On Adaptive Sketch-and-Project for Solving Linear Systems

Robert M Gower, Denali Molitor, Jacob Moorman, Deanna Needell

SIAM J. MATRIX ANAL. APPL.
Vol. 42, No. 2, pp. 954–989

ON ADAPTIVE SKETCH-AND-PROJECT FOR SOLVING LINEAR SYSTEMS*

ROBERT M. GOWER†, DENALI MOLITOR‡, JACOB MOORMAN‡, AND DEANNA NEEDELL‡

Abstract. We generalize the concept of adaptive sampling rules to the sketch-and-project method for solving linear systems. Analyzing adaptive sampling rules in the sketch-and-project setting yields convergence results that apply to all special cases at once, including the Kaczmarz and coordinate descent. This eliminates the need to separately analyze analogous adaptive sampling rules in each special case. To deduce new sampling rules, we show how the progress of one step of the sketch-and-project method depends directly on a *sketched residual*. Based on this insight, we derive a (1) max-distance sampling rule, by sampling the sketch with the largest sketched residual, (2) a proportional sampling rule, by sampling proportional to the sketched residual, and finally (3) a capped sampling rule. The capped sampling rule is a generalization of the recently introduced adaptive sampling rules for the Kaczmarz method [Z.-Z. Bai and W.-T. Wu, *SIAM J. Sci. Comput.,* 40 (2018), pp. A592–A606]. We provide a global exponential convergence theorem for each sampling rule and show that the max-distance sampling rule enjoys the fastest convergence. This finding is also verified in extensive numerical experiments that lead us to conclude that the max-distance sampling rule is superior both experimentally and theoretically to the capped sampling rule. We also provide numerical insights into implementing the adaptive strategies so that the per iteration cost is of the same order as using a fixed sampling strategy when the product of the number of sketches with the sketch size is not significantly larger than the number of columns.

Key words. sketch-and-project, adaptive sampling, least squares, randomized Kaczmarz, coordinate descent

AMS subject classifications. 15A06, 15B52, 65F10, 68W20, 65N75, 65Y20, 68Q25, 68W40, 90C20

DOI. 10.1137/19M1285846

1. Introduction.

We consider the fundamental problem of finding an approximate solution to the linear system

$$\mathbf{A}x = b, \tag{1.1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $b \in \mathbb{R}^m$. Given the possibility of multiple solutions, we set out to find a least-norm solution given by

$$x^* \overset{\text{def}}{=} \min_{x\in\mathbb{R}^n} \tfrac{1}{2}\|x\|_{\mathbf{B}}^2 \quad \text{subject to} \quad \mathbf{A}x = b, \tag{1.2}$$

where $\mathbf{B} \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix and $\|x\|_{\mathbf{B}}^2 \overset{\text{def}}{=} \langle \mathbf{B}x, x\rangle$. Here, we consider consistent systems, for which there exists an x that satisfies (1.1).

*Received by the editors September 6, 2019; accepted for publication (in revised form) by D. L. Boley March 5, 2021; published electronically June 22, 2021.

https://doi.org/10.1137/19M1285846

Funding: The work of the first author was supported by DIM Math Innov Region Ile-de-France grant ED574 - FMJH and LabEx LMH grant ANR-11-LABX-0056-LMH. The work of the second, third, and fourth authors was partially supported by National Science Foundation grants CAREER DMS-1348721 and NSF BIGDATA DMS-1740325. The work of the third author was also supported by National Science Foundation grant DGE-1829071.

†Télécom ParisTech, LTCI, Université Paris-Saclay, F-91120, Palaiseau, France (gowerrobert@gmail.com).

‡Department of Mathematics, University of California, Los Angeles, Los Angeles, CA 90024 USA (dmolitor@math.ucla.edu, jdmoorman@math.ucla.edu, deanna@math.ucla.edu).

When the dimensions of $\mathbf{A}$ are large, direct methods for solving (1.2) can be infeasible, and iterative methods are favored. In particular, Krylov subspace iterative methods including the conjugate gradient algorithms [24] are the industrial standard so long as one can afford full matrix vector products and the system matrix fits in memory. On the other hand, if a single matrix vector product is considerably expensive, or $\mathbf{A}$ is too large to fit in memory, then randomized iterative methods such as the randomized Kaczmarz [26, 57] and the coordinate descent method [36, 29] are effective.

1.1. Randomized Kaczmarz. The randomized Kaczmarz method is typically used to solve linear systems of equations in the large data regime, i.e., when the number of samples m is much larger than the dimension n. The Kaczmarz method was originally proposed in 1937 and has seen applications in computer tomography (CT scans), signal processing, and other areas [26, 57, 16, 38]. In each iteration k, the current iterate x^k is projected onto the solution space of a selected row of the linear system of (1.1). Specifically, at each iteration

$$x^{k+1} = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \left\|x - x^k\right\|^2 \quad \text{subject to} \quad \mathbf{A}_{i_k:}x = b_{i_k},$$

where $\mathbf{A}_{i_k:}$ is the row of $\mathbf{A}$ selected at iteration k. Let $\mathbf{A}_{i_k:}^\top$ denote the transpose of this row. The Kaczmarz update can be written explicitly as

$$x^{k+1} = x^k + \frac{b_{i_k} - \langle \mathbf{A}_{i_k:}, x^k\rangle}{\left\|\mathbf{A}_{i_k:}\right\|^2}\mathbf{A}_{i_k:}^\top. \tag{1.3}$$

1.2. Coordinate descent. Coordinate descent is commonly used for optimizing general convex optimization functions when the dimensions are extremely large, since at each iteration only a single coordinate (or dimension) is updated [55, 54]. Here, we consider coordinate descent applied to (1.2). In this setting, it is sometimes referred to as randomized Gauss–Seidel [36, 29].

At iteration k an index $i \in \{1, \dots, n\}$ is selected and the coordinate x_i^k of the current iterate x^k is updated such that the least-squares objective $\|b - \mathbf{A}x\|^2$ is minimized. More formally,

$$x^{k+1} = \underset{x\in\mathbb{R}^n,\, \lambda\in\mathbb{R}}{\operatorname{argmin}} \|b - \mathbf{A}x\|^2 \quad \text{subject to} \quad x = x^k + \lambda\, e^i,$$

where e^i is the ith coordinate vector. Let $\mathbf{A}_{:i}$ denote the ith column of $\mathbf{A}$ and $\mathbf{A}_{:i}^\top$ denote the transpose of this column. The explicit update for coordinate descent applied to (1.2) is given by

$$x^{k+1} = x^k - \frac{\mathbf{A}_{:i_k}^\top(\mathbf{A}x^k - b)}{\left\|\mathbf{A}_{:i_k}\right\|^2}e^{i_k}. \tag{1.4}$$

1.3. Sketch-and-project methods. Sketch-and-project is a general archetypal algorithm that unifies a variety of randomized iterative methods including both randomized Kaczmarz and coordinate descent along with all of their block variants [18]. At each iteration, sketch-and-project methods project the current iterate onto a subsampled or sketched linear system with respect to some norm. Let $\mathbf{B} \in \mathbb{R}^{n\times n}$ be a symmetric positive definite matrix. We will consider the projection with respect to the $\mathbf{B}$-norm given by $\|\cdot\|_{\mathbf{B}} = \sqrt{\langle \cdot, \mathbf{B}\cdot\rangle}$.

Let $\mathbf{S}_i \in \mathbb{R}^{m\times\tau}$ for $i = 1, \dots, q$ be the set of *sketching matrices* where $\tau \in \mathbb{N}$ is the *sketch size*. In general, the set of sketching matrices $\mathbf{S}_i$ could be infinite; however, here, we restrict ourselves to a finite set of $q \in \mathbb{N} = \{1, 2, \dots\}$ sketching matrices. At the kth iteration of the sketch-and-project algorithm, a sketching matrix $\mathbf{S}_i$ is selected and the current iterate x^k is projected onto the solution space of the sketched system $\mathbf{S}_{i_k}^\top \mathbf{A}x = \mathbf{S}_{i_k}^\top b$ with respect to the $\mathbf{B}$-norm. Given a selected index $i_k \in \{1, \dots, q\}$ the sketch-and-project update solves

$$x^{k+1} = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \left\|x - x^k\right\|_{\mathbf{B}}^2 \quad \text{subject to} \quad \mathbf{S}_{i_k}^\top \mathbf{A}x = \mathbf{S}_{i_k}^\top b. \tag{1.5}$$

The closed form solution to (1.5) is given by

$$x^{k+1} = x^k - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_{i_k}(\mathbf{A}x^k - b), \tag{1.6}$$

where

$$\mathbf{H}_i \overset{\text{def}}{=} \mathbf{S}_i(\mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i)^\dagger \mathbf{S}_i^\top \quad \text{for } i = 1, \dots, q, \tag{1.7}$$

and † denotes the pseudoinverse.

One can recover the randomized Kaczmarz method under the sketch-and-project framework by choosing the matrix $\mathbf{B}$ as the identity matrix and sketches $\mathbf{S}_i = e^i$. If instead $\mathbf{B} = \mathbf{A}^\top\mathbf{A}$ and sketches $\mathbf{S}_i = \mathbf{A}e^i = \mathbf{A}_{:i}$, then the resulting method is coordinate descent.

1.4. Sampling of indices. An important component of the methods above is the selection of the index i_k at iteration k. Methods often use independently and identically distributed (i.i.d.) indices, as this choice makes the method and analysis relatively simple [57, 44]. In addition to choosing indices i.i.d. at each iteration, several adaptive sampling methods have also been proposed, which we discuss next. These sampling strategies use information about the current iterate in order to improve convergence guarantees over i.i.d. random sampling strategies at the cost of extra calculation per iteration. Under certain conditions, such strategies can be implemented with only a marginal additional cost per iteration.

1.4.1. Sampling for the Kaczmarz method. The original Kaczmarz method cycles through the rows of the matrix $\mathbf{A}$ and makes projections onto the solution space with respect to each row [26]. In 2009, Strohmer and Vershynin suggested selecting rows with probabilities that are proportional to the squared row norms (i.e., $p_i \propto \|\mathbf{A}_{i:}\|_2^2$) and provided the first proof of exponential convergence of the randomized Kaczmarz method [57].

Several adaptive selection strategies have also been proposed in the Kaczmarz setting. The max-distance Kaczmarz or Motzkin's method selects the index i_k at iteration k that leads to the largest magnitude update [48, 37]. In addition to the max-distance selection rule, Nutini et al. also consider the greedy selection rule that chooses the row corresponding to the maximal residual component, i.e., $i_k = \operatorname{argmax}_i |\mathbf{A}_{i:}x^k - b_i|$ at each iteration, but show that the max-distance Kaczmarz method performs at least as well as this strategy [48]. More sophisticated adaptive methods have also been suggested for randomized Kaczmarz, such as the capped sampling strategies proposed in [3, 4, 5] or the sampling Kaczmarz Motzkin's method of [31, 22].

1.4.2. Sampling for coordinate descent. For coordinate descent, several works have investigated adaptive coordinate selection strategies [51, 47, 44, 1]. As

coordinate descent is not restricted to solving linear systems, these works often consider more general convex loss functions. A common greedy selection strategy for coordinate descent applied to differentiable loss functions is to select the coordinate that corresponds to the maximal gradient component, which is known as the Gauss–Southwell rule [58, 34, 47, 44] or adaptively according to a duality gap [8].

1.4.3. Sampling for sketch-and-project. The problem of determining the optimal fixed probabilities with which to select the index i_k at each iteration k was shown in section 5.1 of [18] to be a convex semidefinite program, which is often a harder problem than solving the original linear system. The problem of determining the optimal adaptive probabilities is even harder as one must consider the effects of the current index selection on the future iterates. Here, instead, we present adaptive sampling rules that are not necessarily optimal but can be efficiently implemented and are proven to converge faster than the fixed nonadaptive rules.

1.5. Choosing the sketches and preconditioning. Another key question is how we should choose the set of sketching matrices. This question has been partially answered in section 5.2 of [20], wherein the authors show that if a preconditioned $\mathbf{A}$ were available, then the set of sketching matrices should be drawn from row partitions or column partitions of this preconditioned matrix. This strategy can be combined with any index sampling rule for an overall faster algorithm. Here, we will assume a set of sketching matrices has been provided, and we focus only on the index sampling rule.

1.6. Additional related works. Various related works consider extensions to solving (1.2) in the randomized Kaczmarz, coordinate descent, and sketch-and-project settings. The following summary of related works is not exhaustive. While we consider consistent linear systems, others have analyzed and extended sketch-and-project methods to handle inconsistent linear systems [52, 61, 53, 35, 15]. An adaptive maximum-residual sampling strategy has also been analyzed for the inconsistent extension [52]. The randomized Kaczmarz method has also been studied in the context of solving systems of linear inequalities [29, 37, 7, 6]. Block and accelerated variants of randomized Kaczmarz and coordinate descent have also been analyzed [55, 39, 43, 42, 33, 40, 45]. Recent works have considered combining ideas from random sketching methods with those from the sketch-and-project framework [50].

2. Contributions. The primary contribution of our work is to generalize the concept of adaptive sampling to the sketch-and-project framework. We introduce adaptive sampling to this framework and perform the first convergence analysis of several adaptive sampling rules. Analyzing adaptive sampling rules in the sketch-and-project setting yields convergence results that apply to all special cases at once, including the Kaczmarz and coordinate descent settings. This eliminates the need to separately analyze analogous adaptive sampling rules in each special case. The sketch-and-project setting also allows for adaptive sampling rules from one special case to be generalized to all others.

We introduce and analyze three different adaptive sampling rules for the general sketch-and-project method: the max-distance sampling rule, the capped adaptive sampling rule, and proportional sampling probabilities. We prove that each of these adaptive methods converge exponentially in mean squared error with convergence guarantees that are strictly faster than the guarantee for the nonadaptive method that samples indices uniformly. We compare the theoretical convergence guarantees as well as empirical performance for these three adaptive methods along with sampling

from a fixed distribution. Theoretically, the max-distance sampling rule has the fastest convergence guarantee of the sampling rules considered. Empirically, the max-distance rule typically performs best out of the adaptive sampling rules considered and can outpace sampling from a fixed distribution even in terms of flops required.

2.1. Key quantity: Sketched loss. As we will see in the general convergence analysis of the sketch-and-project method detailed in section 7, the convergence at each iteration depends on the current iterate x^k and a key quantity known as the sketched loss

$$f_i(x^k) \overset{\text{def}}{=} \left\|\mathbf{A}x^k - b\right\|_{\mathbf{H}_i}^2 \tag{2.1}$$

of the sketch $\mathbf{S}_i$ (recall that $\mathbf{H}_i$, defined in (1.7), is symmetric positive semidefinite and thus $\|\cdot\|_{\mathbf{H}_i} \overset{\text{def}}{=} \sqrt{\langle \cdot, \mathbf{H}_i \cdot \rangle}$ gives a seminorm). This sketched loss was introduced in [56], where the authors show that the sketch-and-project method can be seen as a stochastic gradient method (we expand on this in section 4). We show that using adaptive selection rules based on the sketched losses results in new methods with faster convergence guarantees.

2.2. Max-distance rule. We introduce the max-distance sketch-and-project method, which is a generalization of both the max-distance Kaczmarz method (also known as Motzkin's method) [48, 37, 23], greedy coordinate descent (Gauss–Southwell rule [47]), and all their possible block variants. Nutini et al. showed that the max-distance Kaczmarz method performs at least as well as uniform sampling and the nonuniform sampling method of [57], in which rows are sampled with probabilities proportional to the squared row norms of $\mathbf{A}$ [48]. We extend this result to the general sketch-and-project setting and also show that the max-distance rule leads to a worst-case convergence guarantee that is *strictly* faster than that of any fixed probability distribution. The max-distance rule is additionally at least as fast as the adaptive sampling methods considered. The theoretical and experimental results presented here suggest that the max-distance rule is superior to alternative sampling strategies for sketch-and-project methods. In particular, as adaptive sampling methods are proposed in various settings and for applications, our work suggests that they should be compared with the max-distance sampling strategy [3, 4, 5, 6, 30].

2.3. The capped adaptive rule. A new family of adaptive sampling methods was recently proposed for the Kaczmarz and coordinate descent type methods [3, 4, 5]. We extend these methods to the sketch-and-project setting, which allows for their application in other settings such as for coordinate descent. While introduced in the Kaczmarz setting under the names greedy randomized Kaczmarz and relaxed greedy randomized Kaczmarz, we refer to this suite of methods in general as *capped adaptive* methods because they select indices i whose corresponding sketched losses $f_i(x^k)$ are larger than a capped threshold given by a convex combination of the largest and average sketched losses. These sampling strategies were introduced as "greedy randomized" sampling rules [3, 4, 5]; however, we rename them here to prevent confusion with the greedy max-distance sampling rule. It was proven in [3] that the worst-case convergence guarantee when using the capped adaptive rule is strictly faster than the fixed nonuniform sampling rule given in [57]. In subsection 7.5, we generalize this capped adaptive sampling to sketch-and-project methods and prove that the resulting convergence guarantee of this adaptive rule is slower than that of the max-distance rule. Furthermore, in Appendix A.3, we show that the max-distance rule requires less computation at each iteration than the capped adaptive rule.

2.4. The proportional adaptive rule. We also present a new and much simpler randomized adaptive rule as compared to the capped adaptive rule discussed above, in which indices are sampled with probabilities that are directly *proportional* to their corresponding sketched losses $f_i(x^k)$. We show that this rule gives a resulting convergence that is at least twice as fast as when sampling the sketches uniformly.

