FULL LENGTH PAPER

Series A

Stochastic quasi-gradient methods: variance reduction via Jacobian sketching

Robert M. Gower[1] · Peter Richtárik[2,3,4] · Francis Bach[5]

Abstract

We develop a new family of variance reduced stochastic gradient descent methods for minimizing the average of a very large number of smooth functions. Our method—JacSketch—is motivated by novel developments in randomized numerical linear algebra, and operates by maintaining a stochastic estimate of a Jacobian matrix composed of the gradients of individual functions. In each iteration, JacSketch efficiently updates the Jacobian matrix by first obtaining a random linear measurement of the true Jacobian through (cheap) sketching, and then projecting the previous estimate onto the solution space of a linear matrix equation whose solutions are consistent with the measurement. The Jacobian estimate is then used to compute a variance-reduced unbiased estimator of the gradient. Our strategy is analogous to the way quasi-Newton methods maintain an estimate of the Hessian, and hence our method can be seen as a *stochastic quasi-gradient method*. Our method can also be seen as stochastic gradient descent applied to a *controlled stochastic optimization reformulation* of the original problem, where the control comes from the Jacobian estimates. We prove that for smooth and strongly convex functions, JacSketch converges linearly with a meaningful rate dictated by a single convergence theorem which applies to general sketches. We also provide a refined convergence theorem which applies to a smaller class of sketches, featuring a novel proof technique based on a *stochastic Lyapunov function*. This enables us to obtain sharper complexity results for variants of JacSketch with importance sampling. By specializing our general approach to specific sketching strategies, JacSketch reduces to the celebrated stochastic average gradient (SAGA) method, and its several existing and many new minibatch, reduced memory,

---

Extended author information available on the last page of the article

and importance sampling variants. Our rate for SAGA with importance sampling is the current best-known rate for this method, resolving a conjecture by Schmidt et al. (Proceedings of the eighteenth international conference on artificial intelligence and statistics, AISTATS 2015, San Diego, California, 2015). The rates we obtain for mini-batch SAGA are also superior to existing rates and are sufficiently tight as to show a decrease in total complexity as the minibatch size increases. Moreover, we obtain the first minibatch SAGA method with importance sampling.

Keywords Stochastic gradient descent · Sketching · Variance reduction · Covariates

Mathematics Subject Classification 65Kxx · 90C15 · 90C25

1 Introduction

We consider the problem of minimizing the average of a large number of differentiable functions

$$x^* = \arg\min_{x \in \mathbb{R}^d} \left[f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right], \tag{1}$$

where f is μ—strongly convex and L—smooth. In solving (1), we restrict our attention to first-order methods that use a (variance-reduced) stochastic estimate of the gradient $g^k \approx \nabla f(x^k)$ to take a step towards minimizing (1) by iterating

$$x^{k+1} = x^k - \alpha g^k, \tag{2}$$

where $\alpha > 0$ is a stepsize.

In the context of machine learning, (1) is an abstraction of the *empirical risk minimization* problem; x encodes the parameters/features of a (statistical) model, and f_i is the loss of example/data point i incurred by model x. The goal is to find the model x which minimizes the average loss on the n observations.

Typically, n is so large that algorithms which rely on scanning through all n functions in each iteration are too costly. The need for incremental methods for the training phase of machine learning models has revived the interest in the stochastic gradient descent (SGD) method [27]. SGD sets $g^k = \nabla f_i(x^k)$, where i is an index chosen from $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$ uniformly at random. SGD therefore requires only a single data sample to complete a step and make progress towards the solution. Thus SGD scales well in the number of data samples, which is important in several machine learning applications since there many be a large number of data samples. On the downside, the variance of the stochastic estimates of the gradient produced by SGD does not vanish during the iterative process, which suggests that a decreasing stepsize regime needs to be put into place if SGD is to converge. Furthermore, for SGD to work efficiently, this decreasing stepsize regime needs to be tuned for each application area, which is costly.

1.1 Variance-reduced methods

Stochastic variance-reduced versions of SGD offer a solution to this high variance issue, which improves the theoretical convergence rate and solves the issue with ad hoc stepsize regimes. The first variance reduced method for empirical risk minimization is the stochastic average gradient (SAG) method of Schmidt, Le Roux and Bach [29], closely followed by Finito [7] and Miso [18]. The analysis of SAG is notoriously difficult, which is perhaps due to the estimator of gradient being biased. Soon afterwards, the SAG gradient estimator was modified into an unbiased one, which resulted in the SAGA method [6]. The analysis of SAGA is dramatically simpler than that of SAG. Another popular method is SVRG of Johnson and Zhang [15] (see also S2GD [16]). SVRG enjoys the same theoretical complexity bound as SAGA, but has a much smaller memory footprint. It is based on an inner–outer loop procedure. In the outer loop, a full pass over data is performed to compute the gradient of f at the current point. In the inner loop, this gradient is modified with the use of cheap stochastic gradients, and steps are taken in the direction of the modified gradients. A notable recent addition to the family of variance reduced methods, developed by Nguyen et al. [20], is known as SARAH. Unlike other methods, SARAH does not use an estimator that is unbiased in the last step. Instead, it is unbiased over a long history of the method.

A fundamentally different way of designing variance reduced methods is to use coordinate descent [24,25] to solve the dual. This is what the SDCA method [33] and its various extensions [32] do. The key advantage of this approach is that the dual often has a seperable structure in the coordinate space, which in turn means that each iteration of coordinate descent is cheap. Furthermore, SDCA is a variance-reduced method by design since the coordinates of the gradient tend to zero as one approaches the solution. One of the downsides of SDCA is that it requires calculating Fenchel duals and their derivatives. This issue was later solved by introducing approximations and mapping the dual iterates to the primal space as pointed out in [6]. This resulted in primal variants of SDCA such as dual-free SDCA [31]. A primal-dual variant which enables the use of arbitrary minibatch strategies was developed by Qu et al. [23], and is known as QUARTZ.

Finally, variance reduced methods can also be accelerated, as has been shown for the loop based methods such as Katyusha [3] or using the Universal catalyst [17].

1.2 Gaps in our understanding of SAGA

Despite significant research into variance-reduced stochastic gradient descent methods for solving (1), there are still big gaps in our understanding of variance reduction. For instance, the current theory supporting the SAGA algorithm is far from complete.

SAGA with uniform probabilities enjoys the iteration complexity $\mathcal{O}((n + \frac{L_{\max}}{\mu}) \log \frac{1}{\epsilon})$, where $L_{\max} \stackrel{\text{def}}{=} \max_i L_i$ and L_i is the smoothness constant of f_i. While importance sampling versions of SAGA have proved in practice to produce a speed-up over uniform SAGA [30], a proof of this speed-up has been elusive. It was conjectured by Schmidt et al. [30] that a properly designed importance sampling strategy for SAGA should lead to the rate $\mathcal{O}\left(\left(n + \frac{\bar{L}}{\mu}\right) \log \frac{1}{\epsilon}\right)$, where $\bar{L} = \frac{1}{n} \sum_i L_i$. However, no such

result was proved. This rate is achieved by, for instance, importance sampling variants of SDCA, QUARTZ [23] and SVRG [36]. However, the analysis only applies to a more specialized version of problem (1) (e.g., one needs an explicit strongly convex regularizer).

Second, existing minibatch variants of SAGA do not enjoy the same rate as that offered by methods such as SDCA and QUARTZ. Are the above issues with SAGA unavoidable, or is it the case that our understanding of the method is far from complete? Lastly, no minibatch variant of SAGA with importance sampling is known.

One of the contributions of this paper is giving positive answers to all of the above questions.

1.3 Jacobian sketching: a new approach to variance reduction

Our key contribution in this paper is the introduction of a novel approach—which we call *Jacobian sketching*—to designing and understanding variance-reduced stochastic gradient descent methods for solving (1). We refer to our method by the name *JacSketch*. We shall now briefly introduce some of the key insights motivating our approach. Let $F : \mathbb{R}^d \to \mathbb{R}^n$ be defined by

$$F(x) \stackrel{\text{def}}{=} (f_1(x), \ldots, f_n(x)) \in \mathbb{R}^n, \tag{3}$$

and further let

$$\nabla \mathbf{F}(x) \stackrel{\text{def}}{=} [\nabla f_1(x), \ldots, \nabla f_n(x)] \in \mathbb{R}^{d \times n}, \tag{4}$$

be the Jacobian of F at x.

The starting point of our new approach is the following trivial observation: the gradient of f at x can be computed from the Jacobian $\nabla \mathbf{F}(x)$ by a simple *linear transformation:*

$$\frac{1}{n} \nabla \mathbf{F}(x) e = \nabla f(x), \tag{5}$$

where e is the vector of all ones in $\mathbb{R}^n$. This alone is not useful to come up with a better way of estimating the gradient. Indeed, formula (5) has two issues. First, the Jacobian is *not* available. If we wanted to compute it, we would need to pay the cost of one pass through the data. Second, even if the Jacobian was available, merely multiplying it by the vector of all ones would cost $\mathcal{O}(nd)$ operations, which is again a cost equivalent to one pass over data.

Now, let us replace the vector of all ones in (5) by $e_i \in \mathbb{R}^n$, the unit coordinate/basis vector in $\mathbb{R}^n$. If the index i is chosen randomly from $[n]$, then

$$\nabla \mathbf{F}(x) e_i = \nabla f_i(x), \tag{6}$$

which is a stochastic gradient of f at x. In other words, by performing a *random linear transformation* of the Jacobian, we have arrived at the classical stochastic estimate of

the gradient. This approach does not suffer from the first issue mentioned above as the Jacobian is *not needed* at all in order to compute $\nabla f_i(x)$. Likewise, it does not suffer from the second issue; namely, the cost of computing the stochastic gradient is merely $\mathcal{O}(d)$, and we can avoid a costly pass through the data.[1]

However, this approach suffers from a new issue: by constructing the estimate this way, we *do not learn* from the (random) information collected about the Jacobian in prior iterations, through having access to random linear transformations thereof. In this paper we take the point of view that this is the reason why SGD suffers from large variance. Our approach towards alleviating this problem is to maintain and update an estimate $\mathbf{J} \in \mathbb{R}^{d \times n}$ of the Jacobian $\nabla \mathbf{F}(x)$.

Given $x^k \in \mathbb{R}^d$, ideally we would like $\mathbf{J}$ to satisfy

$$\mathbf{J} = \nabla \mathbf{F}(x^k), \tag{7}$$

that is, we would like it to be equal to the true Jacobian. However, at the same time we do not wish to pay the price of computing it. Hence, assuming we have an estimate $\mathbf{J}^k \in \mathbb{R}^{d \times n}$ of the Jacobian available, we instead pick a random matrix $\mathbf{S}_k \in \mathbb{R}^{n \times \tau}$ from some distribution $\mathcal{D}$ of matrices[2] and consider the following *sketched* version of the linear system (7), with unknown $\mathbf{J}$:

$$\mathbf{J}\mathbf{S}_k = \nabla \mathbf{F}(x^k)\mathbf{S}_k \in \mathbb{R}^{d \times \tau}. \tag{8}$$

This equation generalizes both (5) and (6). The left hand side contains the sketched system matrix $\mathbf{S}_k$ and the unknown matrix $\mathbf{J}$, and the right hand side contains a quantity we can measure (through a random linear measurement of the Jacobian, which we assume is cheap). Of course, the true Jacobian solves (8). However, in general, and in particular when $\tau \ll n$ which is the regime we want to be in for practical reasons, the system (8) will have infinite $\mathbf{J}$ solutions.

We pick a unique solution $\mathbf{J}^{k+1}$ as the closest solution of (8) to our previous estimate $\mathbf{J}^k$, with respect to a weighted Frobenius norm with a positive definite weight matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$:

$$\mathbf{J}^{k+1} = \arg \min_{\mathbf{J} \in \mathbb{R}^{d \times n}} \|\mathbf{J} - \mathbf{J}^k\|_{\mathbf{W}^{-1}}$$
$$\text{subject to} \quad \mathbf{J}\mathbf{S}_k = \nabla \mathbf{F}(x^k)\mathbf{S}_k, \tag{9}$$

where

$$\|\mathbf{X}\|_{\mathbf{W}^{-1}} \overset{\text{def}}{=} \sqrt{\text{Tr}\left(\mathbf{X}\mathbf{W}^{-1}\mathbf{X}^{\top}\right)}. \tag{10}$$

---

[1] For the purposes of this narrative it suffices to assume that stochastic gradients can be sampled at cost $\mathcal{O}(d)$.

[2] We will not bother about the distribution from which it is picked at the moment. It suffices to say that virtually all distributions are supported by our theory. However, if we wish to obtain a practical method, some distributions will make much more sense than others.

In doing so, we have built a learning mechanism whose goal is to maintain good estimates of the Jacobian throughout the run of method (2). These estimates can be used to efficiently estimate the gradient by performing a linear transformation similar to (5), but with $\nabla \mathbf{F}(x)$ replaced by the latest estimate of the Jacobian. In practice, it is important to design sketching matrices so that the Jacobian sketch $\nabla \mathbf{F}(x)\mathbf{S}_k$ can be calculated efficiently.

The "sketch-and-project" strategy (9) for updating our Jacobian estimate is analogous to the way quasi-Newton methods update the estimate of the Hessian (or inverse Hessian) [8,9,12]. From this perspective, our method can be viewed as a *stochastic quasi-gradient method.*[3]

Problem (9) admits the explicit closed-form solution (see Lemma 14):

$$\mathbf{J}^{k+1} = \mathbf{J}^k + (\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k}, \tag{11}$$

where

$$\mathbf{\Pi}_{\mathbf{S}} \overset{\text{def}}{=} \mathbf{S}(\mathbf{S}^\top \mathbf{W}\mathbf{S})^\dagger \mathbf{S}^\top \mathbf{W}, \tag{12}$$

is a projection matrix, and † denotes the Moore–Penrose pseudoinverse.

> The key insight of our work is to propose an efficient Jacobian learning mechanism based on ideas borrowed from recent results in randomized numerical linear algebra.

Having established our update of the Jacobian estimate, we now need to use this to form an estimate of the gradient. Unfortunately, using $\mathbf{J}^{k+1}$ in place of $\nabla \mathbf{F}(x^k)$ in (5) leads to a biased gradient estimate (something we explore later in Sect. 2.5). To obtain an unbiased estimator of the gradient, we introduce a *stochastic relaxation parameter* $\theta_{\mathbf{S}_k} \geq 0$ and use

$$g^k \overset{\text{def}}{=} \frac{1 - \theta_{\mathbf{S}_k}}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n}\mathbf{J}^{k+1} e = \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}\left(\nabla \mathbf{F}(x^k) - \mathbf{J}^k\right)\theta_{\mathbf{S}_k}\mathbf{\Pi}_{\mathbf{S}_k}e, \tag{13}$$

as an approximation of the gradient. Taking expectations in (13) over $\mathbf{S}^k \sim \mathcal{D}$ (for this we use the notation $\mathbb{E}_{\mathcal{D}}[\cdot] \equiv \mathbb{E}_{\mathbf{S}_k \sim \mathcal{D}}[\cdot]$), we get

$$\mathbb{E}_{\mathcal{D}}\left[g^k\right] = \frac{1}{n}\mathbf{J}^k e + \frac{1}{n}(\nabla \mathbf{F}(x^k) - \mathbf{J}^k)\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}_k}\mathbf{\Pi}_{\mathbf{S}_k}e\right]. \tag{14}$$

Thus provided that

$$\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}_k}\mathbf{\Pi}_{\mathbf{S}_k}e\right] = e, \tag{15}$$

---

[3] The term "quasi-gradient methods" was popular in the 1980s [21], and refers to algorithms for solving certain stochastic optimization problems which rely on stochastic estimates of function values and their derivatives. In this paper we give the term a different meaning by drawing a direct link with quasi-Newton methods.

we have $\mathbb{E}_{\mathcal{D}}\left[g^k\right] \overset{(14)}{=} \frac{1}{n}\nabla\mathbf{F}(x^k)e \overset{(5)}{=} \nabla f(x^k)$, and hence, g^k is an unbiased estimate of the gradient. If (15) holds, we say that $\theta_{\mathbf{S}_k}$ is a *bias-correcting random variable* and $\mathbf{S}^k$ is an *unbiased sketch*. Our new *JacSketch* method is method (2) with g^k computed via (13) and the Jacobian estimate updated via (11). This method is formalized in Sect. 2 as Algorithm 1.

This strategy indeed works, as we show in detail in this paper. Under appropriate conditions (on the stepsize α, properties of f and randomness behind the sketch matrices $\mathbf{S}_k$ and so on), the variance of g^k diminishes to zero (e.g., see Lemma 6), which means that JacSketch is a variance-reduced method. We perform an analysis for smooth and strongly convex functions f, and obtain a linear convergence result (Theorem 1). We summarize our complexity results in detail in Sect. 1.5.

1.4 SAGA as a special case of JacSketch

Of particular importance in this paper are *minibatch sketches*, which are sketches of the form $\mathbf{S}_k = \mathbf{I}_{S_k}$, where S_k is a random subset of $[n]$, and $\mathbf{I}_{S_k}$ is a random column submatrix of the $n \times n$ identity matrix with columns indexed by S_k. For minibatch sketches, JacSketch corresponds to minibatch variants of SAGA. Indeed, in this case, and if $\mathbf{W} = \mathrm{Diag}(w_1, \ldots, w_n)$, we have $\mathbf{\Pi}_{\mathbf{S}_k}e = e_{S_k}$, where $e_S = \sum_{i \in S} e_i$ (see Lemma 7). Therefore,

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n}\sum_{i \in S_k}\left(\nabla f_i(x^k) - \mathbf{J}_{:i}^k\right). \tag{16}$$

In view of (11), and since $\mathbf{\Pi}_{\mathbf{S}_k} = \mathbf{I}_{S_k}\mathbf{I}_{S_k}^\top$ (see Lemma 7), the Jacobian estimate gets updated as follows

$$\mathbf{J}_{:i}^{k+1} = \begin{cases} \mathbf{J}_{:i}^k & i \notin S_k, \\ \nabla f_i(x^k) & i \in S_k. \end{cases} \tag{17}$$

Standard uniform SAGA is obtained by setting $S_k = \{i\}$ with probability $1/n$ for each $i \in [n]$, and letting $\theta_{\mathbf{S}_k} \equiv n$. SAGA with arbitrary probabilities is obtained by instead choosing $S_k = \{i\}$ with probability $p_i > 0$ for each $i \in [n]$, and letting $\theta_{\mathbf{S}_k} \equiv \frac{1}{p_i}$. However, virtually all minibatching and importance sampling strategies can be treated as special cases of our general approach.

The theory we develop answers the open questions raised earlier. In particular, we answer the conjecture of Schmidt et al. [30] about the rate of SAGA with importance sampling in the affirmative. In particular, we establish the iteration complexity $(n + \frac{4\bar{L}}{\mu})\log\frac{1}{\epsilon}$. This complexity is obtained for *different* importance sampling distributions that have not been proposed in the current literature for SAGA. In order to achieve this, we develop a new analysis technique which makes use of a *stochastic Lyapunov function* (see Sect. 5). That is, our Lyapunov function has a random element which is independent of the randomness inherited from the iterates of the method. This is unlike any other Lyapunov function used in the analysis of stochastic methods we

are aware of. Further, we prove that SAGA converges with any initial matrix $\mathbf{J}^0$ in place of the matrix of gradients of functions f_i at the starting point. We also show that our results give better rates for minibatch SAGA than are currently known, even for uniform minibatch strategies. We also allow for a family of completely new uniform minibatching strategies which were not considered in connection with SAGA before, and consider also SAGA with importance sampling for minibatches[4] (based on a partition of $[n]$). Lastly, as a special case, our method recovers standard gradient descent, together with the sharp iteration complexity of $\frac{4L}{\mu} \log \frac{1}{\epsilon}$.

Our general approach also enables a novel *reduced memory* variant of SAGA as a special case. Let $\mathbf{S}_k = e_{S_k}$, and choose $\mathbf{W} = \mathbf{I}$. Since $\mathbf{\Pi}_{\mathbf{S}_k} e = e_{S_k}$, the formula for g^k is the same as in the case of SAGA, and is given by (16). What is notably different about this sketch (compared to $\mathbf{I}_{S_k}$) is that, since $\mathbf{\Pi}_{e_{S_k}} = \frac{1}{|S_k|} e_{S_k} e_{S_k}^\top$, the update of the Jacobian estimate is given by

$$\mathbf{J}^{k+1} \overset{(11)}{=} \mathbf{J}^k - \frac{1}{|S_k|} \sum_{i \in S_k} \left(\mathbf{J}_{:i}^k - \nabla f_i(x^k) \right) e_{S_k}^\top.$$

Thus, *the same update is applied to all the columns of $\mathbf{J}^k$ that belong to S_k.* Equivalently, this update can be written as

$$\mathbf{J}_{:j}^{k+1} = \begin{cases} \frac{1}{|S_k|} \sum_{i \in S_k} \nabla f_i(x^k) & \text{if } j \in S_k, \\ \mathbf{J}_{:j}^k & \text{if } j \notin S_k. \end{cases} \tag{18}$$

In particular, if S_k only ever picks sets which correspond to a partition of $[n]$, and we initialize $\mathbf{J}^0$ so that all the columns belonging to the same partition are the same, then they will be the same within in each partition for all k. In such a case, we do not need to maintain all the identical copies. Instead, we can update and use a condensed/compressed version of the Jacobian, with one column per partition set only, to reduce the total memory usage. This method, with non-uniform probabilities, is analyzed in our framework in Sect. 5.6.

1.5 Summary of complexity results

All convergence results obtained in this paper are summarized in Table 1.

Our convergence results depend on several constants which we will now briefly introduce. The precise definitions can be found in the main text. For $\emptyset \neq C \subseteq [n] = \{1, 2, \ldots, n\}$, define $f_C(x) \overset{\text{def}}{=} \frac{1}{|C|} \sum_{i \in C} f_i(x)$. We assume f_C is L_C—smooth.[5] We let $L_i = L_{\{i\}}$, $L = L_{[n]}$, $L_{\max} = \max_i L_i$ and $\bar{L} = \frac{1}{n} \sum_i L_i$. Note that $L_i \leq L_{\max}$, $\bar{L} \leq L_{\max} \leq n\bar{L}$, $L_C \leq \frac{1}{|C|} \sum_{i \in C} L_i$ and $L \leq \bar{L}$. For a sampling[6] $S \subseteq [n]$, we let $\text{supp}(S) = \{C \subseteq [n] : \mathbb{P}[S = C] > 0\}$. That is, the support of a sampling

---

[4] For some prior results on importance sampling for minibatches, in the context of QUARTZ, see [5].

[5] A formal definition can be found in Assumption 4.2.

[6] In this paper, a sampling is a random set-valued mapping with the sets being subsets of $[n]$.

Table 1 Special cases of our JacSketch method, and the associated iteration complexity

ID	Method	Sketch $\mathbf{S} \in \mathbb{R}^{n \times \tau}$ $\mathbf{W} \succ 0$	Iteration complexity ($\times \log \frac{1}{\epsilon}$)	Reference
1	JacSketch	Any unbiased	$\max\left\{\frac{4\mathcal{L}_1}{\mu}, \frac{1}{\kappa} + \frac{4\rho\mathcal{L}_2}{\kappa\mu n^2}\right\}$	Theorem 1
		Any		
2	JacSketch	$\mathbf{I}_S$	$\max_{C \in \text{supp}(S)}\left(\frac{1}{p_C} + \frac{\tau}{n p_C}\frac{4L_C}{\mu}\right)$	Theorem 6
	(Any probabilities for τ—partition)	$\mathbf{I}$		
3	Gradient descent	$\mathbf{I}$	$\frac{4L}{\mu}$	Theorems 1 and 6
		$\mathbf{I}$		Sections 4.6 and 5.6
4	SAGA	$\mathbf{I}_S$	$n + \frac{4L_{\max}}{\mu}$	Theorems 1 and 6
	(Uniform sampling)	$\mathbf{I}$		Sections 4.6 and 5.6
5	SAGA	$\mathbf{I}_S$	$n + \frac{4\bar{L}}{\mu}$	Theorem 6
	(Importance sampling)	$\mathbf{I}$		(129)
6	Minibatch SAGA	$\mathbf{I}_S$	$\max\left\{\frac{4L_{\max}^{\mathcal{G}}}{\mu}, \frac{n}{\tau} + \frac{4\rho}{\mu n}\max_i\left(\frac{L_i}{w_i}\right)\right\}$	Theorem 1
	(τ—uniform sampling)	$\text{Diag}(w_i)$		(100)
7	Minibatch SAGA	$\mathbf{I}_S$	$\max\left\{\frac{4L_{\max}^{\mathcal{G}}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{(n-1)\tau}\frac{4L_{\max}}{\mu}\right\}$	Theorem 1
	(τ—nice sampling)	$\mathbf{I}$		(101)
8	Minibatch SAGA	$\mathbf{I}_S$	$\max\left\{\frac{4L_{\max}^{\mathcal{G}}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{n\tau}\frac{4(\bar{L}+L_{\max})}{\mu}\right\}$	Theorem 1
	(τ—nice sampling)	$\text{Diag}(L_i)$		(102)
9	Minibatch SAGA	$\mathbf{I}_S$	$\frac{n}{\tau} + \frac{4L_{\max}}{\mu}$	Theorem 1
	(τ—partition sampling)	$\mathbf{I}$		(103)
10	Minibatch SAGA	$\text{Diag}(L_i)$	$\frac{n}{\tau} + \frac{4\max_{C\in\text{supp}(S)}\frac{1}{\tau}\sum_{i\in C}L_i}{\mu}$	Theorem 1
	(τ—partition sampling)	$\mathbf{I}_S$		(104)
11	Minibatch SAGA	$\mathbf{I}_S$	$\frac{n}{\tau} + \frac{4\frac{1}{|\text{supp}(S)|}\sum_{C\in\text{supp}(S)}L_C}{\mu}$	Theorem 6
	(Importance τ—partition sampling)	$\mathbf{I}$		(131)

All methods converge linearly. In the iteration complexity column we list the number of iterations sufficient to obtain an ϵ accurate solution, ignoring a $\log\frac{1}{\epsilon}$ factor

are all the sets which are selected by this sampling with positive probability. Finally, $L_{\max}^{\mathcal{G}} = \max_i \frac{1}{c_1} \sum_{C \in \mathrm{supp}(S), i \in C} L_C$, where c_1 is the cardinality of the set $\{C \; : \; C \in \mathrm{supp}(S), i \in C\}$ (which is assumed to be the same for all i). So, $L_{\max}^{\mathcal{G}}$ is the maximum over i of averages of values L_C for those sets C which are picked by S with positive probability and which contain i. Clearly, $L_{\max}^{\mathcal{G}} \leq L_{\max}$ (see Theorem 3).

