Improving SAGA via a Probabilistic Interpolation with Gradient Descent

Adel Bibi ∗
adel.bibi@kaust.edu.sa

Alibek Sailanbayev ∗
alibek.sailanbayev@kaust.edu.sa

Bernard Ghanem ∗
bernard.ghanem@kaust.edu.sa

Robert Mansel Gower †
gowerrobert@gmail.com

Peter Richtárik ‡
peter.richtarik@kaust.edu.sa

Abstract

We develop and analyze a new algorithm for empirical risk minimization, which is the key paradigm for training supervised machine learning models. Our method—SAGD—is based on a probabilistic interpolation of SAGA and gradient descent (GD). In particular, in each iteration we take a gradient step with probability \( q \) and a SAGA step with probability \( 1 - q \). We show that, surprisingly, the total expected complexity of the method (which is obtained by multiplying the number of iterations by the expected number of gradients computed in each iteration) is minimized for a non-trivial probability \( q \). For example, for a well conditioned problem the choice \( q = \frac{1}{(n - 1)^2} \), where \( n \) is the number of data samples, gives a method with an overall complexity which is better than both the complexity of GD and SAGA. We further generalize the results to a probabilistic interpolation of SAGA and minibatch SAGA, which allows us to compute both the optimal probability and the optimal minibatch size. While the theoretical improvement may not be large, the practical improvement is robustly present across all synthetic and real data we tested for, and can be substantial. Our theoretical results suggest that for this optimal minibatch size our method achieves linear speedup in minibatch size, which is of key practical importance as minibatch implementations are used to train machine learning models in practice. This is the first time linear speedup in minibatch size is obtained for a variance reduced gradient-type method by directly solving the primal empirical risk minimization problem.

1 Introduction

The success of modern machine learning models and applications has been pushing forward the development of algorithms for tackling large scale optimization problems. In particular, the development of stochastic methods for minimizing objective functions with a finite sum structure, such as the empirical risk minimization (ERM) problem \([1]\). In the ERM problem each element of the finite sum structure corresponds to the loss of the model evaluated at a particular data point:

\[
\min_{x \in \mathbb{R}^d} f(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x).
\]  

\( \ast \) King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia.

\( \dagger \) LTCI, Télécom-Paristech, Université Paris-Saclay, Paris, France.

\( \ddagger \) King Abdullah University of Science and Technology (KAUST), Thuwal, Saudi Arabia. University of Edinburgh, United Kingdom. Moscow Institute of Physics and Technology, Dolgoprudny, Russia.
In the big data regime, $n$ is typically very large. This finite sum structure can be leveraged to design efficient optimization algorithms. In this paper, we focus on the case where each $f_i$ is $L_i$-smooth and $f$ is $\mu$-strongly convex, e.g., the regularized ridge regression $f_i(x) = \frac{1}{2}(a_i^T x - y_i)^2 + \frac{1}{2} \|x\|_2^2$, regularized logistic loss $f_i(x) = \frac{1}{2} \log(1 + \exp(-y_i a_i^T x)) + \frac{1}{2} \|x\|_2^2$ and regularized conditional random fields [2].

Variance-reduced methods. In the last five years, variance-reduced stochastic gradient-type methods were devised for solving (1), improving upon the running time of the immensely popular stochastic gradient descent (SGD) method [3,4]. The first such method was SAG [5], which operates by keeping track of the average of the latest stochastic gradients associated with each data point. The method uses a biased estimate of the gradient, and while its analysis is notoriously very technical, it enjoys a linear rate on problem (1) as opposed to a sublinear rate of SGD. This was soon followed by SDCA [6], which requires an explicit strongly convex regularizer to be present in (1), and operates by applying the coordinate descent method [7] to the dual problem. The final complexity rate is the same as that of SAG. One of the most popular variance reduced methods is SVRG [8] (see also S2GD [9]), which relies on a few computations of the full gradient of $f$ in an outer loop, and using this gradient to correct the standard SGD estimator in an inner loop, and leads to an estimator with diminishing variance. In a similar spirit to both SAG and SVRG, SAGA [10] is yet another incremental gradient-based method with the same fast linear convergence rate. SAGA modifies SAG so as to make the gradient estimates unbiased, which leads to a much more streamlined analysis. Similar to SAG, SAGA maintains a table of the latest stochastic gradients computed for each data point, which is then used via an averaging and relaxation step to produce the unbiased gradient estimator.

It is well known that all of the above methods can be improved using several orthogonal tricks, such as mini-batching [11–18], importance sampling [7, 17, 19–22], acceleration [15, 21–26], and higher-order information [27–33]. These approaches lead to further speedup and practical benefits. However, we will not further elaborate on these contributions.

Contributions. Recently, Gower, Richtárik and Bach [34] proposed a novel family of variance reduced methods, called JacSketch, based on iterative Jacobian sketching and controlled stochastic optimization reformulations of [1]. Our work is inspired by their results. In particular, we look more deeply into some exotic special cases of JacSketch not considered in [34] and uncover an interesting phenomenon, which we capture theoretically, and also observe in practice.

In particular, we propose and analyze a new algorithm that interpolates between SAGA and GD in a probabilistic sense. That is, we choose a probability $q \in [0,1]$, and then take a gradient step with probability $q$ and a SAGA step with probability $(1-q)$ . We prove that by choosing $q$ dependent on the condition number of the problem (1) the resulting method always has a better total complexity than either SAGA or GD. Indeed let $\kappa := 4L / \mu$ be this condition number where $L := \sum_{i=1}^n L_i$. If problem (1) is well conditioned as captured by the inequality $\kappa \leq n - 1$, then the simple choice of $q = 1 / (n - 1)^2$ leads to a method with better total complexity than both SAGA and GD. On the other hand, if problem (1) is badly conditioned with $\kappa \geq n - 1$, then choosing $q = \mu / 4nL$ gives a method that also enjoys a better total complexity than both SAGA and GD. These results are captured by Theorem 1.