2.5. Efficient implementations. Our adaptive methods come with the added cost of computing the sketched loss $f(x^k)$ of (2.1) at each iteration. Fortunately, the sketched loss can be computed efficiently with certain precomputations as discussed in section 8. We show how the sketched losses can be maintained efficiently via an auxiliary update, leading to reasonably efficient implementations of the adaptive sampling rules. We demonstrate improved performance of the adaptive methods over uniform sampling when solving linear systems with both real and synthetic matrices per iteration and in terms of the flops required.

2.6. Consequences and future work. Our results on adaptive sampling have consequences on many other closely related problems. For instance, an analogous sampling strategy to our proportional adaptive rule has been proposed for coordinate descent in the primal-dual setting for optimizing regularized loss functions [51]. Also a variant of adaptive and greedy coordinate descent has been shown to speed up the solution of the matrix scaling problem [1]. The matrix scaling problem is equivalent to an entropy-regularized version of the optimal transport problem which has numerous applications in machine learning and computer vision [1, 10]. Thus the adaptive methods proposed here may be extended to these other settings such as adaptive coordinate descent for more general smooth optimization [51]. The adaptive methods and the analysis proposed in this paper may also provide insights toward adaptive sampling for other classes of optimization methods such as stochastic gradient, since the randomized Kaczmarz method can be reformulated as stochastic gradient descent (SGD) applied to the least-squares problem [41].

3. Notation. We now introduce notation that will be used throughout. Let Δ_q denote the simplex in $\mathbb{R}^q$, that is,

$$\Delta_q \overset{\text{def}}{=} \left\{ p \in \mathbb{R}^q \ : \ \sum_{i=1}^{q} p_i = 1, \ p_i \geq 0, \ \text{for } i = 1, \ldots, q \right\}.$$

For probabilities $p \in \Delta_q$ and values x_i depending on an index $i = 1, \ldots, q$, we denote $\mathbb{E}_{i \sim p}[x_i] \overset{\text{def}}{=} \sum_{i=1}^{q} p_i x_i$, where $i \sim p$ indicates that i is sampled with probability p_i. At the kth iteration of the sketch-and-project algorithm, a sketching matrix $\mathbf{S}_{i_k}$ is sampled with probability

$$\mathbb{P}[\mathbf{S}_{i_k} = \mathbf{S}_i \mid x^k] = p_i^k \quad \text{for } i = 1, \ldots, q, \tag{3.1}$$

where $p^k \in \Delta_q$ and we use $p^k \overset{\text{def}}{=} (p_1^k, \ldots, p_q^k)$ to denote the vector containing these probabilities. We drop the superscript k when the probabilities do not depend on the iteration.

For any symmetric positive semidefinite matrix $\mathbf{G}$ we write the seminorm induced by $\mathbf{G}$ as $\|\cdot\|_{\mathbf{G}}^2 \overset{\text{def}}{=} \langle \cdot, \mathbf{G} \cdot \rangle$, while $\|\cdot\|$ denotes the standard 2-norm ($\|\cdot\|_2$). For any matrix $\mathbf{M}$, $\|\mathbf{M}\|_F \overset{\text{def}}{=} \sqrt{\sum_{i,j} \mathbf{M}_{ij}^2}$, where $\mathbf{M}_{ij}$ is the jth entry of the ith row of $\mathbf{M}$. We use

$$\lambda_{\min}^{+}(\mathbf{G}) \overset{\text{def}}{=} \min_{v \in \text{Range}(\mathbf{G}), v \neq 0} \frac{\|v\|_{\mathbf{G}}^2}{\|v\|_2^2}$$

to denote the smallest nonzero eigenvalue of $\mathbf{G}$.

3.1. Organization. The remainder of the paper is organized as follows. Sections 4 and 5 provide additional background on the sketch-and-project method and motivation for adaptive sampling in this setting. Section 4 explains how the sketch-and-project method can be reformulated as SGD. The sampling of the sketches can then be seen as importance sampling in the context of SGD. Section 5 provides geometric intuition for the sketch-and-project method and motivates why one would expect adaptive sampling strategies that depend on the sketched losses $f_i(x^k)$ to perform well.

Section 6 introduces the various sketch selection strategies considered throughout the paper, while section 7 provides convergence guarantees for each of the resulting methods. In section 8, we discuss the computational costs of adaptive sketch-and-project for the sketch selection strategies of section 6 and suggest efficient implementations of the methods. Section 9 discusses convergence and computational cost for the special subcases of randomized Kaczmarz and coordinate descent. Performance of adaptive sketch-and-project methods are demonstrated in section 10 for both synthetic and real matrices.

4. Reformulation as importance sampling for stochastic gradient descent. The sketch-and-project method can be reformulated as a stochastic gradient method, as shown in [56]. We use this reformulation to motivate our adaptive sampling as a variant of importance sampling.

Let $p \in \Delta_q$. Consider the stochastic program

$$\min_{x \in \mathbb{R}^d} F(x) \overset{\text{def}}{=} \mathbb{E}_{i \sim p}\left[f_i(x)\right] = \mathbb{E}_{i \sim p}\left[\|\mathbf{A}x - b\|^2_{\mathbf{H}_i}\right]. \tag{4.1}$$

Objective functions $F(x)$ such as the one in (4.1) are common in machine learning, where $f_i(x)$ often represents the loss with respect to a single data point.

When $\mathbb{E}_{i\sim p}[\mathbf{H}_i]$ is invertible, solving (4.1) is equivalent to solving the linear system (1.1). This invertibility condition on $\mathbb{E}_{i\sim p}[\mathbf{H}_i]$ can be significantly relaxed by using the following technical exactness assumption on the probability p and the set of sketches introduced in [56].

Assumption 1. Let $p \in \Delta_q$, $\Sigma \overset{\text{def}}{=} \{\mathbf{S}_1, \ldots, \mathbf{S}_q\}$ be a set of sketching matrices and $\mathbf{H}_i$ as defined in (1.7). We say that the exactness assumption holds for (p, Σ) if

$$\operatorname{Null}\left(\mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right]\right) \subset \operatorname{Null}\left(\mathbf{A}^\top\right).$$

This exactness assumption guarantees[1] that

$$\operatorname{Null}\left(\mathbf{A}\right) = \operatorname{Null}\left(\mathbf{A}^\top \mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right]\mathbf{A}\right). \tag{4.2}$$

This in turn guarantees that the expected sketched loss of the point x is zero if and only if $\mathbf{A}x = b$. Indeed, by taking the derivative of (4.1) and setting it to zero we have that

$$\nabla F(x) = \mathbf{A}^\top \mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right](\mathbf{A}x - b) = \mathbf{A}^\top \mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right]\mathbf{A}(x - x^*) = 0.$$

Thus, every minimizer x of (4.1) is such that

$$x - x^* \in \operatorname{Null}\left(\mathbf{A}^\top \mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right]\mathbf{A}\right) \overset{(4.2)}{=} \operatorname{Null}\left(\mathbf{A}\right), \tag{4.3}$$

[1] This can be shown by applying Lemma B.1 in Appendix B with $\mathbf{G} = \mathbb{E}_{i\sim p}[\mathbf{H}_i]$ and $\mathbf{W} = \mathbf{A}$.

thus $\mathbf{A}(x - x^*) = \mathbf{A}x - b = 0$. As shown in [19] and [56] this exactness assumption holds trivially for most practical sketching techniques.

When the number of f_i functions is large, the SGD method is typically the method of choice for solving (4.1). To view the sketch-and-project update in (1.6) as an SGD method, we sample an index $i_k \sim p$ at each iteration and take a step

$$x^{k+1} = x^k - \nabla^{\mathbf{B}} f_{i_k}(x^k), \tag{4.4}$$

where $\nabla^{\mathbf{B}} f_{i_k}(x^k)$ is the gradient taken with respect to the $\mathbf{B}$-norm. For $f_i(x^k)$ of (2.1), the exact expression of this stochastic gradient is given by

$$\nabla^{\mathbf{B}} f_{i_k}(x^k) = \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_{i_k}(\mathbf{A}x^k - b). \tag{4.5}$$

By plugging (4.5) into (4.4) we can see that the resulting update is equivalent to a sketch-and-project update in (1.6).

Though the indices $i \in \{1, \dots, q\}$ are often sampled uniformly at random for SGD, many alternative sampling distributions have been proposed in order to accelerate convergence, including adaptive sampling strategies [9, 25, 41, 60, 27, 32, 2]. Such sampling strategies give more weight to sampling indices corresponding to a larger loss $f_i(x)$ or a larger gradient norm $\left\|\nabla^{\mathbf{B}} f_i(x)\right\|_{\mathbf{B}}^2$. In the sketch-and-project setting, it is not hard to show[2] that these two sampling strategies result in similar methods since

$$f_i(x) = \|\mathbf{A}x - b\|_{\mathbf{H}_i}^2 = \left\|\nabla^{\mathbf{B}} f_i(x)\right\|_{\mathbf{B}}^2.$$

In general, updating the loss and gradient of every $f_i(x)$ at each iteration can be too expensive. Thus many methods resort to using global approximations of these values such as the Lipschitz constant of the gradient [41] that lead to fixed data-dependent sample distributions. For the sketch-and-project setting, we demonstrate in section 8 that the adaptive sample distributions can be calculated efficiently, with a per-iterate cost on the same order as is required for the sketch-and-project update.

5. Geometric viewpoint and motivational analysis. The sketch-and-project method given in (1.5) can be seen as a method that calculates the next iterate x^{k+1} by projecting the previous iterate x^k onto a random affine space. Indeed, the constraint in (1.5) can be rewritten as

$$\{x \,:\, \mathbf{S}_i^\top \mathbf{A}x = \mathbf{S}_i^\top b\} \quad = \quad x^* + \operatorname{Null}\left(\mathbf{S}_i^\top \mathbf{A}\right). \tag{5.1}$$

In particular, (1.5) is an orthogonal projection of the point x^k onto an affine space that contains x^* with respect to the $\mathbf{B}$-norm. See Figure 5.1 for an illustration. This projection is determined by the following projection operator.

LEMMA 5.1. *Let*

$$\mathbf{Z}_i \overset{def}{=} \mathbf{B}^{-1/2}\mathbf{A}^\top \mathbf{S}_i (\mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i)^\dagger \mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1/2} = \mathbf{B}^{-1/2}\mathbf{A}^\top \mathbf{H}_i \mathbf{A}\mathbf{B}^{-1/2} \tag{5.2}$$

for $i = 1, \dots, q$, which is the orthogonal projection matrix onto $\operatorname{Range}\left(\mathbf{B}^{-1/2}\mathbf{A}^\top \mathbf{S}_i\right)$. *Consequently*

[2] See Lemma 3.1 in [56].

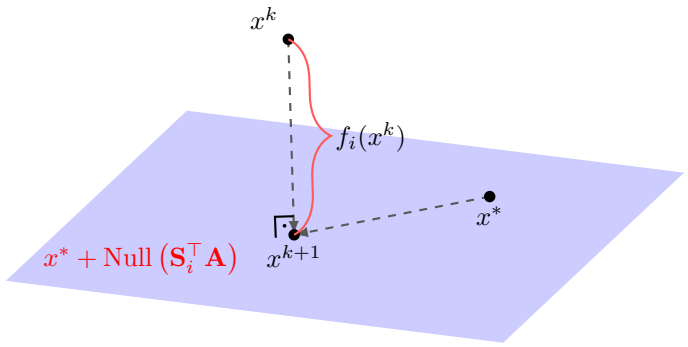


FIG. 5.1. *The geometric interpretation of* (1.5)*, as the projection of* x^k *onto a random affine space that contains* x^**. The distance traveled is given by* $f_i(x^k) = \|x^{k+1} - x^k\|_{\mathbf{B}}^2$.

$$\mathbf{Z}_i\mathbf{Z}_i = \mathbf{Z}_i, \quad \textit{and equivalently} \quad (\mathbf{I} - \mathbf{Z}_i)\mathbf{Z}_i = 0. \tag{5.3}$$

Furthermore we have that $(\mathbf{I} - \mathbf{Z}_i)$ *gives the projection depicted in Figure* 5.1 *since*

$$\mathbf{B}^{1/2}(x^{k+1} - x^*) \;=\; (\mathbf{I} - \mathbf{Z}_{i_k})\mathbf{B}^{1/2}(x^k - x^*). \tag{5.4}$$

Finally we can rewrite the sketched loss as

$$f_i(x) \;=\; \|\mathbf{B}^{1/2}(x - x^*)\|_{\mathbf{Z}_i}^2 \quad \textit{for } i = 1, \dots, q. \tag{5.5}$$

Proof. The proof of (5.3) relies on standard properties of the pseudoinverse and is given in Lemma 2.2 in [18].

As for the proof of (5.4), subtracting x^* from both sides of (1.6) we have that

$$\begin{aligned} x^{k+1} - x^* \quad &= \quad x^k - x^* - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_{i_k}(\mathbf{A}x^k - b) \\ &\overset{\mathbf{A}x^* = b}{=} x^k - x^* - \mathbf{B}^{-1/2}\mathbf{B}^{-1/2}\mathbf{A}^\top \mathbf{H}_{i_k}\mathbf{A}\mathbf{B}^{-1/2}\mathbf{B}^{1/2}(x^k - x^*) \\ &\overset{(5.2)}{=} x^k - x^* - \mathbf{B}^{-1/2}\mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k - x^*). \end{aligned} \tag{5.6}$$

It now only remains to multiply both sides by $\mathbf{B}^{1/2}$.

Finally the proof of (5.5) follows by using $\mathbf{A}x^* = b$ together with the definitions of $\mathbf{H}_i$ and $\mathbf{Z}_i$ given in (1.7) and (5.2) so that

$$f_i(x) = \|\mathbf{A}(x - x^*)\|_{\mathbf{H}_i}^2 = \|x - x^*\|_{\mathbf{A}^\top \mathbf{H}_i \mathbf{A}}^2 \overset{(5.2)}{=} \left\|\mathbf{B}^{1/2}(x - x^*)\right\|_{\mathbf{Z}_i}^2. \tag{5.7}$$

∎

With the explicit expression for the projection operator we can calculate the progress made by a single iteration of the sketch-and-progress method. The convergence proofs later on in section 7 will rely heavily on Lemmas 5.2 and 5.3.

LEMMA 5.2. *Let* $x^k \in \mathbb{R}^d$ *and let* x^{k+1} *be given by* (1.5)*. Then the squared magnitude of the update is*

$$\left\|x^{k+1} - x^k\right\|_{\mathbf{B}}^2 = f_{i_k}(x^k), \tag{5.8}$$

and the error from one iteration to the next decreases according to

$$\left\|x^{k+1} - x^*\right\|_{\mathbf{B}}^2 = \left\|x^k - x^*\right\|_{\mathbf{B}}^2 - f_{i_k}(x^k). \tag{5.9}$$

Proof. We begin by deriving (5.9). Taking the squared norm in (5.4) we have

$$\begin{aligned}
\left\|x^{k+1}-x^*\right\|_{\mathbf{B}}^2 &= \left\|(\mathbf{I}-\mathbf{B}^{-1/2}\mathbf{Z}_{i_k}\mathbf{B}^{1/2})(x^k-x^*)\right\|_{\mathbf{B}}^2 \\
&= \left\|(\mathbf{I}-\mathbf{Z}_{i_k})\mathbf{B}^{1/2}(x^k-x^*)\right\|_2^2 \\
&= \left\langle \mathbf{B}^{1/2}(x^k-x^*), (I-\mathbf{Z}_{i_k})(I-\mathbf{Z}_{i_k})\mathbf{B}^{1/2}(x^k-x^*)\right\rangle \\
&\overset{(5.3)}{=} \left\langle \mathbf{B}^{1/2}(x^k-x^*), (I-\mathbf{Z}_{i_k})\mathbf{B}^{1/2}(x^k-x^*)\right\rangle \\
&= \left\|x^k-x^*\right\|_{\mathbf{B}}^2 - \left\langle \mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k-x^*), \mathbf{B}^{1/2}(x^k-x^*)\right\rangle \\
&\overset{(5.5)}{=} \left\|x^k-x^*\right\|_{\mathbf{B}}^2 - f_i(x^k). \qquad (5.10)
\end{aligned}$$

Finally we establish (5.8) by subtracting x^k from both sides of (1.6) so that

$$x^{k+1}-x^k = -\mathbf{B}^{-1/2}\mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k-x^*).$$

It now remains to take the squared $\mathbf{B}$-norm and use (5.5). ∎

Equation (5.8) shows that the distance traveled from x^k to x^{k+1} is given by the sketch residual $f_{i_k}(x^k)$, as we have depicted in Figure 5.1. Furthermore, (5.9) shows that the contraction of the error $x^{k+1}-x^*$ is given by $-f_{i_k}(x^k)$. Consequently Lemma 5.2 indicates that in order to make the most progress in one step, or maximize the distance traveled, we should choose i_k corresponding to the largest sketched loss $f_{i_k}(x^k)$. We refer to this greedy sketch selection as the max-distance rule, which we explore in detail in subsection 6.3.

Next we give the expected decrease in the error.

LEMMA 5.3. *Let $p^k \in \Delta_q$. Consider the iterates of the sketch-and-project method given in* (1.6) *where $i_k \sim p_i^k$ as is done in Algorithm* 6.2. *It follows that*

$$\mathbb{E}_{i\sim p^k}\left[\left\|x^{k+1}-x^*\right\|_{\mathbf{B}}^2 \mid x^k\right] = \left\|x^k-x^*\right\|_{\mathbf{B}}^2 - \mathbb{E}_{i\sim p^k}\left[f_i(x^k)\right].$$

Proof. The result follows by taking the expectation over (5.9) conditioned on x^k. ∎

Lemma 5.3 suggests choosing adaptive probabilities so that $\mathbb{E}_{i\sim p^k}\left[f_i(x^k)\right]$ is large. This analysis motivates the adaptive methods described in subsection 6.2.