General theorem. Theorem 1 is our most general result, allowing for any(unbiased) sketch **S** (see (15)), and any weight matrix $\mathbf{W} \succ 0$. The resulting iteration complexity given by this theorem is

$$\max\left\{\frac{4\mathcal{L}_1}{\mu}, \frac{1}{\kappa} + \frac{4\rho\mathcal{L}_2}{\kappa\mu n^2}\right\} \times \log\left(\frac{1}{\epsilon}\right),$$

and is also presented in the first row of Table 1. This result depends on two *expected smoothness* constants $\mathcal{L}_1$ (measuring the expected smoothness of the stochastic gradient of our stochastic reformulation; see Assumption 3.1) and $\mathcal{L}_2$ (measuring the expected smoothness of the Jacobian; see Assumption 3.2). The complexity also depends on the *stochastic contraction number* κ (see (48)) and the *sketch residual* ρ (see (37) and (55)). We devote considerable effort to give simple formulas for these constants under some specialized settings (for special combinations of sketches **S** and weight matrices **W**). In fact, the entire Sect. 4 is devoted to this. In particular, all rows of Table 1 where the last column mentions Theorem 1 arise as special cases of the general iteration complexity in the first row.

– *Gradient descent* As a starting point, in row 3 we highlight that one can recover gradient descent as a special case of JacSketch with the choice $\mathbf{S} = \mathbf{I}$ (with probability 1) and $\mathbf{W} = \mathbf{I}$. We get the rate $\frac{4L}{\mu} \log \frac{1}{\epsilon}$, which is tight.
– *SAGA with uniform sampling* Let us now focus on a slightly more interesting special case: row 4. We see that SAGA with uniform probabilities appears as a special case, and enjoys the rate $\left(n + \frac{4L_{\max}}{\mu}\right) \log \frac{1}{\epsilon}$, recovering an existing result.
– *SAGA with importance sampling* Unfortunately, the generality of Theorem 1 comes at a cost: we are not able to obtain an importance sampling version of SAGA as a special case which would have a better iteration complexity than uniform SAGA. This will be remedied by our second complexity theorem, which we shall discuss later below.
– *Minibatch SAGA* Rows 6–11 correspond to minibatch versions of SAGA. In particular, row 6 contains a general statement (albeit still a special case of the statement in row 1), covering virtually all minibatch strategies. Rows 7–11 specialize this result to two particular minibatch sketches (i.e., $\mathbf{S} = \mathbf{I}_S$), each with two choices of **W**. The first sketch corresponds to samplings S which choose from among all subsets of $[n]$ uniformly at random. This sampling is known in the literature as τ-nice sampling [22,25]. The second sketch corresponds to S being a τ—partition sampling. This sampling picks uniformly at random subsets of $[n]$ which form a partition of $[n]$, and are all of cardinality τ. The complexities in rows 7 and 8 are comparable (each can be slightly better than the other, depending on the values of the smoothness constants $\{L_i\}$). On the other hand, in the case of τ—partition, the

choice $\mathbf{W} = \text{Diag}(L_i)$ is better than $\mathbf{W} = \mathbf{I}$: the complexity in row 10 is better than that in row 9 because $\max_{C \in \text{supp}(S)} \frac{1}{\tau} \sum_{i \in C} L_i \leq L_{\max}$.

– *Optimal minibatch size for SAGA* Our analysis for mini-batch SAGA also gives the first iteration complexities that interpolate between the $(n + \frac{4L_{\max}}{\mu}) \log \frac{1}{\epsilon}$ complexity of SAGA and the $\frac{4L}{\mu} \log \frac{1}{\epsilon}$ complexity of gradient descent, as τ increases from 1 to n. Indeed, consider the complexity in rows 7 and 8 for $\tau = 1$ and $\tau = n$. Our iteration complexity of mini-batch SAGA is the first result that is precise enough to inform an optimal mini-batch size (see Sect. 6.2). In contrast, the previous best complexity result for mini-batch SAGA [14] interpolates between $(n + \frac{4L_{\max}}{\mu}) \log \frac{1}{\epsilon}$ and $\frac{4L_{\max}}{\mu} \log \frac{1}{\epsilon}$ as τ increases from 1 to n, and thus is not precise enough as to inform the best minibatch size. We make a more detailed comparison between our results and [14] in Sect. 4.7.

Specialized theorem We now move to the second main complexity result of our paper: Theorem 6. The general complexity statement is listed in row 2 of Table 1:

$$\max_{C \in \text{supp}(S)} \left(\frac{1}{p_C} + \frac{\tau}{np_C} \frac{4L_C}{\mu} \right) \times \log \left(\frac{1}{\epsilon} \right), \tag{19}$$

where $p_C = \mathbb{P}[S = C]$. This theorem is a refined result specialized to minibatch sketches ($\mathbf{S} = \mathbf{I}_S$) with τ—partition samplings S. This is a sampling which picks subsets of $[n]$ of size τ forming a partition of $[n]$, uniformly at random. This theorem also includes gradient descent as special case since when $S = [n]$ with probability 1 (hence, $p_{[n]} = 1$) we have that $\tau = n$ and $L_{[n]} = L$. Hence, (19) specializes to $\frac{4L}{\mu} \log \frac{1}{\epsilon}$. But more importantly, our focus on τ—partition samplings enables us to provide stronger iteration complexity guarantees for non-uniform probabilities.

– *SAGA with importance sampling* The first remarkable special case of (19) is summarized in row 5, and corresponds to SAGA with importance sampling. The complexity obtained, $(n + \frac{4\bar{L}}{\mu}) \log \frac{1}{\epsilon}$, answers a conjecture of Schmidt et al. [30] in the affirmative. In this case, the support of S are the singletons $\{1\}, \{2\}, \ldots, \{n\}$, $p_{\{i\}} = p_i$ for all i, $\tau = 1$ and $L_{\{i\}} = L_i$. Optimizing the complexity bound over the probabilities $p_1, \ldots, p_n$, we obtain the importance sampling $p_i = \frac{\mu n + 4\tau L_i}{\sum_j \mu n + 4\tau L_j}$.

– *Minibatch SAGA with importance sampling* In row 11 we state the complexity for a minibatch SAGA method with importance sampling. This is the first result for this method in the literature. Note that by comparing rows 4 and 10, we can conclude that the complexity of minibatch SAGA with importance sampling is better than for minibatch SAGA with uniform probabilities. Indeed, this is because[7]

$$\frac{1}{|\text{supp}(S)|} \sum_{C \in \text{supp}(S)} L_C \leq \bar{L} \leq \max_{C \in \text{supp}(S)} \frac{1}{\tau} \sum_{i \in C} L_i. \tag{20}$$

---

[7] We prove inequality (20) in the Appendix; see Lemma 13.

1.6 Outline of the paper

We present an alternative narrative motivating the development of JacSketch in Sect. 2. This narrative is based on a novel technical tool which we call *controlled stochastic optimization reformulations* of problem (1). We then develop a general convergence theory of JacSketch in Sect. 3. This theory admits practically any sketches **S** (including minibatch sketches mentioned in the introduction) and weight matrices **W**. The main result in this section is Theorem 1. In Sect. 4 we specialize the general results to minibatch sketches. Here we also compute the various constants appearing in the general complexity result for JacSketch for specific classes of minibatch samplings. In Sect. 5 we develop an alternative theory for JacSketch, one based on a novel *stochastic Lyapunov function*. The main result in this section is Theorem 6. Computational experiments are included in Sect. 6.

1.7 Notation

We will introduce notation when and as needed. If the reader would like to recall any notation, for ease of reference we have a notation glossary in Sect. 1. As a general rule, all matrices are written in upper-case bold letters. By $\log t$ we refer to the natural logarithm of t.

2 Controlled stochastic reformulations

In this section we provide an alternative narrative behind the development of JacSketch; one through the lens of what we call *controlled stochastic reformulations*.

We design our family of methods so that two keys properties are satisfied, namely *unbiasedness*, $\mathbb{E}\left[g^k\right] = \nabla f(x^k)$, and *diminishing variance:* $\mathbb{E}\left[\left\|g^k - \nabla f(x^k)\right\|_2^2\right]$ $\longrightarrow 0$ as $x^k \to x^*$. These are both favoured statistical properties. Moreover, currently only methods that have diminishing variance exhibt fast linear convergence (exponential decay of the error) on strongly convex problems. On the other hand, unbiasedness is not necessary for a fast method in practice since several biased stochastic gradient methods such as SAG [29] perform well in practice. Still, the absence of bias greatly facilitates the analysis of JacSketch.

2.1 Stochastic reformulation using sketching

It will be useful to formalize the condition mentioned in Sect. 1.3 which leads to g^k being an unbiased estimator of the gradient.

Assumption 2.1 (*Unbiased sketch*) Let **W** $\succ$ 0 be a weighting matrix and let $\mathcal{D}$ be the distribution from which the sketch matrices **S** are drawn. There exists a random variable $\theta_{\mathbf{S}}$ such that

$$\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}}\right] e = e. \tag{21}$$

When this assumption is satisfied, we say that $(\mathbf{S}, \theta_{\mathbf{S}}, \mathbf{W})$ constitutes an "unbiased sketch", and we call $\theta_{\mathbf{S}}$ the bias-correcting random variable. When the triple is obvious from the context, sometimes we shall simply say that $\mathbf{S}$ is an unbiased sketch.

The first key insight of this section is that besides producing unbiased estimators of the gradient, unbiased sketches produce *unbiased estimators of the loss function* as well. Indeed, by simply observing that $f(x) = \frac{1}{n} \langle F(x), e \rangle$, we get

$$f(x) \overset{(1)}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x) = \frac{1}{n} \langle F(x), e \rangle \overset{(21)}{=} \frac{1}{n} \langle F(x), \mathbb{E}_{\mathcal{D}}[\theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}} e] \rangle$$

$$= \mathbb{E}_{\mathcal{D}} \left[\frac{1}{n} \langle F(x), \theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}} e \rangle \right].$$

In other words, we can rewrite the finite-sum optimization problem (1) as an equivalent stochastic optimization problem where the randomness comes from $\mathcal{D}$ rather than from the representation-specific uniform distribution over the n loss functions:

$$\min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}_{\mathcal{D}}[f_{\mathbf{S}}(x)], \qquad \text{where} \qquad f_{\mathbf{S}}(x) \overset{\text{def}}{=} \frac{\theta_{\mathbf{S}}}{n} \langle F(x), \mathbf{\Pi}_{\mathbf{S}} e \rangle. \tag{22}$$

The stochastic optimization problem (22) is a *stochastic reformulation* of the original problem (1). Further, the stochastic gradient of this reformulation is given by

$$\nabla f_{\mathbf{S}}(x) = \frac{\theta_{\mathbf{S}}}{n} \nabla \mathbf{F}(x) \mathbf{\Pi}_{\mathbf{S}} e. \tag{23}$$

With these simple observations, our options at designing stochastic gradient-type algorithms for (1) have suddenly broadened dramatically. Indeed, we can now solve the problem, at least in principle, by applying SGD to any stochastic reformulation:

$$x^{k+1} = x^k - \alpha \nabla f_{\mathbf{S}_k}(x^k). \tag{24}$$

But now we have a parameter to play with, namely, the distribution of $\mathbf{S}$. The choice of this parameter will influence both the iteration complexity of the resulting method as well as the cost of each iteration. We now give a few examples of possible choices of $\mathcal{D}$ to illustrate this.

Example 1 (gradient descent) Let $\mathbf{S}$ be equal to $\mathbf{I}$ (or any other $n \times n$ invertible matrix) with probability 1 and let $\mathbf{W} \succ 0$ be chosen arbitrarily. Then $\theta_{\mathbf{S}} \equiv 1$ is bias-correcting since

$$\mathbb{E}_{\mathcal{D}}[\theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}} e] = \mathbf{\Pi}_{\mathbf{S}} e \overset{(12)}{=} \mathbf{S}(\mathbf{S}^\top \mathbf{W} \mathbf{S})^\dagger \mathbf{S}^\top \mathbf{W} e = \mathbf{S} \mathbf{S}^{-1} \mathbf{W}^{-1} (\mathbf{S}^\top)^{-1} \mathbf{S}^\top \mathbf{W} e = \mathbf{I} e = e.$$

With this setup, the SGD method (24) becomes *gradient descent:*

$$x^{k+1} = x^k - \alpha \nabla f_{\mathbf{S}_k}(x^k) \overset{(5)+(23)}{=} x^k - \alpha \nabla f(x^k). \tag{25}$$

Example 2 (SGD with non-uniform sampling) Let $\mathbf{S} = e_i$ (unit basis vector in $\mathbb{R}^n$) with probability $p_i > 0$ and let $\mathbf{W} = \mathbf{I}$. Then $\theta_{e_i} = 1/p_i$ is bias-correcting since

$$\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}} e\right] \stackrel{(12)}{=} \sum_{i=1}^{n} p_i \frac{1}{p_i} e_i (e_i^\top e_i)^{-1} e_i^\top e = \sum_{i=1}^{n} e_i e_i^\top e = \mathbf{I} e = e.$$

Let $S_k = \{i_k\}$ be picked at iteration k. Then the SGD method (24) becomes *SGD with non-uniform sampling:*

$$x^{k+1} = x^k - \alpha \nabla f_{\mathbf{S}_k}(x^k) \stackrel{(23)}{=} x^k - \frac{\alpha}{np_{i_k}} \nabla f_{i_k}(x^k). \tag{26}$$

Note that with this setup, and when $p_i = 1/n$ for all i, the stochastic reformulation is identical to the original finite-sum problem. This is the case because $f_{e_i}(x) = f_i(x)$.

Example 3 (minibatch SGD) Let $\mathbf{S} = e_S = \sum_{i \in S} e_i$, where $S = C \subseteq [n]$ with probability p_C. Let $\mathbf{W} = \mathbf{I}$. Assume that the cardinality of the set $\{C \subseteq [n] : C \in \text{supp}(S), \ i \in C\}$ does not depend on i (and is equal to $c_1 > 0$). Then $\theta_{e_S} = 1/(c_1 p_S)$ is bias-correcting since

$$\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}} e\right] \stackrel{(12)}{=} \sum_{C \in \text{supp}(S)} p_C \frac{1}{c_1 p_C} e_C \underbrace{(e_C^\top e_C)^{-1}}_{|C|} \underbrace{e_C^\top e}_{|C|} = \sum_{C \in \text{supp}(S)} \frac{1}{c_1} e_C = e.$$

Note that $\mathbf{\Pi}_{e_S} e = e_S$. Assume that set S_k is picked in iteration k. Then the SGD method (24) becomes *minibatch SGD with non-uniform sampling:*

$$x^{k+1} = x^k - \alpha \nabla f_{\mathbf{S}_k}(x^k) \stackrel{(23)}{=} x^k - \frac{\alpha}{nc_1} \sum_{i \in S_k} \frac{1}{p_{S_k}} \nabla f_i(x^k). \tag{27}$$

Finally, note that gradient descent (25) is a special case of (27) if we set $p_{[n]} = 1$ and $p_C = 0$ for all other subsets C of $[n]$. Likewise, SGD with non-uniform probabilities (26) is a special case of (27) if we set $p_{\{i\}} = p_i > 0$ for all i and $p_C = 0$ for all other subsets C of $[n]$.

2.2 The controlled stochastic reformulation

Though SGD applied to the stochastic reformulation can generate several known algorithms in special cases, there is no reason to believe that the gradient estimates g^k will have diminishing variance (excluding the extreme case such as gradient descent). Here we handle this issue using *control variates*, a commonly used tool to reduce variance in Monte Carlo methods [13] and introduced in [35] for designing variance reduced stochastic gradient algorithm.

Given a random function $z_{\mathbf{S}}(x)$, we introduce the *controlled stochastic reformulation*:

$$\min_{x \in \mathbb{R}^d} f(x) = \mathbb{E}_{\mathcal{D}} \left[f_{\mathbf{S},z}(x) \right], \quad \text{where} \quad f_{\mathbf{S},z}(x) \stackrel{\text{def}}{=} f_{\mathbf{S}}(x) - z_{\mathbf{S}}(x) + \mathbb{E}_{\mathcal{D}} \left[z_{\mathbf{S}}(x) \right]. \tag{28}$$

Since

$$\nabla f_{\mathbf{S},z}(x) \stackrel{\text{def}}{=} \nabla f_{\mathbf{S}}(x) - \nabla z_{\mathbf{S}}(x) + \mathbb{E}_{\mathcal{D}} \left[\nabla z_{\mathbf{S}}(x) \right] \tag{29}$$

is an unbiased estimator of the gradient $\nabla f(x)$, we can apply SGD to the controlled stochastic reformulation instead, which leads to the method

$$x^{k+1} = x^k - \alpha (\nabla f_{\mathbf{S}_k}(x) - \nabla z_{\mathbf{S}_k}(x) + \mathbb{E}_{\mathcal{D}} \left[\nabla z_{\mathbf{S}}(x) \right]).$$

Reformulation (22) and method (24) is recovered as a special case with the choice $z_{\mathbf{S}}(x) \equiv 0$. However, we now have the extra freedom to choose $z_{\mathbf{S}}(x)$ so as to control the variance of this stochastic gradient. In particular, if $\nabla z_{\mathbf{S}}(x)$ and $\nabla f_{\mathbf{S}}(x)$ are sufficiently correlated, then (29) will have a smaller variance than $\nabla f_{\mathbf{S}}(x)$. For this reason, we choose a linear model for $z_{\mathbf{S}}(x)$ that mimicks the stochastic function $f_{\mathbf{S}}(x)$.

Let $\mathbf{J} \in \mathbb{R}^{d \times n}$ be a matrix of parameters of the following linear model

$$z_{\mathbf{S}}(x) \stackrel{\text{def}}{=} \frac{\theta_{\mathbf{S}}}{n} \left\langle \mathbf{J}^\top x, \mathbf{\Pi}_{\mathbf{S}} e \right\rangle, \quad \nabla z_{\mathbf{S}}(x) = \frac{\theta_{\mathbf{S}}}{n} \mathbf{J} \mathbf{\Pi}_{\mathbf{S}} e. \tag{30}$$

Note that this linear model has the same structure as $f_{\mathbf{S}}(x)$ in (22) except that $F(x)$ has been replaced by the linear function $\mathbf{J}^\top x$.[8] If $\mathbf{S}$ is an unbiased sketch (see (21)), we get $\mathbb{E}_{\mathcal{D}} \left[\nabla z_{\mathbf{S}}(x) \right] = \frac{1}{n} \mathbf{J} e$, which plugged into (28) and (29) together with the definition (22) of $f_{\mathbf{S}}$ gives the following unbiased estimate of $f(x)$ and $\nabla f(x)$:

$$f_{\mathbf{S},\mathbf{J}}(x) \stackrel{\text{def}}{=} f_{\mathbf{S},z}(x) = \frac{\theta_{\mathbf{S}}}{n} \left\langle F(x) - \mathbf{J}^\top x, \mathbf{\Pi}_{\mathbf{S}} e \right\rangle + \frac{1}{n} \left\langle \mathbf{J}^\top x, e \right\rangle, \tag{31}$$

and

$$\nabla f_{\mathbf{S},\mathbf{J}}(x) \stackrel{\text{def}}{=} \nabla f_{\mathbf{S},z}(x) = \frac{\theta_{\mathbf{S}}}{n} (\nabla \mathbf{F}(x) - \mathbf{J}) \mathbf{\Pi}_{\mathbf{S}} e + \frac{1}{n} \mathbf{J} e. \tag{32}$$

We collect this observation that (32) is unbiased in the following lemma for future reference.

Lemma 1 *If* $\mathbf{S}$ *is an unbiased sketch (see Definition* 2.1*), then*

$$\mathbb{E}_{\mathcal{D}} \left[\nabla f_{\mathbf{S},\mathbf{J}}(x) \right] = \nabla f(x), \tag{33}$$

---

[8] SVRG is also built on a linear covariate model [15].

for every $\mathbf{J} \in \mathbb{R}^{d \times n}$ *and* $x \in \mathbb{R}^d$. *That is,* (32) *is an unbiased estimate of the gradient* (1).

Now it remains to choose the matrix $\mathbf{J}$, which we do by minimizing the variance of our gradient estimate.

2.3 The Jacobian estimate, variance reduction and the sketch residual

Since (32) gives an unbiased estimator of $\nabla f(x)$ for all $\mathbf{J} \in \mathbb{R}^{d \times n}$, we can attempt to choose $\mathbf{J}$ that minimizes its variance. Minimizing the variance of (32) in terms of $\mathbf{J}$ will, for all sketching matrices of interest, lead to $\mathbf{J} = \nabla \mathbf{F}(x)$. This follows because

$$
\begin{aligned}
& \mathbb{E}_{\mathcal{D}} \left[\left\| \nabla f_{\mathbf{S}, \mathbf{J}}(x) - \nabla f(x) \right\|_2^2 \right] \\
& \stackrel{(32)}{=} \mathbb{E}_{\mathcal{D}} \left[\left\| \frac{1}{n} \mathbf{J}(\mathbf{I} - \theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}})e - \frac{1}{n} \nabla \mathbf{F}(x)(\mathbf{I} - \theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}})e \right\|_2^2 \right] \\
& = \frac{1}{n^2} \mathbb{E}_{\mathcal{D}} \left[\| (\mathbf{J} - \nabla \mathbf{F}(x))(\mathbf{I} - \theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}})e \|_2^2 \right] \\
& = \frac{1}{n^2} \mathrm{Tr} \left((\mathbf{J} - \nabla \mathbf{F}(x))^\top (\mathbf{J} - \nabla \mathbf{F}(x)) \mathbf{B} \right) \\
& = \frac{1}{n^2} \| \mathbf{J} - \nabla \mathbf{F}(x) \|_{\mathbf{B}}^2,
\end{aligned}
\tag{34}
$$

where

$$
\mathbf{B} \stackrel{\text{def}}{=} \mathbb{E}_{\mathcal{D}} \left[(\mathbf{I} - \theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}}) e e^\top (\mathbf{I} - \theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}}^\top) \right] \stackrel{(21)}{=} \mathbb{E}_{\mathcal{D}} \left[\theta_{\mathbf{S}}^2 \mathbf{\Pi}_{\mathbf{S}} e e^\top \mathbf{\Pi}_{\mathbf{S}}^\top \right] - e e^\top \succeq 0,
\tag{35}
$$

and we have used the weighted Frobenius norm with weight matrix $\mathbf{B}$ (see (10)).

For most distributions $\mathcal{D}$ of interest, the matrix $\mathbf{B}$ is positive definite.[9] Letting $v_{\mathbf{S}} \stackrel{\text{def}}{=} (\mathbf{I} - \theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}})e$, we can bound the largest eigenvalue of matrix $\mathbf{B}$ via Jensen's inequality as follows:

$$
\lambda_{\max}(\mathbf{B}) \stackrel{(35)}{=} \lambda_{\max}(\mathbb{E}_{\mathcal{D}} \left[v_{\mathbf{S}} v_{\mathbf{S}}^\top \right]) \leq \mathbb{E}_{\mathcal{D}} \left[\lambda_{\max}(v_{\mathbf{S}} v_{\mathbf{S}}^\top) \right] = \mathbb{E}_{\mathcal{D}} \left[\| v_{\mathbf{S}} \|_2^2 \right].
$$

Combined with (34), we get the following bound on the variance of $\nabla f_{\mathbf{S}, \mathbf{J}}$:

$$
\mathbb{E}_{\mathcal{D}} \left[\left\| \nabla f_{\mathbf{S}, \mathbf{J}}(x) - \nabla f(x) \right\|_2^2 \right] \leq \frac{\mathbb{E}_{\mathcal{D}} \left[\| v_{\mathbf{S}} \|_2^2 \right]}{n^2} \| \mathbf{J} - \nabla \mathbf{F}(x) \|_{\mathbf{I}}^2.
$$

This suggests that the variance is low when $\mathbf{J}$ is close to the true Jacobian $\nabla \mathbf{F}(x)$, and when the second moment of $v_{\mathbf{S}}$ is small. If $\mathbf{S}$ is an unbiased sketch, then $\mathbb{E}_{\mathcal{D}}[v_{\mathbf{S}}] = 0$,

---

[9] Excluding such trivial cases as when $\mathbf{S}$ is an invertible matrix and $\theta_{\mathbf{S}} = 1$ with probability one, in which case $\mathbf{B} = 0$.

and hence $\mathbb{E}_{\mathcal{D}}\left[\|v_{\mathbf{S}}\|_2^2\right]$ is the variance of $v_{\mathbf{S}}$. So, the lower the variance of $\frac{1}{n}\theta_{\mathbf{S}}\boldsymbol{\Pi}_{\mathbf{S}}e$ as an estimator of $\frac{1}{n}e$, the lower the variance of $\nabla f_{\mathbf{S},\mathbf{J}}(x)$ as an estimator of $\nabla f(x)$.