Motivated by this result, we then modify our method by replacing the GD step with a minibatch SAGA step, with minibatch size $\tau$. Since minibatch SAGA with $\tau = n$ reduces to GD, this is a generalization of our previous method. However, we now have an extra free parameter, $\tau$, and can optimize for it as well. We compute the optimal $(q^*, \tau^*)$ pair minimizing the total complexity of the algorithm using the more relaxed approximation $L_i = L_{\max} := \max_{i=1,...,n} L_i$ of the smoothness constants. Our results show that the optimal minibatch size often is non-trivial, that is, it is neither 1 nor $n$. The resulting method is better than SAGA and minibatch SAGA with any minibatch size, since the choice $(q, \tau) = (0, 1)$ reduces to SAGA and the choice $(q, \tau) = (0, \tau)$ reduces to minibatch SAGA. We show that computing the optimal $(q^*, \tau^*)$ pair can be performed cheaply.

Lastly, we conduct extensive experiments on both synthetic and real datasets to demonstrate that our method is better in practice than both SAGA and GD. We also demonstrate experimentally that the theory matches the experiments in predicting the optimal $(\tau^*)$ given $q^*$. The experiments also suggest that any parallel implementation of our method achieves linear speedup in minibatch size $\tau$. SAGD is the first variance-reduced method for (1) with this property. Linear or superlinear speedup in minibatch size was previously only demonstrated for the SDNA method [13], which applies to...
the dual of a regularized version of (1). However, the super-linear speedup of SDNA is for iteration complexity, and does not take the cost of each iteration into consideration.

2 Probabilistic interpolation of SAGA and gradient descent

Loosely inspired by SVRG [8], which computes a gradient step in an outer loop and then performs variance reduced SGD steps in an inner loop, we propose the following modification of SAGA:

\[
x^{k+1} = \begin{cases} 
\text{Gradient step at } x^k \text{ with a stepsize } \alpha & \text{with prob } q, \\
\text{SAGA step at } x^k \text{ with a stepsize } \alpha & \text{with prob } (1-q), 
\end{cases}
\]

where \( q \in [0,1] \). Every so often (i.e., with “small” probability \( q \)), the current table (matrix) of gradient estimates \( J^k \in \mathbb{R}^{d \times n} \) is reset to consist of the true gradients \( J^k \leftarrow [f'_1(x^k), f'_2(x^k), \ldots, f'_n(x^k)] \), where \( f'_i(.) \in \mathbb{R}^d \), and with probability \( (1-q) \) one column of the table is updated such that is \( J^k = f'_1(x^k) \). However, we observe two major differences between the proposed algorithm and SVRG: (i) the number of inner and outer loop iterations are not fixed but are randomly controlled by \( q \), and (ii) with this new method, all the samples that are used to build the unbiased estimator of the gradient are used to update the full gradient estimate. This is in contrast with SVRG, where only one of the samples of the gradient step update in the inner loop is later used to compute the total gradient in the outer loop.

Note that a somewhat similar albeit fundamentally different strategy is employed in the CSGS method [10] of Lei and Jordan, which modifies SVRG so that the outer loop estimates the gradient via a minibatch, as opposed to the full batch. From this perspective, we work with SAGA instead of SVRG, and include a probabilistic interpolation with minibatch SAGA in order to reduce variance more aggressively every so often, instead of doing this in an outer loop, as in CSGS.

To analyze this interpolation scheme, we consider the total complexity \( \Omega^{q,n} = \Omega^{q,n}_1 \times \Omega^{q,n}_2 \) where \( \Omega^{q,n}_1 \) denotes the iteration count and \( \Omega^{q,n}_2 \) is the average complexity per iteration. Given that computing the \( f'_i \) gradient is the dominating cost, it is clear that under such a sampling \( \Omega^{q,n}_2 = q(n-1)+1 \). That is, on average \( q(n-1)+1 \) gradients are computed per iteration. In what follows we showcase that in fact there exists a non trivial simple choice for \( q \) that is better than both SAGA and GD.

**Theorem 1.** For the update (5), consider the Lyapunov function

\[
\Psi^k \triangleq \|x^k - x^*\|^2 + \alpha \frac{2L_{\max}(q(n-1) + 1)}{\mu} \|J^k - \nabla F(x^*)\|^2_F,
\]

where \( J^k \) is the table of gradients and \( \nabla F(x^*) = [f'_1(x^*), f'_2(x^*), \ldots, f'_n(x^*)] \). If the problem is well conditioned with \( \frac{4L}{\mu} \leq n-1 \), then by choosing \( q = \frac{1}{(n-1)^2} \) and using the stepsize

\[
\alpha = \frac{1}{4(1 - \frac{2}{n})L_{\max} + \mu(n-1)},
\]

the expected total complexity to achieve \( \epsilon \) accuracy, that is \( \mathbb{E}[\Psi^k] \leq \epsilon \Psi^0 \), is given by

\[
\Omega^{\frac{1}{4(1 - \frac{2}{n})L_{\max} + \mu(n-1)}}^{q,n} = \left( n + \frac{n - 2}{n - 1} \frac{4L_{\max}}{\mu} \right) \log \left( \frac{1}{\epsilon} \right),
\]

On the other hand, if the problem is badly conditioned with \( \frac{4L}{\mu} \geq n-1 \), then by choosing \( q = \frac{\mu}{4nL} \) and using the stepsize

\[
\alpha = \frac{1}{4L \mu n \left( (4L + \frac{\mu}{n}) - \mu \right)} 
\]

the total complexity is

\[
\Omega^{\frac{4L + \frac{\mu}{n} - \mu}{4L \mu n} - \mu}^{q,n} = \left( n + \frac{4L_{\max}}{\mu} \left( 1 - \frac{1}{4L + \frac{\mu}{n} - \mu} \right) \right) \log \left( \frac{1}{\epsilon} \right).
\]

The details of the proof are left to the supplementary material. Comparing the total complexity of SAGA, which is given by \( \Omega^{q,n} = (n + 4L_{\max}/\mu) \log(1/\epsilon) \), to the total complexity results
As for (6) we have from
$$4 n \sum_{i=1}^{n} \phi_{i} \xrightarrow{\theta \to \infty}$$
converge to the complexity of SAGA asymptotically as performance than both SAGA and GD. Though we note that both complexity results (4) and (6) are particularly interested in this question since there are currently no guidelines on how to choose the minibatch size for SAGA type methods. It has been observed in practice that using a small minibatch for SAGA is better in terms of effective passes over the data, as compared to sampling a single data point at each iteration (that is, the standard SAGA method). But using too large a minibatch can degrade the performance. Thus the need for theoretical guidelines.

