

Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm

Deanna Needell · Nathan Srebro · Rachel Ward

Received: 14 February 2014 / Accepted: 17 January 2015 / Published online: 3 February 2015
© Springer-Verlag Berlin Heidelberg and Mathematical Optimization Society 2015

Abstract We obtain an improved finite-sample guarantee on the linear convergence of stochastic gradient descent for smooth and strongly convex objectives, improving from a quadratic dependence on the conditioning $(L/\mu)^2$ (where L is a bound on the smoothness and μ on the strong convexity) to a linear dependence on L/μ . Furthermore, we show how reweighting the sampling distribution (i.e. importance sampling) is necessary in order to further improve convergence, and obtain a linear dependence in the average smoothness, dominating previous results. We also discuss importance sampling for SGD more broadly and show how it can improve convergence also in other scenarios. Our results are based on a connection we make between SGD and the *randomized Kaczmarz algorithm*, which allows us to transfer ideas between the separate bodies of literature studying each of the two methods. In particular, we recast the randomized Kaczmarz algorithm as an instance of SGD, and apply our results to prove its exponential convergence, but to the solution of a weighted least squares problem rather than the original least squares problem. We then present a modified Kaczmarz algorithm with partially biased sampling which does converge to the original least squares solution with the same exponential convergence rate.

D. Needell (✉)
Department of Mathematical Sciences, Claremont McKenna College,
Claremont, CA, USA
e-mail: dneedell@cmc.edu

N. Srebro
Toyota Technological Institute at Chicago, Chicago, IL, USA
e-mail: nati@ttic.edu

R. Ward
Department of Mathematics, University of Texas at Austin, Austin, TX, USA
e-mail: rward@math.utexas.edu

Keywords Distribution reweighting · Importance sampling · Kaczmarz method · Stochastic gradient descent

Mathematics Subject Classification 65B99 · 52A99 · 60G99 · 62L20

1 Introduction

This paper connects two algorithms which until now have remained remarkably disjoint in the literature: the randomized Kaczmarz algorithm for solving linear systems and the stochastic gradient descent (SGD) method for optimizing a convex objective using unbiased gradient estimates. The connection enables us to make contributions by borrowing from each body of literature to the other. In particular, it helps us highlight the role of weighted sampling for SGD and obtain a tighter guarantee on the linear convergence regime of SGD.

Recall that stochastic gradient descent is a method for minimizing a convex objective $F(\mathbf{x})$ based on access to unbiased stochastic gradient estimates, i.e. to an estimate \mathbf{g} for the gradient at a given point \mathbf{x} , such that $\mathbb{E}[\mathbf{g}] = \nabla F(\mathbf{x})$. Viewing $F(\mathbf{x})$ as an expectation $F(\mathbf{x}) = \mathbb{E}_i[f_i(\mathbf{x})]$, the unbiased gradient estimate can be obtained by drawing i and using its gradient: $\mathbf{g} = \nabla f_i(\mathbf{x})$. SGD originated as “Stochastic Approximation” in the pioneering work of Robbins and Monroe [41], and has recently received renewed attention for confronting very large scale problems, especially in the context of machine learning [2, 4, 31, 42]. Classical analysis of SGD shows a polynomial rate on the sub-optimality of the objective value, $F(\mathbf{x}_k) - F(\mathbf{x}_\star)$, namely $1/\sqrt{k}$ for non-smooth objectives, and $1/k$ for smooth, or non-smooth but strongly convex objectives. Such convergence can be ensured even if the iterates \mathbf{x}_k do not necessarily converge to a unique optimum \mathbf{x}_\star , as might be the case if $F(\mathbf{x})$ is not strongly convex. Here we focus on the strongly convex case, where the optimum is unique, and on convergence of the iterates \mathbf{x}_k to the optimum \mathbf{x}_\star .

Bach and Moulines [1] recently provided a non-asymptotic bound on the convergence of the iterates in strongly convex SGD, improving on previous results of this kind [26, Section 2.2] [3, Section 3.2] [31, 43]. In particular, they showed that if each $f_i(\mathbf{x})$ is smooth and if \mathbf{x}_\star is a minimizer of (almost) all $f_i(\mathbf{x})$, i.e. $\mathbb{P}_i(\nabla f_i(\mathbf{x}_\star) = 0) = 1$, then $\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_\star\|$ goes to zero exponentially, rather than polynomially, in k . That is, reaching a desired accuracy of $\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_\star\|^2 \leq \varepsilon$ requires a number of steps that scales only logarithmically in $1/\varepsilon$. Bach and Moulines’s bound on the required number of iterations further depends on the average *squared* conditioning number $\mathbb{E}[(L_i/\mu)^2]$, where L_i is the Lipschitz constant of $\nabla f_i(\mathbf{x})$ (i.e. $f_i(\mathbf{x})$ are “ L -smooth”), and $F(\mathbf{x})$ is μ -strongly convex. If \mathbf{x}_\star is not an exact minimizer of each $f_i(\mathbf{x})$, the bound degrades gracefully as a function of $\sigma^2 = \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|^2$, and includes an unavoidable term that behaves as σ^2/k .

In a seemingly independent line of research, the *Kaczmarz method* was proposed as an iterative method for solving (usually overdetermined) systems of linear equations [19]. The simplicity of the method makes it useful in a wide array of applications ranging from computer tomography to digital signal processing [16, 18, 27]. Recently, Strohmer and Vershynin [46] proposed a variant of the Kaczmarz method using a

random selection method which select rows with probability proportional to their squared norm, and showed that using this selection strategy, a desired accuracy of ε can be reached in the noiseless setting in a number of steps that scales like $\log(1/\varepsilon)$ and *linearly* in the condition number.

1.1 Importance sampling in stochastic optimization

From a birds-eye perspective, this paper aims to extend the notion of importance sampling from stochastic sampling methods for numerical linear algebra applications, to more general stochastic convex optimization problems. Strohmer and Vershynin's incorporation of importance sampling into the Kaczmarz setup [46] is just one such example, and most closely related to the SGD set-up. But importance sampling has also been considered in stochastic coordinate-descent methods [33,40]. There also, the weights are proportional to some power of the Lipschitz constants (of the gradient coordinates).

Importance sampling has also played a key role in designing sampling-based low-rank matrix approximation algorithms—both row/column based sampling and entry-wise sampling—where it goes by the name of *leverage score* sampling. The resulting sampling methods are again proportional to the squared Euclidean norms of rows and columns of the underlying matrix. See [5,9,25,44], and references therein for applications to the column subset selection problem and matrix completion. See [14,24,48] for applications of importance sampling to the Nyström method.

Importance sampling has also been introduced to the compressive sensing framework, where it translates to sampling rows of an orthonormal matrix proportionally to their squared inner products with the rows of a second orthonormal matrix in which the underlying signal is assumed sparse. See [20,39] for more details.

1.2 Contributions

Inspired by the analysis of Strohmer and Vershynin and Bach and Moulines, we prove convergence results for stochastic gradient descent as well as for SGD variants where gradient estimates are chosen based on a *weighted sampling distribution*, highlighting the role of importance sampling in SGD.

We first show (Corollary 2.2 in Sect. 2) that without perturbing the sampling distribution, we can obtain a linear dependence on the *uniform conditioning* ($\sup L_i/\mu$), but it is *not possible* to obtain a linear dependence on the *average conditioning* $\mathbb{E}[L_i/\mu]$. This is a quadratic improvement over the previous results [1] in regimes where the components have similar Lipschitz constants.

We then turn to importance sampling, using a weighted sampling distribution. We show that weighting components proportionally to their Lipschitz constants L_i , as is essentially done by Strohmer and Vershynin, can reduce the dependence on the conditioning to a linear dependence on the average conditioning $\mathbb{E}[L_i/\mu]$. However, this comes at an increased dependence on the residual σ^2 . But, we show that by only *partially biasing* the sampling towards L_i , we can enjoy the best of both worlds, obtaining a linear dependence on the average conditioning $\mathbb{E}[L_i/\mu]$, without amplifying the dependence on the residual. Thus, using importance sampling, we obtain a guarantee

dominating, and improving over the previous best-known results [1] (Corollary 3.1 in Sect. 2).

In Sect. 4, we consider the benefits of reweighted SGD also in other scenarios and regimes. We show how also for smooth but not-strongly-convex objectives, importance sampling can improve a dependence on a uniform bound over smoothness, ($\sup L_i$), to a dependence on the average smoothness $\mathbb{E}[L_i]$ —such an improvement is not possible without importance sampling. For non-smooth objectives, we show that importance sampling can eliminate a dependence on the variance in the Lipschitz constants of the components. In parallel work we recently became aware of, Zhao and Zhang [51] also consider importance sampling for non-smooth objectives, including composite objectives, suggesting the same reweighting as we obtain here.

Finally, in Sect. 5, we turn to the Kaczmarz algorithm, explain how it is an instantiation of SGD, and how using partially biased sampling improves known guarantees in this context as well. We show that the randomized Kaczmarz method with uniform i.i.d. row selection can be recast as an instance of preconditioned Stochastic Gradient Descent acting on a re-weighted least squares problem and through this connection, provide exponential convergence rates for this algorithm. We also consider the Kaczmarz algorithm corresponding to SGD with hybrid row selection strategy which shares the exponential convergence rate of Strohmer and Vershynin [46] while also sharing a small error residual term of the SGD algorithm. This presents a clear tradeoff between convergence rate and the convergence residual, not present in other results for the method.