Let us now return to the identity (34) and its role in choosing $\mathbf{J}$. Minimizing the variance in a single step is overly ambitious, since it requires setting $\mathbf{J} = \nabla\mathbf{F}(x)$, which is costly. So instead, we propose to minimize (34) iteratively. But first, to make (34) more manageable, we upper-bound it using a norm defined by the weight matrix $\mathbf{W}$ as follows

$$\|\mathbf{J} - \nabla\mathbf{F}(x)\|_{\mathbf{B}}^2 \le \rho \|\mathbf{J} - \nabla\mathbf{F}(x)\|_{\mathbf{W}^{-1}}^2, \tag{36}$$

where

$$\rho \stackrel{\text{def}}{=} \lambda_{\max}\left(\mathbf{W}^{1/2}\mathbf{B}\mathbf{W}^{1/2}\right) \ge 0 \tag{37}$$

is the largest eigenvalue of $\mathbf{W}^{1/2}\mathbf{B}\mathbf{W}^{1/2}$. We refer to the constant ρ as the *sketch residual*, and it is a key constant affecting the convergence rate of JacSketch as captured by Theorem 1. The sketch residual ρ represents how much information is "lost" on average due to sketching and due to how well $\mathbf{W}^{-1}$ approximates $\mathbf{B}$. We develop formulae and estimates of the sketch residual for several specific sketches of interest in Sect. 4.5.

Example 4 (Zero sketch residual) Consider the setup from Example 1 (gradient descent). That is, let $\mathbf{S}$ be invertible with probability one and let $\theta_{\mathbf{S}} = 1$ be the bias-reducing variable. Then $\boldsymbol{\Pi}_{\mathbf{S}}e = e$ and hence $\mathbf{B} = 0$, which means that $\rho = 0$.

Example 5 (Large sketch residual) Consider the setup from Example 2 (SGD with non-uniform probabilities). That is, let $\mathbf{S} = e_i$ (unit basis vector in $\mathbb{R}^n$) with probability $p_i > 0$ and let $\mathbf{W} = \mathbf{I}$. Then $\theta_{e_i} = 1/p_i$ is a bias-reducing variable, and it is easy to show that $\mathbf{B} = \text{Diag}(1/p_1, \ldots, 1/p_n) - ee^\top$. If we choose $p_i = 1/n$ for all i, then $\rho = n$.

We have switched from the $\mathbf{B}$ norm to a user-controlled $\mathbf{W}^{-1}$ norm because minimizing under the $\mathbf{B}$ norm will prove to be impractical because $\mathbf{B}$ is a dense matrix for most all practical sketches. With this norm change we now have the option to set $\mathbf{W}$ as a sparse matrix (e.g., the identity, or a diagonal matrix), as we explain in Remark 1 further down. However, the theory we develop allows for any symmetric positive definite matrix $\mathbf{W}$.

We can now minimize (36) iteratively by only using a single sketch of the true Jacobian at each iteration. Suppose we have a current estimate $\mathbf{J}^k$ of the true Jacobian and a sketch of the true Jacobian $\nabla\mathbf{F}(x^k)\mathbf{S}_k$. With this we can calculate an improved Jacobian estimate using a projection step

$$\mathbf{J}^{k+1} = \underset{\mathbf{J}\in\mathbb{R}^{d\times n}, \mathbf{Y}\in\mathbb{R}^{m\times\tau}}{\arg\min} \frac{1}{2}\left\|\mathbf{J} - \nabla\mathbf{F}(x^k)\right\|_{\mathbf{W}^{-1}}^2 \quad \text{subject to} \quad \mathbf{J} = \mathbf{J}^k + \mathbf{Y}\mathbf{S}_k^\top\mathbf{W}, \tag{38}$$

the solution of which, as it turns out, depends on $\nabla\mathbf{F}(x^k)$ through its sketch $\nabla\mathbf{F}(x^k)\mathbf{S}_k$ only. That is, we choose the next Jacobian estimate $\mathbf{J}^{k+1}$ as close as possible to the

true Jacobian $\nabla \mathbf{F}(x^k)$ while restricted to a matrix subspace that passes through $\mathbf{J}^k$. Thus in light of (36), the variance is decreasing. The explicit solution to (38) is given by

$$\mathbf{J}^{k+1} = \mathbf{J}^k - (\mathbf{J}^k - \nabla \mathbf{F}(x^k))\mathbf{\Pi}_{\mathbf{S}_k}. \tag{39}$$

See Lemma B.1 in the appendix of an extended preprint version of this paper [10] or Theorem 4.1 in [12] for the proof. Note that, as alluded to before, $\mathbf{J}^{k+1}$ depends on $\nabla \mathbf{F}(x^k)$ through its sketch only. Note that (39) updates the Jacobian estimate by re-using the sketch $\nabla \mathbf{F}(x^k)\mathbf{S}_k$ which we also use when calculating the stochastic gradient (32).

Note that (39) gives the same formula for $\mathbf{J}^{k+1}$ as (11) which we obtained by solving (9); i.e., by projecting $\mathbf{J}^k$ onto the solution set of (8). This is not a coincidence. In fact, the optimization problems (9) and (38) are mutually dual. This is also formally stated in Lemma B.1 in [10].

In the context of solving linear systems, this was observed in [11]. Therein, (9) is called the sketch-and-project method, whereas (38) is called the *constrain-and-approximate* problem. In this sense, the Jacobian sketching narrative we followed in Sect. 1.3 is dual to the Jacobian sketching narrative we are pursuing here.

Remark 1 (On the weight matrix and the cost) Loosely speaking, the denser the weighting matrix $\mathbf{W}$, the higher the computational cost for updating the Jacobian using (39). Indeed, the sparsity pattern of $\mathbf{W}$ controls how many elements of the previous Jacobian estimate $\mathbf{J}^k$ need to be updated. This can be seen by re-arranging (39) as

$$\mathbf{J}^{k+1} = \mathbf{J}^k + \mathbf{Y}_k \mathbf{S}_k^\top \mathbf{W}, \tag{40}$$

where $\mathbf{Y}_k = (\nabla \mathbf{F}(x^k)\mathbf{S}_k - \mathbf{J}^k\mathbf{S}_k)(\mathbf{S}_k^\top \mathbf{W}\mathbf{S}_k)^\dagger \in \mathbb{R}^{d \times \tau}$. Although we have no control over the sparsity of $\mathbf{Y}_k$, the matrix $\mathbf{S}_k^\top \mathbf{W}$ can be sparse when both $\mathbf{S}_k$ and $\mathbf{W}$ are sparse. This will be key in keeping the update (40) at a cost propotional to $d \times \tau$, as opposed to $n \times d$ when $\mathbf{W}$ is dense. This is why we consider a diagonal matrix $\mathbf{W} = \mathrm{Diag}(w_1, \ldots, w_n)$ in all of the special complexity results in Table 1. While it is clear that some non-diagonal sparse matrices $\mathbf{W}$ could also be used, we leave such considerations to future work.

2.4 JacSketch algorithm

Combining formula (32) for the stochastic gradient of the controlled stochastic reformulation with formula (39) for the update of the Jacobian estimate, we arrive at our JacSketch algorithm (Algorithm 1).

Typically, one should not implement the algorithm as presented above. The most efficient implementation of JacSketch will depend heavily on the structure of $\mathbf{W}$, distribution $\mathcal{D}$ and so on. For instance, in the special case of minibatch SAGA, as presented in Sect. 1.4, the update of the Jacobian (77) has a particularly simple form. That is, we maintain a single matrix $\mathbf{J} \in \mathbb{R}^{d \times n}$ and keep replacing its columns by the appropriate

---

Algorithm 1 JacSketch: Variance Reduced Gradient Method via Jacobian Sketching

---

1: **Input:** $(\mathcal{D}, \mathbf{W}, \theta_{\mathbf{S}})$
2: **Initialize:** $x^0 \in \mathbb{R}^d$, Jacobian estimate $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$, stepsize $\alpha > 0$
3: **for** $k = 0, 1, 2, \ldots$ **do**
4:     Sample a fresh copy $\mathbf{S}_k \sim \mathcal{D}$
5:     Calculate $\nabla\mathbf{F}(x^k)\mathbf{S}_k$ ▷ Sketch the Jacobian
6:     $\mathbf{J}^{k+1} = \mathbf{J}^k + (\nabla\mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k} = \mathbf{J}^k(\mathbf{I} - \mathbf{\Pi}_{\mathbf{S}_k}) + \nabla\mathbf{F}(x^k)\mathbf{\Pi}_{\mathbf{S}_k}$ ▷ Update Jacobian estimate
7:     $g^k = \frac{1}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n}(\nabla\mathbf{F}(x^k) - \mathbf{J}^k)\mathbf{\Pi}_{\mathbf{S}_k} e = \frac{1 - \theta_{\mathbf{S}_k}}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n}\mathbf{J}^{k+1} e$ ▷ Update gradient estimate
8:     $x^{k+1} = x^k - \alpha g^k$ ▷ Take a step

---

stochastic gradients, as computed. Moreover, in the case of linear predictors, as is well known, a much more memory-efficient implementation is possible. In particular, if $f_i(x) = \phi_i(a_i^\top x)$ for some loss function ϕ_i and a data vector $a_i \in \mathbb{R}^d$ and all i, then $\nabla f_i(x) = \phi_i'(a_i^\top x)a_i$, which means that the gradient always points in the same direction. In such a situation, it is sufficient to keep track of the scalar loss derivatives $\phi_i'(a_i^\top x)$ only. Similar comments can be made about the step (16) for computing the gradient estimate g^k.

2.5 A window into biased estimates and SAG

We will now take a small detour from the main flow of the paper to develop an alternative viewpoint of Algorithm 1 and also make a bridge to biased methods such as SAG [29].

The simple observation that

$$\nabla f(x^k) = \frac{1}{n}\nabla\mathbf{F}(x^k)e, \tag{41}$$

suggests that $\hat{g}^k = \frac{1}{n}\mathbf{J}^{k+1}e$, where $\mathbf{J}^{k+1} \approx \nabla\mathbf{F}(x^k)$ would give a good estimate of the gradient. To decrease the variance of $\hat{g}^k$, we can also use the same update of the Jacobian estimate (39) since

$$
\begin{aligned}
\mathbb{E}\left[\left\|\hat{g}^k - \nabla f(x^k)\right\|_2^2\right] &= \frac{1}{n^2}\mathbb{E}\left[\left\|(\mathbf{J}^{k+1} - \nabla\mathbf{F}(x^k))e\right\|_2^2\right] \\
&= \frac{1}{n^2}\mathbb{E}\left[\left\|(\mathbf{J}^{k+1} - \nabla\mathbf{F}(x^k))\mathbf{W}^{-1/2}\mathbf{W}^{1/2}e\right\|_2^2\right] \\
&\leq \frac{e^\top\mathbf{W}e}{n^2}\mathbb{E}\left[\left\|\mathbf{J}^{k+1} - \nabla\mathbf{F}(x^k)\right\|_{\mathbf{W}^{-1}}^2\right].
\end{aligned}
$$

Thus, if $\mathbb{E}\left[\left\|\mathbf{J}^{k+1} - \nabla\mathbf{F}(x^k)\right\|_{\mathbf{W}^{-1}}^2\right]$ converges to zero, so will $\mathbb{E}\left[\left\|\hat{g}^k - \nabla f(x^k)\right\|_2^2\right]$. Though unfortunately, the combination of the gradient estimate $\hat{g}^k = \frac{1}{n}\mathbf{J}^{k+1}e$ and a Jacobian estimate updated via (39) will almost always give a biased estimator. For example, if we define $\mathcal{D}$ by setting $\mathbf{S} = e_i$ with probability $\frac{1}{n}$ and let $\mathbf{W} = \mathbf{I}$, then we recover the celebrated SAG method [29] and its biased estimator of the gradient.

The issue with using $\frac{1}{n}\mathbf{J}^{k+1}e$ as an estimator of the gradient is that it decreases the variance too aggressively, neglecting the bias. However, this can be fixed by trading off variance for bias. One way to do this is to introduce the random variable $\theta_{\mathbf{S}}$ as a *stochastic relaxation parameter*

$$\hat{g}^k = \frac{1 - \theta_{\mathbf{S}_k}}{n}\mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n}\mathbf{J}^{k+1}e. \tag{42}$$

If $\theta_{\mathbf{S}}$ is bias correcting, we recover the unbiased SAGA estimator (13). By allowing $\theta_{\mathbf{S}}$ to be closer to one, however, we will get more bias and lower variance. We leave this strategy of building biased estimators for future work. It is conceivable that SAG could be analyzed using reasonably small modifications of the tools developed in this paper. Doing this would be important due to at least four reasons: (i) SAG was the first variance-reduced method for problem (1), (ii) the existing analysis of SAG is not satisfying, (iii) one may be able to obtain a better rate, (iv) one may be able to develop and analyze novel variants of SAG.

3 Convergence analysis for general sketches

In this section we establish a convergence theorem (Theorem 1) which applies to general sketching matrices $\mathbf{S}$ (that is, arbitrary distributions $\mathcal{D}$ from which they are sampled). By design, we keep the setting in this section general, and only deal with specific instantiations and special cases in Sect. 4.

3.1 Two expected smoothness constants

We first formulate two *expected smoothness* assumptions tying together f, its Jacobian $\nabla \mathbf{F}(x)$ and the distribution $\mathcal{D}$ from which we pick sketch matrices $\mathbf{S}$. These assumptions, and the associated expected smoothness constants, play a key role in the convergence result.

Our first assumption concerns the expected smoothness of the stochastic gradients $\nabla f_{\mathbf{S}}$ of the stochastic reformulation (22).[10]

Assumption 3.1 (*Expected smoothness of the stochastic gradient*) There is a constant $\mathcal{L}_1 > 0$ such that

$$\mathbb{E}_{\mathcal{D}}\left[\left\|\nabla f_{\mathbf{S}}(x) - \nabla f_{\mathbf{S}}(x^*)\right\|_2^2\right] \leq 2\mathcal{L}_1(f(x) - f(x^*)), \quad \forall x \in \mathbb{R}^d. \tag{43}$$

It is easy to see from (23) and (32) that

---

[10] A similar relation to (43) holds for the stochastic optimization reformulation of linear systems studied by Richtárik and Takáč [26]. Therein, this relation holds as an identity with $\mathcal{L}_1 = 1$ (see Lemma 3.3 in [26]). However, the function $f_{\mathbf{S}}$ considered there is entirely different and, moreover, $f(x^*) = 0$ and $\nabla f_{\mathbf{S}}(x^*) = 0$ for all $\mathbf{S}$.

$$\|\nabla f_{\mathbf{S}}(x) - \nabla f_{\mathbf{S}}(y)\|_2^2 = \frac{1}{n^2} \|(\nabla \mathbf{F}(x) - \nabla \mathbf{F}(y))\theta_{\mathbf{S}} \mathbf{\Pi}_{\mathbf{S}} e\|_2^2$$
$$= \left\|\nabla f_{\mathbf{S},\mathbf{J}}(x) - \nabla f_{\mathbf{S},\mathbf{J}}(y)\right\|_2^2 \qquad (44)$$

for all $\mathbf{J} \in \mathbb{R}^{d \times n}$ and $x, y \in \mathbb{R}^d$, and hence the expected smoothness assumption can equivalently be understood from the point of view of the controlled stochastic reformulation. The above assumption is not particularly restrictive. Indeed, in Theorem 2 we provide formulae for $\mathcal{L}_1$ for smooth functions f and for a class of minibatch samplings $\mathbf{S} = \mathbf{I}_S$. These formulae can be seen as proofs that Assumption 3.1 is satisfied for a large class of practically relevant sketches $\mathbf{S}$ and functions f.

Our second expected smoothness assumption concerns the Jacobian of F.

Assumption 3.2 (*Expected smoothness of the Jacobian*) There is a constant $\mathcal{L}_2 > 0$ such that

$$\mathbb{E}_{\mathcal{D}} \left[\left\|(\nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{S}}\right\|_{\mathbf{W}^{-1}}^2 \right] \le 2\mathcal{L}_2 (f(x) - f(x^*)), \quad \forall x \in \mathbb{R}^d, \quad (45)$$

where the norm is the weighted Frobenius norm defined in (10).

It is easy to see (see Lemma 4, Eq. (60)) that for any matrix $\mathbf{M} \in \mathbb{R}^{d \times n}$, we have $\mathbb{E}_{\mathcal{D}} \left[\|\mathbf{M}\mathbf{\Pi}_{\mathbf{S}}\|_{\mathbf{W}^{-1}}^2 \right] = \|\mathbf{M}\|_{\mathbb{E}_{\mathcal{D}}[\mathbf{H}_{\mathbf{S}}]}^2$, where

$$\mathbf{H}_{\mathbf{S}} \stackrel{\text{def}}{=} \mathbf{S}(\mathbf{S}^\top \mathbf{W} \mathbf{S})^\dagger \mathbf{S}^\top \stackrel{(12)}{=} \mathbf{\Pi}_{\mathbf{S}} \mathbf{W}^{-1}. \qquad (46)$$

Therefore, (45) can be equivalently written in the form

$$\left\|\nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*)\right\|_{\mathbb{E}_{\mathcal{D}}[\mathbf{H}_{\mathbf{S}}]}^2 \le 2\mathcal{L}_2 (f(x) - f(x^*)), \quad \forall x \in \mathbb{R}^d, \qquad (47)$$

which suggests that the above condition indeed measures the variation/smoothness of the Jacobian under a specific weighted Frobenius norm.

3.2 Stochastic contraction number

By the *stochastic contraction number* associated with $\mathbf{W}$ and $\mathcal{D}$ we mean the constant defined by

$$\kappa = \kappa(\mathcal{D}, \mathbf{W}) \stackrel{\text{def}}{=} \lambda_{\min}(\mathbb{E}_{\mathcal{D}}[\mathbf{\Pi}_{\mathbf{S}}]). \qquad (48)$$

In the next lemma we show that $0 \le \kappa \le 1$ for all distributions $\mathcal{D}$ for which the expectation (48) exists.

Lemma 2 *For all distributions $\mathcal{D}$, we have the bounds $0 \le \kappa \le 1$.*

Proof It is not difficult to show that $\mathbf{W}^{1/2}\mathbf{H}_{\mathbf{S}}\mathbf{W}^{1/2} \stackrel{(46)}{=} \mathbf{W}^{1/2}\mathbf{\Pi}_{\mathbf{S}}\mathbf{W}^{-1/2}$ is the orthogonal projection matrix that projects onto $\mathrm{Range}\left(\mathbf{W}^{1/2}\mathbf{S}\right)$. Consequently, $0 \preceq$

$\mathbf{W}^{1/2}\mathbf{H_S}\mathbf{W}^{1/2} \preceq \mathbf{I}$ and, after taking expectation, we get $0 \preceq \mathbf{W}^{1/2}\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]\mathbf{W}^{1/2} \preceq \mathbf{I}$. Finally, this implies that

$$0 \le \lambda_{\max}(\mathbf{I} - \mathbf{W}^{1/2}\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]\mathbf{W}^{1/2}) = 1 - \lambda_{\min}(\mathbf{W}^{1/2}\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]\mathbf{W}^{1/2}) \le 1. \quad (49)$$

$\square$

In our convergence theorem we will assume that $\kappa > 0$. This can be achieved by choosing a suitable distribution $\mathcal{D}$ and it holds trivially for all the examples we develop. The condition $\kappa > 0$ essentially says that the distribution is sufficiently rich. This contraction number was first proposed in [11] in the context of randomized algorithms for solving linear systems. We refer the reader to that work for details on sufficient assumptions about $\mathcal{D}$ guaranteeing $\kappa > 0$. Below we give an example.

Example 6 Let $\mathbf{W} \succ 0$, and let $\mathcal{D}$ be given by setting $\mathbf{S} = e_i$ with probability $p_i > 0$. Then

$$\kappa \overset{(48)}{=} \lambda_{\min}\left(\mathbf{W}^{1/2}\mathbb{E}_{\mathcal{D}}[\mathbf{\Pi_S}]\mathbf{W}^{-1/2}\right) = \lambda_{\min}\left(\sum_{i=1}^{n} \frac{p_i}{e_i^{\top}\mathbf{W}e_i}\mathbf{W}^{1/2}e_i e_i^{\top}\mathbf{W}^{1/2}\right).$$

Since the vectors $\mathbf{W}^{1/2}e_i$ span $\mathbb{R}^n$ and $p_i > 0$ for all i, the matrix is positive definite and hence $\kappa > 0$. In particular, when $\mathbf{W} = \mathbf{I}$, then the expected projection matrix is equal to $\text{Diag}(p_1, \ldots, p_n)$ and $\kappa = \min_i p_i > 0$. If instead of unit basis vectors $\{e_i\}$ we use vectors that span $\mathbb{R}^n$, using similar arguments we can also conclude that $\kappa > 0$.

3.3 Convergence theorem

Our main convergence result, which we shall present shortly, holds for μ-strongly convex functions. However, it turns out our results hold for the somewhat larger family of functions that are quasi-strongly convex.

Assumption 3.3 (*Quasi-strong convexity*) Function f for some $\mu > 0$ satisfies

$$f(x^*) \ge f(x) + \langle \nabla f(x), x^* - x \rangle + \frac{\mu}{2}\|x^* - x\|_2^2, \quad \forall x \in \mathbb{R}^d, \quad (50)$$

where $x^* = \arg\min_{x\in\mathbb{R}^d} f(x)$.

We are now ready to present the main result of this section.

Theorem 1 (Convergence of JacSketch for General Sketches) *Let $\mathbf{W} \succ 0$. Let f satisfy Assumption 3.3. Let Assumption 2.1 be satisfied (i.e., $\mathbf{S}$ is an unbiased sketch and $\theta_{\mathbf{S}}$ is the associated bias-correcting random variable). Let the expected smoothness assumptions be satisfied: Assumptions 3.1 and 3.2. Assume that $\kappa > 0$. Let the sketch residual be defined as in (37), i.e.,*

$$\rho = \rho(\theta_{\mathbf{S}}, \mathcal{D}, \mathbf{W}) \overset{(37)}{=} \lambda_{\max}\left(\mathbf{W}^{1/2}\left(\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}}^2\mathbf{\Pi_S}ee^{\top}\mathbf{\Pi_S}\right] - ee^{\top}\right)\mathbf{W}^{1/2}\right) \ge 0. \quad (51)$$

Choose any $x^0 \in \mathbb{R}^d$ and $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$. Let $\{x^k, \mathbf{J}^k\}_{k \geq 0}$ be the random iterates produced by JacSketch (Algorithm 1). Consider the Lyapunov function

$$\Psi^k \stackrel{def}{=} \left\| x^k - x^* \right\|_2^2 + \frac{\alpha}{2\mathcal{L}_2} \left\| \mathbf{J}^k - \nabla \mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2 . \tag{52}$$

If the stepsize satisfies

$$0 \leq \alpha \leq \min \left\{ \frac{1}{4\mathcal{L}_1}, \frac{\kappa}{4\mathcal{L}_2\rho/n^2 + \mu} \right\}, \tag{53}$$

then

$$\mathbb{E}\left[\Psi^k\right] \leq (1 - \mu\alpha)^k \cdot \Psi^0, \tag{54}$$

If we choose α to be equal to the upper bound in (53), then

$$k \geq \max\left\{ \frac{4\mathcal{L}_1}{\mu}, \frac{1}{\kappa} + \frac{4\rho\mathcal{L}_2}{\kappa\mu n^2} \right\} \log\left(\frac{1}{\epsilon}\right) \Rightarrow \mathbb{E}\left[\Psi^k\right] \leq \epsilon\Psi^0. \tag{55}$$

Recall that the iteration complexity expression from (55) is listed in row 1 of Table 1.

The Lyapunov function we use is simply the sum of the squared distance between x^k to the optimal x^* and the distance of our Jacobian estimate $\mathbf{J}^k$ to the optimal Jacobian $\nabla \mathbf{F}(x^*)$. Hence, the theorem says that both the iterates $\{x^k\}$ and the Jacobian estimates $\{\mathbf{J}^k\}$ converge.