The complexity of our algorithm is detailed in the following theorem:

Algorithm 1 SAGD: Interpolation between SAGA and minibatch SAGA

**Initialization:** $x^{0} \in \mathbb{R}^{d}$, $q$, batch size $\tau$, $\theta = \frac{n}{q(\tau-1)+1}$, stepsize $\alpha$ and randomly initialize table of $\phi_{i} \in \mathbb{R}^{d}$ $\forall i \in \{1, \ldots, n\}$

**while** $error > \epsilon$ **do**

(i) with probability $\frac{q}{\binom{n}{\tau}}$, sample $\tau$ indices $I_{k}$

Calculate $f_{j}^{\tau}(x^{k})$ for $j \in I_{k}$

$x^{k+1} \leftarrow x^{k} - \alpha \left( \frac{\theta}{n} \sum_{j \in I_{k}} (f_{j}^{\tau}(x^{k}) - \phi_{j}) + \frac{1}{n} \sum_{i=1}^{n} \phi_{i} \right)$; \hspace{1cm} // minibatch SAGA step

Update the table: $\phi_{j} = \begin{cases} \phi_{j}, & \text{if } j \not\in I_{k} \\ f_{j}^{\tau}(x^{k}), & \text{if } j \in I_{k} \end{cases}$

(ii) with probability $1 \frac{1}{n}$, select a sample $j$ uniformly at random;

Calculate $f_{j}(x^{k})$

$x^{k+1} \leftarrow x^{k} - \alpha \left( \frac{\theta}{n} (f_{j}(x^{k}) - \phi_{j}) + \frac{1}{n} \sum_{i=1}^{n} \phi_{i} \right)$; \hspace{1cm} // SAGA step

Update the table: $\phi_{i} = \begin{cases} \phi_{i}, & \text{if } i \neq j \\ f_{j}(x^{k}), & \text{if } i = j \end{cases}$

end

and (6), we see that the total complexity of our method is always better than that of SAGA. Indeed, this is trivially true for (4) and as for (6) it follows from $4 \frac{L}{\mu} \geq n - 1$ that

$$\frac{4 L}{\mu} \geq n - 1 \Rightarrow \frac{4 L}{\mu} + 1 - \frac{1}{n} \geq 1 \Leftrightarrow 0 \leq \frac{1}{\frac{4 L}{\mu} + 1 - \frac{1}{n}} \leq 1.$$ (7)

Thus the simple choice of the interpolation parameter $q$ given in Theorem 1 exhibits strictly better performance than both SAGA and GD. Though we note that both complexity results (4) and (6) converge to the complexity of SAGA asymptotically as $n \to \infty$. Indeed this is again trivial for (4).

As for (6) we have from $4 \frac{L}{\mu} \geq n - 1$ that

$$1 - \frac{1}{\frac{4 L}{\mu} + 1 - \frac{1}{n}} \geq 1 - \frac{1}{n - 1 + \frac{1}{n}} = 1 - \frac{1}{n - \frac{n}{n}} \frac{n}{n} \frac{n}{n} \to \infty 1,$$

and consequently $\Omega \frac{L}{4 \sqrt{\mu}} n$ is asymptotically lower bounded by the complexity of SAGA. Finally since $\Omega \frac{L}{4 \sqrt{\mu}} n$ is always upper bounded by the complexity of SAGA (which follows from (7)) we have that

$$\lim_{n \to \infty} \Omega \frac{L}{4 \sqrt{\mu}} n = (n + 4 L_{\max}/\mu) \log(1/\epsilon).$$

It is salient to note that we do not claim that such $q$ is optimal, it is only an example that a non-trivial interpolation exists that improves upon SAGA and GD. Such a result motivates the direction towards finding the optimal $q^*$ that directly minimizes the total complexity $\Omega n$ in a more general setting.

3 Probabilistic interpolation of SAGA and minibatch SAGA

Now let us consider the more general method that interpolates between SAGA and minibatch SAGA:

$$x^{k+1} = \begin{cases} \tau\text{-minibatch SAGA step of } x^{k} \text{ with stepsize } \alpha, & \text{with prob } q, \\ \text{SAGA step of } x^{k} \text{ with stepsize } \alpha, & \text{with prob } 1 - q. \end{cases}$$ (8)

Note that when $\tau = n$, the model in (8) reduces to the interpolation between gradient descent and SAGA described in (2). In this section, we investigate the values of $(q, \tau)$ that achieve the best total complexity. We are particularly interested in this question since there are currently no guidelines on how to choose the minibatch size for SAGA type methods. It has been observed in practice that using a small minibatch for SAGA is better in terms of effective passes over the data, as compared to sampling a single data point at each iteration (that is, the standard SAGA method). But using too large a minibatch can degrade the performance. Thus the need for theoretical guidelines.