2 SGD for strongly convex smooth optimization

We consider the problem of minimizing a smooth convex function,

$$\mathbf{x}_\star = \underset{\mathbf{x}}{\operatorname{argmin}} F(\mathbf{x}) \quad (2.1)$$

where $F(\mathbf{x})$ is of the form $F(\mathbf{x}) = \mathbb{E}_{i \sim \mathcal{D}} f_i(\mathbf{x})$ for smooth functionals $f_i : \mathcal{H} \rightarrow \mathcal{R}$ over $\mathcal{H} = \mathcal{R}^d$ endowed with the standard Euclidean norm $\|\cdot\|_2$, or over a Hilbert space \mathcal{H} with the norm $\|\cdot\|_2$. Here i is drawn from some *source distribution* \mathcal{D} over an arbitrary probability space. Throughout this manuscript, unless explicitly specified otherwise, expectations will be with respect to indices drawn from the source distribution \mathcal{D} . That is, we write $\mathbb{E} f_i(\mathbf{x}) = \mathbb{E}_{i \sim \mathcal{D}} f_i(\mathbf{x})$. We also denote by σ^2 the “residual” quantity at the minimum,

$$\sigma^2 = \mathbb{E} \|\nabla f_i(\mathbf{x}_\star)\|_2^2.$$

We will instate the following assumptions on the function F :

- (1) Each f_i is continuously differentiable and the gradient function ∇f_i has Lipschitz constant L_i ; that is, $\|\nabla f_i(\mathbf{x}) - \nabla f_i(\mathbf{y})\|_2 \leq L_i \|\mathbf{x} - \mathbf{y}\|_2$ for all vectors \mathbf{x} and \mathbf{y} .
- (2) F has strong convexity parameter μ ; that is, $\langle \mathbf{x} - \mathbf{y}, \nabla F(\mathbf{x}) - \nabla F(\mathbf{y}) \rangle \geq \mu \|\mathbf{x} - \mathbf{y}\|_2^2$ for all vectors \mathbf{x} and \mathbf{y} .

We denote $\sup L$ the supremum of the support of L_i , i.e. the smallest L such that $L_i \leq L$ a.s., and similarly denote $\inf L$ the infimum. We denote the average Lipschitz constant as $\bar{L} = \mathbb{E}L_i$.

A unbiased gradient estimate for $F(\mathbf{x})$ can be obtained by drawing $i \sim \mathcal{D}$ and using $\nabla f_i(\mathbf{x})$ as the estimate. The SGD updates with (fixed) step size γ based on these gradient estimates are then given by:

$$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - \gamma \nabla f_{i_k}(\mathbf{x}_k) \tag{2.2}$$

where $\{i_k\}$ are drawn i.i.d. from \mathcal{D} . We are interested in the distance $\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2$ of the iterates from the unique minimum, and denote the initial distance by $\varepsilon_0 = \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2$.

Bach and Moulines [1, Theorem 1] considered this setting¹ and established that

$$k = 2 \log(\varepsilon/\varepsilon_0) \left(\frac{\mathbb{E}L_i^2}{\mu^2} + \frac{\sigma^2}{\mu^2 \varepsilon} \right) \tag{2.3}$$

SGD iterations of the form (2.2), with an appropriate step-size, are sufficient to ensure $\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq \varepsilon$, where the expectations is over the random sampling. As long as $\sigma^2 = 0$, i.e. the same minimizer \mathbf{x}_\star minimizes all components $f_i(\mathbf{x})$ (though of course it need not be a unique minimizer of any of them), this yields linear convergence to \mathbf{x}_\star , with a graceful degradation as $\sigma^2 > 0$. However, in the linear convergence regime, the number of required iterations scales with the expected *squared* conditioning $\mathbb{E}L_i^2/\mu^2$. In this paper, we reduce this quadratic dependence to a linear dependence. We begin with a guarantee ensuring linear dependence, though with a dependence on $\sup L/\mu$ rather than $\mathbb{E}L_i/\mu$:

Theorem 2.1 *Let each f_i be convex where ∇f_i has Lipschitz constant L_i , with $L_i \leq \sup L$ a.s., and let $F(\mathbf{x}) = \mathbb{E}f_i(\mathbf{x})$ be μ -strongly convex. Set $\sigma^2 = \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|_2^2$, where $\mathbf{x}_\star = \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$. Suppose that $\gamma < \frac{1}{\sup L}$. Then the SGD iterates given by (2.2) satisfy:*

$$\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq [1 - 2\gamma\mu(1 - \gamma \sup L)]^k \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2 + \frac{\gamma\sigma^2}{\mu(1 - \gamma \sup L)}, \tag{2.4}$$

where the expectation is with respect to the sampling of $\{i_k\}$.

If we are given a desired tolerance, $\|\mathbf{x} - \mathbf{x}_\star\|_2^2 \leq \varepsilon$, and we know the Lipschitz constants and parameters of strong convexity, we may optimize the step-size γ , and obtain:

Corollary 2.2 *For any desired ε , using a step-size of*

$$\gamma = \frac{\mu\varepsilon}{2\varepsilon\mu \sup L + 2\sigma^2}$$

¹ Bach and Moulines’s results are somewhat more general. Their Lipschitz requirement is a bit weaker and more complicated, but in terms of L_i yields (2.3). They also study the use of polynomial decaying step-sizes, but these do not lead to improved runtime if the target accuracy is known ahead of time.

we have that after

$$k = 2 \log(2\varepsilon_0/\varepsilon) \left(\frac{\sup L}{\mu} + \frac{\sigma^2}{\mu^2\varepsilon} \right) \tag{2.5}$$

SGD iterations, $\mathbb{E}\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq \varepsilon$, where $\varepsilon_0 = \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2$ and where the expectation is with respect to the sampling of $\{i_k\}$.

Proof Substituting $\gamma = \frac{\mu\varepsilon}{2\varepsilon\mu \sup L + 2\sigma^2}$ into the second term of (2.4) and simplifying gives the bound

$$\frac{\gamma\sigma^2}{\mu(1 - \gamma \sup L)} \leq \varepsilon/2.$$

Now asking that

$$[1 - 2\gamma\mu(1 - \gamma \sup L)]^k \varepsilon_0 \leq \varepsilon/2,$$

substituting for γ , and rearranging to solve for k , shows that we need k such that

$$k \log \left(1 - \frac{\mu^2\varepsilon(\mu\varepsilon \sup L + 2\sigma^2)}{2(\mu\varepsilon \sup L + \sigma^2)^2} \right) \leq -\log \left(\frac{2\varepsilon_0}{\varepsilon} \right).$$

Utilizing the fact that $-1/\log(1 - x) \leq 1/x$ for $0 < x \leq 1$ and rearranging again yields the requirement that

$$k \geq \log \left(\frac{2\varepsilon_0}{\varepsilon} \right) \cdot \frac{2(\mu\varepsilon \sup L + \sigma^2)^2}{\mu^2\varepsilon(\mu\varepsilon \sup L + 2\sigma^2)}.$$

Noting that this inequality holds when $k \geq 2 \log \left(\frac{2\varepsilon_0}{\varepsilon} \right) \cdot \frac{\mu\varepsilon \sup L + \sigma^2}{\mu^2\varepsilon}$ yields the stated number of steps k in (2.5). Since the expression on the right hand side of (2.4) decreases with k , the corollary is proven. \square

Proof sketch The crux of the improvement over Bach and Moulines is in a tighter recursive equation. Bach and Moulines rely on the recursion

$$\|\mathbf{x}_{k+1} - \mathbf{x}_\star\|_2^2 \leq \left(1 - 2\gamma\mu + 2\gamma^2 L_i^2 \right) \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 + 2\gamma^2\sigma^2,$$

whereas we use the Co-Coercivity Lemma 8.1, with which we can obtain the recursion

$$\|\mathbf{x}_{k+1} - \mathbf{x}_\star\|_2^2 \leq \left(1 - 2\gamma\mu + 2\gamma^2\mu L_i \right) \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 + 2\gamma^2\sigma^2,$$

where L_i is the Lipschitz constant of the component used in the current iterate. The significant difference is that one of the factors of L_i (an upper bound on the second derivative), in the third term inside the parenthesis, is replaced by μ (a lower bound on the second derivative of F). A complete proof can be found in the ‘‘Appendix’’.

Comparison to results of Bach and Moulines Our bound (2.5) replaces the dependence on the average square conditioning ($\mathbb{E}L_i^2/\mu^2$) with a linear dependence on the uniform conditioning ($\sup L/\mu$). When all Lipschitz constants L_i are of similar magnitude, this is a quadratic improvement in the number of required iterations. However, when different components f_i have widely different scaling, i.e. L_i are highly variable, the supremum might be larger than the average square conditioning.

Tightness Considering the above, one might hope to obtain a linear dependence on the average conditioning $\bar{L}/\mu = \mathbb{E}L_i/\mu$. However, as the following example shows, this is not possible. Consider a uniform source distribution over $N + 1$ quadratics, with the first quadratic f_1 being $\frac{N}{2}(\mathbf{x}[1] - b)^2$ and all others being $\frac{1}{2}\mathbf{x}[2]^2$, and $b = \pm 1$. Any method must examine f_1 in order to recover \mathbf{x} to within error less than one, but by uniformly sampling indices i , this takes $(N + 1)$ iterations in expectation. It is easy to verify that in this case, $\sup L_i = L_1 = N$, $\bar{L} = 2\frac{N}{N+1} < 2\mathbb{E}L_i^2 = N$, and $\mu = \frac{N}{N+1}$. For large N , a linear dependence on \bar{L}/μ would mean that a constant number of iterations suffice (as $\bar{L}/\mu = 2$), but we just saw that any method that sampled i uniformly must consider at least $(N + 1)$ samples in expectation to get non-trivial error. Note that both $\sup L_i/\mu = N + 1$ and $\mathbb{E}L_i^2/\mu^2 \simeq N + 1$ indeed correspond to the correct number of iterations required by SGD.

We therefore see that the choice between a dependence on the average quadratic conditioning $\mathbb{E}L_i^2/\mu^2$, or a linear dependence on the uniform conditioning $\sup L/\mu$, is unavoidable. A linear dependence on the average conditioning \bar{L}/μ is not possible with any method that samples from the source distribution \mathcal{D} . In the next section, we will show how we can obtain a linear dependence on the average conditioning \bar{L}/μ , using importance sampling, i.e. by sampling from a modified distribution.

3 Importance sampling

We will now consider stochastic gradient descent, where gradient estimates are sampled from a weighted distribution.