3.4 Projection lemmas and the stochastic contraction number κ

In this section we collect some basic results on projections. Recall from (12) that $\mathbf{\Pi_S} = \mathbf{S}(\mathbf{S}^\top \mathbf{W} \mathbf{S})^\dagger \mathbf{S}^\top \mathbf{W}$ and from (46) that $\mathbf{H_S} = \mathbf{S}(\mathbf{S}^\top \mathbf{W} \mathbf{S})^\dagger \mathbf{S}^\top$.

Lemma 3

$$\mathbf{\Pi_S} \mathbf{W}^{-1} (\mathbf{I} - \mathbf{\Pi_S})^\top = 0. \tag{56}$$

Furthermore,

$$\mathbb{E}_\mathcal{D}\left[\mathbf{\Pi_S} \mathbf{W}^{-1} \mathbf{\Pi_S}^\top\right] = \mathbb{E}_\mathcal{D}\left[\mathbf{H_S}\right] \quad and$$
$$\mathbb{E}_\mathcal{D}\left[(\mathbf{I} - \mathbf{\Pi_S}) \mathbf{W}^{-1} (\mathbf{I} - \mathbf{\Pi_S})^\top\right] = \mathbf{W}^{-1} - \mathbb{E}_\mathcal{D}\left[\mathbf{H_S}\right]. \tag{57}$$

Proof Using the pseudoinverse property $\mathbf{A}^\dagger \mathbf{A} \mathbf{A}^\dagger = \mathbf{A}^\dagger$ we have that

$$\mathbf{\Pi_S} \mathbf{W}^{-1} \mathbf{\Pi_S}^\top \stackrel{(12)}{=} \mathbf{S}(\mathbf{S}^\top \mathbf{W} \mathbf{S})^\dagger \mathbf{S}^\top \mathbf{W} \mathbf{S}(\mathbf{S}^\top \mathbf{W} \mathbf{S})^\dagger \mathbf{S}^\top \stackrel{(46)}{=} \mathbf{\Pi_S} \mathbf{W}^{-1} = \mathbf{H_S}, \tag{58}$$

and as a consequence (56) holds. Moreover,

$$(\mathbf{I} - \mathbf{\Pi_S})\mathbf{W}^{-1}(\mathbf{I} - \mathbf{\Pi_S})^\top \overset{(56)}{=} \mathbf{W}^{-1}(\mathbf{I} - \mathbf{\Pi_S})^\top \overset{(46)}{=} \mathbf{W}^{-1} - \mathbf{H_S}. \tag{59}$$

Finally, taking expectation over (58) and (59) gives (57).   □

Lemma 4 *For any matrices* $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{d \times n}$ *we have the identities*

$$\|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S}) + \mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2 = \|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S})\|_{\mathbf{W}^{-1}}^2 + \|\mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2$$

and

$$\mathbb{E}_{\mathcal{D}}\left[\|\mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2\right] = \|\mathbf{N}\|_{\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]}^2. \tag{60}$$

Furthermore,

$$\mathbb{E}_{\mathcal{D}}\left[\|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S}) + \mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2\right] \le (1 - \kappa)\|\mathbf{M}\|_{\mathbf{W}^{-1}}^2 + \|\mathbf{N}\|_{\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]}^2. \tag{61}$$

Proof First, note that

$$\begin{aligned}\|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S}) + \mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2 &= \|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S})\|_{\mathbf{W}^{-1}}^2 + \|\mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2 \\ &\quad + \mathrm{Tr}\left(\mathbf{M}^\top \mathbf{N}\mathbf{\Pi_S}\mathbf{W}^{-1}(\mathbf{I} - \mathbf{\Pi_S})^\top\right) \\ &\overset{(56)}{=} \|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S})\|_{\mathbf{W}^{-1}}^2 + \|\mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2.\end{aligned}$$

By taking expectations in $\mathcal{D}$, we get

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[\|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S}) + \mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2\right] &= \mathbb{E}_{\mathcal{D}}\left[\|\mathbf{M}(\mathbf{I} - \mathbf{\Pi_S})\|_{\mathbf{W}^{-1}}^2\right] + \mathbb{E}_{\mathcal{D}}\left[\|\mathbf{N}\mathbf{\Pi_S}\|_{\mathbf{W}^{-1}}^2\right] \\ &\overset{(57)}{=} \|\mathbf{M}\|_{\mathbf{W}^{-1} - \mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]}^2 + \|\mathbf{N}\|_{\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]}^2 \\ &\le (1 - \kappa)\|\mathbf{M}\|_{\mathbf{W}^{-1}}^2 + \|\mathbf{N}\|_{\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]}^2,\end{aligned}$$

where in the last step we used the estimate

$$\begin{aligned}\mathbf{W}^{-1} - \mathbb{E}_{\mathcal{D}}[\mathbf{H_S}] &= \mathbf{W}^{-1/2}(\mathbf{I} - \mathbf{W}^{1/2}\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]\mathbf{W}^{1/2})\mathbf{W}^{-1/2} \\ &\preceq \lambda_{\max}(\mathbf{I} - \mathbf{W}^{1/2}\mathbb{E}_{\mathcal{D}}[\mathbf{H_S}]\mathbf{W}^{1/2})\mathbf{W}^{-1} \overset{(49)}{=} (1 - \kappa)\mathbf{W}^{-1}.\end{aligned}$$

□

3.5 Key lemmas

We first establish two lemmas. The first lemma provides an upper bound on the quality of new Jacobian estimate in terms of the quality of the current estimate and function

suboptimality. If the second term on the right hand side was not there, the lemma would be postulating a contraction on the quality of the Jacobian estimate.

Lemma 5 *Let Assumption* 3.2 *be satisfied. Then iterates of Algorithm* 1 *satisfy*

$$\mathbb{E}_{\mathcal{D}}\left[\left\|\mathbf{J}^{k+1} - \nabla\mathbf{F}(x^*)\right\|_{\mathbf{W}^{-1}}^2\right] \le (1 - \kappa)\left\|\mathbf{J}^k - \nabla\mathbf{F}(x^*)\right\|_{\mathbf{W}^{-1}}^2 \\ + 2\mathcal{L}_2(f(x^k) - f(x^*)), \qquad (62)$$

where κ is defined in (48).

Proof Subtracting $\nabla\mathbf{F}(x^*)$ from both sides of (39) gives

$$\mathbf{J}^{k+1} - \nabla\mathbf{F}(x^*) \overset{(39)}{=} \underbrace{(\mathbf{J}^k - \nabla\mathbf{F}(x^*))}_{\mathbf{M}}(\mathbf{I} - \mathbf{\Pi}_{\mathbf{S}_k}) + \underbrace{(\nabla\mathbf{F}(x^k) - \nabla\mathbf{F}(x^*))}_{\mathbf{N}}\mathbf{\Pi}_{\mathbf{S}_k}. \quad (63)$$

Taking norms on both sides, then expectation with respect to $\mathbf{S}_k$ and then using Lemma 4, we get

$$\mathbb{E}_{\mathcal{D}}\left[\left\|\mathbf{J}^{k+1} - \nabla\mathbf{F}(x^*)\right\|_{\mathbf{W}^{-1}}^2\right] \overset{(61)}{\le} (1 - \kappa)\|\mathbf{M}\|_{\mathbf{W}^{-1}}^2 + \|\mathbf{N}\|_{\mathbb{E}_{\mathcal{D}}[\mathbf{H}_{\mathbf{S}_k}]}^2 \\ \overset{(45)}{\le} (1 - \kappa)\left\|\mathbf{J}^k - \nabla\mathbf{F}(x^*)\right\|_{\mathbf{W}^{-1}}^2 \\ + 2\mathcal{L}_2(f(x^k) - f(x^*)).$$

□

We now bound the second moment of g^k. The lemma implies that as x^k approaches x^* and $\mathbf{J}^k$ approaches $\nabla\mathbf{F}(x^*)$, the variance of g^k approaches zero. This is a key property of JacSketch which elevates it into the ranks of variance-reduced methods.

Lemma 6 *Let* $\mathbf{S}$ *be an unbiased sketch. Let Assumption* 3.1 *be satisfied (i.e., assume that inequality* (43) *holds for some* $\mathcal{L}_1 > 0$*). Then the second moment of the estimated gradient is bounded by*

$$\mathbb{E}_{\mathcal{D}}\left[\left\|g^k\right\|_2^2\right] \le 4\mathcal{L}_1(f(x^k) - f(x^*)) + 2\frac{\rho}{n^2}\left\|\mathbf{J}^k - \nabla\mathbf{F}(x^*)\right\|_{\mathbf{W}^{-1}}^2, \qquad (64)$$

where ρ is defined in (51).

Proof Adding and subtracting $\frac{\theta_{\mathbf{S}_k}}{n}\nabla\mathbf{F}(x^*)\mathbf{\Pi}_{\mathbf{S}_k}e$ in (13) gives

$$g^k = \underbrace{\frac{1}{n}\mathbf{J}^k e - \frac{\theta_{\mathbf{S}_k}}{n}(\mathbf{J}^k - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{S}_k}e}_{b} + \underbrace{\frac{\theta_{\mathbf{S}_k}}{n}(\nabla\mathbf{F}(x^k) - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{S}_k}e}_{a}.$$

Taking norms on both sides and using the bound $\|a + b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2$ gives

$$
\left\| g^k \right\|_2^2 \leq \underbrace{\frac{2}{n^2} \left\| (\nabla \mathbf{F}(x^k) - \nabla \mathbf{F}(x^*)) \mathbf{\Pi}_{\mathbf{S}_k} \theta_{\mathbf{S}_k} e \right\|_2^2}_{a^k}
$$
$$
+ \underbrace{\frac{2}{n^2} \left\| \theta_{\mathbf{S}_k} (\mathbf{J}^k - \nabla \mathbf{F}(x^*)) \mathbf{\Pi}_{\mathbf{S}_k} e - \mathbf{J}^k e \right\|_2^2}_{b^k}. \tag{65}
$$

In view of Assumption 3.1 (combine (43) and (44)), we have

$$
\mathbb{E}_{\mathcal{D}} \left[a^k \right] \leq 4\mathcal{L}_1 (f(x^k) - f(x^*)), \tag{66}
$$

where the expectation is taken with respect to $\mathbf{S}_k$. Let us now bound $\mathbb{E}_{\mathcal{D}} \left[b^k \right]$. Using the fact that $\nabla \mathbf{F}(x^*)e = 0$, we can write

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D}} \left[b^k \right] &= \frac{2}{n^2} \mathbb{E}_{\mathcal{D}} \left[\left\| (\mathbf{J}^k - \nabla \mathbf{F}(x^*))\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} e - (\mathbf{J}^k - \nabla \mathbf{F}(x^*))e \right\|_2^2 \right] \\
&= \frac{2}{n^2} \mathbb{E}_{\mathcal{D}} \left[\left\| (\mathbf{J}^k - \nabla \mathbf{F}(x^*))(\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})e \right\|_2^2 \right] \\
&= \frac{2}{n^2} \mathbb{E}_{\mathcal{D}} \left[e^\top (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})^\top (\mathbf{J}^k - \nabla \mathbf{F}(x^*))^\top (\mathbf{J}^k - \nabla \mathbf{F}(x^*))(\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})e \right] \\
&= \frac{2}{n^2} \mathbb{E}_{\mathcal{D}} \left[\mathrm{Tr} \left(e^\top (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})^\top (\mathbf{J}^k - \nabla \mathbf{F}(x^*))^\top (\mathbf{J}^k - \nabla \mathbf{F}(x^*))(\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})e \right) \right] \\
&= \frac{2}{n^2} \mathbb{E}_{\mathcal{D}} \left[\mathrm{Tr} \left(e^\top (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})^\top \mathbf{W}^{1/2} \mathbf{W}^{-1/2} (\mathbf{J}^k - \nabla \mathbf{F}(x^*))^\top \right. \right. \\
&\quad \left. \left. (\mathbf{J}^k - \nabla \mathbf{F}(x^*)) \mathbf{W}^{-1/2} \mathbf{W}^{1/2} (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})e \right) \right] \\
&= \frac{2}{n^2} \mathbb{E}_{\mathcal{D}} \left[\mathrm{Tr} \left(\mathbf{W}^{-1/2} (\mathbf{J}^k - \nabla \mathbf{F}(x^*))^\top (\mathbf{J}^k - \nabla \mathbf{F}(x^*)) \mathbf{W}^{-1/2} \mathbf{W}^{1/2} (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I}) e e^\top \right. \right. \\
&\quad \left. \left. (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})^\top \mathbf{W}^{1/2} \right) \right] \\
&= \frac{2}{n^2} \mathrm{Tr} \left(\mathbf{W}^{-1/2} (\mathbf{J}^k - \nabla \mathbf{F}(x^*))^\top (\mathbf{J}^k - \nabla \mathbf{F}(x^*)) \mathbf{W}^{-1/2} \mathbb{E}_{\mathcal{D}} \left[\mathbf{W}^{1/2} (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I}) e e^\top \right. \right. \\
&\quad \left. \left. (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})^\top \mathbf{W}^{1/2} \right] \right).
\end{aligned}
$$

If we now let $v = \mathbf{W}^{1/2} (\theta_{\mathbf{S}_k} \mathbf{\Pi}_{\mathbf{S}_k} - \mathbf{I})e$ and $\mathbf{M} = (\mathbf{J}^k - \nabla \mathbf{F}(x^*)) \mathbf{W}^{-1/2}$, then we can continue:

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D}} \left[b^k \right] &= \frac{2}{n^2} \mathrm{Tr} \left(\mathbf{M}^\top \mathbf{M} \mathbb{E}_{\mathcal{D}} \left[vv^\top \right] \right) \leq \frac{2}{n^2} \mathrm{Tr} \left(\mathbf{M}^\top \mathbf{M} \right) \lambda_{\max} \left(\mathbb{E}_{\mathcal{D}} \left[vv^\top \right] \right) \\
&\overset{(10)}{=} \frac{2}{n^2} \left\| \mathbf{J}^k - \nabla \mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2 \lambda_{\max} \left(\mathbb{E}_{\mathcal{D}} \left[vv^\top \right] \right) \\
&= \frac{2\rho}{n^2} \left\| \mathbf{J}^k - \nabla \mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2, \tag{67}
\end{aligned}
$$

where in the last step we have used the assumption that $\theta_{\mathbf{S}_k}$ is bias-correcting:

$$\lambda_{\max}\left(\mathbb{E}_{\mathcal{D}}\left[vv^\top\right]\right) \overset{(21)}{=} \lambda_{\max}\left(\mathbf{W}^{1/2}\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}_k}^2\,\mathbf{\Pi}_{\mathbf{S}_k}ee^\top\mathbf{\Pi}_{\mathbf{S}_k}^\top\right]\mathbf{W}^{1/2} - \mathbf{W}^{1/2}ee^\top\mathbf{W}^{1/2}\right)$$
$$\overset{(51)}{=} \rho. \tag{68}$$

It now only remains to substitute (66) and (67) into (65) to arrive at (64). $\qquad\square$

3.6 Proof of Theorem 1

With the help of the above lemmas, we now proceed to the proof of the theorem. In view of (50), we have

$$\langle \nabla f(y), y - x^* \rangle \ge f(y) - f(x^*) + \frac{\mu}{2}\left\| y - x^* \right\|_2^2. \tag{69}$$

By using the relationship $x^{k+1} = x^k - \alpha g^k$, the fact that g^k is an unbiased estimate of the gradient $\nabla f(x^k)$, and using one-point strong convexity (69), we get

$$\mathbb{E}_{\mathcal{D}}\left[\left\| x^{k+1} - x^* \right\|_2^2\right] \overset{(2)}{=} \mathbb{E}_{\mathcal{D}}\left[\left\| x^k - x^* - \alpha g^k \right\|_2^2\right]$$
$$\overset{(33)}{=} \left\| x^k - x^* \right\|_2^2 - 2\alpha\left\langle \nabla f(x^k), x^k - x^* \right\rangle + \alpha^2\mathbb{E}_{\mathcal{D}}\left[\left\| g^k \right\|_2^2\right]$$
$$\overset{(69)}{\le} (1 - \alpha\mu)\left\| x^k - x^* \right\|_2^2 + \alpha^2\mathbb{E}_{\mathcal{D}}\left[\left\| g^k \right\|_2^2\right]$$
$$- 2\alpha(f(x^k) - f(x^*)). \tag{70}$$

Next, applying Lemma 6 leads to the estimate

$$\mathbb{E}_{\mathcal{D}}\left[\left\| x^{k+1} - x^* \right\|_2^2\right] \overset{(64)}{\le} (1 - \alpha\mu)\left\| x^k - x^* \right\|_2^2 + 2\alpha\left(2\alpha\mathcal{L}_1 - 1\right)\left(f(x^k) - f(x^*)\right)$$
$$+ 2\alpha^2\frac{\rho}{n^2}\left\| \mathbf{J}^k - \nabla\mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2. \tag{71}$$

Let $\sigma = 1/(2\mathcal{L}_2)$. Adding $\sigma\alpha\mathbb{E}_{\mathcal{D}}\left[\left\| \mathbf{J}^{k+1} - \nabla\mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2\right]$ to both sides of the above inequality and substituting in the definition of Ψ^k from (52), it follows that

$$\mathbb{E}_{\mathcal{D}}\left[\Psi^{k+1}\right] \overset{(71)}{\le} (1 - \alpha\mu)\left\| x^k - x^* \right\|_2^2 + 2\alpha\left(2\alpha\mathcal{L}_1 - 1\right)\left(f(x^k) - f(x^*)\right)$$
$$+ 2\alpha^2\frac{\rho}{n^2}\left\| \mathbf{J}^k - \nabla\mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2 + \sigma\alpha\mathbb{E}_{\mathcal{D}}\left[\left\| \mathbf{J}^{k+1} - \nabla\mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2\right]$$

$$\overset{\text{(Lemma 5)}}{\leq} (1 - \alpha\mu) \left\| x^k - x^* \right\|_2^2 + 2\alpha \underbrace{(\mathcal{L}_2\sigma + 2\alpha\mathcal{L}_1 - 1)}_{\text{I}}(f(x^k) - f(x^*))$$

$$+ \sigma\alpha \underbrace{\left(1 - \kappa + 2\frac{\alpha\rho}{\sigma n^2}\right)}_{\text{II}} \left\| \mathbf{J}^k - \nabla\mathbf{F}(x^*) \right\|_{\mathbf{W}^{-1}}^2. \tag{72}$$

We now choose α so that I ≤ 0 and II $\leq 1 - \alpha\mu$, which can be written as

$$\alpha \leq \frac{1 - \mathcal{L}_2\sigma}{2\mathcal{L}_1} \quad \text{and} \quad \alpha \leq \frac{\kappa}{2\rho/(\sigma n^2) + \mu}. \tag{73}$$

If α satisfies the above two inequalities, then (72) takes on the simplified form $\mathbb{E}_{\mathcal{D}}\left[\Psi^{k+1}\right] \leq (1 - \alpha\mu)\Psi^k$. By taking expectation again and using the tower rule, we get $\mathbb{E}\left[\Psi^k\right] \leq (1 - \alpha\mu)^k\Psi^0$. Note that as long as $k \geq \frac{1}{\alpha\mu} \log \frac{1}{\epsilon}$, we have $\mathbb{E}\left[\Psi^k\right] \leq \epsilon\Psi^0$. Recalling that $\sigma = 1/(2\mathcal{L}_2)$, and choosing α to be the minimum of the two upper bounds (73) gives the upper bound on (53), which in turn leads to (55). □

4 Minibatch sketches

In this section we focus on special cases of Algorithm 1 where one computes $\nabla f_i(x^k)$ for $i \in S^k$, where S^k is a random subset (mini-batch) of $[n]$ chosen in each iteration according to some fixed probability law. As we have seen in the introduction, this is achieved by choosing $\mathbf{S}_k = \mathbf{I}_{S_k}$.

We say that $\mathbf{S}$ is a *minibatch sketch* if $\mathbf{S} = \mathbf{I}_S$ for some random set (sampling) S, where $\mathbf{I}_S \in \mathbb{R}^{n \times |S|}$ is a column submatrix of the $n \times n$ identity matrix $\mathbf{I}$ associated with columns indexed by the set S. That is, the distribution $\mathcal{D}$ from which the sketches $\mathbf{S}$ are sampled is defined by

$$\mathbb{P}\left[\mathbf{S} = \mathbf{I}_C\right] = p_C, \quad C \subseteq [n],$$

where $\sum_{C \subseteq [n]} p_C = 1$ and $p_C \geq 0$ for all C.

4.1 Samplings

We now formalize the notion of a random set, which we will refer to by the name sampling. A *sampling* is a random set-valued mapping with values being the subsets of $[n]$. A sampling S is uniquely characterized by the probabilities $p_C \overset{\text{def}}{=} \mathbb{P}[S = C]$ associated with every subset C of $[n]$.

Definition 1 (*Types of samplings*) We say that sampling S is non-vacuous if $\mathbb{P}[S = \emptyset] = 0$ (i.e., $p_\emptyset = 0$). Let $p_i \overset{\text{def}}{=} \mathbb{P}[i \in S] = \sum_{C:i \in C} p_C$. We say that S is proper if $p_i > 0$ for all i. We say that S is uniform if $p_i = p_j$ for all i, j. We say that S is τ—uniform if it is uniform and $|S| = \tau$ with probability 1. In particular, the

unique sampling which assigns equal probabilities to all subsets of $[n]$ of cardinality τ and zero probabilities to all other subsets is called the τ—nice sampling.

We refer the reader to [22,25] for a background reading on samplings and their properties.

Definition 2 (*Support*) The support of a sampling S is the set of subsets of $[n]$ which are chosen by S with positive probability: $\text{supp}(S) \overset{\text{def}}{=} \{C : p_C > 0\}$. We say that S has uniform support if

$$c_1 \overset{\text{def}}{=} |\{C \in \text{supp}(S) : i \in C\}| = |\{C \in \text{supp}(S) : j \in C\}|$$

for all $i, j \in [n]$. In such a case we say that the support is c_1—uniform.

To illustrate the above concepts, we now list a few examples with $n = 4$.

Example 7 The sampling defined by setting $p_{\{1,2\}} = p_{\{3,4\}} = 0.5$ is non-vacuous, proper, 2—uniform ($p_i = 0.5$ for all i and $|S| = 2$ with probability 1), and has 1—uniform support. If we change the probabilities to $p_{\{1,2\}} = 0.4$ and $p_{\{3,4\}} = 0.6$, the sampling is no longer uniform (since $p_1 = 0.4 \neq 0.6 = p_3$), but it still has 1—uniform support, is proper and non-vacuous. Hence, a sampling with uniform support need not be uniform. On the other hand, a uniform sampling need not have uniform support. As an example, consider sampling S defined via $p_{\{1\}} = 0.4$, $p_{\{2,3\}} = p_{\{3,4\}} = p_{\{2,4\}} = 0.2$. It is uniform (since $p_i = 0.4$ for all i). However, while element 1 appears in a single set of its support, elements 2, 3 and 4 each appear in two sets. So, this sampling does not have uniform support.

Example 8 A uniform sampling need not be τ—uniform for any τ. For example, the sampling defined by setting $p_{\{1,2,3,4\}} = 0.5$, $p_{\{1,2\}} = 0.25$ and $p_{\{3,4\}} = 0.25$ is uniform (since $p_i = 0.75$ for all i), but as it assigns positive probabilities to sets of at least two different cardinalities, it is not τ—uniform for any τ.

Example 9 Further, the sampling defined by setting $p_{\{1,2\}} = 1/6$, $p_{\{1,3\}} = 1/6$, $p_{\{1,4\}} = 1/6$, $p_{\{2,3\}} = 1/6$, $p_{\{2,4\}} = 1/6$, $p_{\{3,4\}} = 1/6$ is non-vacuous, 2—uniform ($p_i = 1/2$ for all i and $|S| = 2$ with probability 1), and has 3—uniform support. The sampling defined by setting $p_{\{1,2\}} = 1/3$, $p_{\{2,3\}} = 1/3$, $p_{\{3,1\}} = 1/3$ is non-vacuous, proper, 2—uniform ($p_i = 2/3$ for all i and $|S| = 2$ with probability 1) and has 2—uniform support.

Note that a sampling with uniform support is necessarily proper as long as $c_1 > 0$. However, it need not be non-vacuous. For instance, the sampling S defined by setting $p_\emptyset = 1$ has 0—uniform support and is vacuous. From now on, we only consider samplings with the following properties.

Assumption 4.1 S is non-vacuous and has c_1—uniform support with $c_1 \geq 1$.

Note that if S is a non-vacuous sampling with 1—uniform support, then its support is necessary a partition of $[n]$. We shall pay specific attention to such samplings in Sect. 5 as for them we can develop a stronger analysis than that provided by Theorem 1.

4.2 Minibatch sketches and projections

In the next result we describe some basic properties of the projection matrix $\mathbf{\Pi_S} = \mathbf{S}(\mathbf{S}^\top\mathbf{WS})^\dagger\mathbf{S}^\top\mathbf{W}$ associated with a minibatch sketch $\mathbf{S}$.