6. Selection rules. Motivated by Lemmas 5.2 and 5.3, we might think that sampling rules that prioritize larger entries of the sketched loss should converge faster. From this point we take two alternatives: (1) choose the i_k that maximizes the decrease (subsection 6.3) or (2) choose a probability distribution that prioritizes the biggest decrease (subsection 6.2). Below, we describe several sketch-and-project sampling strategies (fixed, adaptive, and greedy) and analyze their convergence in section 7. The adaptive and greedy sampling strategies require knowledge of the current sketched loss vector at each iteration. Calculating the sketched loss from scratch is expensive, thus in section 8 we will show how to efficiently calculate the new sketched loss $f(x^{k+1})$ using the previous sketched loss $f(x^k)$.

6.1. Fixed sampling. We first recall the standard nonadaptive sketch-and-project method that will be used as a comparison for the greedy and adaptive versions. In the nonadaptive setting the sketching matrices are sampled from a fixed distribution that is independent of the current iterate x^k. For reference, the details of the nonadaptive sketch-and-project method are provided in Algorithm 6.1.

Algorithm 6.1. Nonadaptive sketch-and-project.

input: $x^0 \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $p \in \Delta_q$, and a set of sketching matrices $\mathbf{S} = [\mathbf{S}_1, \dots, \mathbf{S}_q]$
for $k = 0, 1, 2, \dots$ **do**
  $i_k \sim p_i$
  $x^{k+1} = x^k - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_{i_k}(\mathbf{A}x^k - b)$
output: last iterate x^{k+1}

Algorithm 6.2. Adaptive sketch-and-project.

input: $x^0 \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and a set of sketching matrices $\mathbf{S} = [\mathbf{S}_1, \dots, \mathbf{S}_q]$
for $k = 0, 1, 2, \dots$ **do**
  $f_i(x^k) = \left\|\mathbf{A}x^k - b\right\|^2_{\mathbf{H}_i}$ for $i = 1, \dots, q$
  Calculate $p^k \in \Delta_q$ ▷ Typically based on $f(x^k)$
  $i_k \sim p^k_i$
  $x^{k+1} = x^k - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_{i_k}(\mathbf{A}x^k - b)$
output: last iterate x^{k+1}

6.2. Adaptive probabilities. Equation (5.9) motivates selecting indices that correspond to larger sketched losses with higher probability. We refer to such sampling strategies as adaptive sampling strategies, as they depend on the current iterate and its corresponding sketched loss values. In the adaptive setting, we sample indices at the kth iteration with probabilities given by $p^k \in \Delta_q$. Adaptive sketch-and-project is detailed in Algorithm 6.2.

6.3. Max-distance rule. We refer to the greedy sketch selection rule given by

$$i_k \in \operatorname*{argmax}_{i=1,\dots,q} f_i(x^k) = \operatorname*{argmax}_{i=1,\dots,q} \left\|\mathbf{A}x^k - b\right\|^2_{\mathbf{H}_i} \tag{6.1}$$

as the max-distance selection rule. If multiple indices lead to the maximal sketched loss, any of these indices can be chosen. Per iteration, the max-distance rule leads to the best decrease in mean squared error. The max-distance sketch-and-project method is described in Algorithm 6.3. This greedy selection strategy has been studied for several specific choices of $\mathbf{B}$ and sketching methods. For example, in the Kaczmarz setting, this strategy is typically referred to as max-distance Kaczmarz or Motzkin's method [21, 48, 37]. For coordinate descent, this selection strategy is the Gauss–Southwell rule [44, 47]. We provide a convergence analysis for the general sketch-and-project max-distance selection rule in Theorem 7.7. We further show that max-distance selection leads to a convergence rate that is strictly faster than the resulting convergence rate when sampling from any fixed distribution. While the max-distance rule leads to the fastest convergence for a single iteration, we cannot guarantee that it leads to the fastest convergence overall, as the sketch chosen at each iteration affects the resulting iterate and thus all subsequent iterations.

7. Convergence. We now present convergence results for the max-distance selection rule, uniform sampling, and adaptive sampling with probabilities proportional to the sketched loss. We summarize the convergence rate guarantees discussed throughout section 7 in Table 7.1. Note that these convergence guarantees are upper bounds and thus may not reflect the expected performance of each selection rule.

Algorithm 6.3. Max-distance sketch-and-project.

input: $x^0 \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and a set of sketching matrices $\mathbf{S} = [\mathbf{S}_1, \dots, \mathbf{S}_q]$
for $k = 0, 1, 2, \dots$ **do**
  $f_i(x^k) = \left\|\mathbf{A}x^k - b\right\|^2_{\mathbf{H}_i}$ for $i = 1, \dots, q$
  $i_k = \arg\max_{i=1,\dots,q} f_i(x^k)$
  $x^{k+1} = x^k - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_{i_k}(\mathbf{A}x^k - b)$
output: last iterate x^{k+1}

Though they are only upper bounds on the mean squared error, there is merit in comparing convergence guarantees between methods, since there is currently no known way to compare the mean squared errors directly. We observe in section 10 that the adaptive methods with faster convergence guarantees also converge faster in practice. Our first step in the analysis is to establish an invariance property of the iterates. The restriction to this invariant set allows for a tighter convergence analysis.

DEFINITION 7.1. *Define the set*

$$\Omega \overset{def}{=} \{x \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right) : f_i(x) = 0 \text{ for some } i \in \{1, \dots, q\}\},$$

where $f_i(x)$ is as defined in (2.1).

We now show that if the initial iterate x^0 is chosen from Ω, then all subsequent sketch-and-project iterates x^k remain in Ω. One can ensure that $x^0 \in \Omega$ by applying a sketch-and-project update (equation (1.6)) to any initial point in Range $\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$.

LEMMA 7.2. *If $x^0 \in \Omega$, as defined in Definition* 7.1*, then $x^k \in \Omega$.*

Proof. We first show that if $x^0 \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$, then $x^k - x^* \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$ for $k \geq 0$.[3] First note that $x^* \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$. This follows by taking the Lagrangian of (1.2) given by

$$L(x, \lambda) = \tfrac{1}{2}\left\|x\right\|^2_{\mathbf{B}} + \langle \lambda, \mathbf{A}x - b\rangle .$$

Taking the derivative with respect to x, setting to zero, and isolating x gives

$$x^* = -\mathbf{B}^{-1}\mathbf{A}^\top\lambda \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right). \tag{7.1}$$

Consequently $x^* - x^0 \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$. Assuming that $x^k - x^* \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$ holds, by induction we have that

$$x^{k+1} - x^* \overset{(1.6)}{=} x^k - x^* - \underbrace{\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_{i_k}(\mathbf{S}_{i_k}^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_{i_k})^\dagger\mathbf{S}_{i_k}^\top(\mathbf{A}x^k - b)}_{\in \text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)}. \tag{7.2}$$

Thus $x^{k+1} - x^*$ is the difference of two elements in the subspace Range $\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$ and thus $x^{k+1} - x^* \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$. Since $x^k - x^*, x^* \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$ for all $k \geq 0$, we have that $x^k \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$ for all $k \geq 0$.

We now show that for $f_i(x)$ as defined in (2.1),

$$f_{i_k}(x^{k+1}) = 0 \quad \forall\, k \geq 0.$$

[3] This result was first presented in [19]. We present and prove it here for completeness.

Recall from (5.5) that we can write

$$f_{i_k}(x^{k+1}) = \left\|\mathbf{B}^{1/2}(x^{k+1}-x^*)\right\|^2_{\mathbf{Z}_{i_k}} = \left\langle \mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^{k+1}-x^*), \mathbf{B}^{1/2}(x^{k+1}-x^*)\right\rangle. \tag{7.3}$$

By (5.4) and Lemma 5.1, we have that the above is equal to zero:

$$\begin{aligned}\mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^{k+1}-x^*) &\overset{(5.4)}{=} \mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k - \mathbf{B}^{-1/2}\mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k-x^*) - x^*)\\ &= \mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k-x^*) - \mathbf{Z}_{i_k}\mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k-x^*))\\ &\overset{(5.3)}{=} \mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k-x^*) - \mathbf{Z}_{i_k}\mathbf{B}^{1/2}(x^k-x^*))\\ &= 0.\end{aligned}$$

∎

We also make use of the following fact. For a symmetric positive semidefinite random matrix $\mathbf{M}\in\mathbb{R}^{n\times n}$ drawn from some probability distribution $\mathcal{D}$ and for any vector $v\in\mathbb{R}^n$

$$\mathbb{E}_{\mathcal{D}}\left[\|v\|^2_{\mathbf{M}}\right] = \mathbb{E}_{\mathcal{D}}\left[\langle v, \mathbf{M}v\rangle\right] = \langle v, \mathbb{E}_{\mathcal{D}}\left[\mathbf{M}\right]v\rangle = \|v\|^2_{\mathbb{E}_{\mathcal{D}}[\mathbf{M}]}. \tag{7.4}$$

7.1. Important spectral constants. We define two key spectral constants in the following definition that will be used to express our forthcoming rates of convergence.

DEFINITION 7.3. *Let Ω be the set defined in Definition* 7.1. *Define*

$$\sigma^2_\infty(\mathbf{B},\mathbf{S}) \overset{def}{=} \min_{v\in\Omega}\max_{i=1,\dots,q}\frac{\left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbf{Z}_i}}{\|v\|^2_{\mathbf{B}}}. \tag{7.5}$$

Let $p\in\Delta_q$ and let

$$\sigma^2_p(\mathbf{B},\mathbf{S}) \overset{def}{=} \min_{v\in\Omega}\frac{\left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbb{E}_{i\sim p}[\mathbf{Z}_i]}}{\|v\|^2_{\mathbf{B}}}. \tag{7.6}$$

Next we show that $\sigma^2_\infty(\mathbf{B},\mathbf{S})$ and $\sigma^2_p(\mathbf{B},\mathbf{S})$ can be used to lower bound $\max_i f_i(x)$ and $\mathbb{E}_{i\sim p}[f_i(x)]$, respectively. This result will allow us to develop (5.9) and Lemma 5.3 into a recurrence later on.

LEMMA 7.4. *Let $p\in\Delta_q$ and consider the iterates x^k given by Algorithm* 6.2 *with $x^0\in\Omega$ when using any adaptive sampling rule. The spectral constants* (7.5) *and* (7.6) *are such that*

$$\max_{i=1,\dots,q} f_i(x^k) \quad\geq\quad \sigma^2_\infty(\mathbf{B},\mathbf{S})\left\|x^k-x^*\right\|^2_{\mathbf{B}}, \tag{7.7}$$

$$\mathbb{E}_{i\sim p}\left[f_i(x^k)\right] \quad\geq\quad \sigma^2_p(\mathbf{B},\mathbf{S})\left\|x^k-x^*\right\|^2_{\mathbf{B}}. \tag{7.8}$$

Proof. From the invariance provided by Lemma 7.2 we have that $x^k-x^*\in$ Range$\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$ and consequently

$$\begin{aligned}\frac{\max_{i=1,\dots,q} f_i(x^k)}{\|x^k-x^*\|^2_{\mathbf{B}}} &\overset{(5.5)}{=} \max_{i=1,\dots,q}\frac{\left\|\mathbf{B}^{1/2}(x^k-x^*)\right\|^2_{\mathbf{Z}_i}}{\|x^k-x^*\|^2_{\mathbf{B}}}\\ &\geq \min_{v\in\text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)}\max_{i=1,\dots,q}\frac{\left\|\mathbf{B}^{1/2}v\right\|_{\mathbf{Z}_i}}{\|v\|^2_{\mathbf{B}}} \overset{(7.5)}{=} \sigma^2_\infty(\mathbf{B},\mathbf{S}) \quad \forall k.\end{aligned} \tag{7.9}$$

Analogously we have that

$$\frac{\mathbb{E}_{i\sim p}\left[f_i(x^k)\right]}{\|x^k-x^*\|_{\mathbf{B}}^2} \overset{(5.5)}{=} \frac{\mathbb{E}_{i\sim p}\left[\left\|\mathbf{B}^{1/2}(x^k-x^*)\right\|_{\mathbf{Z}_i}^2\right]}{\|x^k-x^*\|_{\mathbf{B}}^2}$$

$$\geq \min_{v\in \mathrm{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)} \frac{\mathbb{E}_{i\sim p}\left[\left\|\mathbf{B}^{1/2}v\right\|_{\mathbf{Z}_i}^2\right]}{\|v\|_{\mathbf{B}}^2} \overset{(7.6)+(7.4)}{=} \sigma_p^2(\mathbf{B},\mathbf{S}). \tag{7.10}$$

Thus (7.7) and (7.8) follow by rearranging (7.9) and (7.10), respectively. ∎

Finally, we show that $\sigma_p^2(\mathbf{B},\mathbf{S})$ and $\sigma_\infty^2(\mathbf{B},\mathbf{S})$ are always less than one, and if the exactness Assumption 1 holds, then they are both strictly greater than zero. One obvious disadvantage of sampling from a fixed distribution is that it is possible to sample the same index twice in a row. Since the current iterate already lies in the solution space with respect to the previous sketch, no progress is made in such an update. For adaptive distributions that only assign nonzero probabilities to nonzero sketched loss values, the same index will never be chosen twice in a row since the sketched loss corresponding to the previous iterate will always be zero (Lemma 7.2). This fact allows us to derive convergence rates for adaptive sampling strategies that are strictly better than those for fixed sampling strategies and motivates the definition of γ, given in (7.11). The value γ arises in the convergence analysis of the capped-adaptive sampling strategy and allows for the comparison of the convergence guarantees for the sampling strategies that are summarized in Table 7.1.

LEMMA 7.5. *Let $p\in\Delta_q$ and the set of sketching matrices $\{\mathbf{S}_1,\dots,\mathbf{S}_q\}$ be such that the exactness Assumption 1 holds. Define*

$$\gamma \overset{def}{=} \frac{1}{1-\min_{i=1,\dots,q} p_i} \geq 1. \tag{7.11}$$

We then have the following relations:

$$0 < \lambda_{\min}^+\left(\mathbb{E}_{i\sim p}\left[\mathbf{Z}_i\right]\right) \leq \sigma_p^2(\mathbf{B},\mathbf{S}) \leq \gamma\sigma_p^2(\mathbf{B},\mathbf{S}) \leq \sigma_\infty^2(\mathbf{B},\mathbf{S}) \leq 1.$$

Proof. Using the definition of $\mathbf{Z}_i$ given in (5.2) and the fact that $\mathbf{B}$ is symmetric positive definite, we have

$$\mathrm{Null}\left(\mathbb{E}_{i\sim p}\left[\mathbf{Z}_i\right]\right) \overset{(5.2)}{=} \mathrm{Null}\left(\mathbf{B}^{-1/2}\mathbf{A}^\top\mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right]\mathbf{A}\mathbf{B}^{-1/2}\right)$$
$$= \mathrm{Null}\left(\mathbf{A}^\top\mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right]\mathbf{A}\mathbf{B}^{-1/2}\right) \overset{Lemma\ B.1}{=} \mathrm{Null}\left(\mathbf{A}\mathbf{B}^{-1/2}\right),$$

where we applied Lemma B.1 in the appendix with $\mathbf{G}=\mathbb{E}_{i\sim p}\left[\mathbf{H}_i\right]$ and $\mathbf{W}=\mathbf{A}$. Taking the orthogonal complement of the above we have that

$$\mathrm{Range}\left(\mathbb{E}_{i\sim p}\left[\mathbf{Z}_i\right]\right) = \mathrm{Range}\left(\mathbf{B}^{-1/2}\mathbf{A}^\top\right). \tag{7.12}$$

Using the above we then have

$$\sigma_p^2(\mathbf{B},\mathbf{S}) \overset{(7.6)}{=} \min_{v\in\Omega} \frac{\left\|\mathbf{B}^{1/2}v\right\|_{\mathbb{E}_{i\sim p}[\mathbf{Z}_i]}^2}{\|v\|_{\mathbf{B}}^2}$$
$$\overset{(7.12)}{\geq} \min_{\mathbf{B}^{1/2}v\in\mathrm{Range}(\mathbb{E}_{i\sim p}[\mathbf{Z}_i])} \frac{\left\|\mathbf{B}^{1/2}v\right\|_{\mathbb{E}_{i\sim p}[\mathbf{Z}_i]}^2}{\|v\|_{\mathbf{B}}^2} = \lambda_{\min}^+\left(\mathbb{E}_{i\sim p}\left[\mathbf{Z}_i\right]\right) > 0.$$

Since $\gamma \geq 1$, we have that $\sigma_p^2(\mathbf{B},\mathbf{S}) \leq \gamma \sigma_p^2(\mathbf{B},\mathbf{S})$.
Furthermore,

$$\begin{aligned}\sigma_p^2(\mathbf{B},\mathbf{S}) &\overset{(7.6)}{=} \min_{v\in\Omega} \frac{\left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbb{E}_{i\sim p}[\mathbf{Z}_i]}}{\|v\|^2_{\mathbf{B}}} \\ &\overset{(7.4)}{=} \min_{v\in\Omega} \frac{\mathbb{E}_{i\sim p}\left[\left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbf{Z}_i}\right]}{\|v\|^2_{\mathbf{B}}} \\ &= \min_{v\in\Omega} \frac{\sum_{i=1}^q p_i \left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbf{Z}_i}}{\|v\|^2_{\mathbf{B}}}.\end{aligned}$$

Since $v \in \Omega$, there exists j such that $\left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbf{Z}_j} = 0$. Thus,

$$\begin{aligned}\sigma_p^2(\mathbf{B},\mathbf{S}) &= \min_{v\in\Omega} \frac{\sum_{i\neq j} p_i \left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbf{Z}_i}}{\|v\|^2_{\mathbf{B}}} \\ &\leq \sum_{i\neq j} p_i \min_{v\in\Omega} \max_{i=1,\ldots,q} \frac{\left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbf{Z}_i}}{\|v\|^2_{\mathbf{B}}} \\ &\leq \frac{1}{\gamma}\sigma_\infty^2(\mathbf{B},\mathbf{S}).\end{aligned}$$

Finally, using the fact that the matrix $\mathbf{Z}_i$ is an orthogonal projection (Lemma 5.1), we have that

$$\begin{aligned}\sigma_\infty^2(\mathbf{B},\mathbf{S}) &= \min_{v\in\Omega} \max_{i=1,\ldots,q} \frac{\left\|\mathbf{B}^{1/2}v\right\|^2_{\mathbf{Z}_i}}{\|v\|^2_{\mathbf{B}}} \\ &\overset{(5.3)}{=} \min_{v\in\Omega} \max_{i=1,\ldots,q} \frac{\left\|\mathbf{Z}_i\mathbf{B}^{1/2}v\right\|^2}{\left\|\mathbf{B}^{1/2}v\right\|^2} \\ &\leq \min_{v\in\Omega} \max_{i=1,\ldots,q} \frac{\left\|\mathbf{B}^{1/2}v\right\|^2}{\left\|\mathbf{B}^{1/2}v\right\|^2} = 1. \qquad \square\end{aligned}$$

7.2. Sampling from a fixed distribution. We first present a convergence result for the sketch-and-project method when the sketches are drawn from a fixed sampling distribution. This result will later be used as a baseline for comparison against the adaptive sampling strategies.