The complexity of our algorithm is detailed in the following theorem:
Theorem 2. (Convergence of SAGD) Consider Algorithm 1 for solving problem (7). Assume that the functions $f_i$ are $L_i$-smooth, and that $f$ is strongly convex with parameter $\mu > 0$. Consider the Lyapunov function:

$$\psi^k \triangleq \|x^k - x^*\|^2_2 + \frac{\theta \alpha}{2n L_{\max}} \|J^k - \nabla F(x^*)\|^2_F,$$

where $\theta = \frac{n}{q(\tau - 1) + 1}$. Let

$$L_1 = \frac{1}{(q(\tau - 1) + 1)^2} \left( \left(q \left( \frac{\tau(n - \tau)}{n - 1} - 1 \right) + 1 \right) L_{\max} + n q \left( \frac{\tau(n - 1)}{n - 1} \right) \right),$$

which is known as the expected smoothness constant, and let

$$\rho = \begin{cases} \theta^2 \left( \frac{1 - n}{n} + q \left( \frac{n - \tau}{n - 1} \right) \right), & q \theta^2 < \frac{n - 1}{\tau - 1} \\ \theta^2 \left( \frac{1 - n}{n} + q \left( \frac{n - \tau}{n - 1} \right) \right) + n \left( \theta^2 q \left( \frac{n - 1}{n - 1} \right) - 1 \right), & q \theta^2 > \frac{n - 1}{\tau - 1}. \end{cases}$$

Choose $0 \leq q \leq 1$ and $\tau \in [n]$. If the stepsize is given by

$$\alpha = \min \left\{ \frac{1}{4L_1}, \frac{n}{4L_{\max} \rho + \mu n} \right\},$$

then the expected total complexity in order to achieve $\epsilon$ accuracy, that is $\mathbb{E}[\psi^k] \leq \epsilon \psi^0$, is

$$\Omega^{q,\tau} = (q(\tau - 1) + 1) \max \left\{ \frac{4L_1}{\mu}, \theta + \frac{4\rho L_{\max}}{\mu n} \right\} \log \left( \frac{1}{\epsilon} \right).$$

The proof is left for the supplementary material.

4 Theoretical analysis

In this section, we revisit the total complexity result from Theorem 2 and derive the optimal $(q^*, \tau^*)$ pair that minimizes the total complexity (12). In order to simplify the expression in (12), we use the lax estimate $L_i = L_{\max}$ for the individual smoothness constants for $\forall i = 1 \ldots n$. First we note that

Lemma 1. $\frac{4L_1}{\mu} \geq 0$, $\theta + \frac{4\rho L_{\max}}{\mu n} \geq 0$ and $q(\tau - 1) + 1 \geq 0 \ \forall q \in [0, 1]$ and $\forall \tau$.

In light of Lemma 1, the total complexity (12) can be rewritten in the form of $\Omega^{q,\tau} = \max \{g_1, g_2\} \log(1/\epsilon)$ where $g_1 \triangleq \left( \frac{4L_1}{\mu} \right) (q(\tau - 1) + 1)$ and $g_2 \triangleq \left( \theta + \frac{4\rho L_{\max}}{\mu n} \right) (q(\tau - 1) + 1)$.

To obtain a $\tau$ and $q$ that optimize the complexity, we need to analyze how a change in $\tau$ or $q$ affects the functions $g_1$ and $g_2$.

Lemma 2. The function $g_1$ is monotonically increasing in $q \in [0, 1]$.

Lemma 3. The function $g_2$ is concave in $q$ in the interval $[q_-, q_+]$ and monotonically decreasing in $q \in [0, q_-] \cup [q_+, 1]$, where $q_{\pm} = \frac{n + 2(1-n)\sqrt{n}\sqrt{4(1-n)+n\tau}}{2(n-1)(\tau-1)}$.

Note that $q_{\pm}$ are the solutions to the polynomial $q \theta^2 = \frac{n - 1}{\tau - 1}$ in which $\rho$ changes sign. Moreover, it is easy to show that $q_{\pm}$ are valid probabilities through the following Lemma:

Lemma 4. For $\tau \geq 4$, $q_{\pm}$ are respectively increasing and decreasing functions in $\tau$; therefore,

$$\frac{1}{n - 1} \leq q_- \leq \frac{n + 1 - 2\sqrt{n}}{3(n - 1)} \leq \frac{1}{3} \leq \frac{n + 1 + 2\sqrt{n}}{3(n - 1)} \leq q_+ \leq 1.$$

The previous Lemma asserts that $q_-$ and $q_+$ are valid probabilities for all values of $\tau$. At last, we analyze the points of intersections between $g_1$ and $g_2$ in the following Lemma.

Lemma 5. The functions $g_1$ and $g_2$ intersect at exactly $g_{i1} = \frac{n - 1}{(\tau - 1)(\tau - 1) + 1 - n}$ for $\tau \in [\tau_{\min}, \tau_{\max}]$ and at $g_{i2} = \frac{n - 4L_{\max}}{4L_{\max} \tau(\tau - 1)}$ for $\tau \in (\tau_{\max}, n)$ where $\tau_{\min} = \frac{n}{4L_{\max} \mu} + 1 - \frac{\mu}{4L_{\max}}$, $\tau_{\max} = \min \left( \tau_{\max}, n \right)$ $1 > n \geq \frac{4L_{\max}}{\mu} + n \geq \frac{4L_{\max}}{\mu}$ and lastly $\tau_{\max} = \left( \frac{n(n - 1)\mu}{4L_{\max} \mu} \right)$.
Figure 1: Shows two different examples of the functions $g_1$ and $g_2$. As proven in Lemmas\textsuperscript{2} and \textsuperscript{3}, $g_1$ is monotonically increasing $\forall q \in [0, 1]$ while $g_2$ is concave on $[q_-, q_+]$ and decreasing elsewhere. It is also to be noted that the two functions intersect at most one point.

Note that $\frac{4L_{\max}}{\mu}$ is an indicator function such that $\frac{4L_{\max}}{\mu} = 1$ if $n > \frac{4L_{\max}}{\mu}$ and zero elsewhere. It is important to note that for a given $\tau$ the two functions intersect at exactly one point. This is due to the fact that $q_{11}$ and $q_{22}$ occur in a non-overlapping intervals of $\tau$. Lastly, and as an illustration to the behaviour of the two functions $g_1$ and $g_2$, we plot in Figure 1 both functions $g_1$ and $g_2$ under two different choices of $n, \mu, L_{\max}$. Note that $g_1$ and $g_2$ exhibit the behaviour described in Lemmas\textsuperscript{2} and \textsuperscript{3}. Now, we are ready to present the third main result.

