}$ (I)
Hildreth, Theorem 2.2	$\frac{1}{1+\frac{(2-\alpha)\alpha\mu^2}{1+\alpha^2(m-1)}}$	$\frac{1}{1+\frac{2\mu^2}{1+\sqrt{4m-3}}}$ (II)
Osward, Theorem 2.3	$1-\frac{1}{\log(2m)C^2(B)}$	$1-\frac{1}{\log(2m)C^2(B)}$ (III)
Dai, Theorem 3.1	$1-\frac{\alpha(2-\alpha)}{(2+\alpha^2m)\sigma(B)^2}$	$1-\frac{\sqrt{2}}{4m\sigma(B)^2}$ (IV)

TABLE 1. Comparison of cyclic Kaczmarz convergence rate

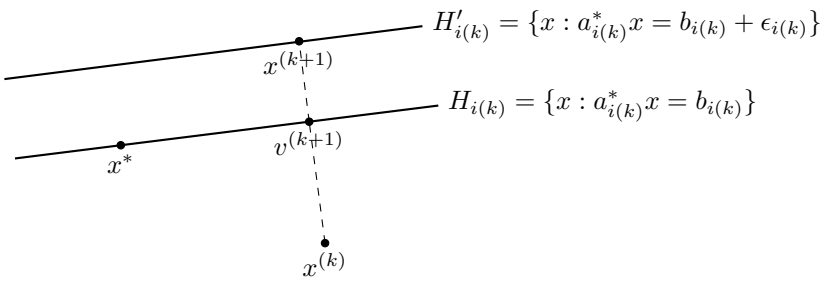
(3.3)

$$\|x^{(k+m)}-x^*\|^2 \le \frac{m}{m+\mu^2}\|x^{(k)}-x^*\|^2+\frac{m(-\sum_{l=1}^m \frac{\epsilon_l^2}{\|a_l\|^2}+\sum_{l=k}^{k+m-1}\frac{2\epsilon_{i(l)}\|x^{(l)}\|}{\|a_{i(l)}\|})+\|\sum_{l=1}^m \frac{\epsilon_l}{\|a_l\|^2}a_l\|^2}{m+\mu^2},$$

where $i(l)$ is defined as in (1.1) *and μ as in* (3.2).

PROOF. Without loss of generality, we assume $x^*=0$.

For an arbitrary k, $x^{(k+1)}$ is the projection of $x^{(k)}$ onto the perturbed hyperplane $H'_{i(k)}=\{x: a_{i(k)}^* x = b_{i(k)}+\epsilon_{i(k)}\}$. Suppose $v^{(k+1)}$ is the projection of $x^{(k)}$ onto $H_{i(k)}$. See the picture below.



The orthogonality gives

$$\|v^{(k+1)}\|^2+\|v^{(k+1)}-x^{(k)}\|^2=\|x^{(k)}\|^2 \tag{3.4}$$

and a simple calculation reaches

$$x^{(k+1)}=v^{(k+1)}+\frac{\epsilon_{i(k)}}{\|a_{i(k)}\|^2}a_{i(k)}. \tag{3.5}$$

(3.4) and (3.5) together implies

(3.6)

$$\|x^{(k)}\|^2-\|x^{(k+1)}-\frac{\epsilon_{i(k)}}{\|a_{i(k)}\|^2}a_{i(k)}-x^{(k)}\|^2=\|x^{(k+1)}-\frac{\epsilon_{i(k)}}{\|a_{i(k)}\|^2}a_{i(k)}\|^2 \ge \left(\|x^{(k+1)}\|-\frac{\epsilon_{i(k)}}{\|a_{i(k)}\|}\right)^2.$$

Let the index in (3.6) run from k to $k+m-1$ and add all these terms up we get

(3.7)

$$\|x^{(k)}\|^2-\sum_{l=k}^{k+m-1}\|x^{(l+1)}-\frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2}a_{i(l)}-x^{(l)}\|^2 \ge \|x^{(k+m)}\|^2+\sum_{l=k}^{k+m-1}\frac{\epsilon_{i(l)}^2}{\|a_{i(l)}\|^2}-\sum_{l=k}^{k+m-1}\frac{2\epsilon_{i(l)}\|x^{(l)}\|}{\|a_{i(l)}\|}$$

On the other hand, suppose $H_j = H_{x^{(k+m)}}$, and define

$$r = \max\{l < k+m : i(l) = j\}.$$

It is clear that $k \le r$.