THEOREM 7.6. *Consider Algorithm* 6.1 *with* $x^0 \in \Omega$ *for some set of probabilities* $p \in \Delta_q$. *It follows that*

$$\mathbb{E}\left[\left\|x^k - x^*\right\|^2_{\mathbf{B}}\right] \leq \left(1 - \sigma_p^2(\mathbf{B},\mathbf{S})\right)^k \left\|x^0 - x^*\right\|^2_{\mathbf{B}}.$$

Proof. Combining Lemma 5.3 and (7.8) of Lemma 7.4 we have that

$$\begin{aligned}\mathbb{E}_{i_k\sim p}\left[\left\|x^{k+1} - x^*\right\|^2_{\mathbf{B}} \mid x^k\right] &\overset{Lemma\ 5.3}{=} \left\|x^k - x^*\right\|^2_{\mathbf{B}} - \mathbb{E}_{i_k\sim p}\left[f_i(x^k)\right] \\ &\overset{(7.8)}{\leq} \left(1 - \sigma_p^2(\mathbf{B},\mathbf{S})\right)\left\|x^k - x^*\right\|^2_{\mathbf{B}}.\end{aligned}$$

Taking the full expectation and unrolling the recurrence, we arrive at Theorem 7.6. $\square$

There are several natural and previously studied choices for fixed sampling distributions, for example, sampling the indices uniformly at random. Another choice is to pick $p \in \Delta_q$ in order to maximize $\sigma_p^2(\mathbf{B}, \mathbf{S})$, but this results in a convex semidefinite program (see section 5.1 in [18]). The authors of [18] suggest convenient probabilities such that $p_i \sim \left\|\mathbf{A}^\top \mathbf{S}_i\right\|^2_{\mathbf{B}^{-1}}$ for which $\sigma_p^2(\mathbf{B}, \mathbf{S})$ reduces to the scaled condition number.

7.3. Max-distance selection. The following theorem provides a convergence guarantee for the max-distance selection rule of subsection 6.3. To our knowledge, this is the first analysis of the max-distance rule for general sketch-and-project methods.

THEOREM 7.7. *The iterates of max-distance sketch-and-project method in Algorithm* 6.3 *satisfy*

$$\left\|x^k - x^*\right\|^2_{\mathbf{B}} \leq (1 - \sigma_\infty^2(\mathbf{B}, \mathbf{S}))^k \left\|x^0 - x^*\right\|^2_{\mathbf{B}},$$

where $\sigma_\infty(\mathbf{B}, \mathbf{S})$ *is defined as in* (7.5) *of Definition* 7.3.

Proof. Combining (5.9) and (7.7) we have that

$$\begin{aligned}\left\|x^{k+1} - x^*\right\|^2_{\mathbf{B}} &\overset{(5.9)}{=} \left\|x^k - x^*\right\|^2_{\mathbf{B}} - \max_{i=1,\ldots,q} f_i(x^k) \\ &\overset{(7.7)}{\leq} \left(1 - \sigma_\infty^2(\mathbf{B}, \mathbf{S})\right) \left\|x^k - x^*\right\|^2_{\mathbf{B}}.\end{aligned}$$

Unrolling the recurrence gives Theorem 7.7. ∎

Since the max-distance rule makes the best possible update at each iteration, it has the fastest convergence guarantee possible under the analysis considered.

7.4. The proportional adaptive rule. We now consider the adaptive sampling strategy in which indices are sampled with probabilities proportional to the sketched loss values. For this sampling strategy, we derive a convergence rate that is strictly faster than that of Theorem 7.6 for uniform sampling.

THEOREM 7.8. *Consider Algorithm* 6.2 *with* $p^k = \frac{f(x^k)}{\|f(x^k)\|_1}$ *and* $x^0 \in \Omega$ *with* Ω *as defined in Definition* 7.1. *Let* $u = (\frac{1}{q}, \ldots, \frac{1}{q}) \in \Delta_q$ *and* $\sigma_u^2(\mathbf{B}, \mathbf{S})$ *be as defined in* (7.6). *Let* $\mathbb{VAR}_u[\cdot]$ *denote the variance taken with respect to the uniform distribution over indices* $i \in \{1, \ldots, q\}$. *It follows that for* $k \geq 1$,

$$\mathbb{E}\left[\left\|x^{k+1} - x^*\right\|^2_{\mathbf{B}} \mid x^k\right] \leq \left(1 - (1 + q^2 \mathbb{VAR}_u\left[p_i^k\right])\sigma_u^2(\mathbf{B}, \mathbf{S})\right) \left\|x^k - x^*\right\|^2_{\mathbf{B}}. \tag{7.13}$$

Furthermore we have that

$$\mathbb{E}\left[\left\|x^{k+1} - x^*\right\|^2_{\mathbf{B}}\right] \leq \left(1 - \left(1 + \tfrac{1}{q}\right)\sigma_u^2(\mathbf{B}, \mathbf{S})\right)^k \mathbb{E}\left[\left\|x^1 - x^*\right\|^2_{\mathbf{B}}\right]. \tag{7.14}$$

Proof. Let $\mathbb{E}_u[\cdot]$ denote the expectation taken with respect to the uniform distribution over indices $i \in \{1, \ldots, q\}$. First note that

$$\mathbb{VAR}_u\left[f_i(x^k)\right] = \mathbb{E}_u\left[(f_i(x^k))^2\right] - \mathbb{E}_u\left[f_i(x^k)\right]^2 = \frac{1}{q}\sum (f_i(x^k))^2 - \frac{1}{q^2}\left(\sum f_i(x^k)\right)^2. \tag{7.15}$$

Given that $p^k = \frac{f(x^k)}{\|f(x^k)\|_1}$,

$$\begin{aligned}
\mathbb{E}_{i\sim p^k}\left[f_i(x^k)\right] &= \sum_{i=1}^q p_i^k f_i(x^k) \\
&= \sum_{i=1}^q \frac{(f_i(x^k))^2}{\sum_{i=1}^q f_i(x^k)} \\
&\overset{(7.15)}{=} \frac{q\mathbb{VAR}_u\left[f_i(x^k)\right] + \frac{1}{q}\left(\sum f_i(x^k)\right)^2}{\sum_{i=1}^q f_i(x^k)} \\
&= \left(q^2\mathbb{VAR}_u\left[\frac{f_i(x^k)}{\sum_{i=1}^q f_i(x^k)}\right] + 1\right)\frac{1}{q}\sum_{i=1}^q f_i(x^k). \qquad (7.16)
\end{aligned}$$

Recalling that $p_i^k = \frac{f_i(x^k)}{\sum_{i=1}^q f_i(x^k)}$ and using Lemma 5.3 we have that

$$\mathbb{E}\left[\left\|x^{k+1}-x^*\right\|_{\mathbf{B}}^2 \mid x^k\right] \leq \left\|x^k-x^*\right\|_{\mathbf{B}}^2 - (1+q^2\mathbb{VAR}_u\left[p_i^k\right])\sigma_u^2(\mathbf{B},\mathbf{S})\left\|x^k-x^*\right\|_{\mathbf{B}}^2.$$

Furthermore, due to Lemma 7.2 we have that $p_{i_k}^{k+1} = 0$. Therefore

$$\begin{aligned}
\mathbb{VAR}_u\left[p_i^{k+1}\right] &= \frac{1}{q}\sum_{i=1}^q\left(p_i^{k+1} - \frac{1}{q}\sum_{s=1}^q p_s^{k+1}\right)^2 \\
&= \frac{1}{q}\sum_{i=1}^q\left(p_i^{k+1}-\frac{1}{q}\right)^2 \geq \frac{1}{q}\left(p_{i_k}^{k+1}-\frac{1}{q}\right)^2 = \frac{1}{q^3}.
\end{aligned}$$

This lower bound on the variance gives the following upper bound on (7.13):

$$\mathbb{E}\left[\left\|x^{k+1}-x^*\right\|_{\mathbf{B}}^2 \mid x^k\right] \leq \left(1-\left(1+\tfrac{1}{q}\right)\sigma_u^2(\mathbf{B},\mathbf{S})\right)\left\|x^k-x^*\right\|_{\mathbf{B}}^2.$$

Taking the expectation and unrolling the recursion gives (7.14). ∎

Thus by sampling proportional to the sketched losses the sketch-and-project method enjoys a strictly faster convergence rate as compared to sampling uniformly. How much faster depends on the variance of the adaptive probabilities through $1 + q^2\mathbb{VAR}_u\left[p_i^k\right]$, which in turn depends on the variance of the sketched losses.

This same variance term is used in [51] to analyze the convergence of an adaptive sampling strategy based on the dual residuals for coordinate descent applied to regularized loss functions and in [49] for adaptive sampling in the block-coordinate Frank–Wolfe algorithm for optimizing structured support vector machines.

7.5. Capped adaptive sampling. We now extend the capped adaptive sampling method and convergence guarantees of [3, 4, 5] for the randomized Kaczmarz and coordinate descent settings to the general sketch-and-project setting; see Algorithm 7.1. Let $p \in \Delta_q$ be a fixed reference probability. At each iteration k an index set $\mathcal{W}_k$ is constructed on line 4 of Algorithm 7.1 that contains indices whose sketched losses are sufficiently close to the maximal sketched loss and that are at least as large as $\mathbb{E}_{i\sim p}\left[f_i(x^k)\right]$. At each iteration, the adaptive probabilities p_i^k are zero for all indices that are not included in the set $\mathcal{W}_k$. The input parameter $\theta \in [0, 1]$ controls how aggressive the sampling method is. In particular, if $\theta = 1$, the method reduces to max-distance sampling. As θ approaches 0, the sampling method remains adaptive, as only indices corresponding to sketched losses larger than $\mathbb{E}_{i\sim p}\left[f_i(x^k)\right]$ are sampled with nonzero probability. Bai and Wu originally introduced an adaptive randomized

Algorithm 7.1. Capped adaptive sketch-and-project.

input: $x^0 \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $p \in \Delta_q$, $\theta \in [0,\,1]$ and a set of sketching matrices $\{\mathbf{S}_1, \ldots, \mathbf{S}_q\}$
for $k = 0, 1, 2, \ldots$ **do**
  $f_i(x^k) = \left\|\mathbf{A}x^k - b\right\|^2_{\mathbf{H}_i}$ for $i = 1, \ldots, q$.
  $\mathcal{W}_k = \left\{i \mid f_i(x^k) \geq \theta \max_{j=1,\ldots,q} f_j(x^k) + (1-\theta)\mathbb{E}_{j\sim p}\left[f_j(x^k)\right]\right\}$
  Choose $p^k \in \Delta_q$ such that $\text{support}(p^k) \subset \mathcal{W}_k$
  $i_k \sim p^k$
  $x^{k+1} = x^k - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{H}_{i_k}(\mathbf{A}x^k - b)$
output: last iterate x^{k+1}

Kaczmarz method with $\theta = 1/2$ [3] and generalized this to allow for the more general choice of $\theta \in [0,\,1]$[4].

Algorithm 7.1 generalizes and improves upon the methods proposed in [3, 4, 5] in several ways. We generalize the methods from the randomized Kaczmarz setting to the more general sketch-and-project setting. We additionally allow for the use of any fixed reference probability distribution $p \in \Delta_q$, whereas the methods of [3, 4, 5] use a specific reference probability when identifying the set of indices that will be selected with nonzero probability. Last, we allow for the use of any adaptive sampling strategy such that the probabilities p_i^k are zero outside of the set $\mathcal{W}_k$ whereas the methods proposed in [3, 4, 5] specify that a specific adaptive probability be used. However, this restriction is unnecessary in proving the accompanying convergence result Theorem 7.10.

Below, we provide two convergence guarantees for Algorithm 7.1. Theorem 7.9 provides a convergence guarantee in terms of the spectral constants $\sigma^2_\infty(\mathbf{B}, \mathbf{S})$ and $\sigma^2_p(\mathbf{B}, \mathbf{S})$ of Definition 7.3 and the parameter θ. Theorem 7.10 provides a generalization of the convergence rate derived in [4].

THEOREM 7.9. *Consider Algorithm* 7.1 *with* $x^0 \in \Omega$, *where* Ω *is as defined in Definition* 7.1. *Let* $p \in \Delta_q$ *be a fixed* reference probability *and* $\theta \in [0, 1]$. *Let*

$$\mathcal{W}_k = \left\{i \mid f_i(x^k) \geq \theta \max_{j=1,\ldots,q} f_j(x^k) + (1-\theta)\mathbb{E}_{j\sim p}\left[f_j(x^k)\right]\right\}. \tag{7.17}$$

It follows that

$$\mathbb{E}\left[\left\|x^k - x^*\right\|^2_{\mathbf{B}}\right] \leq \left(1 - \theta\sigma^2_\infty(\mathbf{B}, \mathbf{S}) - (1-\theta)\sigma^2_p(\mathbf{B}, \mathbf{S})\right)^k \left\|x^0 - x^*\right\|^2_{\mathbf{B}}. \tag{7.18}$$

Proof. First note that $\mathcal{W}_k$ is not empty since

$$\max_{j=1,\ldots,q} f_j(x^k) \geq \mathbb{E}_{j\sim p}\left[f_j(x^k)\right],$$

and thus $\arg\max_{j=1,\ldots,q} f_j(x^k) \in \mathcal{W}_k$. Since $p_i^k = 0$ for all $i \notin \mathcal{W}_k$, Lemma 5.3 gives that

$$\mathbb{E}_{i\sim p^k}\left[\left\|x^{k+1} - x^*\right\|^2_{\mathbf{B}} \mid x^k\right] = \left\|x^{k+1} - x^*\right\|^2_{\mathbf{B}} - \sum_{i\in\mathcal{W}_k} p_i^k f_i(x^k). \tag{7.19}$$

We additionally have

$$\sum_{i\in\mathcal{W}_k} f_i(x^k)p_i^k \overset{(7.17)}{\geq} \sum_{i\in\mathcal{W}_k}\left(\theta \max_{j=1,\dots,q} f_j(x^k) + (1-\theta)\mathbb{E}_{j\sim p}\left[f_j(x^k)\right]\right)p_i^k$$

$$= \theta \max_{j=1,\dots,q} f_j(x^k) + (1-\theta)\mathbb{E}_{j\sim p}\left[f_j(x^k)\right] \tag{7.20}$$

$$\overset{Lemma\ 7.4}{\geq} \left(\theta\sigma_\infty^2(\mathbf{B},\mathbf{S}) + (1-\theta)\sigma_p^2(\mathbf{B},\mathbf{S})\right)\left\|x^k - x^*\right\|_{\mathbf{B}}^2. \tag{7.21}$$

Using (7.21) to bound (7.19) and taking the expectation gives the result. ∎

The resulting convergence rate is a convex combination of the spectral constant $\sigma_\infty^2(\mathbf{B},\mathbf{S})$ which corresponds to the max-distance convergence rate guarantee and $\sigma_p^2(\mathbf{B},\mathbf{S})$ corresponding to the convergence rate guarantee for the fixed reference probabilities p. This convex combination is in terms of the parameter θ and we can see that as θ approaches 1 the method and convergence guarantee approach that of max-distance. When θ is close to 0, the convergence guarantee approaches that of a fixed distribution, but still filters out sketches with sketched losses less than $\mathbb{E}_{j\sim p}\left[f_j(x^k)\right]$. This suggests that for $\theta \approx 0$ the convergence rate guarantee is loose.

We now explicitly extend the analysis of Bai and Wu's work of [3, 4, 5] to derive a convergence rate guarantee for our more general Algorithm 7.1.