**Theorem 3.** The optimal $q^*$ of the total complexity in (12) is either one of the intersection points $(q_{11}, q_{22})$ or $q_-$ or $q = 1$. That is $q^* \in \{1, q_-, q_{11}, q_{22}\}$; therefore,

$$\tau^* = \arg \min_{\tau} \left( \Omega^1, \Omega^{q_-, \tau}, \Omega^{q_{11}, \tau}, \Omega^{q_{22}, \tau} \right)$$

(13)

$$q^* = \arg \min_{q} \left( \Omega^1, \Omega^{1, \tau}, \Omega^{q_-, \tau}, \Omega^{q_{11}, \tau}, \Omega^{q_{22}, \tau} \right)$$

(14)

**Proof.** The proof is a direct implication of the previous lemmas. \hfill \Box

**Discussion.** A key consequence of Theorem 3 is that SAGA ($q = 0$) is not optimal in the sense of total complexity. Moreover, Theorem 3 provides a consistent way to compute the optimal $(q^*, \tau^*)$ that interpolates between SAGA and minibatch SAGA that minimizes the total complexity. Such a pair is computed once before the algorithm. Moreover, solving (13) may seem to be expensive since it requires the computation of $4n$ scalar values. However, the computation is in fact much smaller than $4n$. As discussed in Lemma 3 the intervals of $\tau$ in which the intersection points $(q_{11}, q_{22})$ occur are in fact non-overlapping. That is, computing $g_{11}$ and $g_{22}$ require at most $n$ scalar computations. Moreover, for $q = 1$ the total complexity is $\Omega^1, \max \{g_1 = \frac{4L_{\max}}{\mu}, g_2 = n + \frac{4L_{\max}}{\mu} n-\tau \}$, respectively. Since for $q = \tau = 1$, $g_2 > g_1$; therefore, the optimal minibatch size $\tau^*$ occurs at the intersection point $\tau = \text{round} \left(1 + \frac{\mu(n-1)}{4L_{\max}} \right)$. Therefore, we only need to compute a single scalar for when $q = 1$. Moreover, this result is particularly important as it provides a formula for the optimal minibatch size of SAGA. Lastly, as for $q_-$, it is only computed for when $\tau \geq 4$. Therefore, the total number of scalar computations required is at most $2(n-1)$ and they are computed once before running the algorithm. Therefore, (13) can be simplified to:

$$\tau^* = \arg \min_{\tau} \left( \Omega^{1, \text{round}(1 + \frac{\mu(n-1)}{4L_{\max}})}, \Omega^{q_-, \tau \geq 4, \Omega^{q_{11}, \tau \in [\tau_{\min}, \tau_{\max}]}, \Omega^{q_{22}, \tau \in (\tau_{\max}, n)} \right).$$

**5 Experiments**

In this section, we conduct extensive experiments comparing the performance of SAGD to SAGA as a baseline. The comparisons against GD were omitted as GD was too slow compared to SAGA. The experiments were conducted on several synthetic and real datasets demonstrating that not only SAGD is better in practice but also that SAGD can achieve linear speed up with minibatch size $\tau^*$ for a parallel implementation. We have also conducted experiments to verify how accurately the theory
Figure 2: Shows a comparison between SAGA and our method with optimal \((q^*, \tau^*)\) pair. We also compare them both against \((q^*, \frac{\tau}{2})\) and \((q^*, 2\tau^*)\). The first row shows a comparison in time while the second row shows the same experiment as compared in the total number of effective passes.

predicts the practical optimal batch size \(\tau^*\). In this section, we consider the problem of regularized ridge regression:

\[
f(x) = \frac{1}{2n} \|Ax - y\|^2 + \frac{\lambda}{2} \|x\|^2, \tag{15}\]

where \(A \in \mathbb{R}^{n \times d}\), \(y \in \mathbb{R}^n\) are the data and \(\lambda > 0\) is the regularization parameter. However, further experiments on the regularized logistic loss are left for the supplementary material. The code was implemented in MATLAB16a and all experiments were run on a dual processor 2.66GHz machine.

Any experiments that compare time discarded the time required to read data and the time to perform random sampling. This is due to the assumption that all such data can be hashed and achieved with \(O(1)\) complexity. This will eliminate any dependency on the platform. As for the synthetic data, the rows of \(A\) and \(y\) were sampled from the standard Gaussian distribution \(\mathcal{N}(0, 1)\) with a regularization parameter set to \(\lambda = \frac{1}{n}\). In some few experiments, the data \(A\) and \(y\) were sampled from a uniform distribution with somewhat different choice of \(\lambda\). The details and reasons behind such setups for the experiments will be detailed later. In all experiments, the rows of \(A\) were normalized such that \(\|A(i,:)|_2 = 1 \ \forall i\). Note that for ridge regression, we have \(L_i = \|A(i,:)|_2^2 + \lambda = 1 + \lambda \ \forall i\) and that \(\mu = 1/n \lambda_{\min}(A^T A) + \lambda\). In all experiments, we compute a high accuracy solution \(x^*\) by solving the linear system in (15). The metric used in all experiments is the error \(\|x - x^*\|_2\).