Let $y^{(k+m)} = P_{H_j}(x^{(k+m)})$, and $y^{(r)} = P_{H_j}(x^{(r)}) = v^{(r+1)}$. By the definition of μ,

$$\begin{aligned}
&\mu\|x^{(k+m)}\| \le \|x^{(k+m)} - y^{(k+m)}\| \le \|x^{(k+m)} - y^{(r)}\| \\
&\le \sum_{l=r+1}^{k+m-1} \|x^{(l+1)} - x^{(l)} - \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\| + \|x^{(r+1)} - y^{(r)} + \sum_{l=r+1}^{k+m-1} \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\| \\
&= \sum_{l=r+1}^{k+m-1} \|x^{(l+1)} - x^{(l)} - \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\| + \| \sum_{l=r}^{k+m-1} \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\| \\
&\le \sum_{l=k+1}^{k+m-1} \|x^{(l+1)} - x^{(l)} - \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\| + \| \sum_{l=k}^{k+C-1} \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\|.
\end{aligned} \tag{3.8}$$

Square (3.8), we get

$$\mu^2\|x^{(k+m)}\|^2 \le m \sum_{l=k+1}^{k+m-1} \|x^{(l+1)} - x^{(l)} - \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\|^2 + \| \sum_{l=k}^{k+m-1} \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\|^2. \tag{3.9}$$

Combine (3.7) and (3.9), we have

$$\begin{aligned}
&\mu^2\|x^{(k+m)}\|^2 \\
\le& m \left[\|x^{(k)}\|^2 - \|x^{(k+m)}\|^2 - \sum_{l=k}^{k+m-1} \frac{\epsilon_{i(l)}^2}{\|a_{i(l)}\|^2} + \sum_{l=k}^{k+m-1} \frac{2\epsilon_{i(l)}\|x^{(l)}\|}{\|a_{i(l)}\|} \right] + \| \sum_{l=k}^{k+m-1} \frac{\epsilon_{i(l)}}{\|a_{i(l)}\|^2} a_{i(l)}\|^2 \\
=& m \left[\|x^{(k)}\|^2 - \|x^{(k+m)}\|^2 - \sum_{l=1}^{m} \frac{\epsilon_l^2}{\|a_l\|^2} + \sum_{l=k}^{k+m-1} \frac{2\epsilon_{i(l)}\|x^{(l)}\|}{\|a_{i(l)}\|} \right] + \| \sum_{l=1}^{m} \frac{\epsilon_l}{\|a_l\|^2} a_l\|^2
\end{aligned}$$

which can be simplified to the desired result. □

The interested reader can generalize this theorem for any relaxation parameter $\alpha_k \in (0, 2)$.

4. Connection with statistical learning methods

The connection between the Kaczmarz algorithm and learning theory is established as early as in 1972 with the Perceptron convergence theorem [**30**]. The randomized Kaczmarz algorithm, by definition, is more connected to some statistical methods as we will see below. Many good work that explores this connection includes Needell et al [**34**] on stochastic gradient descent, and Lin and Zhou [**26**] on least square regression setting. See the rest of this section for more references.

4.1. Incremental Gradient Method. We consider least square problem of the form

$$\begin{aligned} &\text{minimize } f(x) = \sum_{i=1}^{m} f_i(x) \\ &\text{subject to } x \in \mathbb{R}^d. \end{aligned} \tag{4.1}$$

Least squares problems often arise in contexts of learning theory where we are trying to fit data with a model. In problems where there are many data blocks, and particularly in neural network training problems, one does not wait to process the entire data set before updating; instead, one cycles through the data blocks in sequence and update the estimate of x after each data block is processed. The iteration formula is

$$\begin{aligned} x^{(k+1)} &= x^{(k)} - \alpha_k \nabla f_{i(k)}(x^{(k)}) \\ &\text{with } i(k) = (k \mod m) + 1. \end{aligned} \tag{4.2}$$

This is called the *incremental gradient method* [**3**]. Notice that the Kaczmarz algorithm (1.1) is a special case of it where $f_i(x) = (\langle a_i, x\rangle - b_i)^2$. Moreover, the incremental gradient method also covers the block Kaczmarz algorithm [**7, 33**] case by letting $f_i(x) = \|A_i x - b_i'\|^2$ where A_i is a submatrx of certain rows of A.

The incremental gradient method has a more general form [**4**], and is similar to the stochastic gradient method, which will be discussed below.

4.2. Stochastic Gradient Descent. The gradient descent method is an iterative algorithm to find a local minimum of the cost function $F(x)$. With an initial estimate $x^{(0)}$, each iterative step is

$$x^{(k+1)} = x^{(k)} - \alpha_k \nabla F(x^{(k)}). \tag{4.3}$$

A recurring problem in machine learning is that large training sets are necessary for good generalization, but large training sets are also more computationally expensive. In other scenarios like online machine learning, data becomes available in a sequential order and one simply does not have access to the entire training data set at once.