3.1 Reweighting a distribution

For a weight function $w(i)$ which assigns a non-negative weight $w(i) \geq 0$ to each index i , the weighted distribution $\mathcal{D}^{(w)}$ is defined as the distribution such that

$$\mathbb{P}_{\mathcal{D}^{(w)}}(I) \propto \mathbb{E}_{i \sim \mathcal{D}}[\mathbb{1}_I(i)w(i)],$$

where I is an event (subset of indices) and $\mathbb{1}_I(\cdot)$ its indicator function. For a discrete distribution \mathcal{D} with probability mass function $p(i)$ this corresponds to weighting the probabilities to obtain a new probability mass function:

$$p^{(w)}(i) \propto w(i)p(i).$$

Similarly, for a continuous distribution, this corresponds to multiplying the density by $w(i)$ and renormalizing.

One way to construct the weighted distribution $\mathcal{D}^{(w)}$, and sample from it, is through *rejection sampling*: sample $i \sim \mathcal{D}$, and accept with probability $w(i)/W$, for some $W \geq \sup_i w(i)$. Otherwise, reject and continue to re-sample until a suggestion i is accepted. The accepted samples are then distributed according to $\mathcal{D}^{(w)}$.

We use $\mathbb{E}^{(w)}[\cdot] = E_{i \sim \mathcal{D}^{(w)}}[\cdot]$ to denote an expectation where indices are sampled from the weighted distribution $\mathcal{D}^{(w)}$. An important property of such an expectation is that for any quantity $X(i)$ that depends on i :

$$\mathbb{E}^{(w)} \left[\frac{1}{w(i)} X(i) \right] = \mathbb{E}[X(i)] / \mathbb{E}[w(i)], \tag{3.1}$$

where recall that the expectations on the r.h.s. are with respect to $i \sim \mathcal{D}$. In particular, when $\mathbb{E}[w(i)] = 1$, we have that $\mathbb{E}^{(w)} \left[\frac{1}{w(i)} X(i) \right] = \mathbb{E}X(i)$. In fact, we will consider only weights s.t. $\mathbb{E}[w(i)] = 1$, and refer to such weights as *normalized*.

3.2 Reweighted SGD

For any normalized weight function $w(i)$, we can weight each component f_i , defining:

$$f_i^{(w)}(\mathbf{x}) = \frac{1}{w(i)} f_i(\mathbf{x}) \tag{3.2}$$

and obtain

$$F(\mathbf{x}) = \mathbb{E}^{(w)}[f_i^{(w)}(\mathbf{x})]. \tag{3.3}$$

The representation (3.3) is an equivalent, and equally valid, stochastic representation of the objective $F(\mathbf{x})$, and we can just as well base SGD on this representation. In this case, at each iteration we sample $i \sim \mathcal{D}^{(w)}$ and then use $\nabla f_i^{(w)}(\mathbf{x}) = \frac{1}{w(i)} \nabla f_i(\mathbf{x})$ as an unbiased gradient estimate. SGD iterates based on the representation (3.3), which we will also refer to as w -weighted SGD, are then given by

$$\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k - \frac{\gamma}{w(i_k)} \nabla f_{i_k}(\mathbf{x}_k) \tag{3.4}$$

where $\{i_k\}$ are drawn i.i.d. from $\mathcal{D}^{(w)}$.

The important observation here is that all SGD guarantees are equally valid for the w -weighted updates (3.4)—the objective is the same objective $F(\mathbf{x})$, the sub-optimality is the same, and the minimizer \mathbf{x}_* is the same. We do need, however, to calculate the relevant quantities controlling SGD convergence with respect to the modified components $f_i^{(w)}$ and the weighted distribution $\mathcal{D}^{(w)}$.

3.3 Strongly convex smooth optimization using weighted SGD

We now return to the analysis of strongly convex smooth optimization and investigate how re-weighting can yield a better guarantee. To do so, we must analyze the relevant quantities involved.

The Lipschitz constant $L_i^{(w)}$ of each component $f_i^{(w)}$ is now scaled, and we have, $L_i^{(w)} = \frac{1}{w(i)}L_i$. The supremum is given by:

$$\sup L_{(w)} = \sup_i L_i^{(w)} = \sup_i \frac{L_i}{w(i)}. \tag{3.5}$$

It is easy to verify that (3.5) is minimized by the weights

$$w(i) = \frac{L_i}{\bar{L}}, \tag{3.6}$$

and that with this choice of weights

$$\sup L_{(w)} = \sup_i \frac{L_i}{L_i/\bar{L}} = \bar{L}. \tag{3.7}$$

Note that the average Lipschitz constant $\bar{L} = \mathbb{E}[L_i] = \mathbb{E}^{(w)}[L_i^{(w)}]$ is invariant under weightings.

Before applying Corollary 2.2, we must also calculate:

$$\begin{aligned} \sigma_{(w)}^2 &= \mathbb{E}^{(w)} \left[\|\nabla f_i^{(w)}(\mathbf{x}_\star)\|_2^2 \right] = \mathbb{E}^{(w)} \left[\frac{1}{w(i)^2} \|\nabla f_i(\mathbf{x}_\star)\|_2^2 \right] \\ &= \mathbb{E} \left[\frac{1}{w(i)} \|\nabla f_i(\mathbf{x}_\star)\|_2^2 \right] = \mathbb{E} \left[\frac{\bar{L}}{L_i} \|\nabla f_i(\mathbf{x}_\star)\|_2^2 \right] \leq \frac{\bar{L}}{\inf L} \sigma^2. \end{aligned} \tag{3.8}$$

Now, applying Corollary 2.2 to the w -weighted SGD iterates (3.4) with weights (3.6), we have that, with an appropriate stepsize,

$$\begin{aligned} k &= 2 \log(2\varepsilon_0/\varepsilon) \left(\frac{\sup L_{(w)}}{\mu} + \frac{\sigma_{(w)}^2}{\mu^2\varepsilon} \right) \\ &\leq 2 \log(2\varepsilon_0/\varepsilon) \left(\frac{\bar{L}}{\mu} + \frac{\bar{L}}{\inf L} \cdot \frac{\sigma^2}{\mu^2\varepsilon} \right) \end{aligned} \tag{3.9}$$

iterations are sufficient for $\mathbb{E}^{(w)} \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq \varepsilon$, where \mathbf{x}_\star , μ and ε_0 are exactly as in Corollary 2.2.

3.4 Partially biased sampling

If $\sigma^2 = 0$, i.e. we are in the “realizable” situation, with true linear convergence, then we also have $\sigma_{(w)}^2 = 0$. In this case, we already obtain the desired guarantee: linear convergence with a linear dependence on the average conditioning \bar{L}/μ , strictly improving over Bach and Moulines. However, the inequality in (3.8) might be tight in the presence of components with very small L_i that contribute towards the residual error (as might well be the case for a small component). When $\sigma^2 > 0$, we therefore get a dissatisfying scaling of the second term, relative to Bach and Moulines, by a factor of $\bar{L}/\inf L$.

Fortunately, we can easily overcome this factor. To do so, consider sampling from a distribution which is a *mixture* of the original source distribution and its re-weighting using the weights (3.6). That is, sampling using the weights:

$$w(i) = \frac{1}{2} + \frac{1}{2} \cdot \frac{L_i}{\bar{L}}. \tag{3.10}$$

We refer to this as *partially biased sampling*. Using these weights, we have

$$\sup_i L_{(w)} = \sup_i \frac{1}{\frac{1}{2} + \frac{1}{2} \cdot \frac{L_i}{\bar{L}}} L_i \leq 2\bar{L} \tag{3.11}$$

and

$$\sigma_{(w)}^2 = \mathbb{E} \left[\frac{1}{\frac{1}{2} + \frac{1}{2} \cdot \frac{L_i}{\bar{L}}} \|\nabla f_i(\mathbf{x}_\star)\|_2^2 \right] \leq 2\sigma^2. \tag{3.12}$$

Plugging these into Corollary 2.2 we obtain:

Corollary 3.1 *Let each f_i be convex where ∇f_i has Lipschitz constant L_i and let $F(\mathbf{x}) = \mathbb{E}_{i \sim \mathcal{D}}[f_i(\mathbf{x})]$, where $F(\mathbf{x})$ is μ -strongly convex. Set $\sigma^2 = \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|_2^2$, where $\mathbf{x}_\star = \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$. For any desired ε , using a stepsize of*

$$\gamma = \frac{\mu\varepsilon}{4(\varepsilon\mu\bar{L} + \sigma^2)}$$

we have that after

$$k = 4 \log(2\varepsilon_0/\varepsilon) \left(\frac{\bar{L}}{\mu} + \frac{\sigma^2}{\mu^2\varepsilon} \right) \tag{3.13}$$

iterations of w -weighted SGD (3.4) with weights specified by (3.10), $\mathbb{E}^{(w)}\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq \varepsilon$, where $\varepsilon_0 = \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2$ and $\bar{L} = \mathbb{E}L_i$.

We now obtain the desired linear scaling on \bar{L}/μ , without introducing any additional factor to the residual term, except for a constant factor of two. We thus obtain a result which dominates Bach and Moulines (up to a factor of 2) and substantially improves upon it (with a linear rather than quadratic dependence on the conditioning).

One might also ask whether the previous best known result (2.3) could be improved using weighted sampling. The relevant quantity to consider is the average square Lipschitz constant for the weighted representation (3.3):

$$\overline{L^2_{(w)}} \doteq \mathbb{E}^{(w)} \left[\left(L_i^{(w)} \right)^2 \right] = \mathbb{E}^{(w)} \left[\frac{L_i^2}{w(i)^2} \right] = \mathbb{E} \left[\frac{L_i^2}{w(i)} \right]. \tag{3.14}$$

Interestingly, this quantity is minimized by the same weights as $\sup L_{(w)}$, given by (3.6), and with these weights we have:

$$\overline{L^2_{(w)}} = \mathbb{E} \left[\frac{L_i^2}{L_i/\overline{L}} \right] = \overline{L} \mathbb{E} L_i = \overline{L}^2. \tag{3.15}$$

Again, we can use the partially biased weights give in (3.10), which yields $\overline{L^2_{(w)}} \leq 2\overline{L}^2$ and also ensures $\sigma^2_{(w)} \leq 2\sigma^2$. In any case, we get a dependence on $\overline{L}^2 = (\mathbb{E} L_i)^2 \leq \mathbb{E}[L_i^2]$ instead of $\overline{L}^2 = \mathbb{E}[L_i^2]$, which is indeed an improvement. Thus, the Bach and Moulines guarantee is also improved by using biased sampling, and in particular the partially biased sampling specified by the weights (3.10). However, relying on Bach and Moulines we still have a quadratic dependence on $(\overline{L}/\mu)^2$, as opposed to the linear dependence we obtain in Corollary 3.1.