Lemma 7 *Let* $\mathbf{W} = \mathrm{Diag}(w_1, \dots, w_n)$. *Let S be any sampling,* $\mathbf{S} = \mathbf{I}_S$ *be the associated minibatch sketch, and let* $\mathbf{P}$ *be the probability matrix[11] associated with sampling S:* $\mathbf{P}_{ij} = \mathbb{P}[i \in S \,\&\, j \in S]$. *Then*

  (i) $\mathbf{\Pi_S} = \mathbf{I}_S\mathbf{I}_S^\top$. *This is a diagonal matrix with the ith diagonal element equal to 1 if $i \in S$, and 0 if $i \notin S$.*
 (ii) $\mathbf{\Pi_S}e = e_S \overset{def}{=} \sum_{i\in S} e_i$.
(iii) $\mathbb{E}_{\mathcal{D}}\left[\mathbf{\Pi_S}ee^\top\mathbf{\Pi_S}\right] = \sum_{C\subseteq[n]} p_C e_C e_C^\top = \mathbf{P}$
 (iv) $\mathbb{E}_{\mathcal{D}}\left[\mathbf{\Pi_S}\right] = \mathrm{Diag}(\mathbf{P})$
  (v) *The stochastic contraction number defined in* (48) *is given by* $\kappa = \min_i p_i$
 (vi) *Let S satisfy Assumption* 4.1. *Then the random variable*

$$\theta_{\mathbf{S}} \overset{def}{=} \frac{1}{c_1 p_S}, \tag{74}$$

  defined on $\mathrm{supp}(S)$, *is bias-correcting. That is,* $\mathbb{E}_{\mathcal{D}}\left[\mathbf{\Pi_S}\theta_S e\right] = e$.

Proof (i) This follows by noting that $\mathbf{I}_S^\top\mathbf{WI}_S$ is the $|S|\times|S|$ diagonal matrix with diagonal entries corresponding to w_i for $i \in S$, which in turn can be used to show that $(\mathbf{I}_S^\top\mathbf{WI}_S)^{-1}\mathbf{I}_S^\top\mathbf{W} = \mathbf{I}_S^\top$.

 (ii) This follows from (i) by noting that $\mathbf{I}_S^\top e$ is the vector of all ones in $\mathbb{R}^{|S|}$.

(iii) Using (ii), we have $\mathbf{\Pi_S}ee^\top\mathbf{\Pi_S} = e_S e_S^\top$. By linearity of expectation, $\left(\mathbb{E}_{\mathcal{D}}\left[e_S e_S^\top\right]\right)_{ij} = \mathbb{E}_{\mathcal{D}}\left[(e_S e_S^\top)_{ij}\right] = \mathbb{E}_{\mathcal{D}}\left[1_{i,j\in S}\right] = \mathbb{P}[i \in S \,\&\, j \in S] = \mathbf{P}_{ij}$, where $1_{i,j\in S} = 1$ if $i,j \in S$ and $1_{i,j\in S} = 0$ otherwise.

 (iv) This follows from (i) by taking expectations of the diagonal elements of $\mathbf{\Pi_S}$.

  (v) Follows from (iv).

 (vi) Indeed,

$$\mathbb{E}_{\mathcal{D}}\left[\theta_{\mathbf{S}}\mathbf{\Pi_S}e\right] \overset{\text{(ii)}}{=} \sum_{C\in\mathrm{supp}(S)} p_C\theta_C e_C \overset{\text{(74)}}{=} \frac{1}{c_1}\sum_{C\in\mathrm{supp}(S)} e_C = e, \tag{75}$$

where the last equation follows from the assumption that the support of S is c_1—uniform. $\qquad\square$

The following simple observation will be useful in the computation of the constant $\mathcal{L}_1$. The proof is straightforward and involves a double counting argument.

Lemma 8 *Let S be a sampling satisfying Assumption* 4.1. *Moreover, assume that S is a τ—uniform sampling. Then* $\frac{|\mathrm{supp}(S)|}{c_1} = \frac{n}{\tau}$. *Consequently,* $\kappa = p_1 = p_2 = \cdots = p_n = \frac{\tau}{n} = \frac{c_1}{|\mathrm{supp}(S)|}$, *where κ is the stochastic contraction number associated with the minibatch sketch* $\mathbf{S} = \mathbf{I}_S$.

---

[11] The notion of a probability matrix associated with a sampling was first introduced in [25] in the context of parallel coordinate descent methods, and further studied in [22].

4.3 JacSketch for minibatch sampling = minibatch SAGA

As we have mentioned in Sect. 1.4 already, JacSketch admits a particularly simple form for minibatch sketches, and corresponds to known and new variants of SAGA. Assume that S satisfies Assumption 4.1 and let $\mathbf{W} = \text{Diag}(w_1, \dots, w_n)$. In view of Lemma 7(vi), this means that the random variable $\theta_\mathbf{S} = \frac{1}{c_1 p_S}$ is bias-correcting, and due to Lemma 7(ii), we have $\mathbf{\Pi}_{\mathbf{S}_k} e = e_{S_k} = \sum_{i \in S_k} e_i$. Therefore,

$$g^k \overset{(13)}{=} \frac{1}{n} \mathbf{J}^k e + \frac{\theta_{\mathbf{S}_k}}{n} \sum_{i \in S_k} (\nabla f_i(x^k) - \mathbf{J}^k_{:i})$$

$$= \frac{1}{n} \left(\sum_{i \notin S_k} \mathbf{J}^k_{:i} + \sum_{i \in S_k} \left(1 - \frac{1}{c_1 p_{S_k}}\right) \mathbf{J}^k_{:i} + \frac{1}{c_1 p_{S_k}} \nabla f_i(x^k) \right). \tag{76}$$

By Lemma 7(i), $\mathbf{\Pi}_{\mathbf{S}_k} = \mathbf{I}_{S_k} \mathbf{I}_{S_k}^\top$. In view of (11), the Jacobian estimate gets updated as follows

$$\mathbf{J}^{k+1}_{:i} = \begin{cases} \mathbf{J}^k_{:i} & i \notin S_k, \\ \nabla f_i(x^k) & i \in S_k. \end{cases} \tag{77}$$

The resulting minibatch SAGA method is formalized as Algorithm 2.

Algorithm 2 JacSketch: Mini-batch SAGA

1: **Parameters:** Sampling S satisfying Assumption 4.1, $\mathbf{W} = \text{Diag}(w_1, \dots, w_n)$, stepsize $\alpha > 0$
2: **Initialization:** Choose $x^0 \in \mathbb{R}^d$, $\mathbf{J}^0 \in \mathbb{R}^{d \times n}$ ▷ Initialization
3: **for** $k = 0, 1, 2, \dots$ **do**
4:     Sample a fresh set $S_k \sim S$
5:     $g^k = \frac{1}{n} \mathbf{J}^k e + \frac{1}{n c_1 p_{S_k}} \sum_{i \in S_k} (\nabla f_i(x^k) - \mathbf{J}^k_{:i})$ ▷ Update gradient estimate
6:     $\mathbf{J}^{k+1}_{:i} = \begin{cases} \mathbf{J}^k_{:i} & i \notin S_k, \\ \nabla f_i(x^k) & i \in S_k. \end{cases}$ ▷ Update Jacobian estimate
7:     $x^{k+1} = x^k - \alpha g^k$ ▷ Take a step

Below we specialize the formula for g^k to a few interesting special cases.

Example 10 (Standard SAGA) Standard uniform SAGA is obtained by setting $S_k = \{i\}$ with probability $1/n$ for each $i \in [n]$. Since the support of this sampling is 1—uniform, we set $c_1 = 1$. This leads to the gradient estimate

$$g^k = \frac{1}{n} \mathbf{J}^k e + \nabla f_i(x^k) - \mathbf{J}^k_{:i}. \tag{78}$$

Example 11 (Non-uniform SAGA) However, we can use non-uniform probabilities instead. Let $S_k = \{i\}$ with probability $p_i > 0$ for each $i \in [n]$. Since the support of this sampling is 1—uniform, we have $c_1 = 1$. So, the gradient estimate has the form

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{np_i}(\nabla f_i(x^k) - \mathbf{J}^k_{:i}). \tag{79}$$

Example 12 (Uniform minibatch SAGA, version 1) Let $C_1, \ldots, C_q$ be nonempty subsets of forming a partition $[n]$. Let $S_k = C_j$ with probability $p_{C_j} > 0$. The support of this sampling is 1—uniform, and hence we can choose $c_1 = 1$. This leads to the gradient estimate

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{np_{C_j}} \sum_{i \in C_j} (\nabla f_i(x^k) - \mathbf{J}^k_{:i}).$$

Example 13 (Uniform minibatch SAGA, version 2) Let S_k be chosen uniformly at random from all subsets of $[n]$ of cardinality $\tau \geq 2$. That is, $\mathbf{S}_k$ is the τ-nice sampling, and the probabilities are equal to $p_{S_k} = 1/\binom{n}{\tau}$. This sampling has c_1—uniform support with $c_1 = \binom{n-1}{\tau-1} = \frac{\tau}{n}\binom{n}{\tau}$. Thus, $nc_1 p_{S_k} = \tau$, and we have

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{\tau} \sum_{i \in S_k} (\nabla f_i(x^k) - \mathbf{J}^k_{:i}). \tag{80}$$

Example 14 (Gradient descent) Consider the same situation as in Example 13, but with $\tau = n$. That is, we choose $S_k = [n]$ with probability 1, and $c_1 = 1$. Then

$$g^k = \frac{1}{n}\mathbf{J}^k e + \frac{1}{n} \sum_{i=1}^{n} (\nabla f_i(x^k) - \mathbf{J}^k_{:i}) = \nabla f(x^k).$$

4.4 Expected smoothness constants $\mathcal{L}_1$ and $\mathcal{L}_2$

Here we compute the expected smoothness constants $\mathcal{L}_1$ and $\mathcal{L}_2$ in the case of $\mathbf{S}$ being a minibatch sketch $\mathbf{S} = \mathbf{I}_S$, and assuming that f is convex and smooth. We first formalize the notion of smoothness we will use.

Assumption 4.2 For $\emptyset \neq C \subseteq [n]$ define

$$f_C(x) \overset{\text{def}}{=} \frac{1}{|C|} \sum_{i \in C} f_i(x). \tag{81}$$

For each $\emptyset \neq C \subseteq [n]$ and all $x \in \mathbb{R}^d$, the function f_C is L_C—smooth and convex. That is, there exists $L_C \geq 0$ such that the following inequality holds

$$\|\nabla f_C(x) - \nabla f_C(x^*)\|_2^2 \\ \leq 2L_C\left(f_C(x) - f_C(x^*) - \langle \nabla f_C(x^*), x - x^*\rangle\right), \quad \forall x \in \mathbb{R}^d. \tag{82}$$

Let $L_i = L_{\{i\}}$ for $i \in [n]$.

The above assumption is somewhat non-standard. Note that, however, if we instead assume that each f_i is convex and L_i-smooth, then the above assumption holds for $L_C = \frac{1}{|C|} \sum_{i \in C} L_i$. In some cases, however, we may have better estimates of the constants L_C than those provided by the averages of the L_i values. The value of these constants will have a direct influence on $\mathcal{L}_1$ and $\mathcal{L}_2$, which is why we work with this more refined assumption instead.

Lemma 9 (Smoothness of the Jacobian) *Assume that f_i is convex and L_i—smooth for all $i \in [n]$. Define $L_{\max} \overset{def}{=} \max_i L_i$ and $\mathbf{D}_L \overset{def}{=} \mathrm{Diag}(L_1, \ldots, L_n) \in \mathbb{R}^{n \times n}$. Then*

$$\left\| \nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*) \right\|_{\mathbf{D}_L^{-1}}^2 \leq 2n(f(x) - f(x^*)), \quad \forall x \in \mathbb{R}^d. \tag{83}$$

Proof Indeed,

$$\left\| \nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*) \right\|_{\mathbf{D}_L^{-1}}^2$$

$$\overset{(10)}{=} \left\| (\nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*)) \mathbf{D}_L^{-1/2} \right\|^2 \overset{(10)}{=} \sum_{i=1}^n \frac{1}{L_i} \left\| \nabla f_i(x) - \nabla f_i(x^*) \right\|_2^2$$

$$\leq 2 \sum_{i=1}^n (f_i(x) - f_i(x^*) - \langle \nabla f_i(x^*), x - x^* \rangle) \overset{(1)}{=} 2n(f(x) - f(x^*)),$$

where in the last step we used the fact that $\sum_{i=1}^n \nabla f_i(x^*) = n \nabla f(x^*) = 0$. □

Theorem 2 (Expected smoothness) *Let $\mathbf{S} = \mathbf{I}_S$ be a minibatch sketch where S is a sampling satisfying Assumption 4.1 (in particular, the support of S is c_1—uniform). Consider the bias-correcting random variable $\theta_{\mathbf{S}}$ given in (74). Further, let f satisfy Assumption 4.2. Then the expected smoothness assumptions (Assumptions 3.1 and 3.2) are satisfied with constants $\mathcal{L}_1$ and $\mathcal{L}_2$ given by[12]*

$$\mathcal{L}_1 = \frac{1}{nc_1^2} \max_i \left\{ \sum_{C \in \mathrm{supp}(S) \,:\, i \in C} \frac{|C| L_C}{p_C} \right\}, \quad \mathcal{L}_2 = n \max_i \left\{ \frac{p_i L_i}{w_i} \right\}, \tag{84}$$

where $L_i = L_{\{i\}}$. If moreover, S is τ—nice sampling, then[13]

$$\mathcal{L}_1 = L_{\max}^{\mathcal{G}} \overset{def}{=} \max_i \left\{ \frac{1}{c_1} \sum_{C \in \mathrm{supp}(S) \,:\, i \in C} L_C \right\}, \quad \mathcal{L}_2 = \tau \max_i \left\{ \frac{L_i}{w_i} \right\}. \tag{85}$$

---

[12] Recall that $p_i = \mathbb{P}[i \in S]$ for $i \in [n]$, $p_C = \mathbb{P}[S = C]$ for $C \subseteq [n]$ and $\mathbf{W} = \mathrm{Diag}(w_1, \ldots, w_n) \succ 0$.

[13] Note that $c_1 = |\{C \in \mathrm{supp}(S) \,:\, 1 \in C\}|$, and hence $\mathcal{L}_1$ has the form of a maximum over averages.

Proof Let $\mathbf{R} = \nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*)$ and $A = \mathbb{E}_{\mathcal{D}}\left[\|\nabla f_{\mathbf{S}}(x) - \nabla f_{\mathbf{S}}(x^*)\|_2^2\right]$. Then

$$A \overset{(44)}{=} \mathbb{E}_{\mathcal{D}}\left[\frac{\theta_{\mathbf{S}}^2}{n^2}\|\mathbf{R}\mathbf{\Pi}_{\mathbf{S}}e\|_2^2\right] \overset{(74)}{=} \sum_{C \in \text{supp}(S)} \frac{p_C}{c_1^2 p_C^2 n^2}\|\mathbf{R}\mathbf{\Pi}_{I_C}e\|_2^2$$

$$= \sum_{C \in \text{supp}(S)} \frac{1}{c_1^2 p_C n^2}\text{Tr}\left(e^\top \mathbf{\Pi}_{I_C}^\top \mathbf{R}^\top \mathbf{R}\mathbf{\Pi}_{I_C}e\right)$$

$$= \sum_{C \in \text{supp}(S)} \frac{1}{c_1^2 p_C n^2}\text{Tr}\left(\mathbf{R}^\top \mathbf{R}\mathbf{\Pi}_{I_C}ee^\top \mathbf{\Pi}_{I_C}^\top\right)$$

$$\overset{\text{Lem } 7\text{(iii)}}{=} \sum_{C \in \text{supp}(S)} \frac{1}{c_1^2 p_C n^2}\text{Tr}\left(\mathbf{R}^\top \mathbf{R}e_C e_C^\top\right)$$

$$= \sum_{C \in \text{supp}(S)} \frac{1}{c_1^2 p_C n^2}\left\|(\nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*))e_C\right\|_2^2$$

$$= \sum_{C \in \text{supp}(S)} \frac{|C|^2}{c_1^2 p_C n^2}\left\|\nabla f_C(x) - \nabla f_C(x^*)\right\|_2^2.$$

Using (82) and (81), we can continue:

$$A \overset{(82)}{\le} \sum_{C \in \text{supp}(S)} \frac{2L_C|C|^2}{c_1^2 p_C n^2}(f_C(x) - f_C(x^*) - \langle\nabla f_C(x^*), x - x^*\rangle)$$

$$\overset{(81)}{=} \frac{2}{c_1^2 n^2}\sum_{C \in \text{supp}(S)} \frac{L_C|C|^2}{p_C}\frac{1}{|C|}\sum_{i \in C}(f_i(x) - f_i(x^*) - \langle\nabla f_i(x^*), x - x^*\rangle)$$

$$= \frac{2}{c_1^2 n^2}\sum_{C \in \text{supp}(S)}\sum_{i \in C}(f_i(x) - f_i(x^*) - \langle\nabla f_i(x^*), x - x^*\rangle)\frac{L_C|C|}{p_C}$$

$$= \frac{2}{c_1^2 n^2}\sum_{i=1}^{n}\sum_{C \in \text{supp}(S)\,:\,i \in C}(f_i(x) - f_i(x^*) - \langle\nabla f_i(x^*), x - x^*\rangle)\frac{L_C|C|}{p_C}$$

$$= \frac{2}{c_1^2 n^2}\sum_{i=1}^{n}(f_i(x) - f_i(x^*) - \langle\nabla f_i(x^*), x - x^*\rangle)\sum_{C \in \text{supp}(S)\,:\,i \in C}\frac{L_C|C|}{p_C}$$

$$\le \frac{2}{c_1^2 n}\max_{i}\left\{\sum_{C \in \text{supp}(S)\,:\,i \in C}\frac{L_C|C|}{p_C}\right\}\frac{1}{n}\sum_{i=1}^{n}(f_i(x) - f_i(x^*)$$

$$- \langle\nabla f_i(x^*), x - x^*\rangle), \tag{86}$$

where in this last inequality we have used convexity of f_i for $i \in [n]$. Since

$$\frac{1}{n}\sum_{i=1}^{n}\left(f_i(x) - f_i(x^*) - \langle\nabla f_i(x^*), x - x^*\rangle\right)$$

$$= f(x) - f(x^*) - \langle \nabla f(x^*), x - x^* \rangle = f(x) - f(x^*),$$

the formula for $\mathcal{L}_1$ now follows by comparing (86) to (43). In order to establish the formula for $\mathcal{L}_2$, we estimate

$$
\begin{aligned}
\mathbb{E}_{\mathcal{D}} \left[\| \mathbf{R}\mathbf{\Pi_S} \|_{\mathbf{W}^{-1}}^2 \right] &\overset{(10)}{=} \mathbb{E}_{\mathcal{D}} \left[\left\| \mathbf{R}\mathbf{\Pi_S}\mathbf{W}^{-1/2} \right\|_{\mathbf{I}}^2 \right] \overset{(10)}{=} \mathrm{Tr}\left(\mathbf{R}^\top \mathbf{R} \mathbb{E}_{\mathcal{D}} \left[\mathbf{\Pi_S} \mathbf{W}^{-1} \mathbf{\Pi_S}^\top \right] \right) \\
&\overset{(57)}{=} \mathrm{Tr}\left(\mathbf{R}^\top \mathbf{R} \mathbb{E}_{\mathcal{D}} \left[\mathbf{H_S} \right] \right) \\
&= \mathrm{Tr}\left(\mathbf{D}_L^{-1/2} \mathbf{R}^\top \mathbf{R} \mathbf{D}_L^{-1/2} \mathbf{D}_L^{1/2} \mathbb{E}_{\mathcal{D}} \left[\mathbf{H_S} \right] \mathbf{D}_L^{1/2} \right) \\
&\leq \| \mathbf{R} \|_{\mathbf{D}_L^{-1}}^2 \, \lambda_{\max}\left(\mathbf{D}_L^{1/2} \mathbb{E}_{\mathcal{D}} \left[\mathbf{H_S} \right] \mathbf{D}_L^{1/2} \right) \\
&\overset{(83)}{\leq} 2n\lambda_{\max}\left(\mathbf{D}_L^{1/2} \mathbb{E}_{\mathcal{D}} \left[\mathbf{H_S} \right] \mathbf{D}_L^{1/2} \right) (f(x^k) - f(x^*)).
\end{aligned}
\tag{87}
$$

From Lemma 7(iv) we have $\mathbb{E}_{\mathcal{D}} \left[\mathbf{H_S} \right] = \mathbb{E}_{\mathcal{D}} \left[\mathbf{\Pi_S} \right] \mathbf{W}^{-1} = \mathbf{P}\mathbf{W}^{-1} = \mathrm{Diag}\left(\frac{p_1}{w_1}, \ldots, \frac{p_n}{w_n} \right)$, and hence $\mathbf{D}_L^{1/2} \mathbb{E}_{\mathcal{D}} \left[\mathbf{H_S} \right] \mathbf{D}_L^{1/2} = \mathrm{Diag}\left(\frac{p_1 L_1}{w_1}, \ldots, \frac{p_n L_n}{w_n} \right)$. Comparing to the definition of $\mathcal{L}_2$ in (45) to (87), we conclude that

$$
\mathcal{L}_2 = n\lambda_{\max}\left(\mathbf{D}_L^{1/2} \mathbf{P}\mathbf{W}^{-1} \mathbf{D}_L^{1/2} \right) = n \max_i \left\{ \frac{p_i L_i}{w_i} \right\}.
$$

The specialized formulas (85) for τ—nice sampling follow as special cases of the general formulas (84) since $\frac{|C|}{p_C} = \frac{1}{\tau}\binom{n}{\tau} = \frac{n!}{(\tau-1)!(n-\tau)!} = n\binom{n-1}{\tau-1} = nc_1$ and $p_i = \tau/n$ for all i. $\qquad \square$

In the next result we establish some inequalities relating the quantities L, $L_{\max}$, L_C and $L_{\max}^{\mathcal{G}}$. In particular, the results says that for a certain family of samplings S (the same for which we have defined the quantity $L_{\max}^{\mathcal{G}}$ in (85)), the expected smoothed constant $L_{\max}^{\mathcal{G}}$ is lower-bounded by the average of L_C over $C \in \mathcal{G} = \mathrm{supp}(S)$, and upper-bounded by $L_{\max}$.

Theorem 3 *Let S be a τ—uniform sampling ($\tau \geq 1$) with c_1—uniform support ($c_1 \geq 1$). Let $\mathcal{G} = \mathrm{supp}(S)$. Then*

$$
f(x) = \frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} f_C(x).
\tag{88}
$$

Moreover,

$$
L \leq \frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} L_C \leq L_{\max}^{\mathcal{G}} \leq L_{\max}.
\tag{89}
$$

The last inequality holds without the need to assume τ—uniformity.

Proof Using the fact that S has c_1—uniform support, and utilizing a double-counting argument, we observe that $\sum_{C \in \mathcal{G}} |C| f_C(x) = c_1 \sum_{i=1}^n f_i(x)$. Multiplying both sides by $\frac{1}{nc_1}$, and since $|C| = \tau$ for all $C \in \mathcal{G}$, we get $\frac{\tau |\mathcal{G}|}{c_1 n} \frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} f_C(x) = \frac{1}{n} \sum_{i=1}^n f_i(x) = f(x)$. To obtain (88), it now only remains to use the identity

$$\frac{\tau |\mathcal{G}|}{c_1 n} = 1 \tag{90}$$

which was shown in Lemma 8. The first inequality in (89) follows from (88) using standard arguments (identical to those that lead to the inequality $L \leq \bar{L}$).

Let us now establish the second inequality in (89). Define $L_i^{\mathcal{G}} \stackrel{\text{def}}{=} \frac{1}{c_1} \sum_{C \in \mathcal{G} : i \in C} L_C$. Again using a double-counting argument we observe that $\tau \sum_{C \in \mathcal{G}} L_C = c_1 \sum_{i=1}^n L_i^{\mathcal{G}}$. Multiplying both sides of this equality by $\frac{|\mathcal{G}|}{c_1 n}$ and using identity (90), we get $\frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} L_C = \frac{1}{n} \sum_{i=1}^n L_i^{\mathcal{G}} \leq \max_i L_i^{\mathcal{G}} = L_{\max}^{\mathcal{G}}$. We will now establish the last inequality by proving that $L_i^{\mathcal{G}} \leq L_{\max}$ for any i:

$$
\begin{aligned}
L_i^{\mathcal{G}} &= \frac{1}{c_1} \sum_{C \in \mathcal{G} : i \in C} L_C \leq \frac{1}{c_1} \sum_{C \in \mathcal{G} : i \in C} \frac{1}{|C|} \sum_{i \in C} L_i \\
&\leq \frac{1}{c_1} \sum_{C \in \mathcal{G} : i \in C} \frac{1}{|C|} \sum_{i \in C} L_{\max} \\
&= L_{\max} \frac{1}{c_1} \sum_{C \in \mathcal{G} : i \in C} \underbrace{\frac{1}{|C|} \sum_{i \in C} 1}_{=1} \leq L_{\max} \frac{1}{c_1} \underbrace{\sum_{C \in \mathcal{G} : i \in C} 1}_{=1} \leq L_{\max}.
\end{aligned}
$$

Note that we did not need to assume τ—uniformity to prove that $L_{\max}^{\mathcal{G}} \leq L_{\max}$. $\square$

4.5 Estimating the sketch residual ρ

In this section we compute the sketch residual ρ for several classes of samplings S. Let $\mathcal{G} = \text{supp}(S)$. We will assume throughout this section that S is non-vacuous, has c_1—uniform support (with $c_1 \geq 1$), and is τ—uniform.