The insight of *stochastic gradient descent* (SGD) is that the cost function is an expectation. The gradient of the expectation may be estimated using a small set of samples. To be specific, let

$$F(x) = \mathbb{E} f_i(x) (= \frac{1}{N} \sum f_i(x) \text{ for example}).$$

We can sample a *minibatch* of data $I \in \{1, 2 \cdots, N\}$ drawn uniformly from the training set. The minibatch size $n = |I|$ is typically chosen to be a relatively small number, ranging from 1 to a few hundred. The SGD update is to approximate the gradient in (4.3) by $\frac{1}{n}\nabla \sum_{i\in I} f_i(x^{(k)})$, as

$$x^{(k+1)} = x^{(k)} - \alpha_k \frac{1}{n} \nabla \sum_{i \in I_k} f_i(x^{(k)})), \tag{4.4}$$

where α_k is called the step size or learning rate. The stochastic gradient method plays an important role in machine learning. Nearly all of deep learning is powered by SGD [**15**].

The randomized Kaczmarz algorithm is a special case of SGD with minibatch size $n = 1$ as mentioned in [**34**]. To reduce to the randomized Kaczmarz case, let $F(x) = \frac{1}{2}\|Ax - b\|^2 = \mathbb{E}_i\left[\frac{1}{2p_i}(\langle a_i, x\rangle - b_i)^2\right]$, where the probability of picking i is $p_i = \frac{\|a_i\|^2}{\|A\|_F^2}$, and $f_i(x) = \frac{1}{2p_i}(\langle a_i, x\rangle - b_i)^2$. Now (4.4) becomes

$$x^{(k+1)} = x^{(k)} - \alpha_k \|A\|_F^2 \frac{\langle a_{i(k)}, x^{(k)}\rangle - b_{i(k)}}{\|a_{i(k)}\|^2} a_{i(k)}, \tag{4.5}$$

which is exactly (1.2) if we let the learning rate $\alpha_k = \frac{1}{\|A\|_F^2}$.

Understanding the randomized Kaczmarz algorithm as SGD allows to obtain improved methods and results for the randomized Kaczmarz method. Some results are available in [**34**].

References

1. S. Agmon *The relaxation method for linear inequalities.* Canadian Journal of Mathematics 6.3 (1954): 382-392.
2. H. H. Bauschke, and J. M. Borwein. *On projection algorithms for solving convex feasibility problems.* SIAM review 38.3 (1996): 367-426.
3. D. Bertsekas. *A new class of incremental gradient methods for least squares problems.* SIAM Journal on Optimization 7.4 (1997): 913-926.
4. D. Bertsekas. *Incremental gradient, subgradient, and proximal methods for convex optimization: A survey.* Optimization for Machine Learning 2010.1-38 (2011): 3.
5. L. M. Bregman. *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming.* USSR computational mathematics and mathematical physics 7.3 (1967): 200-217.
6. Y. Censor. *Row-action methods for huge and sparse systems and their applications.* SIAM review 23.4 (1981): 444-466.
7. Y. Censor, P. B. Eggermont, and D. Gordon. *Strong underrelaxation in Kaczmarz's method for inconsistent systems.* Numerische Mathematik 41.1 (1983): 83-92.
8. Y. Censor, and Eli Tom. *Convergence of string-averaging projection schemes for inconsistent convex feasibility problems.* Optimization Methods and Software 18.5 (2003): 543-554.
9. X. Chen, and A. M. Powell. *Almost sure convergence of the Kaczmarz algorithm with random measurements.* Journal of Fourier Analysis and Applications 18.6 (2012): 1195-1214.
10. X. Chen, and A. M. Powell. *Randomized Subspace Actions and Fusion Frames.* Constructive Approximation 43.1 (2016): 103-134.
11. L. Dai, and Thomas B. Schön. *On the exponential convergence of the Kaczmarz algorithm.* IEEE Signal Processing Letters 22.10 (2015): 1571-1574.
12. Y. Eldar and D. Needell. *Acceleration of randomized Kaczmarz method via the Johnson-Lindenstrauss lemma.* Numerical Algorithms, 58 (2011), no. 2, 163–177.
13. H. G. Feichtinger, C. Cenker, M. Mayer, H. Steier, and Thomas Strohmer. *New variants of the POCS method using affine subspaces of finite codimension with applications to irregular sampling.* In Applications in Optical Science and Engineering, pp. 299-310. International Society for Optics and Photonics, 1992.
14. J. L. Goffin. *The relaxation method for solving systems of linear inequalities.* Mathematics of Operations Research (1980): 388-414.
15. I. Goodfellow, Y. Bengio and A. Courville. *Deep Learning.* MIT Press 2016.
16. R. Gordon, R. Bender, and G. T. Herman. *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography.* Journal of theoretical Biology 29.3 (1970): 471-481.
17. L. G. Gubin, B. T. Polyak, and E. V. Raik. *The method of projections for finding the common point of convex sets.* USSR Computational Mathematics and Mathematical Physics 7.6 (1967): 1-24.
18. G. T. Herman, *Fundamentals of computerized tomography: image reconstruction from projections.* Springer Science & Business Media, 2009.