3.5 Implementing importance sampling

As discussed above, when the magnitudes of L_i are highly variable, importance sampling is necessarily in order to obtain a dependence on the average, rather than worst-case, conditioning. In some applications, especially when the Lipschitz constants are known in advance or easily calculated or bounded, such importance sampling might be possible by directly sampling from $\mathcal{D}^{(w)}$. This is the case, for example, in trigonometric approximation problems or linear systems which need to be solved repeatedly, or when the Lipschitz constant is easily computed from the data, and multiple passes over the data are needed anyway. We do acknowledge that in other regimes, when data is presented in an online fashion, or when we only have sampling access to the source distribution \mathcal{D} (or the implied distribution over gradient estimates), importance sampling might be difficult.

One option that could be considered, in light of the above results, is to use rejection sampling to simulate sampling from $\mathcal{D}^{(w)}$. For the weights (3.6), this can be done by accepting samples with probability proportional to $L_i / \sup L$. The overall probability of accepting a sample is then $\overline{L} / \sup L$, introducing an additional factor of $\sup L / \overline{L}$. This results in a sample complexity with a linear dependence on $\sup L$, as in Corollary 2.2 (for the weights (3.10), we can first accept with probability 1/2, and then if we do not accept, perform this procedure). Thus, if we are presented samples from \mathcal{D} , and the cost of obtaining the sample dominates the cost of taking the gradient step, we do not gain (but do not lose much either) from rejection sampling. We might

still gain from rejection sampling if the cost of operating on a sample (calculating the actual gradient and taking a step according to it) dominates the cost of obtaining it and (a bound on) the Lipschitz constant.

3.6 A family of partially biased schemes

The choice of weights (3.10) corresponds to an equal mix of uniform and fully biased sampling. More generally, we could consider sampling according to any one of a family of weights which interpolate between uniform and fully biased sampling:

$$w^\lambda(i) = \lambda + (1 - \lambda) \frac{L_i}{\bar{L}}, \quad \lambda \in [0, 1]. \tag{3.16}$$

To be concrete, we summarize below the a template algorithm for SGD with partially biased sampling:

Algorithm 3.1: Stochastic Gradient Descent with Partially Biased Sampling

Input:

- Initial estimate $\mathbf{x}_0 \in \mathbb{R}^d$
- Bias parameter $\lambda \in [0, 1]$
- Step size $\gamma > 0$
- Tolerance parameter $\delta > 0$
- Access to the source distribution \mathcal{D}
- If $\lambda < 1$: bounds on the Lipschitz constants L_i ; the weights $w^\lambda(i)$ derived from them (see eq. 3.16); and access to the weighted distribution $\mathcal{D}^{(\lambda)}$.

Output: Estimated solution $\hat{\mathbf{x}}$ to the problem $\min_{\mathbf{x}} F(\mathbf{x})$

$k \leftarrow 0$
repeat
 $k \leftarrow k + 1$
 Draw an index $i \sim \mathcal{D}^{(\lambda)}$.
 $\mathbf{x}_k \leftarrow \mathbf{x}_{k-1} - \frac{\gamma}{w^\lambda(i)} \nabla f_i(\mathbf{x}_{k-1})$
until $\nabla F(\mathbf{x}) \leq \delta$
 $\hat{\mathbf{x}} \leftarrow \mathbf{x}_k$

For arbitrary $\lambda \in [0, 1]$, we have the bounds

$$\sup_i L_{(w^\lambda)} = \sup_i \frac{L_i}{\lambda + (1 - \lambda) \frac{L_i}{\bar{L}}} \leq \min \left(\frac{\bar{L}}{1 - \lambda}, \frac{\sup_i L_i}{\lambda} \right)$$

and

$$\sigma^2_{(w^\lambda)} = \mathbb{E} \left[\frac{1}{\lambda + (1 - \lambda) \frac{L_i}{\bar{L}}} \|\nabla f_i(\mathbf{x}_\star)\|_2^2 \right] \leq \max \left(\frac{1}{\lambda}, \frac{\bar{L}}{(1 - \lambda) \inf_i L_i} \right) \sigma^2$$

Plugging these quantities into Corollary 2.2, we obtain:

Corollary 3.2 *Let each f_i be convex where ∇f_i has Lipschitz constant L_i and let $F(\mathbf{x}) = \mathbb{E}_{i \sim \mathcal{D}}[f_i(\mathbf{x})]$, where $F(\mathbf{x})$ is μ -strongly convex. Set $\sigma^2 = \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|_2^2$, where $\mathbf{x}_\star = \operatorname{argmin}_{\mathbf{x}} F(\mathbf{x})$. For any desired ε , using a stepsize of*

$$\gamma = \frac{\mu\varepsilon}{2\varepsilon\mu \min\left(\frac{\bar{L}}{1-\lambda}, \frac{\sup_i L_i}{\lambda}\right) + 2 \max\left(\frac{1}{\lambda}, \frac{\bar{L}}{(1-\lambda)\inf_i L_i}\right)\sigma^2}$$

we have that after

$$k = 2 \log(2\varepsilon_0/\varepsilon) \left(\frac{\min\left(\frac{\bar{L}}{1-\lambda}, \frac{\sup_i L_i}{\lambda}\right)}{\mu} + \frac{\max\left(\frac{1}{\lambda}, \frac{\bar{L}}{(1-\lambda)\inf_i L_i}\right)\sigma^2}{\mu^2\varepsilon} \right)$$

iterations of w -weighted SGD (3.4) with partially biased weights (3.16), $\mathbb{E}^{(w)}\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq \varepsilon$, where $\varepsilon_0 = \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2$ and $\bar{L} = \mathbb{E}L_i$.

In this corollary, even if λ is close to 1, i.e. we add only a small amount of bias to the sampling, we obtain a bound with a linear dependence on the average conditioning \bar{L}/μ (multiplied by a factor of $\frac{1}{\lambda}$), since we can bound $\min\left(\frac{\bar{L}}{1-\lambda}, \frac{\sup_i L_i}{\lambda}\right) \leq \frac{\bar{L}}{1-\lambda}$.

4 Importance sampling for SGD in other scenarios

In the previous section, we considered SGD for smooth and strongly convex objectives, and were particularly interested in the regime where the residual σ^2 is low, and the linear convergence term is dominant. Weighted SGD could of course be relevant also in other scenarios, and we now briefly survey them, as well as relate them to our main scenario of interest.

4.1 Smooth, not strongly convex

When each component f_i is convex, non-negative, and has an L_i -Lipschitz gradient, but the objective $F(\mathbf{x})$ is not necessarily strongly convex, then after

$$k = O\left(\frac{(\sup L)\|\mathbf{x}_\star\|_2^2}{\varepsilon} \cdot \frac{F(\mathbf{x}_\star) + \varepsilon}{\varepsilon}\right) \tag{4.1}$$

iterations of SGD with an appropriately chosen step-size we will have $F(\bar{\mathbf{x}}_k) \leq F(\mathbf{x}_\star) + \varepsilon$, where $\bar{\mathbf{x}}_k$ is an appropriate averaging of the k iterates Srebro et al. [45]. The relevant quantity here determining the iteration complexity is again $\sup L$. Furthermore, Srebro et al. [45], relying on an example from Foygel and Srebro [13], point out that the dependence on the supremum is unavoidable and *cannot* be replaced with the average Lipschitz constant \bar{L} . That is, if we sample gradients according to the source distribution \mathcal{D} , we must have a linear dependence on $\sup L$.

The only quantity in the bound (4.1) that changes with a re-weighting is $\sup L$ —all other quantities ($\|\mathbf{x}_*\|_2^2$, $F(\mathbf{x}_*)$, and the sub-optimality ε) are invariant to re-weightings. We can therefore replace the dependence on $\sup L$ with a dependence on $\sup L_{(w)}$ by using a weighted SGD as in (3.4). As we already calculated, the optimal weights are given by (3.6), and using them we have $\sup L_{(w)} = \bar{L}$. In this case, there is no need for partially biased sampling, and we obtain that with an appropriate step-size,

$$k = O\left(\frac{\bar{L}\|\mathbf{x}_*\|_2^2}{\varepsilon} \cdot \frac{F(\mathbf{x}_*) + \varepsilon}{\varepsilon}\right) \tag{4.2}$$

iterations of weighed SGD updates (3.4) using the weights (3.6) suffice.

We again see that using importance sampling allows us to reduce the dependence on $\sup L$, which is unavoidable without biased sampling, to a dependence on \bar{L} .

4.2 Non-smooth objectives

We now turn to non-smooth objectives, where the components f_i might not be smooth, but each component is G_i -Lipschitz. Roughly speaking, G_i is a bound on the first derivative (gradient) of f_i , while L_i is a bound on the second derivatives of f_i . Here, the performance of SGD depends on the second moment $\overline{G^2} = \mathbb{E}[G_i^2]$. The precise iteration complexity depends on whether the objective is strongly convex or whether \mathbf{x}_* is bounded, but in either case depends linearly on $\overline{G^2}$ (see e.g. [32,43]).