THEOREM 7.10. *Consider Algorithm* 7.1 *with* $x^0 \in \Omega$, *where* Ω *is as defined in Definition* 7.1. *Let* $p \in \Delta_q$ *be a set of fixed* reference probabilities *and* $\theta \in [0,1]$. *Let*

$$\gamma \overset{def}{=} \frac{1}{\max_{i=1,\dots,q}\sum_{j=1,\,j\neq i}^{q} p_j} > 1.$$

It follows for $k \geq 1$ *that*

$$\mathbb{E}\left[\left\|x^k - x^*\right\|_{\mathbf{B}}^2\right]$$
$$\leq \left(1 - (\theta\gamma + (1-\theta))\,\sigma_p^2(\mathbf{B},\mathbf{S})\right)^{k-1}\left(1 - \theta\sigma_\infty^2(\mathbf{B},\mathbf{S}) - (1-\theta)\sigma_p^2(\mathbf{B},\mathbf{S})\right)\left\|x^0 - x^*\right\|_{\mathbf{B}}^2, \tag{7.22}$$

where the expectation is taken with respect to the probabilities prescribed by Algorithm 7.1.

Proof. By Lemma 7.2, at least one of the sketched losses is guaranteed to be zero for each iteration $k \geq 1$. Making the conservative assumption that this sketched loss corresponds to the smallest probability $\hat{p}_{i_k}^k$, for an adaptive sampling strategy that assigns $p_i^k = 0$ to sketches $\mathbf{S}_i$ with a sketched loss $f_i(x^k) = 0$ we have that

$$\frac{\max_{j=1,\dots,q} f_j(x^{k+1})}{\mathbb{E}_{j\sim p}\left[f_j(x^{k+1})\right]} \geq \gamma. \tag{7.23}$$

Combining this with (7.20),

$$\sum_{i\in\mathcal{W}_k} f_i(x^{k+1})p_i^{k+1} \geq \left(\theta\frac{\max_{j=1,\dots,q} f_j(x^{k+1})}{\mathbb{E}_{j\sim p}\left[f_j(x^{k+1})\right]} + (1-\theta)\right)\mathbb{E}_{j\sim p}\left[f_j(x^{k+1})\right]$$
$$\overset{(7.23)}{\geq} (\theta\gamma + (1-\theta))\,\mathbb{E}_{j\sim p}\left[f_j(x^{k+1})\right]$$
$$\overset{(7.6)}{\geq} (\theta\gamma + (1-\theta))\,\sigma_p^2(\mathbf{B},\mathbf{S}). \tag{7.24}$$

Consequently for $k \geq 1$, by (7.19), we then have

$$\mathbb{E}\left[\left\|x^{k+1} - x^*\right\|_{\mathbf{B}}^2 \mid x^k\right] \leq \left\|x^k - x^*\right\|_{\mathbf{B}}^2 - (\theta\gamma + (1-\theta))\,\sigma_p^2(\mathbf{B},\mathbf{S})\left\|x^k - x^*\right\|_{\mathbf{B}}^2.$$

Taking the expectation and unrolling the recursion gives

$$\mathbb{E}\left[\left\|x^{k+1}-x^*\right\|_{\mathbf{B}}^2\right] \leq \left(1-\left(\theta\gamma+(1-\theta)\right)\sigma_p^2(\mathbf{B},\mathbf{S})\right)^{k-1}\left\|x^1-x^*\right\|_{\mathbf{B}}^2.$$

Since, at the very first update, we cannot guarantee that there exists $i \in [1,\dots,q]$ such that $f_i(x^0)=0$, (7.24) is not guaranteed for $k=0$. So instead we use (7.18) to unroll the last step in this recurrence to arrive at (7.22). ∎

The convergence rate for Algorithm 7.1 of Theorem 7.10 is an improvement over the convergence rate guarantee for a fixed probability distribution since $\gamma > 1$. As was the case for Theorem 7.9, the convergence rate is maximized when $\theta = 1$, at which point the resulting method is equivalent to the max-distance sampling strategy of Algorithm 6.3. Further, when $\theta = 1$, Theorem 7.10 guarantees that

$$\mathbb{E}\left[\left\|x^{k}-x^*\right\|_{\mathbf{B}}^2\right] \leq \left(1-\gamma\sigma_p^2(\mathbf{B},\mathbf{S})\right)^{k-1}\left(1-\sigma_\infty^2(\mathbf{B},\mathbf{S})\right)\left\|x^0-x^*\right\|_{\mathbf{B}}^2.$$

For $\theta = 0$, Theorem 7.10 recovers the same convergence guarantee as for sampling according to the nonadaptive probabilities p.

7.6. Convergence summary. Sketch-and-project convergence guarantees with varying sampling strategies are summarized in Table 7.1. Recall Lemma 7.5, which states that under Assumption 1

$$0 < \sigma_p^2(\mathbf{B},\mathbf{S}) \leq \gamma\sigma_p^2(\mathbf{B},\mathbf{S}) \leq \sigma_\infty^2(\mathbf{B},\mathbf{S}) \leq 1.$$

Combining Lemma 7.5 with the convergence guarantees in Table 7.1, we see that adaptive strategies have faster convergence guarantees than sampling with respect to corresponding fixed distributions and the max-distance method has the fastest convergence guarantee of all methods considered. In fact, the max-distance rule has the fastest convergence guarantee possible under the convergence analysis considered. In section 10, we will see that sampling strategies with similar costs and faster convergence guarantees typically outperform those with slower convergence guarantees despite the fact that the derived convergence guarantees are not tight.

8. Implementation tricks and computational complexity. One can perform adaptive sketching with the same order of cost per iteration as the standard nonadaptive sketch-and-project method when τq, the number of sketches q times the sketch size τ, is not significantly larger than the number of columns n. In particular, adaptive sketching methods can be performed for a per-iteration cost of $O(\tau^2 q + \tau n)$,

TABLE 7.1
Summary of convergence guarantees of section 7, where $\gamma = 1/\max_{i=1,\dots,q}\sum_{j=1,j\neq i}^{q} p_i$ *as defined in* (7.11) *and* $\epsilon = \theta(\gamma-1) \leq \theta\frac{1}{q-1}$.

Sampling strategy	Convergence rate bound	Rate bound shown in
Fixed, $p_i^k \equiv p_i$	$1-\sigma_p^2(\mathbf{B},\mathbf{S})$	[18], Theorem 7.6
Max-distance	$1-\sigma_\infty^2(\mathbf{B},\mathbf{S})$	Theorem 7.7
$p_i^k \propto f_i(x^k)$	$1-\left(1+\frac{1}{q}\right)\sigma_u^2(\mathbf{B},\mathbf{S})$	Theorem 7.8
Capped	$1-(1+\epsilon)\,\sigma_p^2(\mathbf{B},\mathbf{S})$	Theorem 7.10

whereas the standard nonadaptive sketch-and-project method has a per-iteration cost of $O(\tau n)$. Appendix A discusses the costs of adaptive sketch-and-project methods in more detail. Pseudocode for efficient implementation is provided in Algorithm A.1.

The main computational costs of adaptive sketch-and-project (Algorithm 6.2) at each iteration come from computing the sketched losses $f_i(x^k)$ of (2.1) and updating the iterate from x^k to x^{k+1} via (1.6). The iterate update for x^k and the formula for the sketched loss $f_i(x^k) = \|\mathbf{A}x - b\|^2_{\mathbf{H}_i}$ both require calculating what we call the *sketched residual*,

$$\mathbf{R}_i^k \overset{\text{def}}{=} \mathbf{C}_i^\top \mathbf{S}_i^\top (\mathbf{A}x^k - b), \tag{8.1}$$

where $\mathbf{C}_i$ is any square matrix satisfying $\mathbf{C}_i\mathbf{C}_i^\top = (\mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i)^\dagger$. The adaptive methods considered here require the sketched residual $\mathbf{R}_i^k$ for each sketch index $i = 1, 2, \ldots, q$ at each iteration. For such adaptive methods, it is possible to update the iterate x^k and compute the sketched losses $f_i(x^k)$ more efficiently if one maintains the set of sketched residuals $\{\mathbf{R}_i^k : i = 1, 2, \ldots, q\}$ in memory.

Different sampling strategies require different amounts of computation as well. Among the adaptive sampling strategies considered here, max-distance sampling requires the least amount of computation followed by sampling proportional to the sketched losses. Capped adaptive sampling requires the most computation. The costs for each sampling strategy are discussed in detail in Appendix A.3 and are summarized in Table A.3.

Remark 1. While the adaptive strategies require calculating the sketched residuals $\{R_i^k : i = 1, \ldots, q\}$ at each iteration, this calculation can be done using the auxiliary update (A.4) in $2\tau^2 q$ flops (see Table A.2), which is significantly less computation than using full gradient descent updates for many choices of sketch size τ and number of sketches q. The gradient descent update for the least-squares problem with step size γ_k is given by

$$\begin{aligned} x^{k+1} &= x^k - \gamma_k \nabla F(x) \\ &= x^k - \gamma_k \mathbf{A}^\top \left(\mathbf{A}x^k - b\right). \end{aligned}$$

Let $r^k \overset{\text{def}}{=} \mathbf{A}^\top \left(\mathbf{A}x^k - b\right)$. This update can be rewritten as

$$x^{k+1} = x^k - \gamma_k r^k$$

and r^k can be updated as

$$\begin{aligned} r^{k+1} &= \mathbf{A}^\top \left(\mathbf{A}x^{k+1} - b\right) \\ &= \mathbf{A}^\top \left(\mathbf{A}\left(x^k - \gamma_k r^k\right) - b\right) \\ &= r^k - \gamma_k \mathbf{A}^\top \mathbf{A} r^k. \end{aligned}$$

The product between $\mathbf{A}^\top \mathbf{A} r^k$ and $\mathbf{A}^\top \mathbf{A} x^k$ both require $O(n^2)$ flops with $\mathbf{A}^\top \mathbf{A}$ precomputed, making a full gradient descent update significantly more expensive than adaptive sketch-and-project updates for which $\tau^2 q \ll n^2$.

When $\tau = 1$ (including both the Kaczmarz and coordinate descent setting), the cost of the adaptive sketch-and-project update is $O(q)$, where q is the number of sketches. For randomized Kaczmarz and coordinate descent, this cost is $O(m) + O(n)$ and $O(n)$, respectively. Note that this is a factor less expensive than the $O(n^2)$ cost of a full gradient update.

9. Summary of consequences for special cases. We now discuss the consequences of the convergence analyses of section 7 and the computational costs detailed in section 8 for the special sketch-and-project subcases of randomized Kaczmarz and coordinate descent. For $\mathbf{C}_i$ as defined in (A.1), in both the randomized Kaczmarz method and coordinate descent, $\mathbf{C}_i$ is a scalar and thus its value is fixed.

9.1. Adaptive Kaczmarz. By choosing the parameter matrix $\mathbf{B} = \mathbf{I}$ and sketching matrices $\mathbf{S}_i = e_i$ for $i = 1, \dots, m$ where $e_i \in \mathbb{R}^n$ is the ith coordinate vector, we arrive at the Kaczmarz method introduced in subsection 1.1. For randomized Kaczmarz, the sketches $\mathbf{S}_i = e_i$ isolate a single row of the matrix $\mathbf{A}$, as $\mathbf{S}_i^\top \mathbf{A} = \mathbf{A}_{i:}$. In this setting, the number of sketches $q = m$ for $\mathbf{A} \in \mathbb{R}^m$, and the sketch size is $\tau = 1$. In order to perform the adaptive update efficiently, the matrices

$$\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i \mathbf{C}_i = \frac{\mathbf{A}_{i:}^\top}{\|\mathbf{A}_{i:}\|} \quad \text{and} \quad \mathbf{C}_i^\top \mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_j \mathbf{C}_j = \frac{\langle \mathbf{A}_{i:}, \mathbf{A}_{j:} \rangle}{\|\mathbf{A}_{i:}\| \, \|\mathbf{A}_{j:}\|} \; \forall\, i, j = 1, 2, \dots m$$

should be precomputed.

In order to succinctly express the convergence rates, we define the diagonal probability matrix $\mathbf{P} = \operatorname{diag}(p_1, \dots, p_m)$ and the normalized matrix $\bar{\mathbf{A}} \overset{\text{def}}{=} \mathbf{D}_{RK}^{-1}\mathbf{A}$, with $\mathbf{D}_{RK} \overset{\text{def}}{=} \operatorname{diag}\left(\|\mathbf{A}_{1:}\|_2, \dots, \|\mathbf{A}_{m:}\|_2\right)$ as in [48]. In the randomized Kaczmarz setting, the projection matrix $\mathbf{Z}_i$ as defined in (5.2) is the orthogonal projection onto the ith row of $\mathbf{A}$ and takes the form

$$\mathbf{Z}_i = \frac{\mathbf{A}_{i:}\mathbf{A}_{i:}^\top}{\|\mathbf{A}_{i:}\|^2}.$$

We then have

$$\mathbb{E}_{i \sim p}\left[\mathbf{Z}_i\right] = \mathbf{D}_{RK}^{-1}\mathbf{A}\mathbf{P}\mathbf{A}^\top \mathbf{D}_{RK}^{-1} = \bar{\mathbf{A}}^\top \mathbf{P} \bar{\mathbf{A}}.$$

The costs and convergence rates for the adaptive sampling strategies discussed in section 6 applied to the Kaczmarz method are summarized in Table 9.1, where we used the notation $\|x\|_\infty \overset{\text{def}}{=} \max_i |x_i|$ for any vector x.

TABLE 9.1

Summary of convergence guarantees and costs of various sampling strategies for the randomized Kaczmarz algorithm. Here, $\gamma = 1/\max_{i=1,\dots,m} \sum_{j=1, j\neq i}^m p_i$ as defined in (7.11), $\mathbf{P} = \operatorname{diag}(p_1, \dots, p_m)$ is a matrix of arbitrary fixed probabilities, and $\bar{\mathbf{A}} \overset{def}{=} \mathbf{D}_{RK}^{-1}\mathbf{A}$, with $\mathbf{D}_{RK} \overset{def}{=} \operatorname{diag}\left(\|\mathbf{A}_{1:}\|_2, \dots, \|\mathbf{A}_{m:}\|_2\right)$. Only leading order flop counts are reported. The number of sketches is q, the sketch size is τ, and the number of rows and columns in the matrix $\mathbf{A}$ is m and n, respectively.

Sampling strategy	Convergence rate bound	Rate bound shown in	Flops per iteration
Uniform	$1 - \frac{1}{m}\lambda_{\min}^+(\bar{\mathbf{A}}^\top \bar{\mathbf{A}})$	[48], Theorem 7.6	$2\min(n,m) + 2n$
$p_i \propto \|\mathbf{A}_{i:}\|_2^2$	$1 - \frac{\lambda_{\min}^+(\mathbf{A}^\top \mathbf{A})}{\|\mathbf{A}\|_F^2}$	[57], Theorem 7.6	$2\min(n,m) + 2n$
Max-distance	$1 - \min\limits_{v \in \text{Range}(\mathbf{A}^\top)} \frac{\|\bar{\mathbf{A}}v\|_\infty}{\|v\|_2}$	[48], Theorem 7.7	$3m + 2n$
$p_i^k \propto f_i(x^k)$	$1 - \frac{m+1}{m}\lambda_{\min}^+(\bar{\mathbf{A}}^\top \bar{\mathbf{A}})$	Theorem 7.8	$5m + 2n$
Capped	$1 - (\theta\gamma + 1)\,\lambda_{\min}^+(\bar{\mathbf{A}}^\top \mathbf{P} \bar{\mathbf{A}})$	[4], Theorem 7.10	$9m + 2n$

9.2. Adaptive coordinate descent. By choosing the parameter matrix $\mathbf{B} = \mathbf{A}^\top\mathbf{A}$ and sketching matrices $\mathbf{S}_i = \mathbf{A}e_i$ for $i = 1, \dots, n$ where $e_i \in \mathbb{R}^m$ is the ith coordinate vector, we arrive at the coordinate descent method introduced in subsection 1.2. In this setting, the number of sketches $q = n$, where n is number of columns in $\mathbf{A}$, and the sketch size is $\tau = 1$.

Coordinate descent uses fewer flops per iteration than indicated by the general computation given in Appendix A.1. This computational savings arises from the sparsity of the matrix $\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_{i_k}\mathbf{C}_{i_k} = e_i/\left\|\mathbf{A}_{:i}\right\|$. As a result, the iterate update of x^k to x^{k+1} using the sketched residuals $\mathbf{R}_{i_k}^k$ requires only $O(1)$ flops instead of $2n$ flops as indicated in the general analysis that is summarized in Table A.2. The cost of a coordinate descent update is dominated by the $2n$ flops required to calculate $\mathbf{R}_{i_k}^k$ either by the auxiliary update of Algorithm A.1 or directly via (8.1).