Further, we assume that $\mathbf{W} = \text{Diag}(w_1, \ldots, w_n)$, and that the bias-correcting random variable θ_S is chosen as $\theta_S = \frac{1}{c_1 p_S} = \frac{|\mathcal{G}|}{c_1}$ (see (75) and Lemma 8). In view of the above, since $\mathbf{\Pi}_{\mathbf{I}_C} e = e_C$, the sketch residual is given by

$$
\rho \overset{(51)}{=} \lambda_{\max} \left(\mathbf{W}^{1/2} \left(\frac{|\mathcal{G}|^2}{c_1^2} \mathbb{E}_{\mathcal{D}} \left[\mathbf{\Pi}_S e e^\top \mathbf{\Pi}_S \right] - e e^\top \right) \mathbf{W}^{1/2} \right)
$$

$$
= \lambda_{\max} \left(\mathbf{W}^{1/2} \left(\frac{|\mathcal{G}|}{c_1^2} \sum_{C \in \mathcal{G}} e_C e_C^\top - e e^\top \right) \mathbf{W}^{1/2} \right)
$$

$$= \lambda_{\max} \left(\left(\frac{|\mathcal{G}|}{c_1^2} \sum_{C \in \mathcal{G}} e_C e_C^\top - e e^\top \right) \mathbf{W} \right), \tag{91}$$

where the last equality follows by permuting the multiplication of matrices within the $\lambda_{\max}$.

In the following text we calculate upper bounds for ρ for τ—partition and τ—nice samplings. Note that Theorem 1 still holds if we use an upper bound of ρ in place of ρ.

Theorem 4 *If S is the τ—partition sampling, then*

$$\rho \leq \frac{n}{\tau} \max_{C \in \mathcal{G}} \sum_{i \in C} w_i. \tag{92}$$

Proof Using Lemma 8, and since $c_1 = 1$, we get $\frac{|\mathcal{G}|}{c_1^2} = \frac{n}{\tau}$. Consequently,

$$\rho \overset{(91)}{\leq} \frac{n}{\tau} \lambda_{\max} \left(\sum_{C \in \mathcal{G}} e_C e_C^\top \mathbf{W} \right) = \frac{n}{\tau} \lambda_{\max} \left(\sum_{C \in \mathcal{G}} e_C w_C^\top \right), \tag{93}$$

where $w_C = \sum_{i \in C} w_i e_i$ and we used that $-\mathbf{W}^{1/2} e e^\top \mathbf{W}^{1/2}$ is negative semidefinite. When $\mathbf{W} = \mathbf{I}$, the above bound is tight. By Gershgorin's theorem, every eigenvalue λ of the matrix is bounded by at least one of the inequalities $\lambda \leq \sum_{i \in C} w_i$ for $C \in \mathcal{G}$. Consequently, from (93) we have that $\rho \leq \frac{n}{\tau} \max_{C \in \mathcal{G}} \sum_{i \in C} w_i$. □

Next we give an useful upper bound on ρ for a large family of uniform samplings (for proof, see "Appendix C").

Theorem 5 *Let $\mathcal{G}$ be a collection of subsets of $[n]$ with the property that the number of sets $C \in \mathcal{G}$ containing distinct elements $i, j \in [n]$ is the same for all i, j. In particular, define*

$$c_2 \overset{def}{=} |\{C : \{1, 2\} \subseteq C, C \in \mathcal{G}\}|. \tag{94}$$

Now define a sampling S by setting $S = C \in \mathcal{G}$ with probability $\frac{1}{|\mathcal{G}|}$. Moreover, assume that the support of S is c_1—uniform. Consider the minibatch sketch $\mathbf{S} = \mathbf{I}_S$.

(i) *If $\mathbf{W} = \text{Diag}(w_1, \ldots, w_n)$, then*

$$\rho \leq \max_{i=1,\ldots,n} \left\{ \left(\frac{|\mathcal{G}|}{c_1} - 1 \right) w_i + \sum_{j \neq i} w_j \left| \frac{|\mathcal{G}| c_2}{c_1^2} - 1 \right| \right\}. \tag{95}$$

(ii) *If $\mathbf{W} = \mathbf{I}$, then*

$$\rho = \max \left\{ \frac{|\mathcal{G}|}{c_1} \left(1 + (n-1)\frac{c_2}{c_1} \right) - n, \; \frac{|\mathcal{G}|}{c_1} \left(1 - \frac{c_2}{c_1} \right) \right\}. \tag{96}$$

Note that as long as $\tau \geq 2$, the τ—nice sampling S satisfies the assumptions of the above theorem. Indeed, $\mathcal{G}$ is the support of S consisting of all subsets of $[n]$ of size τ, $|\mathcal{G}| = \binom{n}{\tau}$, $c_1 = \binom{n-1}{\tau-1}$, and $c_2 = \binom{n-2}{\tau-2}$. As a result, bound (95) simplifies to

$$\rho \leq \left(\frac{n}{\tau} - 1 \right) \max_{i=1,\ldots,n} \left\{ w_i + \frac{1}{n-1} \sum_{j \neq i} w_j \right\}, \tag{97}$$

and (96) simplifies to

$$\rho = \frac{n}{\tau} \frac{n - \tau}{n - 1}. \tag{98}$$

4.6 Calculating the iteration complexity for special cases

In this section we consider minibatch SAGA (Algorithm 2) and calculate its iteration complexity in special cases using Theorem 1 by pulling together the formulas for $\mathcal{L}_1, \mathcal{L}_2, \kappa$ and ρ established in previous sections. In particular, assume S is τ—uniform and has c_1—uniform support with $c_1 \geq 1$. In this case, formula (85) for $\mathcal{L}_1, \mathcal{L}_2$ from Lemma 2 applies and we have $\mathcal{L}_1 = L_{\max}^{\mathcal{G}}$ and $\mathcal{L}_2 = \tau \max_i \left\{ \frac{L_i}{w_i} \right\}$.

Moreover, by Lemma 8, $\kappa = \frac{\tau}{n}$. By Theorem 1, if we use the stepsize

$$\alpha = \min \left\{ \frac{1}{4\mathcal{L}_1}, \frac{\kappa}{4\mathcal{L}_2 \rho/n^2 + \mu} \right\}$$

$$= \frac{1}{4} \min \left\{ \frac{1}{L_{\max}^{\mathcal{G}}}, \frac{1}{\frac{\rho}{n} \max_{j=1,\ldots,n} \left\{ \frac{L_j}{w_j} \right\} + \frac{\mu}{4} \frac{n}{\tau}} \right\}, \tag{99}$$

then the iteration complexity is given by

$$\max \left\{ \frac{4\mathcal{L}_1}{\mu}, \frac{1}{\kappa} + \frac{4\rho \mathcal{L}_2}{\kappa \mu n^2} \right\} \log \left(\frac{1}{\epsilon} \right)$$

$$= \max \left\{ \frac{4L_{\max}^{\mathcal{G}}}{\mu}, \frac{n}{\tau} + \frac{4\rho}{\mu n} \max_i \left\{ \frac{L_i}{w_i} \right\} \right\} \log \left(\frac{1}{\epsilon} \right). \tag{100}$$

Complexity (100) is listed in line 9 of Table 1. The complexities in lines 3, 5 and 10–13 arise as special cases of (100) for specific choices of S:

– In line 3 we have *gradient descent*. This arises for the choice $\mathbf{W} = \mathbf{I}$ and $S = [n]$ with probability 1. In this case, $\tau = n$, $L_{\max}^{\mathcal{G}} = L$ and $\rho = 0$. So, (100) simplifies to $\frac{4L}{\mu} \log \left(\frac{1}{\epsilon} \right)$.
– In line 5 we have *uniform SAGA*. We choose $\mathbf{W} = \mathbf{I}$ and $S = \{i\}$ with probability $1/n$. We have $\tau = 1$ and $L_{\max}^{\mathcal{G}} = L_{\max}$. In view of Theorem 4, $\rho \leq n$. So, (100) simplifies to $\left(n + \frac{4L_{\max}}{\mu} \right) \log \left(\frac{1}{\epsilon} \right)$.

– In line 10 we choose $\mathbf{W} = \mathbf{I}$ and S is the τ-nice sampling. In this case, Theorem 5 says that $\rho = \frac{n}{\tau}\frac{n-\tau}{n-1}$ (see (98)). Therefore, (100) reduces to

$$\max \left\{ \frac{4L^{\mathcal{G}}_{\max}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{(n-1)\tau}\frac{4L_{\max}}{\mu} \right\} \log \left(\frac{1}{\epsilon} \right). \tag{101}$$

– In line 11 we choose $\mathbf{W} = \mathrm{Diag}(L_i)$ and S is the τ-nice sampling. Theorem 5 says that $\rho \leq \frac{n-\tau}{\tau}\left(\frac{n-2}{n-1}L_{\max} + \frac{n}{n-1}\bar{L} \right)$ (see (97)). Therefore, (100) reduces to

$$\max \left\{ \frac{4L^{\mathcal{G}}_{\max}}{\mu}, \frac{n}{\tau} + \frac{n-\tau}{\tau n}\frac{4\left(\frac{n-2}{n-1}L_{\max} + \frac{n}{n-1}\bar{L} \right)}{\mu} \right\} \log \left(\frac{1}{\epsilon} \right). \tag{102}$$

To simplify the above expression, one may further use the bound $\frac{n-2}{n-1}L_{\max} + \frac{n}{n-1}\bar{L} \leq L_{\max} + \bar{L}$. In Table 1 we have listed the complexity in this simplified form.

– In line 12 of Table 1 we let $\mathbf{W} = \mathbf{I}$ and S is the τ-partition sampling. In view of Theorem 4, $\rho \leq \frac{n}{\tau}\tau = n$ and hence (100) reduces to

$$\max \left\{ \frac{4L^{\mathcal{G}}_{\max}}{\mu}, \frac{n}{\tau} + \frac{4L_{\max}}{\mu} \right\} \log \left(\frac{1}{\epsilon} \right). \tag{103}$$

– In line 13 of Table 1 we let $\mathbf{W} = \mathrm{Diag}(L_i)$ and S is the τ-partition sampling. In view of Theorem 4, $\rho \leq \frac{n}{\tau}\max_{C \in \mathcal{G}}\sum_{i \in C} L_i$ and hence (100) reduces to

$$\max \left\{ \frac{4L^{\mathcal{G}}_{\max}}{\mu}, \frac{n}{\tau} + \frac{4\max_{C \in \mathcal{G}}\sum_{i \in C} L_i}{\mu\tau} \right\} \log \left(\frac{1}{\epsilon} \right). \tag{104}$$

Note that the previous bound for $\mathbf{W} = \mathbf{I}$ is better than this bound since $\max_{C \in \mathcal{G}}\sum_{i \in C} L_i \leq \tau L_{\max}$.

4.7 Comparison with previous mini-batch SAGA convergence results

Recently in [14], a method that includes a mini-batch variant of SAGA was proposed. This work is the most closely related to our minibatch SAGA. The methods described in [14] can be cast in our framework. In the language of our paper, in [14] the authors update the Jacobian estimate according to (77), where S_k is sampled according to a uniform probability with $p_i = \tau/n$, for all $i = 1, \ldots, n$. What [14] do differently is that instead of introducing the bias-corecting random variable $\theta_{\mathbf{S}}$ to maintain an unbiased gradient estimate, the gradient estimate is updated using the standard SAGA update (78) and this sampling process is done independently of how S_k is sampled for the Jacobian update. Thus at every iteration a gradient $\nabla f_i(x^k)$ is sampled to compute (78), but is then discarded and not used to update the Jacobian update so as

to maintain the independence between $\mathbf{J}^k$ and g^k. By introducing the bias-correcting random variable $\theta_{\mathbf{S}}$ in our method we avoid the data-hungry strategy used in [14].

The analysis provided in [14] shows that, by choosing the stepsize appropriately, the expectation of a Lyapunov function similar to (52) is less than $\epsilon > 0$ after

$$\frac{1}{2}\left(\frac{n}{\tau} + K + \sqrt{\frac{n^2}{\tau^2} + K^2}\right)\log\left(\frac{1}{\epsilon}\right) \tag{105}$$

iterations, where $K \stackrel{\text{def}}{=} \frac{4L_{\max}}{\mu}$. When $\tau = 1$ this gives an iteration complexity of $O(n + K)\log\frac{1}{\epsilon}$, which is essentially the same complexity as the standard SAGA method. The main issue with this complexity is that it decreases only very modestly as τ increases. In particular, on the extreme end when $\tau = n$, since $K \geq 4$, we can approximate $(1 + K)^2 \approx 1 + K^2$ and the resulting complexity (105) becomes

$$\left(1 + \frac{4L_{\max}}{\mu}\right)\log\left(\frac{1}{\epsilon}\right).$$

Yet we know that $\tau = n$ corresponds to gradient descent, and thus the iteration complexity should be $O(\frac{L}{\mu}\log(1/\epsilon))$, which is what we recover in the analysis of all our mini-batch variants. In Fig. 1a–c in the experiments in Sect. 6 we illustrate how (105) descreases very modestly as τ increases.

5 A refined analysis with a stochastic Lyapunov function

In this section we perform a refined analysis of JacSketch applied with a minibatch sketch $\mathbf{S} = \mathbf{I}_S$ where the sampling S is over partitions of $[n]$ into sets of size τ.[14]

Assumption 5.1 Let $\mathcal{G}$ be a partition of $[n]$ into sets of size τ. Assume that the sampling S picks sets from the partition $\mathcal{G}$ uniformly at random. That is, $p_C \stackrel{\text{def}}{=} \mathbb{P}[S = C]$ for $C \in \mathcal{G} = \text{supp}(S)$. A sampling with these properties is called a τ—partition sampling.

In the terminology introduced in Sect. 4.1, a τ—partition sampling is non-vacuous, proper and τ—uniform. Its support is a partition of $[n]$, and is 1—uniform. It satisfies Assumption 4.1. Restricting our attention to τ—partition samplings will allow us to perform a more in-depth analysis of JacSketch using a *stochastic Lyapunov function*.

One of the key reasons why we restrict our attention to τ-partition samplings is the fact that

$$\mathbf{I}_{C_1}^\top \mathbf{I}_{C_2} = \begin{cases} \mathbf{I} \in \mathbb{R}^{\tau \times \tau}, & C_1 = C_2, \\ 0 \in \mathbb{R}^{\tau \times \tau}, & C_1 \neq C_2, \end{cases} \tag{106}$$

---

[14] This is only possible when n is a multiple of τ.

for $C_1, C_2 \in \mathcal{G}$. Recall from Lemma 7 that if $\mathbf{W} = \mathbf{I}$, then $\mathbf{\Pi}_{\mathbf{I}_C} = \mathbf{I}_C \mathbf{I}_C^\top$. Consequently, for $C_1, C_2 \in \mathcal{G}$ we have

$$C_1 \neq C_2 \Rightarrow \mathbf{\Pi}_{\mathbf{I}_{C_1}} \mathbf{\Pi}_{\mathbf{I}_{C_2}} = 0, \quad C_1 = C_2 \Rightarrow (\mathbf{I} - \mathbf{\Pi}_{\mathbf{I}_{C_1}}) \mathbf{\Pi}_{\mathbf{I}_{C_2}} = 0. \quad (107)$$

This orthogonality property will be fundamental for controlling the convergence of the gradient estimate in Lemma 10.

5.1 Convergence theorem

Recall from (32) that the stochastic gradient of the controlled stochastic reformulation (28) of the original finite-sum problem (1) is given by

$$\nabla f_{\mathbf{I}_S, \mathbf{J}}(x) = \frac{1}{n} \mathbf{J} e + \frac{1}{p_S n} (\nabla \mathbf{F}(x) - \mathbf{J}) \mathbf{\Pi}_{\mathbf{I}_S} e \quad (108)$$

provided that we use the minibatch sketch $\mathbf{S} = \mathbf{I}_S$ and bias-correcting variable $\theta_{\mathbf{S}} = \theta_{\mathbf{I}_S} = 1/p_S$ given by Lemma 7(vi). This object will appear in our Lyapunov function, evaluated at $x = x^*$ and $\mathbf{J} = \mathbf{J}^k$. We are now ready to present the main result of this section.

Theorem 6 (Convergence for minibatch sketches with τ-partition samplings) *Let*

(i) **$\mathbf{S}$** *be a minibatch sketch (i.e., $\mathbf{S} = \mathbf{I}_S$),[15] where S is a τ—partition sampling with support $\mathcal{G} = \mathrm{supp}(S)$.*

(ii) *$f_C \stackrel{\mathrm{def}}{=} \frac{1}{|C|} \sum_{i \in C} f_i$ be L_C—smooth and μ—strongly convex (for $\mu > 0$) for all $C \in \mathcal{G}$.*

(iii) *$\mathbf{W} = \mathbf{I}$, $\theta_{\mathbf{S}} = \frac{1}{p_S}$.*

(iv) *$\{x^k, \mathbf{J}^k\}$ be the iterates produced by JacSketch.*

Consider the stochastic Lyapunov function

$$\Psi_S^k \stackrel{\mathrm{def}}{=} \left\| x^k - x^* \right\|_2^2 + 2\sigma_S \alpha \left\| \frac{1}{n} \mathbf{J}^k e - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2, \quad (109)$$

where $\sigma_S = \frac{n}{4\tau L_S}$ is a stochastic Lyapunov constant. If we use a stepsize that satisfies

$$\alpha \leq \min_{C \in \mathcal{G}} \frac{p_C}{\mu + \frac{4L_C \tau}{n}}, \quad (110)$$

then

$$\mathbb{E}\left[\Psi_S^k \right] \leq (1 - \mu\alpha)^k \cdot \mathbb{E}\left[\Psi_S^0 \right]. \quad (111)$$

---

[15] We can alternatively set $\mathbf{S} = e_S$ and the same results will hold.

This means that if we choose the stepsize equal to the upper bound (110), *then*

$$k \geq \max_{C \in \mathcal{G}} \left\{ \frac{1}{p_C} + \frac{4L_C}{\mu} \frac{\tau}{np_C} \right\} \log \left(\frac{1}{\epsilon} \right) \Rightarrow \mathbb{E} \left[\Psi_S^k \right] \leq \epsilon \cdot \mathbb{E} \left[\Psi_S^0 \right]. \qquad (112)$$

5.2 Gradient estimate contraction

Here we will show that our gradient estimate contracts in the following sense.

Lemma 10 *Let S be the τ—partition sampling, and $\sigma(S) \stackrel{def}{=} \sigma_S \geq 0$ be any non-negative random variable. Then*

$$\mathbb{E} \left[\sigma_S \left\| \frac{1}{n} \mathbf{J}^{k+1} e - \nabla f_{\mathbf{I}_S, \mathbf{J}^{k+1}}(x^*) \right\|_2^2 \right] \leq \mathbb{E} \left[\sigma_S (1 - p_S) \left\| \frac{1}{n} \mathbf{J}^k e - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]$$

$$+ \mathbb{E} \left[\sigma_S p_S \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]. \qquad (113)$$

Proof For simplicity, in this proof we let $\nabla \mathbf{F}^k = \nabla \mathbf{F}(x^k)$ and $\nabla \mathbf{F}^* = \nabla \mathbf{F}(x^*)$. Rearranging (108), we have

$$\frac{1}{n} \mathbf{J}^{k+1} e - \nabla f_{\mathbf{I}_S, \mathbf{J}^{k+1}}(x^*) \stackrel{(108)}{=} \frac{1}{np_S} (\mathbf{J}^{k+1} - \nabla \mathbf{F}^*) \mathbf{\Pi}_{\mathbf{I}_S} e$$

$$\stackrel{(39)}{=} \frac{1}{np_S} \left(\mathbf{J}^k - (\mathbf{J}^k - \nabla \mathbf{F}^k) \mathbf{\Pi}_{\mathbf{I}_{S_k}} - \nabla \mathbf{F}^* \right) \mathbf{\Pi}_{\mathbf{I}_S} e$$

$$= \frac{1}{np_S} (\mathbf{J}^k - \nabla \mathbf{F}^*)(\mathbf{I} - \mathbf{\Pi}_{\mathbf{I}_{S_k}}) \mathbf{\Pi}_{\mathbf{I}_S} e$$

$$+ \frac{1}{np_S} (\nabla \mathbf{F}^k - \nabla \mathbf{F}^*) \mathbf{\Pi}_{\mathbf{I}_{S_k}} \mathbf{\Pi}_{\mathbf{I}_S} e. \qquad (114)$$

Taking norm squared on both sides gives

$$\left\| \frac{1}{n} \mathbf{J}^{k+1} e - \nabla f_{\mathbf{I}_S, \mathbf{J}^{k+1}}(x^*) \right\|_2^2 = \frac{1}{n^2 p_S^2} \underbrace{\left\| \overbrace{(\mathbf{J}^k - \nabla \mathbf{F}^*)}^{\mathbf{A}} (\mathbf{I} - \mathbf{\Pi}_{\mathbf{I}_{S_k}}) \mathbf{\Pi}_{\mathbf{I}_S} e \right\|_2^2}_{\mathrm{I}}$$

$$+ \frac{1}{n^2 p_S^2} \underbrace{\left\| \overbrace{(\nabla \mathbf{F}^k - \nabla \mathbf{F}^*)}^{\mathbf{R}} \mathbf{\Pi}_{\mathbf{I}_{S_k}} \mathbf{\Pi}_{\mathbf{I}_S} e \right\|_2^2}_{\mathrm{II}}$$

$$+ 2 \frac{1}{n^2 p_S^2} \underbrace{\left\langle (\mathbf{J}^k - \nabla \mathbf{F}^*)(\mathbf{I} - \mathbf{\Pi}_{\mathbf{I}_{S_k}}) \mathbf{\Pi}_{\mathbf{I}_S} e (\nabla \mathbf{F}^k - \nabla \mathbf{F}^*) \mathbf{\Pi}_{\mathbf{I}_{S_k}} \mathbf{\Pi}_{\mathbf{I}_S} e \right\rangle}_{\mathrm{III}}. \qquad (115)$$

First, it follows from (107) that expression III is zero. We now multiply expressions I and II by σ_S and bound certain conditional expectations of these terms. Since S and S_k are independent samplings, we have

$$
\begin{aligned}
\mathbb{E}\left[\frac{\sigma_S}{n^2 p_S^2}\left\|\mathbf{A}(\mathbf{I}-\boldsymbol{\Pi}_{\mathbf{I}_{S_k}})\boldsymbol{\Pi}_{\mathbf{I}_S}e\right\|_2^2 \mid \mathbf{A}\right] &= \sum_{C \in \mathcal{G}}\sum_{C' \in \mathcal{G}} p_C p_{C'} \frac{\sigma_C}{n^2 p_C^2}\left\|\mathbf{A}(\mathbf{I}-\boldsymbol{\Pi}_{\mathbf{I}_{C'}})\boldsymbol{\Pi}_{\mathbf{I}_C}e\right\|_2^2 \\
&\stackrel{(107)}{=} \sum_{C \in \mathcal{G}}\frac{\sigma_C}{n^2 p_C}\left\|\mathbf{A}\boldsymbol{\Pi}_{\mathbf{I}_C}e\right\|_2^2 \sum_{C' \in \mathcal{G},\, C' \neq C} p_{C'} \\
&= \sum_{C \in \mathcal{G}}\frac{\sigma_C}{n^2 p_C}(1-p_C)\left\|\mathbf{A}\boldsymbol{\Pi}_{\mathbf{I}_C}e\right\|_2^2 \\
&= \sum_{C \in \mathcal{G}} p_C \sigma_C (1-p_C)\frac{1}{n^2 p_C^2}\left\|\mathbf{A}\boldsymbol{\Pi}_{\mathbf{I}_C}e\right\|_2^2 \\
&\stackrel{(114)}{=} \mathbb{E}\left[\sigma_S(1-p_S)\left\|\frac{1}{n}\mathbf{J}^k e - \nabla f_{\mathbf{I}_S,\mathbf{J}^k}(x^*)\right\|_2^2 \mid \mathbf{J}^k\right].
\end{aligned}
\tag{116}
$$

Taking conditional expectation over expression II yields

$$
\begin{aligned}
\mathbb{E}\left[\frac{\sigma_S}{n^2 p_S^2}\left\|\mathbf{R}\boldsymbol{\Pi}_{\mathbf{I}_{S_k}}\boldsymbol{\Pi}_{\mathbf{I}_S}e\right\|_2^2 \mid \mathbf{R}, S_k\right] &= \sum_{C \in \mathcal{G}} p_C \frac{\sigma_C}{n^2 p_C^2}\left\|\mathbf{R}\boldsymbol{\Pi}_{\mathbf{I}_{S_k}}\boldsymbol{\Pi}_{\mathbf{I}_C}e\right\|_2^2 \\
&\stackrel{(107)}{=} \frac{\sigma_{S_k}}{n^2 p_{S_k}}\left\|\mathbf{R}\boldsymbol{\Pi}_{\mathbf{I}_{S_k}}\boldsymbol{\Pi}_{\mathbf{I}_{S_k}}e\right\|_2^2 = \frac{\sigma_{S_k}}{n^2 p_{S_k}}\left\|\mathbf{R}\boldsymbol{\Pi}_{\mathbf{I}_{S_k}}e\right\|_2^2 \\
&= \sigma_{S_k} p_{S_k}\left\|\nabla f_{\mathbf{I}_{S_k},\mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_{S_k},\mathbf{J}^k}(x^*)\right\|_2^2,
\end{aligned}
\tag{117}
$$

where in the last equation we used the identity

$$
\begin{aligned}
&\left\|\nabla f_{\mathbf{I}_C,\mathbf{J}}(x) - \nabla f_{\mathbf{I}_C,\mathbf{J}}(y)\right\|_2^2 \\
&= \left\|\frac{1}{np_C}(\nabla \mathbf{F}(x) - \nabla \mathbf{F}(y))\boldsymbol{\Pi}_C e\right\|_2^2, \quad \forall \mathbf{J} \in \mathbb{R}^{d \times n}, \forall C \in \mathcal{G},
\end{aligned}
\tag{118}
$$

which in turn is a specialization of (44) to the minibatch sketch $\mathbf{S} = \mathbf{I}_S$ and the specific choice of the bias-correcting variable $\theta_{\mathbf{S}} = 1/p_S$. It remains to take expectation of (116) and (117), apply the tower property, and combine this with (115). □

5.3 Bounding the second moment of g^k

In the next lemma we bound the second moment of our gradient estimate g^k.