By using weighted SGD we can replace the linear dependence on $\overline{G^2}$ with a linear dependence on $\overline{G_{(w)}^2} = \mathbb{E}^{(w)}[(G_i^{(w)})^2]$, where $G_i^{(w)}$ is the Lipschitz constant of the scaled $f_i^{(w)}$ and is given by $G_i^{(w)} = G_i/w(i)$. Again, this follows directly from the standard SGD guarantees, where we consider the representation (3.3) and use any subgradient from $\partial f_i^{(w)}(\mathbf{x})$.

We can calculate:

$$\overline{G_{(w)}^2} = \mathbb{E}^{(w)}\left[\frac{G_i^2}{w(i)^2}\right] = \mathbb{E}\left[\frac{G_i^2}{w(i)}\right] \tag{4.3}$$

which is minimized by the weights:

$$w(i) = \frac{G_i}{\overline{G}} \tag{4.4}$$

where $\overline{G} = \mathbb{E}G_i$. Using these weights we have $\overline{G_{(w)}^2} = \mathbb{E}[G_i]^2 = \overline{G}^2$. Using importance sampling, we can thus reduce the linear dependence on $\overline{G^2}$ to a linear dependence on \overline{G}^2 . Its helpful to recall that $\overline{G^2} = \overline{G}^2 + \text{Var}[G_i]$. What we save is therefore exactly the variance of the Lipschitz constants G_i .

In parallel work, Zhao and Zhang [51] also consider importance sampling for stochastic optimization for non-smooth objectives. Zhao and Zhang consider a more gen-

eral setting, with a composite objective that is only partially linearized. But also there, the iteration complexity depends on the second moment of the gradient estimates, and the analysis performed above applies (Zhao and Zhang perform a specialized analysis instead).

4.3 Non-realizable regime

Returning to the smooth and strongly convex setting of Sects. 2 and 3, let us consider more carefully the residual term $\sigma^2 = \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|_2^2$. This quantity definitively depends on the weighting, and in the analysis of Sect. 3.3, we avoided increasing it too much, introducing partial biasing for this purpose. However, if this is the dominant term, we might want to choose weights so as to minimize this term. The optimal weights here would be proportional to $\|\nabla f_i(\mathbf{x}_\star)\|_2$. The problem is that we do not know the minimizer \mathbf{x}_\star , and so cannot calculate these weights. Approaches which dynamically update the weights based on the current iterates as a surrogate for \mathbf{x}_\star are possible, but beyond the scope of this paper.

An alternative approach is to bound $\|\nabla f_i(\mathbf{x}_\star)\|_2 \leq G_i$ and so $\sigma^2 \leq \overline{G^2}$. Taking this bound, we are back to the same quantity as in the non-smooth case, and the optimal weights are proportional to G_i . Note that this is a different weighting than using weights proportional to L_i , which optimize the linear-convergence term as studied in Sect. 3.3.

To understand how weighting according to G_i and L_i are different, consider a generalized linear objective where $f_i(\mathbf{x}) = \phi_i(\langle \mathbf{z}_i, \mathbf{x} \rangle)$, and ϕ_i is a scalar function with $|\phi_i'| \leq G_\phi$ and $|\phi_i''| \leq L_\phi$. We have that $G_i \propto \|\mathbf{z}_i\|_2$ while $L_i \propto \|\mathbf{z}_i\|_2^2$. Weighting according to the Lipschitz constants of the gradients, i.e. the “smoothness” parameters, as in (3.6), versus weighting according to the Lipschitz constants of f_i as in (4.4), thus corresponds to weighting according to $\|\mathbf{z}_i\|_2^2$ versus $\|\mathbf{z}_i\|_2$, and are rather different. We can also calculate that weighing by $L_i \propto \|\mathbf{z}_i\|_2^2$ [i.e. following (3.6)], yields $\overline{G_{(w)}^2} = \overline{G^2} > \overline{G^2}$. That is, weights proportional to L_i yield a suboptimal gradient-dependent term (the same dependence as if no weighting at all was used). Conversely, using weights proportional to G_i , i.e. proportional to $\|\mathbf{z}_i\|_2$ yields $\sup L_{(w)} = (E[\sqrt{L_i}])\sqrt{\sup L}$ —a suboptimal dependence, though better than no weighting at all.

Again, as with partially biased sampling, we can weight by the average, $w(i) = \frac{1}{2} \cdot \frac{G_i}{\overline{G}} + \frac{1}{2} \cdot \frac{L_i}{\overline{L}}$ and ensure both terms are optimal up to a factor of two.

5 The least squares case and the randomized Kaczmarz method

A special case of interest is the least squares problem, where

$$F(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n (\langle \mathbf{a}_i, \mathbf{x} \rangle - b_i)^2 = \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2 \tag{5.1}$$

with \mathbf{b} an n -dimensional vector, \mathbf{A} an $n \times d$ matrix with rows \mathbf{a}_i , and $\mathbf{x}_\star = \operatorname{argmin}_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ is the least-squares solution. Writing the least squares problem (5.1) in the form (2.1), we see that the source distribution \mathcal{D} is uniform over $\{1, 2, \dots, n\}$, the components are $f_i = \frac{n}{2} (\langle \mathbf{a}_i, \mathbf{x} \rangle - b_i)^2$, the Lipschitz constants are $L_i = n \|\mathbf{a}_i\|_2^2$, the average Lipschitz constant is $\frac{1}{n} \sum_i L_i = \|\mathbf{A}\|_F^2$, the strong convexity parameter is $\mu = \frac{1}{\|(\mathbf{A}^T \mathbf{A})^{-1}\|_2}$, so that $K(\mathbf{A}) := \bar{L}/\mu = \|\mathbf{A}\|_F^2 \|(\mathbf{A}^T \mathbf{A})^{-1}\|_2$, and the residual is $\sigma^2 = n \sum_i \|\mathbf{a}_i\|_2^2 |\langle \mathbf{a}_i, \mathbf{x}_\star \rangle - b_i|^2$. Note that in the case that \mathbf{A} is not full-rank, one can instead replace μ with the smallest nonzero eigenvalue of $\mathbf{A}^* \mathbf{A}$ as in [23, Equation (3)]. In that case, we instead write $K(\mathbf{A}) = \|\mathbf{A}\|_F^2 \|(\mathbf{A}^T \mathbf{A})^\dagger\|_2$ as the appropriate condition number.

The randomized *Kaczmarz method* [7, 8, 15, 17, 27, 28, 46, 47, 49, 52] for solving the least squares problem (5.1) begins with an arbitrary estimate \mathbf{x}_0 , and in the k th iteration selects a row $i = i(k)$ i.i.d. at random from the matrix \mathbf{A} and iterates by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + c \cdot \frac{b_i - \langle \mathbf{a}_i, \mathbf{x}_k \rangle}{\|\mathbf{a}_i\|_2^2} \mathbf{a}_i, \tag{5.2}$$

where the step size $c = 1$ in the standard method.

Strohmer and Vershynin provided the first non-asymptotic convergence rates, showing that drawing rows proportionally to $\|\mathbf{a}_i\|_2^2$ leads to provable exponential convergence in expectation for the full-rank case [46]. Their method can easily be extended to the case when the matrix is not full-rank to yield convergence to some solution, see e.g. [23, Equation (3)]. Recent works use acceleration techniques to improve convergence rates [6, 10–12, 22, 29, 30, 34–38, 52].

However, one can easily verify that the iterates (5.2) are precisely weighted SGD iterates (3.4) with the fully biased weights (3.6).

The reduction of the quadratic dependence on the conditioning to a linear dependence in Theorem 2.1, as well as the use of biased sampling which we investigate here was motivated by Strohmer and Vershynin’s analysis of the randomized Kaczmarz method. Indeed, applying Theorem 2.1 to the weighted SGD iterates (2.2) for (5.1) with the weights (3.6) and a stepsize of $\gamma = 1$ yields precisely the Strohmer and Vershynin [46] guarantee.

Understanding the randomized Kaczmarz method as SGD allows us also to obtain improved methods and results for the randomized Kaczmarz method:

Using step-sizes As shown by Strohmer and Vershynin [46] and extended by Needell [28], the randomized Kaczmarz method with weighted sampling exhibits exponential convergence, but only to within a radius, or *convergence horizon*, of the least-squares solution. This is because a step-size of $\gamma = 1$ is used, and so the second term in (2.4) does not vanish. It has been shown [8, 15, 30, 47, 49] that changing the step size can allow for convergence inside of this convergence horizon, although non-asymptotic results have been difficult to obtain. Our results allow for finite-iteration guarantees with arbitrary step-sizes and can be immediately applied to this setting. Indeed, applying Theorem 2.1 with the weights (3.6) gives

Corollary 5.1 *Let A be an $n \times d$ matrix with rows \mathbf{a}_i . Set $\mathbf{e} = A\mathbf{x}_\star - \mathbf{b}$, where \mathbf{x}_\star is the minimizer of the problem*

$$\mathbf{x}_\star = \operatorname{argmin}_x \frac{1}{2} \|A\mathbf{x} - \mathbf{b}\|_2^2.$$

Suppose that $c < 1$. Set $a_{\min}^2 = \inf_i \|\mathbf{a}_i\|_2^2$, $a_{\max}^2 = \sup_i \|\mathbf{a}_i\|_2^2$. Then the expected error at the k th iteration of the Kaczmarz method described by (5.2) with row \mathbf{a}_i selected with probability $p_i = \|\mathbf{a}_i\|_2^2 / \|A\|_F^2$ satisfies

$$\mathbb{E} \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq \left[1 - \frac{2c(1-c)}{K(A)} \right]^k \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2 + \frac{c}{1-c} K(A)r, \tag{5.3}$$

with $r = \frac{\sigma^2}{n \|A\|_F^2 a_{\min}^2}$. The expectation is taken with respect to the weighted distribution over the rows.

When e.g. $c = \frac{1}{2}$, we recover the exponential rate of Strohmer and Vershynin [46] up to a factor of 2, and nearly the same convergence horizon. For arbitrary c , Corollary 5.1 implies a tradeoff between a smaller convergence horizon and a slower convergence rate.