**Synthetic Experiments.** In the synthetic experiments, we compare SAGD to SAGA across different choices of \(n\) and \(d\). As can be noted in Figure 2, SAGD with optimal interpolation factor and minibatch size \((q^*, \tau^*)\) consistently outperforms SAGA in time. The same comparison was also conducted against several other non optimal choices of minibatch size, namely \((2\tau^*\) and \(\tau^*/2)\). In practice, SAGD with \((q^*, \tau^*)\) performs very similarly to SAGD with \((q^*, \tau^*/2)\). However, SAGD with \((q^*, \tau^*\) is still favorable as it allows for larger batch sizes suitable for parallelization. Figure 2 also shows a comparison among SAGA, SAGD with \((q^*, \tau^*)\), and SAGD with non optimal minibatch sizes \((\tau^*/2\) and \(2\tau^*)\) on the number of effective data passes. It is clear that SAGD with \((q^*, 2\tau^*)\) always requires a larger number of effective data passes to achieve the same accuracy. More interestingly, both SAGA and SAGD with both \((q^*, \tau^*)\) and \((q^*, \tau^*/2)\) can achieve the same \(\epsilon\) accuracy in the same number of effective data passes. This suggests that upon having a parallel implementation of SAGD, one can achieve a linear speedup in minibatch size \(\tau^*\) outperforming both competitors. This is the first time linear speedup in minibatch size is obtained for a variance reduced gradient-type method by directly solving the primal empirical risk minimization problem. Similarly, we conduct even further synthetic experiments but on larger dimensional problems. Figure 3 demonstrates that SAGD with \((q^*, \tau^*)\) is either slightly faster or similar to SAGA in time. However, similar to the con-
clusion drawn in Figure 2, SAGD with \((q^*, \tau^*)\) can have significant speed ups outperforming SAGA in a parallel implementation. As most of the previous experiments had \(q^*\) as optimal interpolation factor, we showcase the existence for other non trivial \(q^*\). For instance, if we set \(\lambda = 3/(2n)\) for the dataset with \(n = 10000\) and \(d = 100\), the problem’s condition number \(\frac{\lambda}{\mu}\) changes that the optimal probabilistic interpolation is non trivially \(q^* = 0.952\). This further demonstrates the existence of a non-trivial interpolation that can be better than SAGA and any other minibatch version of SAGA. Moreover, we changed the distribution from which \(A\) and \(y\) are sampled to a uniform distribution where we set \(\lambda = 401/(\sqrt{n^3})\) for the experiment with \(n = 100000\) and \(d = 100\). Such a setup also demonstrated the existence of a non trivial \(q^* = 0.847\) as an optimal interpolation factor. In all synthetic experiments, one can observe that SAGD is always at least as good as SAGA with the former enjoying a linear speedup in \(\tau\) with a parallel implementation.

**Sensitivity to Optimal \(\tau^*\).** In this experiment, we demonstrate that the theoretical results of the optimal minibatch size \(\tau^*\) for a given \(q^*\) match the one observed in practice. We conduct experiments on six different synthetic datasets that vary in both the number of samples \(n\) and the problem dimension \(d\). In all experiments, we compare the number of effective data passes required to achieve \(\|x - x^*\|_2 \leq 10^{-10}\) accuracy over multiple choices of \(\tau\). First as can be noted from Figure 3, increasing the minibatch size beyond a certain value will linearly increase the number of required data passes to achieve the same accuracy. Moreover, the theoretical optimal minibatch size \(\tau^*\) is very close to the best \(\tau\) observed in practice. In the few cases where the theoretical \(\tau^*\) does not precisely match the best \(\tau\) observed in practice, the difference in the number of required data passes to achieve \(\epsilon\) accuracy is very small. For instance, for the experiment \((n = 1000, d = 10)\) the \(\tau^*\) minibatch required 34 data passes for achieving \(\epsilon\) accuracy as compared to the 33 passes for the best \(\tau\) in practice. The difference is also 1 data pass between \(\tau^*\) and the best \(\tau\) is for the experiments \((n = 1000, d = 100, 10000)\). As for the experiments \((n = 10000, d = 10, 100)\), the difference is 2 data passes. Lastly for \((n=10000,d = 1000)\), the theory precisely predicts \(\tau^* = 3\) in practice.

**Real Datasets.** In this subsection, we conduct similar types of experiments as the one described in the synthetic datasets but on real data. Three real datasets used in the experiments are from the LIBSVM database [37] namely, *ijcnn1, a9a* and *australian*. We compare SAGA against SAGD with \((q^*, \tau^*)\) and two non optimal minibatch sizes \((q^*, \tau^*/2)\) and \((q^*, 2\tau^*)\). On the *a9a* and *australian* datasets, we set \(\lambda = 10\) to showcase an example of the non trivial \(q = 0.555\) and \(q = 0.534\), respectively. As can be noted again from Figure 3, a direct linear speedup of our proposed algorithm can be achieved with a parallel implementation of SAGD on real datasets too.
6 Conclusion

We have demonstrated that the optimal interpolation between SAGA and gradient descent is not trivially SAGA. We thus presented formulas for the optimal probabilistic interpolation factor $q$ between SAGA and the optimal minibatch SAGA of size $\tau$. We demonstrated experimentally that the proposed algorithm is consistently better than SAGA on both synthetic and real dataset. We have also shown that the theory predicts what is observed in practice for optimal batch size $\tau^*$ of minibatch SAGA. More interestingly, the experiments strongly suggest that one can achieve linear speedup for a parallel implementation of our method.
References

[1] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: from theory to algorithms*. Cambridge University Press, 2014.

[2] M. W. Schmidt, R. Babanezhad, M. O. Ahmed, A. Defazio, A. Clifton, and A. Sarkar, “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, 2015.

[3] H. Robbins and S. Monro, “A stochastic approximation method,” *Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[4] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, “Robust stochastic approximation approach to stochastic programming,” *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1574–1609, 2009.

[5] M. Schmidt, N. Le Roux, and F. Bach, “Minimizing finite sums with the stochastic average gradient,” *Mathematical Programming*, vol. 162, no. 1-2, pp. 83–112, 2017.

[6] S. Shalev-Shwartz and T. Zhang, “Stochastic dual coordinate ascent methods for regularized loss minimization,” *Journal of Machine Learning Research*, vol. 14, no. Feb, pp. 567–599, 2013.

[7] P. Richtárik and M. Takáč, “Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function,” *Mathematical Programming*, vol. 144, pp. 1–38, 2014.