Lemma 11 *The second moment of the gradient estimate is bounded by*

$$
\mathbb{E}\left[\left\|g^k\right\|_2^2 \mid \mathbf{J}^k, x^k\right] \leq 2\mathbb{E}\left[\left\|\nabla f_{\mathbf{I}_S,\mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_S,\mathbf{J}^k}(x^*)\right\|_2^2 \mid \mathbf{J}^k, x^k\right]
$$

$$+ 2\mathbb{E}\left[\left\|\nabla f_{\mathbf{I}_S,\mathbf{J}^k}(x^*) - \frac{1}{n}\mathbf{J}^k e\right\|_2^2 \mid \mathbf{J}^k, x^k\right]. \quad (119)$$

Proof Adding and subtracting $\frac{1}{np_{S_k}}\nabla\mathbf{F}(x^*)\mathbf{\Pi}_{\mathbf{I}_{S_k}}e$ from (108) gives

$$g^k = \frac{1}{n}\mathbf{J}^k e - \frac{1}{np_{S_k}}(\mathbf{J}^k - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{I}_{S_k}}e + \frac{1}{np_{S_k}}(\nabla\mathbf{F}(x^k) - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{I}_{S_k}}e.$$

Taking norm squared on both sides, and using the bound $\|a+b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2$ gives

$$\begin{aligned}
\left\|g^k\right\|_2^2 &\leq \frac{2}{n^2 p_{S_k}^2}\left\|(\nabla\mathbf{F}(x^k) - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{I}_{S_k}}e\right\|_2^2 \\
&\quad + \frac{2}{n^2}\left\|\frac{1}{p_{S_k}}(\mathbf{J}^k - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{I}_{S_k}}e - \mathbf{J}^k e\right\|_2^2 \\
&\overset{(118)}{=} 2\left\|\nabla f_{\mathbf{I}_{S_k},\mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_{S_k},\mathbf{J}^k}(x^*)\right\|_2^2 \\
&\quad + \frac{2}{n^2}\underbrace{\left\|\frac{1}{p_{S_k}}(\mathbf{J}^k - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{I}_{S_k}}e - \mathbf{J}^k e\right\|_2^2}_{A}. \quad (120)
\end{aligned}$$

Taking expectation of the A term, we get

$$\begin{aligned}
\mathbb{E}\left[\left\|\underbrace{\frac{1}{p_S}(\mathbf{J}^k - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{I}_S}e}_{X} - \underbrace{\mathbf{J}^k e}_{\mathbb{E}[X]}\right\|_2^2 \mid \mathbf{J}^k, x^k\right] &\leq \mathbb{E}\left[\left\|\frac{1}{p_S}(\mathbf{J}^k - \nabla\mathbf{F}(x^*))\mathbf{\Pi}_{\mathbf{I}_S}e\right\|_2^2 \mid \mathbf{J}^k, x^k\right] \\
&\overset{(114)}{=} n^2\mathbb{E}\left[\left\|\nabla f_{\mathbf{I}_S,\mathbf{J}^k}(x^*) - \frac{1}{n}\mathbf{J}^k e\right\|_2^2 \mid \mathbf{J}^k, x^k\right],
\end{aligned}$$

where we used the inequality $\mathbb{E}\left[\|X - \mathbb{E}[X]\|_2^2\right] \leq \mathbb{E}\left[\|X\|_2^2\right]$. The result follows by combining the above with (120). □

5.4 Smoothness and strong convexity of $f_{\mathbf{I}_C,\mathbf{J}}$

Recalling the setting of Theorem 6, we assume that each f_C is μ—strongly convex and L_C—smooth:

$$f_C(y) + \langle\nabla f_C(y), x - y\rangle + \frac{\mu}{2}\|x - y\|_2^2 \leq f_C(x)$$

$$\leq f_C(y) + \langle\nabla f_C(y), x - y\rangle + \frac{L_C}{2}\|x - y\|_2^2$$

for all $C \in \mathcal{G}$. It is known (see Section 2.1 in [19]) that the above conditions imply the following inequality:

$$\langle \nabla f_C(x) - \nabla f_C(y), x - y \rangle \geq \frac{\mu L_C}{\mu + L_C} \|x - y\|_2^2$$
$$+ \frac{1}{\mu + L_C} \|\nabla f_C(x) - \nabla f_C(y)\|_2^2, \quad (121)$$

for all $x, y \in \mathbb{R}^d$. A consequence of these assumptions that will be useful to us is that the function $f_{\mathbf{I}_C, \mathbf{J}}$ is $\frac{\tau \mu}{n p_C}$—strongly convex and $\frac{\tau L_C}{n p_C}$—smooth. This can in turn be used to establish the next lemma, which will be used in the proof of Theorem 6:

Lemma 12 *Under the assumptions of Theorem 6 (in particular, assumptions on f and S), we have*

$$\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{\mu}{2} \|x - y\|_2^2$$
$$+ \mathbb{E}_{\mathcal{D}} \left[\frac{n p_S}{2 \tau L_S} \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}}(x) - \nabla f_{\mathbf{I}_S, \mathbf{J}}(y) \right\|_2^2 \right], \quad (122)$$

for all $x, y \in \mathbb{R}^d$ and $\mathbf{J} \in \mathbb{R}^{d \times n}$.

Proof Applying (121) to the function $f_{\mathbf{I}_S, \mathbf{J}}$ gives

$$\langle \nabla f_{\mathbf{I}_S, \mathbf{J}}(x) - \nabla f_{\mathbf{I}_S, \mathbf{J}}(y), x - y \rangle \geq \frac{\tau}{n p_S} \frac{\mu L_S}{\mu + L_S} \|x - y\|_2^2$$
$$+ \frac{n p_S}{\tau (\mu + L_S)} \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}}(x) - \nabla f_{\mathbf{I}_S, \mathbf{J}}(y) \right\|_2^2$$
$$\geq \frac{\tau \mu}{2 n p_S} \|x - y\|_2^2$$
$$+ \frac{n p_S}{2 \tau L_S} \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}}(x) - \nabla f_{\mathbf{I}_S, \mathbf{J}}(y) \right\|_2^2.$$

Taking expectation over both sides over S, noting that $\mathbb{E}_{\mathcal{D}} \left[\frac{1}{p_S} \right] = \sum_{C \in \mathcal{G}} 1 = \frac{n}{\tau}$, and recalling that $\nabla f_{\mathbf{I}_S, \mathbf{J}}(x)$ is an unbiased estimator of $\nabla f(x)$, we get the result. $\square$

5.5 Proof of Theorem 6

Let $\mathbb{E}_k [\cdot]$ denote expectation conditional on $\mathbf{J}^k$ and x^k. We can write

$$\mathbb{E}_k \left[\left\| x^{k+1} - x^* \right\|_2^2 \right]$$
$$\overset{(2)}{=} \mathbb{E}_k \left[\left\| x^k - x^* - \alpha g^k \right\|_2^2 \right]$$
$$\overset{(33)}{=} \left\| x^k - x^* \right\|_2^2 - 2\alpha \left\langle \nabla f(x^k), x^k - x^* \right\rangle + \alpha^2 \mathbb{E}_k \left[\left\| g^k \right\|_2^2 \right]$$

$$\overset{(122)}{\le} (1 - \mu\alpha) \left\| x^k - x^* \right\|_2^2 - \alpha\mathbb{E}_k \left[\frac{np_S}{\tau L_S} \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]$$

$$+ \alpha^2 \mathbb{E}_k \left[\left\| g^k \right\|_2^2 \right]$$

$$\overset{(119)}{\le} (1 - \mu\alpha) \left\| x^k - x^* \right\|_2^2 + 2\alpha^2 \mathbb{E}_k \left[\left\| \frac{1}{n} \mathbf{J}^k e - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]$$

$$+ 2\alpha\mathbb{E}_k \left[\left(\alpha - \frac{np_S}{2\tau L_S} \right) \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]. \tag{123}$$

Next, after taking expectation in (123), applying the tower property, and subsequently adding the term $2\alpha\mathbb{E} \left[\sigma_S \left\| \frac{1}{n} \mathbf{J}^{k+1} e - \nabla f_{\mathbf{I}_S, \mathbf{J}^{k+1}}(x^*) \right\|_2^2 \right]$ to both sides of the resulting inequality, we get

$$\mathbb{E} \left[\Psi_S^{k+1} \right] \le \mathbb{E} \left[(1 - \mu\alpha) \left\| x^k - x^* \right\|_2^2 \right]$$

$$+ 2\alpha\mathbb{E} \left[\left(\alpha - \frac{np_S}{2\tau L_S} \right) \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]$$

$$+ 2\alpha^2 \mathbb{E} \left[\left\| \frac{1}{n} \mathbf{J}^k e - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|^2 \right] + 2\alpha\mathbb{E} \left[\sigma_S \left\| \frac{1}{n} \mathbf{J}^{k+1} e - \nabla f_{\mathbf{I}_S, \mathbf{J}^{k+1}}(x^*) \right\|_2^2 \right]$$

$$\overset{(113)}{\le} \mathbb{E} \left[\underbrace{(1 - \mu\alpha)}_{\text{I}} \left\| x^k - x^* \right\|_2^2 \right]$$

$$+ 2\alpha\mathbb{E} \left[\sigma_S \underbrace{\left(1 - p_S + \frac{\alpha}{\sigma_S} \right)}_{\text{II}} \left\| \frac{1}{n} \mathbf{J}^k e - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]$$

$$+ 2\alpha\mathbb{E} \left[\underbrace{\left(\alpha + \sigma_S p_S - \frac{np_S}{2\tau L_S} \right)}_{\text{III}} \left\| \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^k) - \nabla f_{\mathbf{I}_S, \mathbf{J}^k}(x^*) \right\|_2^2 \right]. \tag{124}$$

Next, we determine a bound on α so that III ≤ 0. Choosing

$$\alpha + \sigma_C p_C - \frac{np_C}{2\tau L_C} \le 0, \quad \forall C \in \mathcal{G} \Rightarrow \alpha \le \frac{np_C}{2\tau L_C} - \sigma_C p_C, \quad \forall C \in \mathcal{G}, \tag{125}$$

guarantees that III ≤ 0, and thus the last term in term in (124) can be safely dropped. Next, to build a recurrence and conclude the convergence proof, we bound the stepsize α so that II $\le$ I; that is,

$$1 - p_C + \frac{\alpha}{\sigma_C} \le 1 - \alpha\mu, \quad \forall C \in \mathcal{G} \Rightarrow \alpha \le \frac{\sigma_C p_C}{\mu\sigma_C + 1}, \quad \forall C \in \mathcal{G}. \tag{126}$$

Consequently,

$$\mathbb{E}\left[\Psi_S^{k+1}\right] \leq \mathbb{E}\left[(1-\mu\alpha)\left\|x^k - x^*\right\|_2^2\right]$$

$$+2\alpha\mathbb{E}\left[\sigma_S(1-\mu\alpha)\left\|\frac{1}{n}\mathbf{J}^k e - \nabla f_{\mathbf{I}_S,\mathbf{J}^k}(x^*)\right\|_2^2\right]$$

$$= (1-\mu\alpha)\mathbb{E}\left[\Psi_S^k\right].$$

Since $\sigma_S = \frac{n}{4\tau L_S}$, in view of (125) and (126) the combined bound on α is

$$\alpha \leq \min\left\{\frac{np_C}{4\tau L_C}, \frac{p_C}{\mu + 4\frac{\tau}{n}L_C}\right\} = \frac{p_C}{\mu + 4\frac{\tau}{n}L_C}, \quad \forall C \in \mathcal{G}.$$

Hence, we have established the recursion (111).

5.6 Calculating the iteration complexity in special cases

In this section we consider the special case of JacSketch analyzed via Theorem 6—minibatch SAGA with τ—partition sampling—and look at further special cases by varying the minibatch size τ and probabilities. Our aim is to justify the complexities appearing in Table 1. In view of Theorem 6 the iteration complexity is given by

$$\max_{C\in\mathcal{G}}\left(\frac{1}{p_C} + \frac{\tau}{np_C}\frac{4L_C}{\mu}\right)\log\left(\frac{1}{\epsilon}\right), \tag{127}$$

where $\mathcal{G} = \text{supp}(S)$. Complexity (127) is listed in line 2 of Table 1. The complexities in lines 4, 6, 8 and 14 arise as special cases of (127) for specific choices of τ and probabilities p_C.

- In line 4 we have *gradient descent*. This is obtained by choosing $\mathcal{G} = \{[n]\}$ (whence $p_{[n]} = 1, \tau = n$ and $L_{[n]} = L$), which is why (127) simplifies to $\left(1 + \frac{4L}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)$.
- In line 6 we consider *uniform SAGA*. That is, we choose $\tau = 1$ and $p_i = 1/n$ for all i. We have $\mathcal{G} = \{\{1\}, \{2\}, \ldots, \{n\}\}$ and $L_{\{i\}} = L_i$. Therefore, (127) simplifies to $\left(n + \frac{4L_{\max}}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)$. This is essentially the same[16] complexity result given in [6].
- In line 8 we consider *SAGA with importance sampling*. This is the same setup as above, except we choose

$$p_i = \frac{\mu n + 4L_i}{\sum_{j=1}^n n\mu + 4L_j}, \tag{128}$$

---

[16] With the difference being that in [6] the iteration complexity is $2\left(n + L_{\max}/\mu\right)\log\left(\frac{1}{\epsilon}\right)$, thus a small constant change.

which is the optimal choice minimizing the complexity bound in $p_1, \ldots, p_n$. With these optimal probabilities, the stepsize bound becomes $\alpha \le \frac{1}{n\mu + 4\bar{L}}$, and by choosing the maximum allowed stepsize the resulting iteration complexity is

$$\left(n + \frac{4\bar{L}}{\mu}\right) \log\left(\frac{1}{\epsilon}\right). \tag{129}$$

Now consider the probabilities $p_i = \frac{L_i}{\sum_{j=1}^n L_j}$ suggested in [30]. Using our bound, these lead to the complexity

$$\max_{i=1,\ldots,n} \left\{ \frac{\sum_{j=1}^n L_j}{L_i} + 4\frac{\sum_{j=1}^n L_j}{\mu n} \right\} \log\frac{1}{\epsilon} = \left(\frac{n\bar{L}}{L_{\min}} + \frac{4\bar{L}}{\mu}\right) \log\left(\frac{1}{\epsilon}\right). \tag{130}$$

Comparing this with (129), we see that this non-uniform sampling offers a significant speed up over uniform sampling if $n\mu \le L_{\min}$. However, our complexity (129) is always better than both the uniform sampling sampling complexity $(n + L_{\max}/\mu) \log\left(\frac{1}{\epsilon}\right)$ and (130).

– Finally, in line 14 of Table 1 we optimize over probabilities p_C directly; that is we extend the importance sampling described above to any τ. Minimizing the complexity bound over the probabilities, and noting that $|\mathcal{G}| = \frac{n}{\tau}$, this leads to the rate

$$\left(\frac{n}{\tau} + \frac{4\frac{1}{|\mathcal{G}|}\sum_{C \in \mathcal{G}} L_C}{\mu}\right) \log\left(\frac{1}{\epsilon}\right). \tag{131}$$

This iteration complexity also applies to the reduced memory variant of SAGA (18). This is because Theorem 6 also holds for sketches $\mathbf{S} = e_S$ where S is a τ—partition sampling. To see this, note that our analysis in this section relies on the orthogonality property (107) which also holds for $\mathbf{S} = e_S$ since (for $\mathbf{W} = \mathbf{I}$) we have:

$$\mathbf{\Pi}_{e_{C_1}} \mathbf{\Pi}_{e_{C_2}} = \frac{1}{\tau} e_{C_1} \underbrace{(e_{C_1}^\top e_{C_2})}_{=0} e_{C_2}^\top \frac{1}{\tau} = 0, \quad \text{for} \quad C_1, C_2 \in \mathcal{G}, \quad C_1 \ne C_2.$$

Lemmas 10, 11 and 12 depend on the sketch through $\nabla f_{\mathbf{S},\mathbf{J}}(x^*)$ only, which in turn depends on the sketch through $\mathbf{\Pi}_{\mathbf{S}} e$, and it is easy to see that if either $\mathbf{S} = \mathbf{I}_S$ or $\mathbf{S} = e_S$, we have $\mathbf{\Pi}_{\mathbf{S}} e = e_S$.

6 Experiments

We perform several experiments to validate the theory, and also test the practical relevance of non-uniform SAGA (79) with the optimized probability distribution (128). All of our code for these experiments was written in Julia and can be found on github in https://github.com/gowerrobert/StochOpt.jl.

In our experiments we test either ridge regression

$$f(x) = \frac{1}{2n} \left\| \mathbf{A}^\top x - y \right\|_2^2 + \frac{\lambda}{2} \|x\|_2^2, \tag{132}$$

or logistic regression

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \log \left(1 + e^{-y_i \langle a_i, x \rangle} \right) + \frac{\lambda}{2} \|x\|_2^2, \tag{133}$$

where $\mathbf{A} = [a_1, \dots, a_n] \in \mathbb{R}^{d \times n}$, $y \in \mathbb{R}^n$ is the given data and $\lambda > 0$ the regularization parameter.

6.1 New non-uniform sampling using optimal probabilities

First we compare non-uniform SAGA using the new optimized importance probabilities (128) against using the probabilities $p_i = L_i / \overline{L}$ as suggested in [30]. When $n\mu$ is significantly smaller than L_i for all i then the two sampling are very similar. But when $n\mu$ is relatively large, then the optimized probabilities (128) can be much closer to a uniform distribution as compared to using $p_i = L_i / \overline{L}$. We illustrate this by solving a ridge regression problem (132), using generated data such that

$$\mathbf{A}^\top x = y + \epsilon, \tag{134}$$

where the elements of $\mathbf{A}$ and x are sampled from the standard Gaussian distribution $\mathcal{N}(0, 1)$, and the elements of ϵ are sampled from $\mathcal{N}(0, 10^{-3})$. It is not hard to see that the smoothness constants $\{L_i\}$ are given by $L_i = \|a_i\|_2^2 + \lambda$ for $i \in [n]$. We scale the columns of $\mathbf{A}$ so that $\|a_1\|_2^2 = 1$ and $\|a_i\|_2^2 = \frac{1}{n^2}$, for $i = 2, \dots, n$, and set the regularization parameter $\lambda = \frac{1}{n^2}$. Consequently, $L_{\max} = 1 + \frac{1}{n^2}$, $L_i = \frac{2}{n^2}$ for $i = 1, \dots, n$, $\overline{L} = \frac{(n+1)^2 - 1}{n^3}$ and $\mu = \frac{1}{n} \lambda_{\min}(\mathbf{A}\mathbf{A}^\top) + \frac{1}{n^2}$. In this case the iteration complexity of non-uniform SAGA with the optimal probabilities (129) is given by
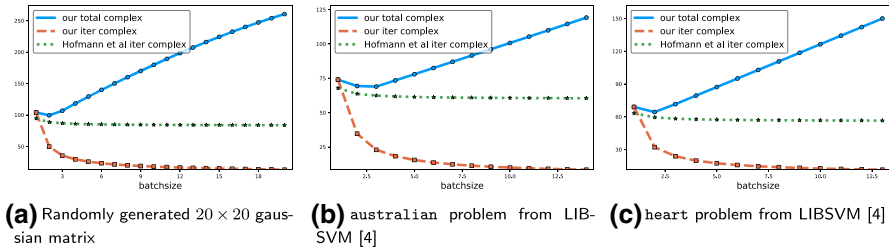
$$\left(n + 4 \frac{(n+1)^2 - 1}{\mu n^3} \right) \log \left(\frac{1}{\epsilon} \right). \tag{135}$$

The complexity (130) which results from using the probabilities $p_i = L_i / \overline{L}$ is given by

$$\frac{(n+1)^2 - 1}{n^3} \left(\frac{n^3}{2} + \frac{4}{\mu} \right) \log \left(\frac{1}{\epsilon} \right). \tag{136}$$

Now we consider the regime where $n \to \infty$, in which case $\mu \to \mathcal{O}(\frac{1}{n^2})$ and consequently (135)$\to \mathcal{O}(n) \log \frac{1}{\epsilon}$ and in contrast (136) $\to \mathcal{O}(n^2) \log \frac{1}{\epsilon}$.

We illustrate this in Fig. 1a–c where we set $n = 10$, $n = 100$ and $n = 1000$, respectively, and plot the complexities given in (135) and (136) . To accompany this

(a) Randomly generated 20×20 gaussian matrix

(b) `australian` problem from LIBSVM [4]

(c) `heart` problem from LIBSVM [4]

Fig. 1 The iteration complexity of minibatch SAGA (80) vs the mini-batch size τ for two ridge regression problems (132). We used $\lambda = L_{\max}/n$

(a) $n = 10$

(b) $n = 100$

(c) $n = 1000$
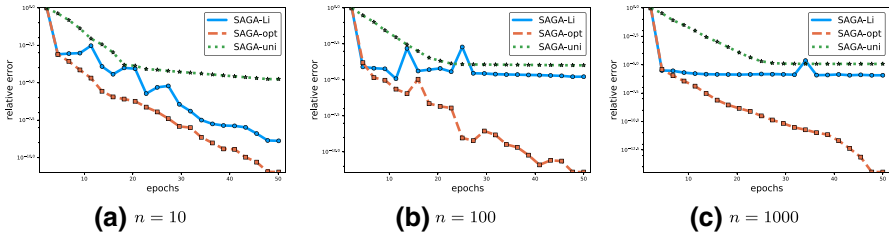
Fig. 2 Comparing the performance of SAGA with importance sampling based on the optimized probabilities (128) (`SAGA-opt`), $p_i = L_i/\overline{L}$ (`SAGA-Li`) and $p_i = 1/n$ (`SAGA-uni`) for an artificially constructed ridge regression problem as n grows. Markers represent monitored points and not the iterations of the algorithms

plot, in Fig. 2a–c we also plot an execution of `SAGA-uni` (SAGA with uniform probabilities), `SAGA-Li` (SAGA with $p_i = L_i/\overline{L}$) and `SAGA-opt` (SAGA with optimized probabilities). In all figures we see that `SAGA-opt` is the fastest method. We can also see that `SAGA-Li` stalls in Fig. 2b and c when n is larger, performing even worst as compared to `SAGA-uni`.

6.2 Optimal mini-batch size

Our analysis of the mini-batch SAGA is precise enough as to inform an optimal mini-batch size. For instance, consider τ—nice sampling and the resulting iteration complexity (102). Theorem 3 states that for any $\tau \in [n]$, the terms within the maximum in (102) are bounded by

$$L_{\max} \geq L_{\max}^{\mathcal{G}} \geq \quad L \tag{137}$$

$$L_{\max} + \frac{\mu n}{4} \geq C(\tau) \stackrel{\text{def}}{=} \frac{1}{\tau} \frac{n-\tau}{n-1} L_{\max} + \frac{\mu}{4} \frac{n}{\tau} \geq \quad \frac{\mu}{4}. \tag{138}$$

Moreover, the upper and lower bounds are realized for $\tau = 1$ and $\tau = n$, respectively. Consequently, for τ small, we have $L_{\max}^{\mathcal{G}} \leq C(\tau)$. On the other hand, for τ large we have $L_{\max}^{\mathcal{G}} \geq C(\tau)$. Furthermore, $C(\tau)$ decreases super-linearly in τ while $L_{\max}^{\mathcal{G}}$ tends to decrease more modestly. Consequently, the point where $L_{\max}^{\mathcal{G}}$ overtakes $C(\tau)$ is often the best for the overall complexity of the method. To better appreciate these observations, we plot the evolution of the iteration complexity (102), the total com-

plexity and the iteration complexity as predicted by Hofmann et al. [14] (see (105)) as τ increases in Fig. 3a–c for three different linear least squares problems. Since each step of mini-batch SAGA computes τ stochastic gradients, so the total complexity is τ times the iteration complexity. In each figure we can see that our iteration complexity initially decreases super-linearly, then at some point the complexity is dominated by $L_{\max}^{\mathcal{G}}$ and the iteration complexity decreases sublinearly. Up to this point we can observe an improvement in overall total complexity. This is in contrast to the iteration complexity given by Hofmann et al. that shows practically no improvement in even the iteration complexity as τ increases.