Uniform row selection The Kaczmarz variant of Strohmer and Vershynin [46] calls for weighted row sampling, and thus requires pre-computing all the row norms. Although certainly possible in some applications, in other cases this might be better avoided. Understanding the randomized Kaczmarz as SGD allows us to apply Theorem 2.1 also with uniform weights (i.e. to the unbiased SGD), and obtain a randomized Kaczmarz using uniform sampling, which converges to the least-squares solution and enjoys finite-iteration guarantees:

Corollary 5.2 *Let A be an $n \times d$ matrix with rows \mathbf{a}_i . Let D be the diagonal matrix with terms $d_{j,j} = \|\mathbf{a}_i\|_2$, and consider the composite matrix $D^{-1}A$. Set $\mathbf{e}_w = D^{-1}(A\mathbf{x}_\star^w - \mathbf{b})$, where \mathbf{x}_\star^w is the minimizer of the weighted least squares problem*

$$\mathbf{x}_\star^w = \operatorname{argmin}_x \frac{1}{2} \|D^{-1}(A\mathbf{x} - \mathbf{b})\|_2^2.$$

Suppose that $c < 1$. Then the expected error after k iterations of the Kaczmarz method described by (5.2) with uniform row selection satisfies

$$\mathbb{E} \|\mathbf{x}_k - \mathbf{x}_\star^w\|_2^2 \leq \left[\left(1 - \frac{2c(1-c)}{K(D^{-1}A)} \right) \right]^k \|\mathbf{x}_0 - \mathbf{x}_\star^w\|_2^2 + \frac{c}{1-c} K(D^{-1}A)r_w,$$

where $r_w = \|\mathbf{e}_w\|_2^2/n$.

Note that the randomized Kaczmarz algorithm with uniform row selection converges exponentially to a *weighted* least-squares solution, to within arbitrary accuracy

by choosing sufficiently small stepsize c . Thus, in general, the randomized Kaczmarz algorithms with uniform and biased row selection converge (up to a convergence horizon) towards different solutions.

Partially biased sampling As in our SGD analysis, using the partially biased sampling weights is applicable also for the randomized Kaczmarz method. Applying Theorem 2.1 using weights (3.10) gives

Corollary 5.3 (Randomized Kaczmarz with partially biased sampling) *Let \mathbf{A} be an $n \times d$ matrix with rows \mathbf{a}_i . Set $\mathbf{e} = \mathbf{A}\mathbf{x}_\star - \mathbf{b}$, where \mathbf{x}_\star is the minimizer of the problem*

$$\mathbf{x}_\star = \operatorname{argmin}_x \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Suppose $c < 1/2$. Then the iterate \mathbf{x}_k of the modified Kaczmarz method described by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2c \cdot \frac{b_i - \langle \mathbf{a}_i, \mathbf{x}_k \rangle}{\|\mathbf{A}\|_F^2/n + \|\mathbf{a}_i\|_2^2} \mathbf{a}_i \tag{5.4}$$

with row \mathbf{a}_i selected with probability $p_i = \frac{1}{2} \cdot \frac{\|\mathbf{a}_i\|_2^2}{\|\mathbf{A}\|_F^2} + \frac{1}{2} \cdot \frac{1}{n}$ satisfies

$$\mathbb{E} \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 \leq \left[1 - \frac{2c(1-2c)}{K(\mathbf{A})} \right]^k \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2 + \frac{cK(\mathbf{A})}{1-2c} \cdot \frac{2\sigma^2}{n\|\mathbf{A}\|_F^2} \tag{5.5}$$

The partially biased randomized Kaczmarz method described above [which does have modified update Eq. (5.4) compared to the standard update Eq. (5.2)] yields the same convergence rate as the fully biased randomized Kaczmarz method [46] (up to a factor of 2), but gives a better dependence on the residual error over the fully biased sampling, as the final term in (5.5) is smaller than the final term in (5.3).

6 Numerical experiments

In this section we present some numerical results for the randomized Kaczmarz algorithm with partially biased sampling, that is, applying Algorithm 3.1 to the least squares problem $F(\mathbf{x}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ (so $f_i(\mathbf{x}) = \frac{n}{2} \langle \mathbf{a}_i, \mathbf{x} \rangle - b_i$) and considering $\lambda \in [0, 1]$. Recall that $\lambda = 0$ corresponds to the randomized Kaczmarz algorithm of Strohmer and Vershynin with fully weighted sampling [46]. $\lambda = 0.5$ corresponds to the partially biased randomized Kaczmarz algorithm outlined in Corollary 5.3. We demonstrate how the behavior of the algorithm depends on λ , the conditioning of the system, and the residual error at the least squares solution. We focus on exploring the role of λ on the convergence rate of the algorithm for various types of matrices \mathbf{A} . We consider five types of systems, described below, each using a $1,000 \times 10$ matrix \mathbf{A} . In each setting, we create a vector \mathbf{x} with standard normal entries. For the described matrix \mathbf{A} and residual \mathbf{e} , we create the system $\mathbf{b} = \mathbf{A}\mathbf{x} + \mathbf{e}$ and run the randomized Kaczmarz method with various choices of λ . Each experiment consists of 100 independent trials and uses the optimal step size as in Corollary 3.2 with $\varepsilon = 0.1$; the plots

show the average behavior over these trials. The settings below show the various types of behavior the Kaczmarz method can exhibit.

- Case 1** Each row of the matrix A has standard normal entries, except the last row which has normal entries with mean 0 and variance 10^2 . The residual vector e has normal entries with mean 0 and variance 0.1^2 .
- Case 2** Each row of the matrix A has standard normal entries. The residual vector e has normal entries with mean 0 and variance 0.1^2 .
- Case 3** The j th row of A has normal entries with mean 0 and variance j . The residual vector e has normal entries with mean 0 and variance 20^2 .
- Case 4** The j th row of A has normal entries with mean 0 and variance j . The residual vector e has normal entries with mean 0 and variance 10^2 .
- Case 5** The j th row of A has normal entries with mean 0 and variance j . The residual vector e has normal entries with mean 0 and variance 0.1^2 .

Figure 1 shows the convergence behavior of the randomized Kaczmarz method in each of these five settings. As expected, when the rows of A are far from normalized, as in Case 1, we see different behavior as λ varies from 0 to 1. Here, weighted sampling ($\lambda = 0$) significantly outperforms uniform sampling ($\lambda = 1$), and the trend is monotonic in λ . On the other hand, when the rows of A are close to normalized, as in Case 2, the various λ give rise to similar convergence rates, as is expected. Out of the λ tested (we tested increments of 0.1 from 0 to 1), the choice $\lambda = 0.7$ gave the worst convergence rate, and again purely weighted sampling gives the best. Still, the worst-case convergence rate was not much worse, as opposed to the situation with uniform sampling in Case 1. Cases 3, 4, and 5 use matrices with varying row norms and cover “high”, “medium”, and “low” noise regimes, respectively. In the high noise regime (Case 3), we find that fully weighted sampling, $\lambda = 0$, is relatively very slow to converge, as the theory suggests, and hybrid sampling outperforms both weighted and uniform selection. In the medium noise regime (Case 4), hybrid sampling still outperforms both weighted and uniform selection. Again, this is not surprising, since hybrid sampling allows a balance between small convergence horizon (important with large residual norm) and convergence rate. As we decrease the noise level (as in Case 5), we see that again weighted sampling is preferred.

Figure 2 shows the number of iterations of the randomized Kaczmarz method needed to obtain a fixed approximation error. For the choice $\lambda = 1$ for Case 1, we cut off the number of iterations after 50,000, at which point the desired approximation error was still not attained. As seen also from Fig. 1, Case 1 exhibits monotonic improvements as we scale λ . For Cases 2 and 5, the optimal choice is pure weighted sampling, whereas Cases 3 and 4 prefer intermediate values of λ .

7 Summary and outlook

We consider this paper as making three contributions: the improved dependence on the conditioning for smooth and strongly convex SGD, the discussion of importance sampling for SGD, and the connection between SGD and the randomized Kaczmarz method.