[8] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” in *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

[9] J. Konečný and P. Richtárik, “S2GD: Semi-stochastic gradient descent methods,” *Frontiers in Applied Mathematics and Statistics*, pp. 1–14, 2017.

[10] A. Defazio, F. Bach, and S. Lacoste-Julien, “SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives,” in *Advances in neural information processing systems*, pp. 1646–1654, 2014.

[11] M. Takáč, A. Bijral, P. Richtárik, and N. Srebro, “Mini-batch primal and dual methods for SVMs,” in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[12] P. Richtárik and M. Takáč, “Parallel coordinate descent methods for big data optimization,” *Mathematical Programming*, vol. 156, no. 1-2, pp. 433–484, 2016.

[13] R. Tappenden, M. Takáč, and P. Richtárik, “On the complexity of parallel coordinate descent,” *Optimization Methods and Software*, vol. 32, no. 2, pp. 372–395, 2018.

[14] J. Konečný, J. Lu, P. Richtárik, and M. Takáč, “Mini-batch semi-stochastic gradient descent in the proximal setting,” *IEEE Journal of Selected Topics in Signal Processing*, 2016.

[15] S. Shalev-Shwartz and T. Zhang, “Accelerated mini-batch stochastic dual coordinate ascent,” in *Advances in Neural Information Processing Systems 26*, pp. 378–385, 2013.

[16] M. Takáč, P. Richtárik, and N. Srebro, “Distributed mini-batch SDCA,” to appear in *Journal of Machine Learning Research* (arXiv:1507.08322), pp. 1–15, 2015.

[17] D. Csiba and P. Richtárik, “Importance sampling for minibatches,” to appear in *Journal of Machine Learning Research* (arXiv:1602.02283), 2016.

[18] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams, “Variance reduced stochastic gradient descent with neighbors,” in *Advances in Neural Information Processing Systems*, pp. 2305–2313, 2015.

[19] P. Richtárik and M. Takáč, “On optimal probabilities in stochastic coordinate descent methods,” *Optimization Letters*, vol. 10, no. 6, pp. 1233–1243, 2016.

[20] Z. Qu, P. Richtárik, and T. Zhang, “Quartz: Randomized dual coordinate ascent with arbitrary sampling,” in *Advances in Neural Information Processing Systems 28*, pp. 865–873, 2015.

[21] P. Richtárik and M. Takáč, “Coordinate descent with arbitrary sampling I: algorithms and complexity,” in *Optimization Methods and Software*, vol. 31, no. 5, pp. 829–857, 2016.

[22] Z. Allen-Zhu, Z. Qu, P. Richtárik, and Y. Yuan, “Even faster accelerated coordinate descent using non-uniform sampling,” in *International Conference on Machine Learning*, pp. 1110–1119, 2016.

[23] O. Fercoq and P. Richtárik, “Accelerated, parallel and proximal coordinate descent,” *SIAM Journal on Optimization*, vol. 25, pp. 1997–2023, 2015.

[24] O. Fercoq, Z. Qu, P. Richtárik, and M. Takáč, “Fast distributed coordinate descent for minimizing non-strongly convex losses,” *IEEE International Workshop on Machine Learning for Signal Processing*, 2014.

[25] Q. Lin, Z. Lu, and L. Xiao, “An accelerated proximal coordinate gradient method,” in *Advances in Neural Information Processing Systems*, pp. 3059–3067, 2014.

[26] Z. Allen-Zhu, “Katyusha: The first direct acceleration of stochastic gradient methods,” in *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1200–1205, ACM, 2017.
[27] M. W. Schmidt, E. V. D. Berg, M. P. Friedlander, and K. Murphy, “Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm,” *International Conference on Artificial Intelligence and Statistics*, vol. 5, pp. 456–463, 2009.

[28] M. Schmidt, D. Kim, and S. Sra, “Projected Newton-type methods in machine learning,” in *Optimization for Machine Learning* (S. Sra, S. Nowozin, and S. J. Wright, eds.), pp. 1–25, The MIT press, 2011.

[29] M. A. Erdogdu and A. Montanari, “Convergence rates of subampled Newton methods,” in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), pp. 3052–3060, Curran Associates, Inc., 2015.

[30] R. M. Gower, D. Goldfarb, and P. Richtárik, “Stochastic block BFGS: Squeezing more curvature out of data,” in *Proceedings of the 33rd International Conference on Machine Learning*, 2016.

[31] P. Moritz, R. Nishihara, and M. I. Jordan, “A linearly-convergent stochastic L-BFGS algorithm,” in *International Conference on Artificial Intelligence and Statistics*, pp. 249–258, 2016.

[32] R. M. Gower, N. Le Roux, and F. Bach, “Tracking the gradients using the Hessian: A new look at variance reducing stochastic methods,” *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, vol. 84, pp. 707–715, 2018.

[33] R. M. Gower, F. Hanzely, F. Richtárik, and S. Stich, “Accelerated stochastic matrix inversion: general theory and speeding up BFGS rules for faster second-order optimization,” *arXiv:1801.05490*, 2018.

[34] R. M. Gower, P. Richtárik, and F. Bach, “Stochastic quasi-gradient methods: variance reduction via Jacobian sketching,” *arXiv:1805.02632*, 2018.

[35] Z. Qu, P. Richtárik, M. Takáč, and O. Fercoq, “SDNA: stochastic dual Newton ascent for empirical risk minimization,” in *Proceedings of The 33rd International Conference on Machine Learning*, pp. 1823–1832, 2016.

[36] L. Lei and M. I. Jordan, “Less than a single pass: Stochastically controlled stochastic gradient,” in *PMLR: Proceedings of Machine Learning Research (AISTATS 2017)*, vol. 54, 2017.

[37] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[38] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course (Applied Optimization)*. Kluwer Academic Publishers, 2004.