Similar to the randomized Kaczmarz case, we define the diagonal probability matrix $\mathbf{P} \overset{\text{def}}{=} \operatorname{diag}(p_1, \dots, p_n)$ and the normalized matrix $\widetilde{\mathbf{A}} \overset{\text{def}}{=} \mathbf{A}\mathbf{D}_{CD}^{-1}$, with $\mathbf{D}_{CD} \overset{\text{def}}{=} \operatorname{diag}\left(\left\|\mathbf{A}_{:1}\right\|_2, \dots, \left\|\mathbf{A}_{:n}\right\|_2\right)$. The projection matrix $\mathbf{Z}_i$ as defined in (5.2) is the projection given by

$$\mathbf{Z}_i = (\mathbf{A}^\top\mathbf{A})^{-1/2}\mathbf{A}^\top\mathbf{A}\frac{e_ie_i^\top}{\left\|\mathbf{A}_{:i}\right\|^2}\mathbf{A}^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A})^{-1/2} = (\mathbf{A}^\top\mathbf{A})^{1/2}\frac{e_ie_i^\top}{\left\|\mathbf{A}_{:i}\right\|^2}(\mathbf{A}^\top\mathbf{A})^{1/2}.$$

We then have

$$\mathbb{E}_{i\sim p}\left[\mathbf{Z}_i\right] = (\mathbf{A}^\top\mathbf{A})^{1/2}\mathbf{D}_{CD}^{-1}\mathbf{P}\mathbf{D}_{CD}^{-1}(\mathbf{A}^\top\mathbf{A})^{1/2}.$$

Note that $\mathbb{E}_{i\sim p}\left[\mathbf{Z}_i\right]$ is similar to $\mathbf{P}\mathbf{D}_{CD}^{-1}\mathbf{A}^\top\mathbf{A}\mathbf{D}_{CD}^{-1} = \mathbf{P}\widetilde{\mathbf{A}}^\top\widetilde{\mathbf{A}}$ and thus

$$\lambda_{\min}^+(\mathbb{E}_{i\sim p}\left[\mathbf{Z}_i\right]) = \lambda_{\min}^+(\mathbf{P}\widetilde{\mathbf{A}}^\top\widetilde{\mathbf{A}}).$$

The costs and convergence rates for the adaptive sampling strategies discussed in section 6 applied to coordinate descent are summarized in Table 9.2.

10. Experiments. We test the performance of various adaptive and nonadaptive sampling strategies in the special sketch-and-project subcases of randomized Kaczmarz and coordinate descent. Despite the fact that the convergence guarantees of section 7 are only upper bounds, empirical results demonstrate that methods

TABLE 9.2

Summary of convergence guarantees and costs of various sampling strategies for adaptive coordinate descent. Here, $\gamma = 1/\max_{i=1,\dots,n}\sum_{j=1,j\neq i}^n p_i$ as defined in (7.11), $\mathbf{P} = \operatorname{diag}(p_1, \dots, p_n)$ is a matrix of arbitrary fixed probabilities, and $\widetilde{\mathbf{A}} = \mathbf{A}\mathbf{D}_{CD}^{-1}$, with $\mathbf{D}_{CD} = \operatorname{diag}\left(\left\|\mathbf{A}_{:1}\right\|_2, \dots, \left\|\mathbf{A}_{:n}\right\|_2\right)$. Only flop counts of leading order are reported.

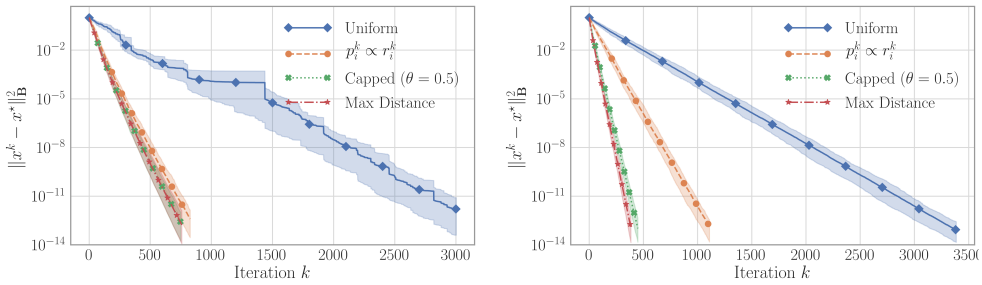
Sampling	Convergence rate bound	Rate bound shown in	Flops per iteration
Uniform	$1 - \frac{1}{n}\lambda_{\min}^+(\widetilde{\mathbf{A}}^\top\widetilde{\mathbf{A}})$	Theorem 7.6	$2n$
$p_i \propto \left\|\mathbf{A}_{:i}\right\|_2^2$	$\left(1 - \frac{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}{\left\|\mathbf{A}\right\|_F^2}\right)$	[29] Theorem 7.6	$2n$
Max-distance	$1 - \min_{v\in\text{Range}(\mathbf{A}^\top)}\frac{\left\|\widetilde{\mathbf{A}}v\right\|_\infty}{\left\|v\right\|_2}$.	Theorem 7.7	$3n$
$p_i^k \propto f_i(x^k)$	$1 - \frac{n+1}{n}\lambda_{\min}^+(\widetilde{\mathbf{A}}^\top\widetilde{\mathbf{A}})$	Theorem 7.8	$5n$
Capped	$1 - (\theta\gamma + 1)\,\lambda_{\min}^+(\mathbf{P}\widetilde{\mathbf{A}}^\top\widetilde{\mathbf{A}})$	Theorem 7.10	$9n$

with better convergence guarantees typically converge faster in practice as well. We report performance via three different metrics: norm-squared error versus iteration, norm-squared error versus approximate flop count, and the worst expected convergence factor. The worst expected convergence factor aims to approximate the spectral constants of Definition 7.3.

Results are averaged over 50 trials. Unless specified otherwise, for synthetic matrices (Figures 10.1 and 10.2), a different matrix $\mathbf{A}$ is used for each trial. In all experiments, a different exact solution x^* and vector b are used in each trial. The exact solutions x^* are generated by

$$x^* = \frac{\mathbf{A}^\top \omega}{\|\mathbf{A}^\top \omega\|_{\mathbf{B}}},$$

where $\omega \in \mathbb{R}^m$ is a vector of i.i.d. random normal entries. Thus $\|x^*\|_{\mathbf{B}}^2 = 1$ is normalized with respect to the $\mathbf{B}$-norm and lies in the row space of $\mathbf{A}$. The latter condition guarantees that x^* is indeed the unique solution to (1.1). We measure the error in terms of the $\mathbf{B}$-norm. Recall that for randomized Kaczmarz $\mathbf{B} = \mathbf{I}$, while for coordinate descent, $\mathbf{B} = \mathbf{A}^\top \mathbf{A}$. The sketch-and-project methods are implemented using the auxiliary update Algorithm A.1 as detailed in Algorithm A.1. For the max-distance sampling rule, if multiple sketches achieve the maximal sketched-loss value, we select the first such sketch.



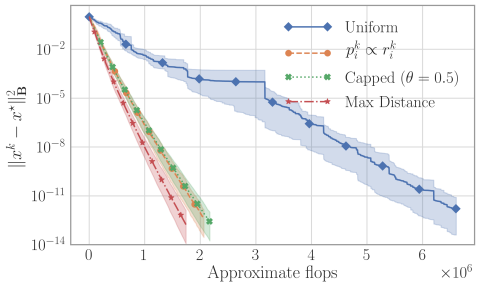
(a) Adaptive randomized Kaczmarz, $\mathbf{A} \in \mathbb{R}^{100\times 1000}$.

(b) Adaptive randomized Kaczmarz, $\mathbf{A} \in \mathbb{R}^{1000\times 100}$.

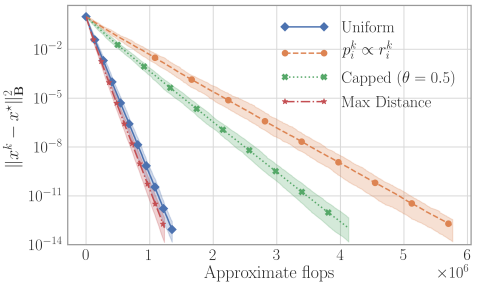
(c) Adaptive coordinate descent, $\mathbf{A} \in \mathbb{R}^{100\times 1000}$.

(d) Adaptive coordinate descent, $\mathbf{A} \in \mathbb{R}^{1000\times 100}$.

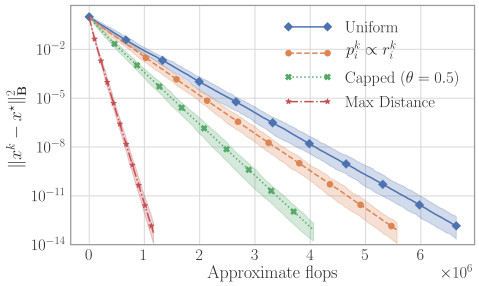
FIG. 10.1. *A comparison between different selection strategies for randomized Kaczmarz and coordinate descent methods. Squared error norms were averaged over* 50 *trials. Confidence intervals indicate the middle* 95% *performance. Subplots on the left show convergence for underdetermined systems, while those on the right show the convergence on overdetermined systems.*



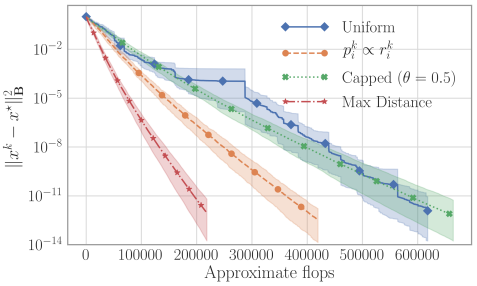
(a) Adaptive randomized Kaczmarz, $\mathbf{A} \in \mathbb{R}^{100\times 1000}$.



(b) Adaptive randomized Kaczmarz, $\mathbf{A} \in \mathbb{R}^{1000\times 100}$.



(c) Adaptive coordinate descent, $\mathbf{A} \in \mathbb{R}^{100\times 1000}$.

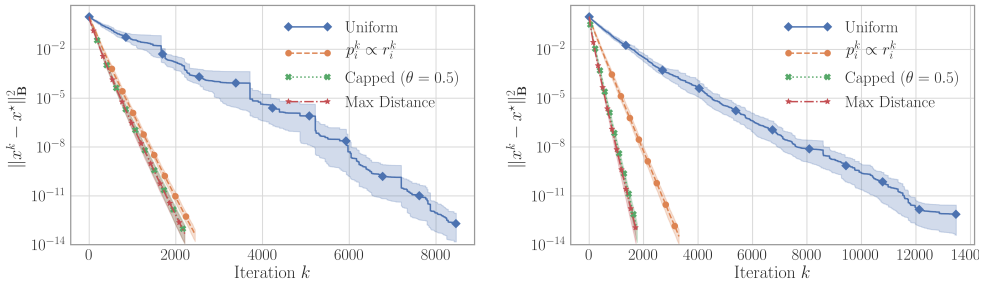


(d) Adaptive coordinate descent, $\mathbf{A} \in \mathbb{R}^{1000\times 100}$.

FIG. 10.2. *A comparison between different selection strategies for randomized Kaczmarz and coordinate descent methods. Squared error norms were averaged over* 50 *trials and are plotted against the approximate flops aggregated over the computations that occur at each iteration. Confidence intervals indicate the middle* 95% *performance. Subplots on the left show convergence for underdetermined systems, while those on the right show the convergence on overdetermined systems.*

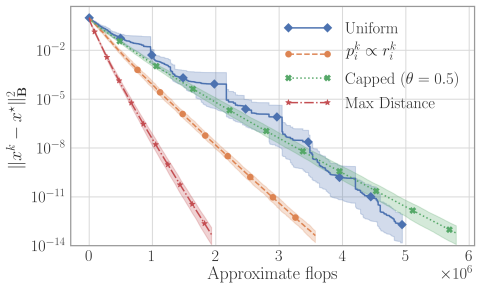
We consider synthetic matrices of size 1000×100 and 100×1000 that are generated with i.i.d. standard Gaussian entries. We additionally test the various adaptive sampling strategies on two large-scale matrices arising from real-world problems. These matrices are available via the SuiteSparse Matrix Collection [11]. The first system (Ash958) is an overdetermined matrix with 958 rows, 292 columns, and 1916 entries [13, 14]. The matrix comes from a survey of the United Kingdom and is part of the original Harwell sparse matrix test collection. The second real matrix we consider is the GEMAT1 matrix, which arises from optimal power flow modeling. This matrix is highly underdetermined and consists of 4929 rows, 10,595 columns, and 47,369 entries [13, 14]. Note that the matrices considered are small enough to be loaded into memory, so direct methods could be used to solve the systems and the precomputational costs for the adaptive sketch-and-project methods are affordable.

10.1. Error per iteration. We first investigate the convergence of the squared norm of the error, $\left\|x^k - x^*\right\|_{\mathbf{B}}^2$ in terms of the number of iterations; see Figure 10.1. The first row of subfigures (Figures 10.1(a) and 10.1(b)) shows convergence for randomized Kaczmarz, while the second row of subfigures (Figures 10.1(c) and 10.1(d)) gives the convergence of various sampling strategies for coordinate descent. The first column of subfigures (Figures 10.1(a) and 10.1(c)) uses an underdetermined system of

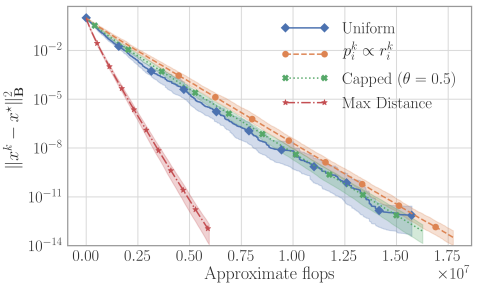


(a) Adaptive coordinate descent.

(b) Adaptive randomized Kaczmarz.



(c) Adaptive coordinate descent.



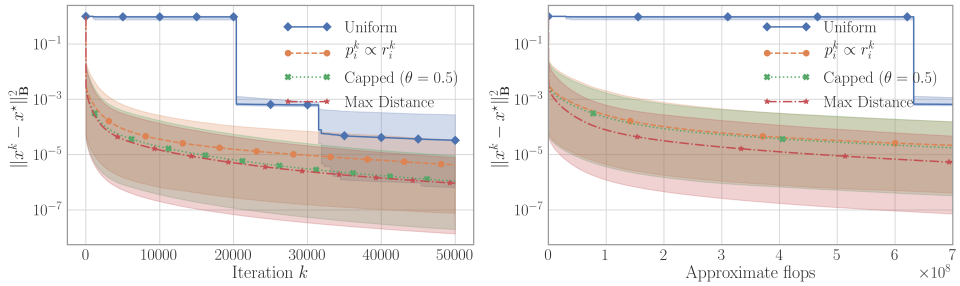
(d) Adaptive randomized kaczmarz.

FIG. 10.3. *A comparison between different selection strategies for randomized Kaczmarz and coordinate descent methods on the Ash*958 *matrix. Squared error norms were averaged over* 50 *trials and plotted against both the iteration and the approximate flops required. Confidence intervals indicate the middle* 95% *performance.*

100×1000 while the second column of subfigures (Figures 10.1(b) and 10.1(d)) considers an overdetermined system of 1000×100. Figures 10.3(c) and 10.3(d) demonstrate convergence per iteration for the Ash958 matrix and Figures 10.4(a) and 10.4(c) for randomized Kaczmarz and coordinate descent applied to the GEMAT1 matrix.

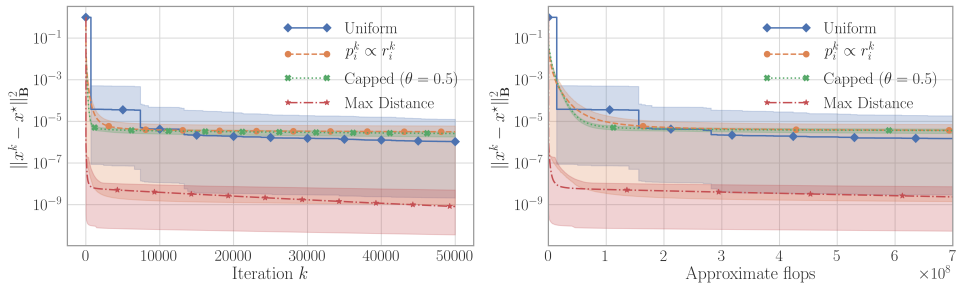
As expected, we see that the max-distance sampling strategy performs at least as well as well as other adaptive sampling strategies and uniform sampling. These experiments provide evidence that, in addition to having the best convergence guarantee, the max-distance rule outperforms other adaptive sampling methods in practice as well. For randomized Kaczmarz applied to underdetermined systems and coordinate descent applied to overdetermined systems, max-distance and the capped adaptive sampling strategies perform similarly in terms of squared error per iteration. The convergence of randomized Kaczmarz for each sampling strategy applied to overdetermined systems is very similar to that of coordinate descent applied to underdetermined systems. Similarly, the convergence of randomized Kaczmarz for each sampling strategy applied to underdetermined systems is very similar to that of coordinate descent applied to overdetermined systems. For the large and underdetermined GEMAT1 matrix, we find that randomized coordinate descent methods have much larger variance in their performance compared to randomized Kaczmarz methods.

10.2. Error versus approximate flops required. If we take into account the number of flops required for each method, the relative performance of the methods



(a) Adaptive randomized Kaczmarz.

(b) Adaptive randomized Kaczmarz.



(c) Adaptive coordinate descent.

(d) Adaptive coordinate descent.

FIG. 10.4. *A comparison between different selection strategies for randomized Kaczmarz and coordinate descent on the GEMAT*1 *matrix. Squared error norms were averaged over* 50 *trials and plotted against both the iteration and the approximate flops required. Confidence intervals indicate the middle* 95% *performance.*

changes significantly. In order to approximate the number of flops required for each sampling strategy, we use the leading order flop counts per iteration given in Tables 9.1 and 9.2. We do not consider the precomputational costs, but only the costs incurred at each iteration. The performance in terms of flops of each sampling strategy is reported in Figure 10.2. Performance on the Ash958 matrix is reported in Figures 10.3(c) and 10.3(d). Performance on the GEMAT1 matrix for randomized Kaczmarz and coordinate descent is reported in Figures 10.4(b) and 10.4(d).