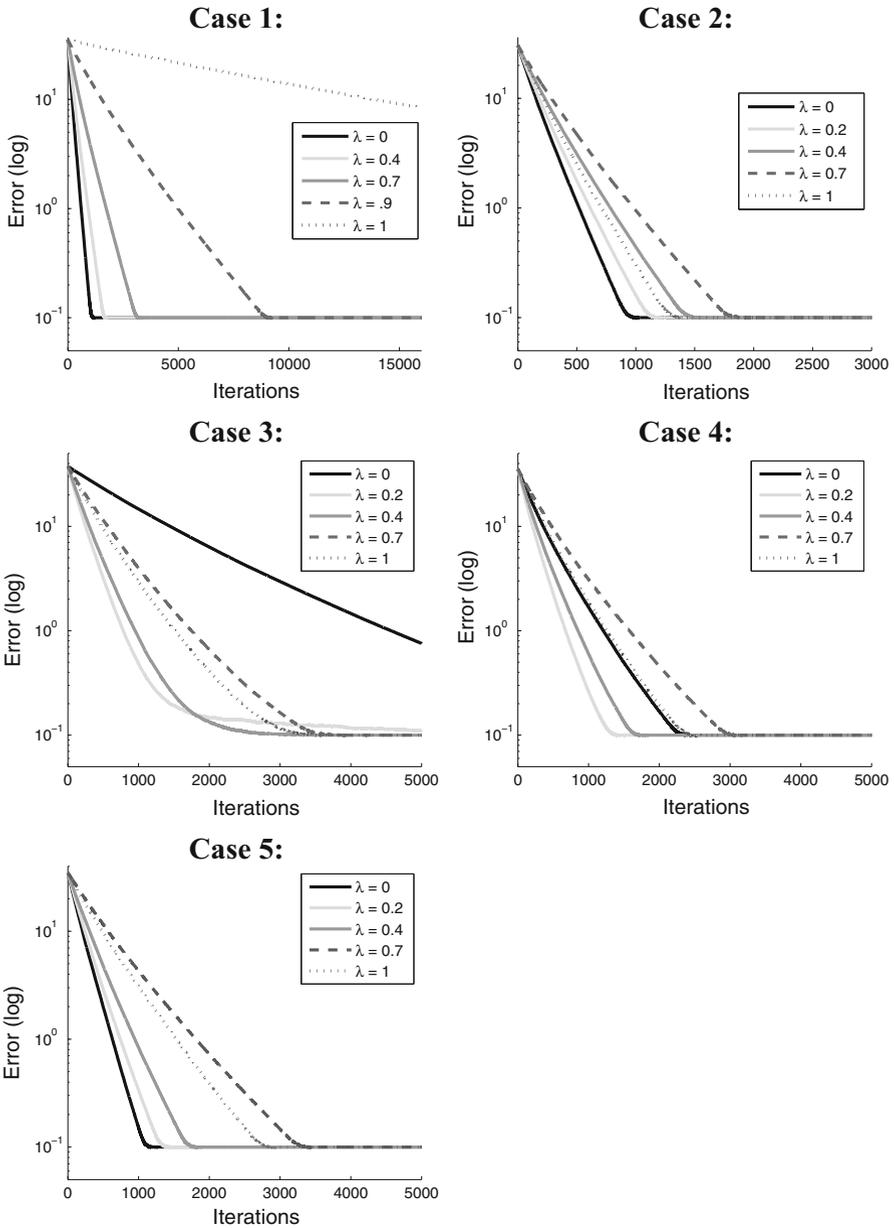
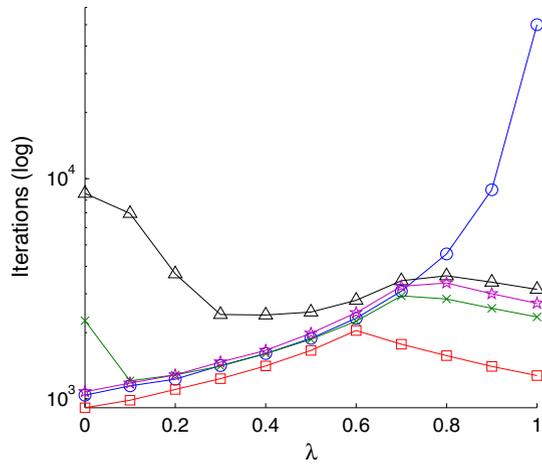


Fig. 1 The convergence rates for the randomized Kaczmarz method with various choices of λ in the five settings described above. The vertical axis is in logarithmic scale and depicts the approximation error $\|x_k - x_\star\|_2^2$ at iteration k (the horizontal axis)

For simplicity, we only considered SGD iterates with a fixed step-size γ . This is enough for getting the optimal iteration complexity if the target accuracy ε is known in advance, which was our approach in this paper. It is easy to adapt the analysis, using

Fig. 2 Number of iterations k needed by the randomized Kaczmarz method with partially biased sampling, for various values of λ , to obtain approximation error $\|x_k - x_\star\|_2^2 \leq \varepsilon = 0.1$ in the five cases described above: Case 1 (blue with circle marker), Case 2 (red with square marker), Case 3 (black with triangle marker), Case 4 (green with x marker), and Case 5 (purple with star marker)



standard techniques, to incorporate decaying step-sizes, which are appropriate if we don't know ε in advance.

We suspect that the assumption of strong convexity can be weakened to *restricted strong convexity* [21, 50] without changing any of the results of this paper; we leave this analysis to future work.

Finally, our discussion of importance sampling is limited to a static reweighting of the sampling distribution. A more sophisticated approach would be to update the sampling distribution dynamically as the method progresses, and as we gain more information about the relative importance of components. Although such dynamic importance sampling is sometimes attempted heuristically, we are not aware of any rigorous analysis of such a dynamic biasing scheme.

Acknowledgments We would like to thank the anonymous reviewers for their useful feedback which significantly improved the manuscript. We would like to thank Chris White for pointing out a simplified proof of Corollary 2.2. DN was partially supported by a Simons Foundation Collaboration grant, NSF CAREER #1348721 and an Alfred P. Sloan Fellowship. NS was partially supported by a Google Research Award. RW was supported in part by ONR Grant N00014-12-1-0743, an AFOSR Young Investigator Program Award, and an NSF CAREER award.

8 Appendix: Proofs

Our main results utilize an elementary fact about smooth functions with Lipschitz continuous gradient, called the co-coercivity of the gradient. We state the lemma and recall its proof for completeness.

8.1 The co-coercivity Lemma

Lemma 8.1 (Co-coercivity) *For a smooth function f whose gradient has Lipschitz constant L ,*

$$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2^2 \leq L \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{x}) - \nabla f(\mathbf{y}) \rangle.$$

Proof Since ∇f has Lipschitz constant L , if \mathbf{x}_\star is the minimizer of f , then (see e.g. [32], page 26)

$$\begin{aligned} \frac{1}{2L} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}_\star)\|_2^2 &= \frac{1}{2L} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{x}_\star)\|_2^2 + \langle \mathbf{x} - \mathbf{x}_\star, \nabla f(\mathbf{x}_\star) \rangle \\ &\leq f(\mathbf{x}) - f(\mathbf{x}_\star); \end{aligned} \tag{8.1}$$

Now define the convex functions

$$G(\mathbf{z}) = f(\mathbf{z}) - \langle \nabla f(\mathbf{x}), \mathbf{z} \rangle, \quad \text{and} \quad H(\mathbf{z}) = f(\mathbf{z}) - \langle \nabla f(\mathbf{y}), \mathbf{z} \rangle,$$

and observe that both have Lipschitz constants L and minimizers \mathbf{x} and \mathbf{y} , respectively. Applying (8.1) to these functions therefore gives that

$$G(\mathbf{x}) \leq G(\mathbf{y}) - \frac{1}{2L} \|\nabla G(\mathbf{y})\|_2^2, \quad \text{and} \quad H(\mathbf{y}) \leq H(\mathbf{x}) - \frac{1}{2L} \|\nabla H(\mathbf{y})\|_2^2.$$

By their definitions, this implies that

$$\begin{aligned} f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{x} \rangle &\leq f(\mathbf{y}) - \langle \nabla f(\mathbf{x}), \mathbf{y} \rangle - \frac{1}{2L} \|\nabla f(\mathbf{y}) - \nabla f(\mathbf{x})\|_2^2 \\ f(\mathbf{y}) - \langle \nabla f(\mathbf{y}), \mathbf{y} \rangle &\leq f(\mathbf{x}) - \langle \nabla f(\mathbf{y}), \mathbf{x} \rangle - \frac{1}{2L} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2^2. \end{aligned}$$

Adding these two inequalities and canceling terms yields the desired result. □

8.2 Proof of Theorem 2.1

With the notation of Theorem 2.1, and where i is the random index chosen at iteration k , and $w = w_\lambda$, we have

$$\begin{aligned} \|\mathbf{x}_{k+1} - \mathbf{x}_\star\|_2^2 &= \|\mathbf{x}_k - \mathbf{x}_\star - \gamma \nabla f_i(\mathbf{x}_k)\|_2^2 \\ &= \|(\mathbf{x}_k - \mathbf{x}_\star) - \gamma(\nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_\star)) - \gamma \nabla f_i(\mathbf{x}_\star)\|_2^2 \\ &= \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 - 2\gamma \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla f_i(\mathbf{x}_k) \rangle \\ &\quad + \gamma^2 \|\nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_\star) + \nabla f_i(\mathbf{x}_\star)\|_2^2 \\ &\leq \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 - 2\gamma \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla f_i(\mathbf{x}_k) \rangle \\ &\quad + 2\gamma^2 \|\nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_\star)\|_2^2 + 2\gamma^2 \|\nabla f_i(\mathbf{x}_\star)\|_2^2 \\ &\leq \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 - 2\gamma \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla f_i(\mathbf{x}_k) \rangle \\ &\quad + 2\gamma^2 L_i \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_\star) \rangle + 2\gamma^2 \|\nabla f_i(\mathbf{x}_\star)\|_2^2, \end{aligned}$$

where we have employed Jensen’s inequality in the first inequality and the co-coercivity Lemma 8.1 in the final line. We next take an expectation with respect to the

choice of i . By assumption, $i \sim \mathcal{D}$ such that $F(\mathbf{x}) = \mathbb{E}f_i(\mathbf{x})$ and $\sigma^2 = \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|^2$. Then $\mathbb{E}\nabla f_i(\mathbf{x}) = \nabla F(\mathbf{x})$, and we obtain:

$$\begin{aligned} \mathbb{E}\|\mathbf{x}_{k+1} - \mathbf{x}_\star\|_2^2 &\leq \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 - 2\gamma \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla F(\mathbf{x}_k) \rangle \\ &\quad + 2\gamma^2 \mathbb{E}[L_i \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_\star) \rangle] + 2\gamma^2 \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|_2^2 \\ &\leq \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 - 2\gamma \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla F(\mathbf{x}_k) \rangle \\ &\quad + 2\gamma^2 \sup_i L_i \mathbb{E} \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla f_i(\mathbf{x}_k) - \nabla f_i(\mathbf{x}_\star) \rangle + 2\gamma^2 \mathbb{E}\|\nabla f_i(\mathbf{x}_\star)\|_2^2 \\ &= \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 - 2\gamma \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla F(\mathbf{x}_k) \rangle \\ &\quad + 2\gamma^2 \sup L \langle \mathbf{x}_k - \mathbf{x}_\star, \nabla F(\mathbf{x}_k) - \nabla F(\mathbf{x}_\star) \rangle + 2\gamma^2 \sigma^2 \end{aligned}$$