19. C. Hildreth. *A quadratic programming procedure.* Naval Research Logistics (NRL) 4.1 (1957): 79-85.
20. I. Halperin, Israel. *The product of projection operators."* Acta Sci. Math.(Szeged) 23.1 (1962): 96-99.
21. Alfredo N. Iusem, and Alvaro R. De Pierro. *On the convergence properties of Hildreth's quadratic programming algorithm.* Mathematical programming 47.1-3 (1990): 37-51.
22. N. Jamil, X. Chen, and A. Cloninger. *Hildreths algorithm with applications to soft constraints for user interface layout.* Journal of Computational and Applied Mathematics 288 (2015): 193-202.
23. S. Kaczmarz. Angenäherte Auflösung von Systemen linearer Gleichungen. Bulletin International de l'Académie Polonaise de Sciences A, (1937), 355–357. English translation in: Approximate solution of systems of linear equations. International Journal of Control 57 (1993), 1269–1271.
24. Arnold Lent, and Yair Censor. *Extensions of Hildreth's row-action method for quadratic programming.* SIAM Journal on Control and Optimization 18.4 (1980): 444-454.
25. D. Leventhal and A.S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. Mathematics of Operations Research, 35 (2010), no. 3, 641–654.
26. J. Lin, and D. Zhou. *Learning theory of randomized Kaczmarz algorithm.* Journal of Machine Learning Research 16 (2015): 3341-3365.
27. D. A. Lorenz, et al. *A sparse Kaczmarz solver and a linearized Bregman method for online compressed sensing.* Image Processing (ICIP), 2014 IEEE International Conference on. IEEE, 2014.
28. A. Ma, D. Needell, and A. Ramdas. *Convergence Properties of the Randomized Extended Gauss–Seidel and Kaczmarz Methods.* SIAM Journal on Matrix Analysis and Applications 36.4 (2015): 1590-1604.
29. J. Mandel. *Convergence of the cyclical relaxation method for linear inequalities.* Mathematical programming 30.2 (1984): 218-228.
30. M. Minsky, and S. Papert. *Perceptron: An introduction to computational geometry.* MIT Press, Cambridge, MA, 1972
31. T. Motzkin, and I. J. Schoenberg. *The relaxation method for linear inequalities.* Canadian Journal of Mathematics 6.3 (1954): 393-404.
32. D. Needell, *Randomized Kaczmarz solver for noisy linear systems.* BIT Numerical Mathematics 50.2 (2010): 395-403.
33. D. Needell, and J. A. Tropp. *Paved with good intentions: Analysis of a randomized block kaczmarz method.* Linear Algebra and its Applications 441 (2014): 199-221.
34. D. Needell, R. Ward, and N. Srebro. *Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm.* Advances in Neural Information Processing Systems. 2014.
35. J. Von Neumann, *Functional Operators. Vol. II. The geometry of orthogonal spaces*, volume 22 (reprint of 1933 notes) of Annals of Math. Studies. Princeton University Press (1950).
36. N. J. Nilsson. *Learning machines.* McGraw-Hill, New York, 1965
37. T. Strohmer, and R. Vershynin. *A randomized Kaczmarz algorithm with exponential convergence.* Journal of Fourier Analysis and Applications, 15 (2009), no. 2, 262–278.
38. K. Tanabe. *Projection method for solving a singular system of linear equations and its applications.* Numerische Mathematik 17.3 (1971): 203-214.
39. P. Oswald, and Weiqi Zhou. *Convergence Estimates for Kaczmarz-Type Methods.* preprint (2015).
40. K. Wei. *Solving systems of phaseless equations via Kaczmarz methods: A proof of concept study.* Inverse Problems 31.12 (2015): 125008.
41. A. Zouzias, and N. M. Freris. *Randomized extended Kaczmarz for solving least squares.* SIAM Journal on Matrix Analysis and Applications 34.2 (2013): 773-793.

Department of Mathematics and Statistics, University of San Francisco, San Francisco, CA 94117

E-mail address: xchen@math.usfca.edu