As discussed in section 8, the adaptive methods are typically more expensive than nonadaptive methods as one must update the sketched residuals $\mathbf{R}_i^k$ for $i = 1, \dots, q$ at each iteration k. Yet even after taking flops into consideration, we find that the max-distance sampling strategy still performs the best overall on the systems considered.

For randomized Kaczmarz applied to an overdetermined synthetic matrix, uniform sampling performance is comparable to max-distance (Figure 10.2(b)). In all other experiments, however, max-distance sampling is the clear winner. Since max-distance sampling performs at least as well per iteration as capped adaptive sampling and sampling with probabilities proportional to the sketched losses, yet the max-distance sampling method is less expensive, it naturally performs the best among the adaptive methods when flop counts are considered.

10.3. Spectral constant estimates. Theorems 7.6 to 7.10 of section 7 provide conservative views of the convergence rates of each method, as the spectral constants of Definition 7.3 give the expected convergence corresponding to the worst possible

TABLE 10.1
Minimal expected step size factor for each sampling method applied to matrices containing i.i.d. Gaussian entries.

Sampling	Randomized Kaczmarz		Coordinate descent	
	1000×100	100×1000	1000×100	100×1000
Uniform	0.00705	0.00667	0.00656	0.00715
$p_i \propto \|\mathbf{A}_{:i}\|_2^2$	0.02019	0.01569	0.01722	0.02014
Capped	0.03885	0.01901	0.01952	0.03878
Max-distance	0.04593	0.01994	0.02171	0.04711

point $x \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}\right)$ as opposed to the iterates x^k. In practice, the convergence at each iteration might perform better than the convergence bounds indicate.

Recall that the convergence rates derived in section 7 are given in terms of spectral constants (Definition 7.3) of the form

$$\sigma_p^2(\mathbf{B}, \mathbf{S}) \overset{\text{def}}{=} \min_{x \in \text{Range}(\mathbf{B}^{-1}\mathbf{A}^\top)} \frac{\mathbb{E}_{i\sim p}\left[f_i(x)\right]}{\|x - x^\star\|_{\mathbf{B}}^2}.$$

We will refer to the value

$$\frac{\mathbb{E}_{i\sim p^k}\left[f_i(x^k)\right]}{\|x^k - x^\star\|_{\mathbf{B}}^2}$$

as the *expected step size factor* and note that larger values indicate superior performance.

The smallest expected step size factor observed for each method provides an estimate and upper bound on the spectral constants in the derived convergence rates. The minimal expected step size factor for each sampling method applied to random Gaussian matrices of size 1000×100 and 100×1000 is reported in Table 10.1. Since these values depend on the matrix $\mathbf{A}$ considered, we use a single random Gaussian matrix of each size. As expected, we find that these values increase from uniform sampling, sampling proportional to the sketched losses, capped adaptive sampling, and finally max-distance selection. In Theorem 7.8, we proved a bound on the convergence rate for sampling proportional to the sketched losses that was twice as fast as the convergence guarantee for uniform sampling. We find that the estimated spectral constants in Table 10.1 for the proportional sampling strategy is also at least twice as large as the estimated spectral constant for uniform sampling.

11. Conclusions. We extend adaptive sampling to the general sketch-and-project setting. The analysis of adaptive sampling rules in the sketch-and-project setting yields results for all special cases (randomized Kaczmarz, coordinate descent, block variants) at once. We present a computationally efficient method for implementing the adaptive sampling strategies using an auxiliary update. For several specific adaptive sampling strategies including max-distance selection, the capped adaptive sampling of [3, 4, 5], and sampling proportional to the sketched residuals, we derive convergence rates and show that the max-distance sampling rule has the fastest convergence guarantee among the sampling methods considered. This superior performance is seen in practice as well for both the randomized Kaczmarz and coordinate descent subcases. We find no evidence that adaptive sampling strategies with costs similar to the max-distance rule have any advantages over the max-distance rule. Adaptive sampling rules that are cheaper than max-distance or accelerated in other

ways remain promising directions for improved convergence [31, 46, 22]; this would include analyzing various computational costs and architectures.

Appendix A. Implementation tricks and computational complexity. We describe how one can perform adaptive sketching with the same order of cost per iteration as the standard nonadaptive sketch-and-project method when τq, the number of sketches q times the sketch size τ, is not significantly larger than the number of columns n. In particular, we show how adaptive sketching methods can be performed for a per-iteration cost of $O(\tau^2 q + \tau n)$, whereas the standard nonadaptive sketch-and-project method has a per-iteration cost of $O(\tau n)$. The precomputations and efficient update strategies presented here are a generalization of those suggested in [3] for the Kaczmarz setting. Precomputational costs are a one-time expense and are independent of the sampling strategy. The precomputational costs depend on the sparsity structure of the sketches and are summarized for randomized Kaczmarz and coordinate descent in Table A.1. The computational costs given in this section may be overestimates of the costs required for specific sketch choices such as when the update is sparse, as is the case in coordinate descent. The special cases of adaptive Kaczmarz and adaptive coordinate descent are analyzed in section 9.

Pseudocode for efficient implementation is provided in Algorithm A.1. Throughout this section, we will frequently omit $O(1)$ and $O(\log(q))$ flop counts since they are insignificant compared to the number of rows m, the number of columns n, and the number of sketches q.

TABLE A.1

Precomputational costs for adaptive randomized Kaczmarz and adaptive coordinate descent. The computational costs assume the previous elements have been computed and give the cost of computing the value for all indices.

Computation	Randomized Kaczmarz		Coordinate descent	
$\mathbf{C}_i$ of (A.1)	$\frac{1}{\|\mathbf{A}_{i:}\|}$	$2mn + O(m)$	$\frac{1}{\|\mathbf{A}_{:i}\|}$	$2mn + O(n)$
$\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i \mathbf{C}_i$	$\frac{\mathbf{A}_{i:}^\top}{\|\mathbf{A}_{i:}\|}$	mn	$\frac{e_i}{\|\mathbf{A}_{:i}\|}$	n
$\mathbf{C}_i^\top \mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_j \mathbf{C}_j$	$\frac{\langle \mathbf{A}_{i:}, \mathbf{A}_{j:} \rangle}{\|\mathbf{A}_{i:}\|\|\mathbf{A}_{j:}\|}$	$m^2 n + O(m^2 + mn)$	$\frac{\langle \mathbf{A}_{:i}, \mathbf{A}_{:j} \rangle}{\|\mathbf{A}_{:i}\|\|\mathbf{A}_{:j}\|}$	$mn^2 + O(mn + n^2)$

Algorithm A.1. Efficient adaptive sampling sketch-and-project.

input: $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $\{\mathbf{S}_i \in \mathbb{R}^{m\times\tau} : i = 1, 2, \ldots, q\}$, $\mathbf{B} \in \mathbf{R}^{n\times n}$, $x^0 \in \text{Range}\left(\mathbf{B}^{-1}\mathbf{A}^\top\right)$,

compute $\mathbf{C}_i = \text{Cholesky}\left((\mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i)^\dagger\right)$ for $i = 1, 2, \ldots, q$

▷ The $\mathbf{C}_i$ can be discarded after line 5.

compute $\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i \mathbf{C}_i \in \mathbb{R}^{n\times\tau}$ for $i = 1, 2, \ldots, q$

compute $\mathbf{C}_i^\top \mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_j \mathbf{C}_j \in \mathbb{R}^{\tau\times\tau}$ for $i, j = 1, 2, \ldots, q$

initialize $\mathbf{R}_i^0 = \mathbf{C}_i^\top \left(\mathbf{S}_i^\top (\mathbf{A}x^0 - b)\right) \in \mathbb{R}^\tau$ for $i = 1, 2, \ldots, q$

for $k = 0, 1, 2, \ldots$ **do**

  compute $f_i(x^k) = \left\|\mathbf{R}_i^k\right\|_2^2$ for $i = 1, 2, \ldots, q$

  sample $i_k \sim p_i^k$, where $p^k \in \Delta_q$ is a function of $f(x^k)$

  update $x^{k+1} = x^k - (\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_{i_k} \mathbf{C}_{i_k})\mathbf{R}_{i_k}^k$

  update $\mathbf{R}_i^{k+1} = \mathbf{R}_i^k - (\mathbf{C}_i^\top \mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_{i_k} \mathbf{C}_{i_k})\mathbf{R}_{i_k}^k$ for $i = 1, 2, \ldots, q$

output: last iterate x^{k+1}

TABLE A.2

Summary of the costs of the of Algorithm A.1 *excluding costs that are specific to the sampling method. The number of sketches is q, the sketch size is τ, and the number of columns in the matrix* $\mathbf{A}$ *is n.*

Per iteration computation	Flops
$f_i(x^k)$ $\forall i$ via (A.2)	$(2\tau-1)q$
x^{k+1} via (A.3)	$2\tau n$
$\mathbf{R}_i^k$ $\forall i$ with auxiliary update, (A.4)	$2\tau^2 q$
$\mathbf{R}_{i_k}^k$ via direct computation, (8.1)	$2\tau n$

(a) Baseline flop counts. Flop counts of $O(1)$ have been omitted.

Stored object	Storage
x^k	n
$\mathbf{R}_i^k \quad \forall i$	τq
$\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_i\mathbf{C}_i \quad \forall i$	$\tau q n$
$\mathbf{C}_i^\top\mathbf{S}_i^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_j\mathbf{C}_j$ $\forall i,j$	$\frac{1}{4}\tau(\tau+1)q(q+1)$
$\mathbf{C}_i^\top\mathbf{S}_i^\top\mathbf{A}$ and $\mathbf{C}_i^\top\mathbf{S}_i^\top b \quad \forall i$	$\tau q(n+1)$

(b) Storage costs.

A.1. Per-iteration cost. The main computational costs of adaptive sketch-and-project (Algorithm 6.2) at each iteration come from computing the sketched losses $f_i(x^k)$ of (2.1) and updating the iterate from x^k to x^{k+1} via (1.6). We now discuss how these steps can be calculated efficiently. A suggested efficient implementation for adaptive sketch-and-project is provided in Algorithm A.1. The costs of each step of an iteration of the adaptive sketch-and-project method are summarized in Table A.2.

Let $\mathbf{C}_i$ be any square matrix satisfying

$$\mathbf{C}_i\mathbf{C}_i^\top = (\mathbf{S}_i^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_i)^\dagger. \tag{A.1}$$

For example, $\mathbf{C}_i$ could be the Cholesky decomposition of $(\mathbf{S}_i^\top\mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_i)^\dagger$. The sketched loss $f_i(x^k)$ and the iterate update from x^k to x^{k+1} can now be written as

$$f_i(x^k) = \left\|\mathbf{S}_i^\top(\mathbf{A}x^k - b)\right\|^2_{\mathbf{C}_i\mathbf{C}_i^\top} = \left\|\mathbf{C}_i^\top\mathbf{S}_i^\top(\mathbf{A}x^k - b)\right\|_2^2$$

and

$$x^{k+1} = x^k - \mathbf{B}^{-1}\mathbf{A}^\top\mathbf{S}_{i_k}\mathbf{C}_{i_k}\mathbf{C}_{i_k}^\top\mathbf{S}_{i_k}^\top(\mathbf{A}x^k - b).$$

Notice that both the iterate update for x^k and the formula for the sketched loss $f_i(x^k)$ share the sketched residual $\mathbf{R}_i^k \overset{\text{def}}{=} \mathbf{C}_i^\top\mathbf{S}_i^\top(\mathbf{A}x^k - b)$ defined in (8.1). In adaptive methods one must compute the sketched residual $\mathbf{R}_i^k$ for $i = 1, 2, \ldots, q$. When sampling from a fixed distribution, however, calculating the sketched losses $f_i(x^k)$ is unnecessary and only the sketched residual $\mathbf{R}_{i_k}^k$ corresponding to the selected index i_k need be computed.

Depending on the sketching matrices $\mathbf{S}_i$ and the matrix $\mathbf{B}$, it is possible to update the iterate x^k and compute the sketched losses $f_i(x^k)$ more efficiently if one maintains the set of sketched residuals $\{\mathbf{R}_i^k : i = 1, 2, \ldots, q\}$ in memory. Using the sketched residuals, the calculations above can be rewritten as

$$f_i(x^k) = \left\|\mathbf{R}_i^k\right\|_2^2 \tag{A.2}$$

and

$$x^{k+1} = x^k - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_{i_k}\mathbf{C}_{i_k}\mathbf{R}_{i_k}^k. \tag{A.3}$$

The sketched residuals for the current iteration $\{\mathbf{R}_i^k : i = 1, 2, \ldots, q\}$ can be computed in two ways, either via an auxiliary update applied to the set of sketched residuals for the previous iteration $\{\mathbf{R}_i^{k-1} : i = 1, 2, \ldots, q\}$ or directly using the iterate x^k. Using the auxiliary update,

$$\begin{aligned}\mathbf{R}_i^{k+1} &= \mathbf{C}_i^\top \mathbf{S}_i^\top (\mathbf{A}x^{k+1} - b) \\ &= \mathbf{C}_i^\top \mathbf{S}_i^\top \left(\mathbf{A}(x^k - \mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_{i_k}\mathbf{C}_{i_k}\mathbf{R}_{i_k}^k) - b\right) \\ &= \mathbf{R}_i^k - \mathbf{C}_i^\top \mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_{i_k}\mathbf{C}_{i_k}\mathbf{R}_{i_k}^k \end{aligned} \tag{A.4}$$

with the initialization

$$\mathbf{R}_i^0 = \mathbf{C}_i^\top \left(\mathbf{S}_i^\top (\mathbf{A}x^0 - b)\right).$$

If the matrix $\mathbf{C}_i^\top \mathbf{S}_i^\top \mathbf{A}\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_j \mathbf{C}_j \in \mathbb{R}^{\tau\times\tau}$ is precomputed for each $i, j = 1, 2, \ldots, q$, the sketched residual $\mathbf{R}_i^k$ can be updated to $\mathbf{R}_i^{k+1}$ for $2\tau^2$ flops for each index i via (A.4). Using the precomputed matrices requires storing $\frac{1}{4}\tau(\tau+1)q(q+1)$ floats.

In the nonadaptive case, one only needs to compute the single sketched residual $\mathbf{R}_{i_k}^k$ as opposed to the entire set of sketched residuals, since the sketched losses $f_i(x^k)$ are not needed. If the matrices

$$\mathbf{C}_i^\top \mathbf{S}_i^\top \mathbf{A} \in \mathbb{R}^{\tau\times n} \quad \text{and} \qquad \mathbf{C}_i^\top \mathbf{S}_i^\top b \in \mathbb{R}^\tau$$

are precomputed for $i = 1, 2, \ldots, q$, computing each sketched residual $\mathbf{R}_i^k$ directly from the iterate x^k costs $2\tau n$ flops via (8.1). If $q\tau > n$, then it is cheaper to compute the sketched residual $\mathbf{R}_{i_k}^k$ using the auxiliary update (A.4) rather than computing it directly from x^k.

From the sketched residual $\mathbf{R}_i^k$, the sketched losses $f_i(x^k)$ can be computed for $2\tau - 1$ flops for each index i via (A.2). If the matrix $\mathbf{B}^{-1}\mathbf{A}^\top \mathbf{S}_i \mathbf{C}_i \in \mathbb{R}^{n\times\tau}$ is precomputed for each $i = 1, 2, \ldots, q$, the iterate x^k can then be updated to x^{k+1} for $2\tau n$ flops via (A.3). These costs are summarized in Table A.2.

A.2. Cost of sampling indices. The cost of computing the sampling probabilities p^k from the sketched losses $f_i(x^k)$ depends on the sampling strategy used. Sampling from a fixed distribution can be achieved with an $O(1)$ cost using precomputations of $O(q)$ [59]. Adaptive strategies sample from a new, unseen distribution at each iteration, which can be achieved with an average of q flops using, for example, inversion by sequential search [28], [12, p. 86]. In practice, the probabilities p_i^k corresponding to each index i are given by a function of the sketched losses $f(x_i^k)$ and normalizing these values is unnecessary. Instead, one can sum the q sketched losses and apply inversion by sequential search with a random value r generated between zero and the sum of these values. This summation requires $q - 1$ flops. Thus, the total cost for sampling from an adaptive probability distribution for the methods considered is approximately $2q$ flops on average. The costs for the sampling strategies discussed in section 6 are summarized in Table A.3. The calculations of these costs are discussed in more detail in Appendix A.3. Costs per iteration including sampling are reported in Table A.4.

A.3. Sampling strategy specific costs. We now detail the calculations that lead to the costs associated with each of the specific sampling strategies that are reported in Table A.3.

TABLE A.3

Rule-specific per-iteration costs of Algorithm A.1. *Only leading order flop counts are reported. The nonsampling flops are those that are independent of the specific adaptive sampling method used and are those that correspond to the steps indicated in Table* A.2(*a*). *The extra flops for sampling are those that are required to calculate the adaptive sampling probabilities* p^k *at each iteration. The number of sketches is* q, *the sketch size is* τ, *and the number of columns in the matrix* $\mathbf{A}$ *is* n.

Sampling strategy	Nonsampling flops	Flops from sampling
Fixed, $p_i^k \equiv p_i \ \forall k$	$2\tau \min(n, \tau q) + 2\tau n$	$O(1)$
Max-distance	$(2\tau^2 + 2\tau - 1)q + 2\tau n$	q if $\tau > 1$ $O(\log(q))$ if $\tau = 1$
$p_i^k \propto f_i(x^k)$		$2q$
Capped		$6q$

TABLE A.4

Summary of convergence guarantees of section 7, *where* $\gamma = 1/\max_{i=1,\dots,m} \sum_{j=1, j\neq i}^{m} p_i$ *as defined in* (7.11) *and* $\epsilon = \theta(\gamma - 1) \leq \theta \frac{1}{m}$. *Flop counts of* $O(\log(q))$ *have been omitted. Flop counts assume all matrices are dense. The number of sketches is* q, *the sketch size is* τ, *and the number of columns in the matrix* $\mathbf{A}$ *is* n.