We now utilize the strong convexity of $F(\mathbf{x})$ and obtain that

$$\begin{aligned} &\leq \|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 - 2\gamma\mu(1 - \gamma \sup L)\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 + 2\gamma^2\sigma^2 \\ &= (1 - 2\gamma\mu(1 - \gamma \sup L))\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 + 2\gamma^2\sigma^2 \end{aligned}$$

when $\gamma \leq \frac{1}{\sup L}$. Recursively applying this bound over the first k iterations yields the desired result,

$$\begin{aligned} \mathbb{E}\|\mathbf{x}_k - \mathbf{x}_\star\|_2^2 &\leq \left(1 - 2\gamma\mu(1 - \gamma \sup L)\right)^k \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2 \\ &\quad + 2 \sum_{j=0}^{k-1} \left(1 - 2\gamma\mu(1 - \gamma \sup L)\right)^j \gamma^2 \sigma^2 \\ &\leq \left(1 - 2\gamma\mu(1 - \gamma \sup L)\right)^k \|\mathbf{x}_0 - \mathbf{x}_\star\|_2^2 + \frac{\gamma\sigma^2}{\mu(1 - \gamma \sup L)}. \end{aligned}$$

References

1. Bach, F., Moulines, E.: Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In: *Advances in Neural Information Processing Systems (NIPS)* (2011)
2. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer (2010)
3. Bottou, L., Bousquet, O.: The tradeoffs of large-scale learning. In: *Optimization for Machine Learning*, p. 351 (2011)
4. Bousquet, O., Bottou, L.: The tradeoffs of large scale learning. In: *Advances in Neural Information Processing Systems*, pp. 161–168 (2007)
5. Boutsidis, C., Mahoney, M., Drineas, P.: An improved approximation algorithm for the column subset selection problem. In: *Proceedings of the Symposium on Discrete Algorithms*, pp. 968–977 (2009)
6. Byrne, C.L.: *Applied Iterative Methods*. A K Peters Ltd., Wellesley, MA. ISBN:978-1-56881-342-4; 1-56881-271-X (2008)
7. Cenkler, C., Feichtinger, H.G., Mayer, M., Steier, H., Strohmer, T.: New variants of the POCS method using affine subspaces of finite codimension, with applications to irregular sampling. In: *Proceedings of SPIE: Visual Communications and Image Processing*, pp. 299–310 (1992)
8. Censor, Y., Eggermont, P.P.B., Gordon, D.: Strong underrelaxation in Kaczmarz’s method for inconsistent systems. *Numer. Math.* **41**(1), 83–92 (1983)
9. Chen, Y., Bhojanapalli, S., Sanghavi, S., Ward, R.: Coherent matrix completion. In: *Proceedings of the 31st International Conference on Machine Learning*, pp. 674–682 (2014)

10. Eggermont, P.P.B., Herman, G.T., Lent, A.: Iterative algorithms for large partitioned linear systems, with applications to image reconstruction. *Linear Algebra Appl.* **40**, 37–67 (1981). doi:[10.1016/0024-3795\(81\)90139-7](https://doi.org/10.1016/0024-3795(81)90139-7). ISSN:0024–3795
11. Eldar, Y.C., Needell, D.: Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma. *Numer. Algorithms* **58**(2), 163–177 (2011). doi:[10.1007/s11075-011-9451-z](https://doi.org/10.1007/s11075-011-9451-z). ISSN: 1017–1398
12. Elfving, T.: Block-iterative methods for consistent and inconsistent linear equations. *Numer. Math.* **35**(1), 1–12 (1980). doi:[10.1007/BF01396365](https://doi.org/10.1007/BF01396365). ISSN:0029–599X
13. Foygel, R., Srebro, N.: Concentration-based guarantees for low-rank matrix reconstruction. In: 24th Annual Conference on Learning Theory (COLT) (2011)
14. Gittens, A., Mahoney, M.: Revisiting the nystrom method for improved large-scale machine learning. In: International Conference on Machine Learning (ICML) (2013)
15. Hanke, M., Niethammer, W.: On the acceleration of Kaczmarz’s method for inconsistent linear systems. *Linear Algebra Appl.* **130**, 83–98 (1990)
16. Herman, G.T.: *Fundamentals of computerized tomography: image reconstruction from projections*. Springer, Berlin (2009)
17. Herman, G.T., Meyer, L.B.: Algebraic reconstruction techniques can be made computationally efficient. *IEEE Trans. Med. Imaging* **12**(3), 600–609 (1993)
18. Hounsfield, G.N.: Computerized transverse axial scanning (tomography): Part 1. Description of system. *Br. J. Radiol.* **46**(552), 1016–1022 (1973)
19. Kaczmarz, S.: Angenäherte auflösung von systemen linearer gleichungen. *Bull. Int. Acad. Polon. Sci. Lett. Ser. A.* **35**, 335–357 (1937)
20. Krahmer, F., Ward, R.: Stable and robust sampling strategies for compressive imaging. *IEEE Trans. Image Proc.* **23**(2), 612–622 (2014)
21. Lai, M., Yin, W.: Augmented ℓ_1 and nuclear-norm models with a globally linearly convergent algorithm. *SIAM J. Imaging Sci.* **6**(2), 1059–1091 (2013)
22. Lee, Y.T., Sidford, A.: Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. *IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 147–156 (2013)
23. Liu, J., Wright, S.J., Sridhar, S.: An asynchronous parallel randomized kaczmarz algorithm. [arXiv:1401.4780](https://arxiv.org/abs/1401.4780) (2014)
24. Ma, P., Yu, B., Mahoney, M.: A statistical perspective on algorithmic leveraging. In: International Conference on Machine Learning (ICML). [arXiv:1306.5362](https://arxiv.org/abs/1306.5362) (2014)
25. Mahoney, M.: Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.* **3**(2), 123–224 (2011)
26. Murata, N.: *A Statistical Study of On-Line Learning*. Cambridge University Press, Cambridge (1998)
27. Natterer, F.: *The Mathematics of Computerized Tomography*, volume 32 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (2001). ISBN:0-89871-493-1. doi:[10.1137/1.9780898719284](https://doi.org/10.1137/1.9780898719284). Reprint of the 1986 original
28. Needell, D.: Randomized Kaczmarz solver for noisy linear systems. *BIT* **50**(2), 395–403 (2010). doi:[10.1007/s10543-010-0265-5](https://doi.org/10.1007/s10543-010-0265-5). ISSN:0006–3835
29. Needell, D., Tropp, J.A.: Paved with good intentions: analysis of a randomized block kaczmarz method. *Linear Algebra Appl.* **441**, 199–221 (2014)
30. Needell, D., Ward, R.: Two-subspace projection method for coherent overdetermined linear systems. *J. Fourier Anal. Appl.* **19**(2), 256–269 (2013)
31. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.* **19**(4), 1574–1609 (2009)
32. Nesterov, Y.: *Introductory Lectures on Convex Optimization*. Kluwer, Boston (2004)
33. Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.* **22**(2), 341–362 (2012)
34. Popa, C.: Extensions of block-projections methods with relaxation parameters to inconsistent and rank-deficient least-squares problems. *BIT* **38**(1), 151–176 (1998). doi:[10.1007/BF02510922](https://doi.org/10.1007/BF02510922). ISSN:0006–3835
35. Popa, C.: Block-projections algorithms with blocks containing mutually orthogonal rows and columns. *BIT* **39**(2), 323–338 (1999). doi:[10.1023/A:1022398014630](https://doi.org/10.1023/A:1022398014630). ISSN:0006–3835
36. Popa, C.: A fast Kaczmarz–Kovarik algorithm for consistent least-squares problems. *Korean J. Comput. Appl. Math.* **8**(1), 9–26 (2001). ISSN:1229–9502

37. Popa, C.: A Kaczmarz–Kovarik algorithm for symmetric ill-conditioned matrices. *An. Științ. Univ. Ovidius Constanța Ser. Mat.* **12**(2), 135–146 (2004). ISSN:1223–723X
38. Popa, C., Preclik, T., Köstler, H., Rüdte, U.: On Kaczmarz’s projection iteration as a direct solver for linear least squares problems. *Linear Algebra Appl.* **436**(2), 389–404 (2012)
39. Rauhut, H., Ward, R.: Sparse Legendre expansions via ℓ_1 -minimization. *J. Approx. Theory* **164**(5), 517–533 (2012)
40. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Math. Program.* **144**(1–2), 1–38 (2012)
41. Robbins, H., Monro, S.: A stochastic approximation method. *Ann. Math. Stat.* **22**, 400–407 (1951)
42. Shalev-Shwartz, S., Srebro, N.: Svm optimization: inverse dependence on training set size. In: *Proceedings of the 25th International Conference on Machine Learning*, pp. 928–935 (2008)
43. Shamir, O., Zhang, T.: Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes. [arXiv:1212.1824](https://arxiv.org/abs/1212.1824) (2012)
44. Spielman, D., Srivastava, N.: Graph sparsification by effective resistances. *SIAM J. Comput.* **40**(6), 1913–1926 (2011)
45. Srebro, N., Sridharan, K., Tewari, A.: Smoothness, low noise and fast rates. In: *Advances in Neural Information Processing Systems*, pp. 2199–2207 (2010)
46. Strohmer, T., Vershynin, R.: A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.* **15**(2), 262–278 (2009). doi:[10.1007/s00041-008-9030-4](https://doi.org/10.1007/s00041-008-9030-4). ISSN:1069–5869
47. Tanabe, K.: Projection method for solving a singular system of linear equations and its applications. *Numer. Math.* **17**(3), 203–214 (1971)
48. Wang, S., Zhang, Z.: Improving cur matrix decomposition and the nystrom approximation via adaptive sampling. *J. Mach. Learn. Res.* **14**, 2729–2769 (2013)
49. Whitney, T.M., Meany, R.K.: Two algorithms related to the method of steepest descent. *SIAM J. Numer. Anal.* **4**(1), 109–118 (1967)
50. Zhang, H., Yin, W.: Gradient methods for convex minimization: better rates under weaker conditions. *CAM Report 13–17*, UCLA (2013)
51. Zhao, P., Zhang, T.: Stochastic optimization with importance sampling. [arXiv:1401.2753](https://arxiv.org/abs/1401.2753) (2014)
52. Zouzias, A., Freris, N.M.: Randomized extended Kaczmarz for solving least-squares. *SIAM J. Matrix Anal. Appl.* **34**(2), 773–793 (2013)