Though these experiments indicate only modest improvements in total complexity, and suggests that $\tau = 2$ or $\tau = 3$ is optimal, we must bear in mind that this corresponds to 10% and 20% of the data for these small dimensional problems. We conjecture that for larger problems, this improvement in total complexity will also be larger.

To use these insights in practice, we need to be able to efficiently determine the τ which corresponds to the point at which the convergence regimes switches from being dominated by $C(\tau)$ to being dominated by $L_{\max}^{\mathcal{G}}$. This surmounts to choosing τ so that $L_{\max}^{\mathcal{G}} = \frac{1}{\tau}\frac{n-\tau}{n-1}L_{\max} + \frac{\mu}{4}\frac{n}{\tau}$. Estimating $L_{\max}$ and μ is often possible, but the cost of computing $L_{\max}^{\mathcal{G}}$ has a combinatorial dependency on n and τ. Thus to have a practical way of choosing τ, we first need to bound $L_{\max}^{\mathcal{G}}$. This can be done for losses with linear classifiers using concentration bounds. We leave this for future work.

6.3 Comparative experiments

We now compare the performance of `SAGA-opt` to several known methods such as `SVRG` [15], `grad` (gradient descent with fixed stepsizes) and `AMprev` (an improved version of `SVRG` that uses second order information) [28]. For the stepsize of `SAGA-opt` and `SAG-opt`, we found the stepsize $\alpha \leq \frac{1}{n\mu+4\bar{L}}$ given by theory to be a bit too conservative. Instead do we away with the 4 and used $\alpha = \frac{1}{n\mu+\bar{L}}$ instead. For the remaining methods we used a grid search over $L_{\max} \times 2^m$ for $m = 21, 19, 17, \ldots, -10, -11$.

To illustrate how biased gradient estimates can perform well in practice, we also test `SAG-opt`: a method that uses the same Jacobian updates as `SAGA-opt`, but instead uses the biased gradient estimate $g^k = \frac{1}{n}\mathbf{J}^{k+1}e$. See Sect. 2.5 for more details on biased gradient estimates.

In Fig. 3a–c we compare the methods on three logistic regression problems (133) based on three different data sets taken from LIBSVM [4]. In all these problems the two methods with optimized non-uniform sampling `SAG-opt` and `SAGA-opt` were faster in terms of both epochs and time. The next best method was `AM-prev`, followed by `SVRG` and `grad`. It is interesting to see how well `SAG-opt` performs in practice, despite having biased gradient estimates. This is why we believe it is important to advance the analyse of biased gradient estimates as future work.
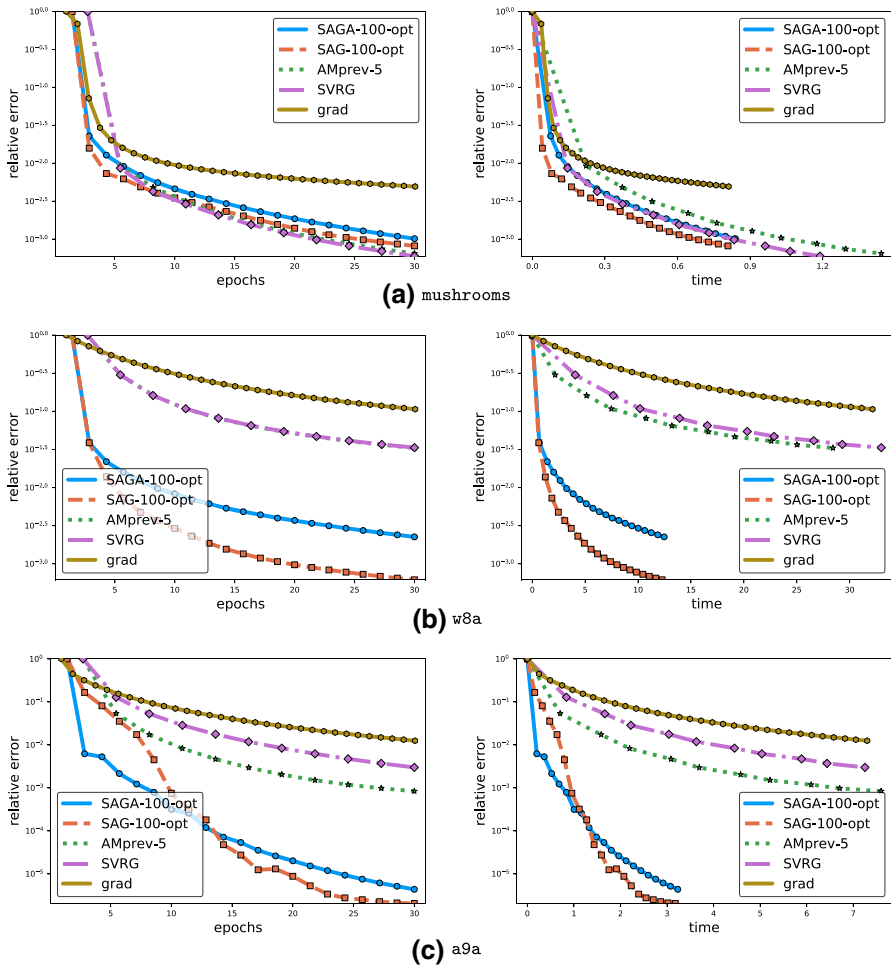
Fig. 3 Comparison of the methods on logistic regression problems (133) with data taken from LIBSVM [4]

7 Conclusion

We now provide a brief summary of some of the key contributions of this paper and a few selected pointers to possible future research directions.

We developed and analyzed JacSketch—a novel family of variance reduced methods based on Jacobian sketching—and provided a link between variance reduction for empirical risk minimization and recent results from the field of randomized numerical linear algebra on sketch-and-project type methods for solving linear systems. In particular, it turns out that variance reduction is obtained by taking an SGD step on a stochastic optimization problem whose solution is the unknown Jacobian. As a consequence of our analysis, we resolved the conjecture of [30] in the affirmative by proving a properly designed importance sampling for SAGA leading to the iteration

complexity of $\mathcal{O}(n + \frac{\bar{L}}{\mu}) \log\left(\frac{1}{\epsilon}\right)$. For this purpose we developed a new proof technique using a *stochastic Lyapunov function*. Our complexity result for uniform mini-batch SAGA perfectly interpolates between the best known convergence rates of SAGA and gradient descent, and is sufficiently precise as to inform the choice of the batch size that minimizes the over all complexity of the method. Additionally we design and analyse a *reduced memory* variant of SAGA as a special case.

For future work we see many possible avenues including the following.

Structured sparse weight matrices One may wish to explore combinations of a weight matrix and different sketches to design new efficient methods further improving iteration complexity. For this the weighting matrix will have to be highly structured (e.g., block diagonal or very sparse) so that the Jacobian update (39) can be computed efficiently.

Bias-variance trade-off One can try to explore the bias-variance trade-off as opposed to merely focus on the extremes only: SAG (minimum variance) and SAGA (no bias). There is also no empirical evidence that unbiased estimators outperform the biased ones.

Johnson–Lindenstrauss sketches One can design completely new methods using different sparse sketches, such as the fast Johnson–Lindenstrauss transform [2] or the Achlioptas transform [1]. The resulting method can then be analyzed through Theorem 1. But first these sketches need to be adapted to ensure we get an efficient method. In particular, computing $\nabla \mathbf{F}(x)\mathbf{S}$ is only efficient if $\mathbf{S}$ most of the rows of $\mathbf{S}$ are zeros.

Appendix A: Proof of inequality (20)

Lemma 13 *Let S be a sampling whose support $\mathcal{G} = \mathrm{supp}(S)$ is a partition of $[n]$. Moreover, assume all sets of this partition have cardinality τ. Then*

$$\frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} L_C \leq \bar{L} \leq \max_{C \in \mathcal{G}} \frac{1}{\tau} \sum_{i \in C} L_i.$$

Proof By assumption, $|\mathcal{G}| = \frac{n}{\tau}$. The first inequality follows from $\sum_{C \in \mathcal{G}} L_C \leq \sum_{C \in \mathcal{G}} \frac{1}{\tau} \sum_{i \in C} L_i = \frac{1}{\tau} \sum_{i=1}^{n} L_i = \frac{n}{\tau} \bar{L}$. On the other hand,

$$\bar{L} = \frac{1}{n} \sum_{i=1}^{n} L_i = \frac{1}{n} \sum_{C \in \mathcal{G}} \sum_{i \in C} L_i = \frac{1}{|\mathcal{G}|} \sum_{C \in \mathcal{G}} \frac{1}{\tau} \sum_{i \in C} L_i \leq \max_{C \in \mathcal{G}} \frac{1}{\tau} \sum_{i \in C} L_i. \qquad \square$$

Appendix B: Duality of sketch-and-project and constrain-and-approximate

Lemma 14 *Let* $\mathbf{J}^k, \nabla \mathbf{F} \in \mathbb{R}^{d \times n}$ *and* $\mathbf{S} \in \mathbb{R}^{n \times \tau}$. *The sketch-and-project problem*

$$\mathbf{J}^{k+1} = \arg \min_{\mathbf{J} \in \mathbb{R}^{d \times n}} \frac{1}{2} \left\| \mathbf{J} - \mathbf{J}^k \right\|_{\mathbf{W}^{-1}}^{2} \quad \text{subject to} \quad \nabla \mathbf{F} \mathbf{S} = \mathbf{J} \mathbf{S}, \qquad (139)$$

and the constrain-and-approximate problem

$$\mathbf{J}^{k+1} = \arg \min_{\mathbf{J} \in \mathbb{R}^{d \times n}, \mathbf{Y} \in \mathbb{R}^{d \times \tau}} \frac{1}{2} \|\mathbf{J} - \nabla \mathbf{F}\|_{\mathbf{W}^{-1}}^{2} \quad \text{subject to} \quad \mathbf{J} = \mathbf{J}^k + \mathbf{Y} \mathbf{S}^\top \mathbf{W}, \quad (140)$$

have the same solution, given by:

$$\mathbf{J}^{k+1} = \mathbf{J}^k - (\mathbf{J}^k - \nabla \mathbf{F}) \mathbf{S} (\mathbf{S}^\top \mathbf{W} \mathbf{S})^\dagger \mathbf{S}^\top \mathbf{W}. \qquad (141)$$

Proof The proof is given in Theorem 4.1 in [12]. $\qquad \square$

Appendix C: Proof of Theorem 5

First we will establish that

$$\frac{|\mathcal{G}|}{c_1^2} \sum_{C \in \mathcal{G}} e_C e_C^\top \mathbf{W} = \frac{|\mathcal{G}| c_2}{c_1^2} \begin{pmatrix} \frac{c_1}{c_2} w_1 & w_2 & \cdots & w_{n-1} & w_n \\ w_1 & \frac{c_1}{c_2} w_2 & \cdots & w_{n-1} & w_n \\ \vdots & & \ddots & & \vdots \\ w_1 & \cdots & & \frac{c_1}{c_2} w_{n-1} & w_n \\ w_1 & w_2 & \cdots & w_{n-1} & \frac{c_1}{c_2} w_n \end{pmatrix}. \qquad (142)$$

Indeed, for every i we have that $e_i^\top \frac{|\mathcal{G}|}{c_1^2} \left(\sum_{C \in \mathcal{G}} e_C e_C^\top \mathbf{W} \right) e_i = w_i \frac{|\mathcal{G}|}{c_1^2} \sum_{C \in \mathcal{G}: i \in C} 1 = w_i \frac{|\mathcal{G}|}{c_1}$, and for every $i \neq j$ we have $e_i^\top \frac{|\mathcal{G}|}{c_1^2} \left(\sum_{C \in \mathcal{G}} e_C e_C^\top \mathbf{W} \right) e_j = w_j \frac{|\mathcal{G}|}{c_1^2} \sum_{C \in \mathcal{G}: i, j \in C} 1 = w_j \frac{|\mathcal{G}| c_2}{c_1^2}$. Using (142), (91) and the Gershgorin circle theorem to bound ρ from above we get $\rho \leq \max_i \left\{ \left(\frac{|\mathcal{G}|}{c_1} - 1 \right) w_i + \sum_{i \neq j} w_j \left| \frac{|\mathcal{G}| c_2}{c_1^2} - 1 \right| \right\}$, as claimed. When

$\mathbf{W} = \mathbf{I}$ we can get tighter results by using that $\left(\frac{|\mathcal{G}|}{c_1^2} \sum_{C \in \mathcal{G}} e_C e_C^\top - e e^\top \right)$ is a *circulant matrix* with associated vector $v = \left(\frac{|\mathcal{G}|}{c_1} - 1, \frac{|\mathcal{G}| c_2}{c_1^2} - 1, \ldots, \frac{|\mathcal{G}| c_2}{c_1^2} - 1 \right) \in \mathbb{R}^n$. There is an elegant formula for calculating eigenvalues λ_j of circulant matrices [34] using v, given by

$$\lambda_j = v_1 + \sum_{k=1}^{n-1} \omega_j^k v_{n-k+1} = \frac{|\mathcal{G}|}{c_1} - 1$$

$$+ \left(\frac{|\mathcal{G}| c_2}{c_1^2} - 1 \right) \sum_{k=1}^{n-1} \omega_j^k, \quad \text{for} \quad j = 0, \ldots, n-1, \quad (143)$$

where $\omega_j = e^{\frac{2\pi i j}{n}}$ are the n-th roots of unity and i is the imaginary number. From (143) we see that there are only two distinct eigenvalues. Namely, for $j = 0$ we have

$$\lambda_0 \overset{(143)}{=} \frac{|\mathcal{G}|}{c_1} - 1 + \left(\frac{|\mathcal{G}| c_2}{c_1^2} - 1 \right)(n-1) = \frac{|\mathcal{G}|}{c_1} \left(1 + (n-1) \frac{c_2}{c_1} \right) - n.$$

The other eigenvalue is given by any $j \neq 0$ since

$$\lambda_j \overset{(143)}{=} \frac{|\mathcal{G}|}{c_1} - 1 - \left(\frac{|\mathcal{G}| c_2}{c_1^2} - 1 \right) + \left(\frac{|\mathcal{G}| c_2}{c_1^2} - 1 \right) \underbrace{\sum_{k=0}^{n-1} \omega_j^k}_{=0} = \frac{|\mathcal{G}|}{c_1} \left(1 - \frac{c_2}{c_1} \right). \qquad \square$$

Appendix D: Notation glossary

See the Table 2.

Table 2 Frequently used notation

$f(x)$	$\frac{1}{n} \sum_{i=1}^n f_i(x)$ (convex loss function $f : \mathbb{R}^d \to \mathbb{R}$)	(1)
x^*	Minimizer of f	(1)
μ	Strong convexity constant of f	Table 1 and Assumption 3.3 and Theorem 6
α	Stepsize	(2)
g^k	Stochastic estimator of $\nabla f(x^k)$	(2), (13), (16), (33)
$[n]$	$\{1, 2, \ldots, n\}$	
$F(x)$	$(f_1(x), \ldots, f_n(x))^\top \in \mathbb{R}^n$ (function $F : \mathbb{R}^d \to \mathbb{R}^n$)	(3)
$\nabla \mathbf{F}(x)$	$[\nabla f_1(x), \ldots, \nabla f_n(x)] \in \mathbb{R}^{d \times n}$ (Jacobian of F at x)	(4)

Table 2 continued

e	$(1, 1, \ldots, 1)^\top \in \mathbb{R}^n$ (vector of all ones)	(5)
f^*/f^k	Shorthand for $f(x^*)$ / $f(x^k)$	
$\mathbf{W}$	$n \times n$ symmetric positive definite "weight" matrix	(10), (12)
$\|\mathbf{X}\|_{\mathbf{W}^{-1}}$	$(\mathrm{Tr}\left(\mathbf{X}\mathbf{W}^{-1}\mathbf{X}^\top\right))^{1/2}$ (weighted Frobenius norm)	(10)
$\mathbf{S}$	A random (sketching) $n \times \tau$ matrix picked from $\mathcal{D}$	
$\mathbf{\Pi_S}$	$\mathbf{S}(\mathbf{S}^\top \mathbf{W}\mathbf{S})^\dagger \mathbf{S}^\top \mathbf{W}$ (stochastic projection matrix)	
$\theta_{\mathbf{S}}$	Bias-correcting random variable	(15) and Assumption 2.1
$\mathbb{E}_{\mathcal{D}}[\cdot]$	$\mathbb{E}_{\mathbf{S}\sim\mathcal{D}}[\cdot]$ (expectation over $\mathbf{S}\sim\mathcal{D}$)	
S or S_k	Sampling (a random subset of $[n]$)	
τ	$\mathbb{E}[|S|]$ (minibatch size)	
C	Subset of $[n]$	
e_C	$\sum_{i\in C} e_i$ (e_i is the ith unit coordinate vector in $\mathbb{R}^d$)	
p_C/p_i	$\mathbb{P}[S=C]/\mathbb{P}[i \in S]$	Sections 1.4 and 4
$\mathbf{I}_C$	Column submatrix of $\mathbf{I}$ with columns indexed by C	Section 4 and Theorem 6
$\mathcal{G} = \mathrm{supp}(S)$	$\{C \subseteq [n] \ : \ p_C > 0\}$ (support of sampling S)	Section 4
f_C	$\frac{1}{|C|}\sum_{i\in C} f_i$ (subsampled loss function)	Section 4 and Theorems 3 and 6
L_C	Smoothness constant of f_C	Sections 1.5 and 4.4 and Theorems 3 and 6
L_i	Smoothness constant of f_i	Sections 1.5 and 4.4
$L_{\max}$	$\max_i L_i$	Sections 1.5 and 4.4 and Theorem 3
L	Smoothness constant of $f = \frac{1}{n}\sum_i f_i$	Sections 1.5 and 4.4 and Theorem 3
$\bar{L}$	$\frac{1}{n}\sum_{i=1} L_i$	Sections 1.5 and 4.4 and Theorem 3
$\mathcal{L}_1$	Expected smoothness constant of the stochastic gradient	Assumption 3.1 and Theorem 1
$\mathcal{L}_2$	Expected smoothness constant of the Jacobian	Assumption 3.2 and Theorem 1
$L_i^{\mathcal{G}}$	$\frac{1}{c_1}\sum_{C \ : \ C\in\mathcal{G},\, i\in C} L_C$	
$L_{\max}^{\mathcal{G}}$	$\max_i L_i^{\mathcal{G}}$ ($=\mathcal{L}_1$ for τ—uniform S with c_1—uniform support)	Sections 1.5 and 4.4 and Theorems 2 and 3
κ	Stochastic contraction number	Section 3.2 and Lemma 2 and Theorem 1
ρ	Sketch residual	(37) and Theorem 1 and Lemma 6
Ψ^k / Ψ_S^k	Lyapunov function/stochastic Lyapunov function	(52)/(109)
c_1	$|\{C \ : \ C \in \mathrm{supp}(S),\, 1 \in C\}|$	Definition 2
c_2	$|\{C \ : \ C \in \mathrm{supp}(S),\, 1 \in C;\, 2 \in C\}|$	(94)

References

1. Achlioptas, D.: Database-friendly random projections: Johnson–Lindenstrauss with binary coins. J. Comput. Syst. Sci. **66**(4), 671–687 (2003)
2. Ailon, N., Chazelle, B.: The fast Johnson–Lindenstrauss transform and approximate nearest neighbors. SIAM J. Comput. **39**(1), 302–322 (2009)
3. Allen-Zhu, Z.: Katyusha: the first direct acceleration of stochastic gradient methods. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing. STOC 2017, pp. 1200–1205 (2017)
4. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 1–27 (2011)
5. Csiba, D., Richtárik, P.: Importance sampling for minibatches. J. Mach. Learn. Res. **19**(1), 962982 (2018)
6. Defazio, A., Bach, F., Lacoste-julien, S.: SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. Adv. Neural Inf. Process. Syst. **27**, 1646–1654 (2014)
7. Defazio, A.J., Caetano, T.S., Domke, J.: Finito: a faster, permutable incremental gradient method for big data problems. In: CoRR arXiv:1407.2710 (2014)
8. Goldfarb, D.: Modification methods for inverting matrices and solving systems of linear algebraic equations. Math. Comput. **26**(120), 829–829 (1972)
9. Goldfarb, D.: A family of variable-metric methods derived by variational means. Math. Comput. **24**(109), 23–26 (1970)
10. Gower, R.M., Richtárik, P., Bach, F.: Stochastic quasi-gradient methods: variance reduction via Jacobian sketching. arXiv:1805.02632 (2018)
11. Gower, R.M., Richtárik, P.: Randomized iterative methods for linear systems. SIAM J. Matrix Anal. Appl. **36**(4), 1660–1690 (2015)
12. Gower, R.M., Richtárik, P.: Randomized quasi-newton updates are linearly convergent matrix inversion algorithms. SIAM J. Matrix Anal. Appl. **38**(4), 1380–1409 (2017)
13. Hickernell, F.J., Lemieux, C., Owen, A.B.: Control variates for quasi-Monte Carlo. Stat. Sci. **20**(1), 1–31 (2005)
14. Hofmann, T., Lucchi, A., Lacoste-Julien, S., McWilliams, B.: Variance reduced stochastic gradient descent with neighbors. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds) NIPS, pp. 2305–2313 (2015)
15. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in Neural Information Processing Systems 26, pp. 315–323. Curran Associates, Inc. (2013)
16. Konečný, J., Richtárik, P.: Semi-stochastic gradient descent methods. Front. Appl. Math. Stat. **3**, 9 (2017)
17. Lin, H., Mairal, J., Harchaoui, Z.: Catalyst acceleration for first-order convex optimization: from theory to practice. J. Mach. Learn. Res. **18**(1), 7854–7907 (2017)
18. Mairal, J.: Incremental majorization–minimization optimization with application to large-scale machine learning. SIAM J. Optim. **25**(2), 829–855 (2015)
19. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course, 1st edn. Springer, Berlin (2014)
20. Nguyen, L.M., Liu, J., Scheinberg, K., Takáč, M.: SARAH: a novel method for machine learning problems using stochastic recursive gradient. In: Precup D., Teh Y.W. (eds) Proceedings of the 34th International Conference on Machine Learning, Vol. 70, pp. 2613–2621. Proceedings of Machine Learning Research (PMLR) (2017)
21. Novikova, N.: A stochastic quasi-gradient method of solving optimization problems in Hilbert space. U.S.S.R. Comput. Math. Math. Phys. **24**(2), 6–16 (1984)
22. Qu, Z., Richtárik, P.: Coordinate descent with arbitrary sampling II: expected separable overapproximation. arXiv:1412.8063 (2014)
23. Qu, Z., Richtárik, P., Zhang, T.: Quartz: Randomized dual coordinate ascent with arbitrary sampling. In: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1. NIPS'15, pp. 865–873. MIT Press, Cambridge (2015)
24. Richtárik, P., Takáč, M.: Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math. Program. **144**(1), 1–38 (2014)

25. Richtárik, P., Takáč, M.: Parallel coordinate descent methods for big data optimization problems. In: Mathematical Programming, pp. 1–52 (2015)
26. Richtárik, P., Takáč, M.: Stochastic reformulations of linear systems: algorithms and convergence theory. arXiv:1706.01108 (2017)
27. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**, 400–407 (1951)
28. Robert, N.L.R., Gower, M., Bach, F.: Tracking the gradients using the Hessian: a new look at variance reducing stochastic methods. In: Proceedings of the 21th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research (2018)
29. Schmidt, M., Le Roux, N., Bach, F.: Minimizing finite sums with the stochastic average gradient. Math. Program. **162**(1), 83–112 (2017)
30. Schmidt, M.W., Babanezhad, R., Ahmed, M.O., Defazio, A., Clifton, A., Sarkar, A.: Non-uniform stochastic average gradient method for training conditional random fields. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9–12, 2015 (2015)
31. Shalev-Shwartz, S.: SDCA without duality, regularization, and individual convexity. arXiv:1602.01582 (2016)
32. Shalev-Shwartz, S., Zhang, T.: Accelerated mini-batch stochastic dual coordinate ascent. Adv. Neural Inf. Process. Syst. **26**, 378–385 (2013)
33. Shalev-Shwartz, S., Zhang, T.: Stochastic dual coordinate ascent methods for regularized loss. J. Mach. Learn. Res. **14**(1), 567–599 (2013). arXiv:1209.1873
34. Varga, R.S.: Eigenvalues of circulant matrices. Pac. J. Math. **1**, 151–160 (1954)
35. Wang, C., Chen, X., Smola, A.J., Xing, E.P.: Variance reduction for stochastic gradient optimization. In: Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds) Advances in Neural Information Processing Systems, vol. 26, pp. 181–189. Curran Associates Inc. (2013)
36. Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. arXiv:1403.4699 (2014)

Affiliations

Robert M. Gower[1] ⬤ · **Peter Richtárik[2,3,4]** · **Francis Bach[5]**

✉ Robert M. Gower
   gowerrobert@gmail.com

[1] LTCI, Telécom Paris, Institut Polytechnique de Paris, Palaiseau, France

[2] King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia

[3] University of Edinburgh, Edinburgh, UK

[4] Moscow Institute of Physics and Technology (MIPT), Dolgoprudny, Russia

[5] INRIA - ENS - PSL Research University, Paris, France