Sampling strategy	Flops per iteration when $\tau > 1$	Flops per iteration when $\tau = 1$
Fixed, $p_i^k \equiv p_i$	$2\tau \min(n, \tau q) + 2\tau n$	$2\min(n, q) + 2n$
Max-distance	$(2\tau^2 + 2\tau)q + 2\tau n$	$3q + 2n$
$p_i^k \propto f_i(x^k)$	$(2\tau^2 + 2\tau + 1)q + 2\tau n$	$5q + 2n$
Capped	$(2\tau^2 + 2\tau + 5)q + 2\tau n$	$9q + 2n$

A.3.1. Sampling from a fixed distribution. When sampling the indices i from a fixed distribution, computing the sketched losses $f_i(x^k)$ is unnecessary and only the sketched residual $\mathbf{R}_{i_k}^k$ of the selected index i_k is needed to update the iterate x^k. If $q\tau > n$, where q is the number of sketches, τ is the sketch size, and n is the number of columns in the matrix $\mathbf{A}$, it is cheaper to compute the sketched residual $\mathbf{R}_{i_k}^k$ using the auxiliary update (A.4) rather than computing it directly from x^k. Ignoring the $O(1)$ cost of sampling from the fixed distribution, the iterate update takes either $4\tau n$ flops if $q\tau > n$ and one maintains the set of sketched residuals via the auxiliary update (A.4) or $2\tau(n + q)$ flops if the sketched residual $\mathbf{R}_{i_k}^k$ is calculated from the iterate x^k directly via (8.1).

A.3.2. Max-distance selection. Performing max-distance selection requires finding the maximum element of the length q vector of sketched losses given in (A.2). In the average case, this costs $q + O(\log q)$ flops, where q flops are used to check each element and $O(\log q)$ flops arise from updates to the running maximal value. For convenience, we ignore the $O(\log q)$ flops and consider the cost of the selection step using the max-distance rule to be q flops. If the sketches $\mathbf{S}_i$ are vectors, or equivalently we have $\tau = 1$, then the sketched residuals $\mathbf{R}_i^k$ are scalars and finding the maximal sketched loss $f_i(x^k)$ is equivalent to finding the sketched residual $\mathbf{R}_i^k$ of maximal magnitude. We can thus save q flops per iteration by skipping the step of computing the sketched losses and instead taking the sketched residual of maximal magnitude.

A.3.3. Sampling proportional to the sketched loss. Sampling indices with probabilities proportional to the sketched losses $f_i(x^k)$ requires approximately $2q$ flops on average using inversion by sequential search.

A.3.4. Capped adaptive sampling. Recall that using capped adaptive sampling requires identifying the set

$$\mathcal{W}_k = \left\{ i \mid f_i(x^k) \geq \theta \max_{j=1,\ldots,q} f_j(x^k) + (1-\theta)\mathbb{E}_{j\sim p}\left[f_j(x^k)\right] \right\}.$$

Sampling with the capped adaptive sampling strategy requires identifying the set $\mathcal{W}_k$ and sampling an index from this set. Identifying the set $\mathcal{W}_k$ requires $q + O(\log q)$ flops to identify the maximal sketched loss $f_i(x^k)$, $2q$ flops to compute the weighted average of the sketched losses $\mathbb{E}_{j\sim p}\left[f_j(x^k)\right]$, $O(1)$ flops to calculate the threshold for the set $\mathcal{W}_k$, and q flops to compare each sketched loss against the threshold. Sampling an index from the set $\mathcal{W}_k$ requires on average $2q$ flops by using inversion by sequential search as discussed in section A.2.[4] Thus, the total cost of the sampling step is $6q + O(\log q)$ flops. When a uniform average is used in place of the weighted average, the expected sketched loss $\mathbb{E}_{j\sim p}\left[f_j(x^k)\right]$ can be computed in just q flops as opposed to $2q$. In that case, the total cost of the sampling step is only $5q + O(\log q)$.

Appendix B. Auxiliary lemma. We now invoke a lemma taken from [17].

LEMMA B.1. *For any matrix* $\mathbf{W}$ *and symmetric positive semidefinite matrix* $\mathbf{G}$ *such that*

$$\text{Null}\left(\mathbf{G}\right) \subset \text{Null}\left(\mathbf{W}^\top\right), \tag{B.1}$$

we have that

$$\text{Null}\left(\mathbf{W}\right) = \text{Null}\left(\mathbf{W}^\top\mathbf{G}\mathbf{W}\right). \tag{B.2}$$

Proof. In order to establish (B.2), it suffices to show the inclusion $\text{Null}\left(\mathbf{W}\right) \supseteq \text{Null}\left(\mathbf{W}^\top\mathbf{G}\mathbf{W}\right)$ since the reverse inclusion trivially holds. Letting $s \in \text{Null}\left(\mathbf{W}^\top\mathbf{G}\mathbf{W}\right)$, we see that $\|\mathbf{G}^{1/2}\mathbf{W}s\|^2 = 0$, which implies $\mathbf{G}^{1/2}\mathbf{W}s = 0$. Consequently

$$\mathbf{W}s \in \text{Null}\left(\mathbf{G}^{1/2}\right) = \text{Null}\left(\mathbf{G}\right) \overset{(B.1)}{\subset} \text{Null}\left(\mathbf{W}^\top\right).$$

Thus $\mathbf{W}s \in \text{Null}\left(\mathbf{W}^\top\right) \cap \text{Range}\left(\mathbf{W}\right)$ which are orthogonal complements which shows that $\mathbf{W}s = 0$. ∎

REFERENCES

[1] B. K. ABID AND R. M. GOWER, *Greedy stochastic algorithms for entropy-regularized optimal transport problems*, in Proceedings of the 21st International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, 2018.

[2] G. ALAIN, A. LAMB, C. SANKAR, A. COURVILLE, AND Y. BENGIO, *Variance Reduction in SGD by Distributed Importance Sampling*, preprint, arXiv:1511.06481, 2015.

[3] Z.-Z. BAI AND W.-T. WU, *On greedy randomized Kaczmarz method for solving large sparse linear systems*, SIAM J. Sci. Comput., 40 (2018), pp. A592–A606, https://doi.org/10.1137/17M1137747.

[4] The analyses of [3, 4] omitted the cost of sampling the index from a new distribution at each iteration, and thus our cost calculations differ by $2q$.

[4] Z.-Z. Bai and W.-T. Wu, *On relaxed greedy randomized Kaczmarz methods for solving large sparse linear systems*, Appl. Math. Lett., 83 (2018), pp. 21–26, https://doi.org/10.1016/j.aml.2018.03.008, http://www.sciencedirect.com/science/article/pii/S0893965918300739.
[5] Z.-Z. Bai and W.-T. Wu, *On greedy randomized coordinate descent methods for solving large linear least-squares problems*, Numer. Linear Algebra Appl., 26 (2019), e2237, https://doi.org/10.1002/nla.2237.
[6] Z.-Z. Bai and W.-T. Wu, *On partially randomized extended Kaczmarz method for solving large sparse overdetermined inconsistent linear systems*, Linear Algebra Appl., 578 (2019), pp. 225–250, https://doi.org/https://doi.org/10.1016/j.laa.2019.05.005.
[7] J. Briskman and D. Needell, *Block Kaczmarz method with inequalities*, J. Math. Imaging Vision, 52 (2015), pp. 385–396.
[8] D. Csiba, Z. Qu, and P. Richtárik, *Stochastic dual coordinate ascent with adaptive probabilities*, in Proceedings of the International Conferences on Machine Learning, 2015.
[9] D. Csiba and P. Richtárik, *Importance sampling for minibatches*, J. Mach. Learn. Res., 19 (2018), pp. 962–982.
[10] M. Cuturi, *Sinkhorn distances: Lightspeed computation of optimal transport*, in Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds., Curran Associates, 2013, pp. 2292–2300, http://papers.nips.cc/paper/4927-sinkhorn-distances-lightspeed-computation-of-optimal-transport.pdf.
[11] T. A. Davis and Y. Hu, *The University of Florida sparse matrix collection*, ACM Trans. Math. Software, 38 (2011), pp. 1–25, https://doi.org/10.1145/2049662.2049663.
[12] L. Devroye, *Non-uniform Random Variate Generation*, Springer-Verlag, New York, 1986.
[13] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Sparse matrix test problems*, ACM Trans. Math. Software, 15 (1989), pp. 1–14.
[14] I. S. Duff, R. G. Grimes, and J. G. Lewis, *Users' Guide for the Harwell-Boeing Sparse Matrix Collection (Release* I*)*, Central Computing Department, Atlas Centre, Rutherford Appleton Laboratory, Oxon, 1992.
[15] B. Dumitrescu, *On the relation between the randomized extended Kaczmarz algorithm and coordinate descent*, BIT, 55 (2015), pp. 1005–1015, https://doi.org/10.1007/s10543-014-0526-9.
[16] R. Gordon, R. Bender, and G. T. Herman, *Algebraic reconstruction techniques (ART) for three-dimensional electron microscopy and X-ray photography*, J. Theoret. Biol., 29 (1970), pp. 471–481.
[17] R. Gower and P. Richtárik, *Linearly Convergent Randomized Iterative Methods for Computing the Pseudoinverse*, preprint, arXiv:1612.06255, 2016.
[18] R. M. Gower and P. Richtárik, *Randomized iterative methods for linear systems*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1660–1690.
[19] R. M. Gower and P. Richtárik, *Stochastic Dual Ascent for Solving Linear Systems*, http://arxiv.org/abs/1512.06890, 2015.
[20] R. M. Gower and P. Richtárik, *Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1380–1409.
[21] M. Griebel and P. Oswald, *Greedy and randomized versions of the multiplicative Schwarz method*, Linear Algebra Appl., 437 (2012), pp. 1596–1610.
[22] J. Haddock and A. Ma, *Greed works: An improved analysis of sampling Kaczmarz–Motzkin*, SIAM J. Math. Data Sci., 3 (2021), pp. 342–368, https://doi.org/10.1137/19M1307044.
[23] J. Haddock and D. Needell, *On Motzkin's method for inconsistent linear systems*, BIT, 59 (2019), pp. 387–401, https://doi.org/10.1007/s10543-018-0737-6.
[24] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. National Bureau of Standards, 49 (1952), pp. 409–436.
[25] R. Johnson and T. Zhang, *Accelerating stochastic gradient descent using predictive variance reduction*, in Advances in Neural Information Processing Systems, 2013, pp. 315–323.
[26] M. S. Kaczmarz, *Angenäherte auflösung von systemen linearer gleichungen*, Bull. Int. Acad. Polonaise Sci. Lettres Classe Sci. Math. Naturelles Sér. A, 35 (1937), pp. 355–357.
[27] A. Katharopoulos and F. Fleuret, *Not all samples are created equal: Deep learning with importance sampling*, in Proceedings of the International Conference on Machine Learning, 2018.
[28] A. Kemp, *Efficient generation of logarithmically distributed pseudo-random variables*, J. R. Stat. Soc. Ser. C Appl. Stat., 30 (1981), pp. 249–253.
[29] D. Leventhal and A. S. Lewis, *Randomized methods for linear constraints: Convergence rates and conditioning*, Math. Oper. Res., 35 (2010), pp. 641–654.
[30] T. Li, T.-J. Kao, D. Isaacson, J. C. Newell, and G. J. Saulnier, *Adaptive Kaczmarz method for image reconstruction in electrical impedance tomography*, Physiol. Meas., 34 (2013), pp. 595–608.

[31] J. A. D. Loera, J. Haddock, and D. Needell, *A sampling Kaczmarz-Motzkin algorithm for linear feasibility*, SIAM J. Sci. Comput., 39 (2017), pp. S66–S87.
[32] I. Loshchilov and F. Hutter, *Online Batch Selection for Faster Training of Neural Networks*, preprint, arXiv:1511.06343, 2015.
[33] Z. Lu and L. Xiao, *On the complexity analysis of randomized block-coordinate descent methods*, Math. Program., 152 (2015), pp. 615–642.
[34] Z.-Q. Luo and P. Tseng, *On the convergence of the coordinate descent method for convex differentiable minimization*, J. Optim. Theory Appl., 72 (1992), pp. 7–35.
[35] A. Ma, D. Needell, and A. Ramdas, *Convergence properties of the randomized extended Gauss–Seidel and Kaczmarz methods*, SIAM J. Matrix Analy. Appl., 36 (2015), pp. 1590–1604.
[36] A. Ma, D. Needell, and A. Ramdas, *Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 1590–1604.
[37] T. S. Motzkin and I. J. Schoenberg, *The relaxation method for linear inequalities*, Canad. J. Math., 6 (1954), pp. 393–404.
[38] F. Natterer, *The Mathematics of Computerized Tomography*, Classics in Appl. Math. 32, SIAM, Philadelphia, 2001, https://doi.org/10.1137/1.9780898719284.
[39] I. Necoara, *Faster Randomized Block Kaczmarz Algorithms*, https://arxiv.org/abs/1902.09946, 2019.
[40] I. Necoara, Y. Nesterov, and F. Glineur, *Random block coordinate descent methods for linearly constrained optimization over networks*, J. Optim. Theory Appl., 173 (2017), pp. 227–254.
[41] D. Needell, N. Srebro, and R. Ward, *Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm*, Math. Program., 155 (2015), pp. 549–573.
[42] D. Needell and J. A. Tropp, *Paved with good intentions: Analysis of a randomized block Kaczmarz method*, Linear Algebra Appl., 441 (2014), pp. 199–221.
[43] D. Needell, R. Zhao, and A. Zouzias, *Randomized block Kaczmarz method with projection for solving least squares*, Linear Algebra Appl., 484 (2015), pp. 322–343.
[44] Y. Nesterov, *Efficiency of coordinate descent methods on huge-scale optimization problems*, SIAM J. Optim., 22 (2012), pp. 341–362.
[45] Y. Nesterov and S. U. Stich, *Efficiency of the accelerated coordinate descent method on structured optimization problems*, SIAM J. Optim., 27 (2017), pp. 110–123.
[46] Y.-Q. Niu and B. Zheng, *A greedy block Kaczmarz algorithm for solving large-scale linear systems*, Appl. Math. Lett., 104 (2020), 106294.
[47] J. Nutini, M. Schmidt, I. Laradji, M. Friedlander, and H. Koepke, *Coordinate descent converges faster with the Gauss-Southwell rule than random selection*, in Proceedings of the International Conference on Machine Learning, 2015, pp. 1632–1641.
[48] J. Nutini, B. Sepehry, I. Laradji, M. Schmidt, H. Koepke, and A. Virani, *Convergence rates for greedy Kaczmarz algorithms, and faster randomized Kaczmarz rules using the orthogonality graph*, Proceedings of the Conference on Uncertainty in Artificial Intelligence, 2016.
[49] A. Osokin, J.-B. Alayrac, I. Lukasewitz, P. Dokania, and S. Lacoste-Julien, *Minding the gaps for block Frank-Wolfe optimization of structured SVMs*, in Proceedings of the 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research 48, 2016, pp. 593–602.
[50] V. Patel, M. Jahangoshahi, and D. A. Maldonado, *An Implicit Representation and Iterative Solution of Randomly Sketched Linear Systems*, https://arxiv.org/abs/1904.11919, 2019.
[51] D. Perekrestenko, V. Cevher, and M. Jaggi, *Faster Coordinate Descent via Adaptive Importance Sampling*, preprint, arXiv:1703.02518, 2017.
[52] S. Petra and C. Popa, *Single projection Kaczmarz extended algorithms*, Numer. Algorithms, 73 (2016), pp. 791–806.
[53] C. Popa, *Characterization of the solutions set of inconsistent least-squares problems by an extended Kaczmarz algorithm*, Korean J. Comput. Appl. Math., 6 (1999), pp. 51–64.
[54] P. Richtárik and M. Takáč, *Distributed coordinate descent method for learning with big data*, J. Mach. Learn. Res., 17 (2016), pp. 1–25.
[55] P. Richtárik and M. Takáč, *Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function*, Math. Program., 144 (2014), pp. 1–38.
[56] P. Richtárik and M. Takáč, *Stochastic Reformulations of Linear Systems: Algorithms and Convergence Theory*, arXiv:1706.01108, 2017.

[57] T. Strohmer and R. Vershynin, *A randomized Kaczmarz algorithm with exponential convergence*, J. Fourier Anal. Appl., 15 (2009), pp. 262–278.
[58] P. Tseng, *Dual ascent methods for problems with strictly convex costs and linear constraints: A unified approach*, SIAM J. Control Optim., 28 (1990), pp. 214–242.
[59] A. J. Walker, *New fast method for generating discrete random numbers with arbitrary frequency distributions*, Electron. Lett., 10 (1974), pp. 127–128, https://doi.org/10.1049/el:19740097.
[60] P. Zhao and T. Zhang, *Stochastic optimization with importance sampling for regularized loss minimization*, in Proceedings of the International Conference on Machine Learning, 2015, pp. 1–9.
[61] A. Zouzias and N. M. Freris, *Randomized extended Kaczmarz for solving least squares*, SIAM J. Matrix Anal. Appl., 34 (2013), pp. 773–793.