[39] R. S. Varga, “Eigenvalues of circulant matrices,” *Pacific J. Math.*, vol. 1, pp. 151–160., 1954.
A Proof of Theorem 1

The proof will follow by simply computing the constants in Theorem 2 and then analyzing the resulting complexity.

When \( \tau = n \) we have that

\[
\theta = \frac{n}{q(\tau - 1) + 1} = \frac{n}{q(n - 1) + 1}
\]

and consequently

\[
\rho = \frac{n(1 - q)}{(q(n - 1) + 1)^2} + n \left( \frac{n^2 q}{(q(n - 1) + 1)^2} - 1 \right) \quad \begin{cases} 0 \leq q \leq \frac{1}{(n - 1)^2} \\ q \geq \frac{1}{(n - 1)^2} \end{cases}
\] (16)

Furthermore from (9) and \( \tau = n \) we have

\[
\mathcal{L}_1 = \frac{(1 - q)L_{\max} + qn^2 L}{(q(n - 1) + 1)^2}.
\] (17)

Inserting (16) and (17) into (11) gives a stepsize of

\[
\alpha = \min \left\{ \frac{1}{4 \mu} \frac{(q(n - 1) + 1)^2}{(1 - q)L_{\max} + qn^2 L} \frac{n(q(n - 1) + 1)}{\mu n^2 + 4 \rho L_{\max}(q(n - 1) + 1)} \right\}
\] (18)

and a total complexity (12) of

\[
\Omega^{n, \tau} = (q(n - 1) + 1) \max \left\{ \frac{4(1 - q)L_{\max} + qn^2 L}{\mu (q(n - 1) + 1)^2}, \frac{n}{q(n - 1) + 1} + \frac{4\rho L_{\max}}{\mu n} \right\} \log \left( \frac{1}{\epsilon} \right)
\] (19)

Now we separate the analysis into two cases depending on the condition number \( \frac{\rho^T}{\mu} \).

Consider the well conditioned case where \( \frac{\rho^T}{\mu} \leq n - 1 \) and let \( q = \frac{1}{(n - 1)^2} \). From (16) we have that

\[
\rho = \frac{n(1 - q)}{(q(n - 1) + 1)^2} = \frac{n(1 - (n - 1)^2)}{(1 - (n - 1)^2)^2(n - 1)^2} = \frac{n(n - 1)^2(n - 1)^2 - 1}{n(n - 1)^2(n - 1)^2} = n - 2,
\]

which together with \( q = \frac{1}{(n - 1)^2} \) when plugged into (18) gives (3), and when plugged into (19) gives

\[
\Omega^{n, \tau} = \max \left\{ \frac{n - 2}{\mu} \frac{4L_{\max}}{n - 1} + \frac{n}{\mu} + \frac{n - 2}{\mu} L_{\max} \right\} \log \left( \frac{1}{\epsilon} \right)
\]

\[
\frac{\rho^T}{\mu} \leq n - 1 \Rightarrow n + \frac{n - 2}{\mu} L_{\max} \log \left( \frac{1}{\epsilon} \right).
\]

Now consider the badly conditioned case where \( \frac{\rho^T}{\mu} \geq n - 1 \) and let \( q = \frac{\mu}{4nL} \). It follows that \( q \leq \frac{1}{(n - 1)^2} \) since

\[
q = \frac{\mu}{4nL} \leq \frac{1}{n(n - 1)} \leq \frac{1}{(n - 1)^2}.
\]

Consequently

\[
\rho = \frac{n(1 - q)}{(q(n - 1) + 1)^2} = \frac{n(1 - \frac{\mu}{4nL})}{(\frac{\mu}{4nL}(n - 1) + 1)^2} = \frac{4n^2 L(4nL - \mu)}{(\mu(n - 1) + 4nL)^2}.
\]
Plugging in the above and \( q = \frac{\mu}{4nL} \) into \( X(q) \) defined in (19) gives

\[
X(q) = \frac{4}{\mu} \frac{(1 - \frac{\mu}{4nL})L_{\text{max}} + \frac{\mu}{4nL}n^2L}{(n - 1) + 1} = \frac{4L_{\text{max}}}{\mu} \frac{4nL - \mu}{n(4L + \mu) - \mu} + \frac{4L}{\mu} \frac{n^2\mu}{n(4L + \mu) - \mu}
\]

(20)

and plugging into \( Y(q) \) gives

\[
Y(q) = n + \frac{4}{\mu} \left( \frac{\mu}{4nL} (n - 1) + 1 \right) \rho L_{\text{max}}
\]

\[
= n + \frac{4L_{\text{max}}}{\mu} \frac{4nL - \mu}{n(4L + \mu) - \mu}.
\]

(21)

Furthermore \( Y(q) \geq X(q) \) which follows by comparing the two and noting that

\[
n \geq \frac{4L}{\mu} \frac{n^2\mu}{n(4L + \mu) - \mu} \iff \frac{n(4L + \mu) - \mu}{n\mu} \geq \frac{4L}{\mu} \iff \frac{4L}{\mu} + \frac{\mu(n - 1)}{n\mu} \geq \frac{4L}{\mu}.
\]

Consequently, the total complexity (19) is dominated by \( Y(q) \) and given by

\[
\Omega^{n,\tau} = \left( n + \frac{4L_{\text{max}}}{\mu} \frac{4nL - \mu}{n(4L + \mu) - \mu} \right) \log \left( \frac{1}{\epsilon} \right).
\]

Taking into consideration that \( Y(q) \geq X(q) \) and that the stepsize is given by

\[
\alpha = \frac{q(n - 1) + 1}{\mu} \min \left\{ \frac{1}{X(q)}, \frac{1}{Y(q)} \right\}
\]

\[
= \frac{n(4L + \mu) - \mu}{4nL} \frac{1}{Y(q)} = \frac{1}{4nL} \frac{\mu(n(4L + \mu) - \mu) + 4L_{\text{max}}(4nL - \mu)}{\mu n(n(4L + \mu) - \mu)^2} + 4L_{\text{max}}(4nL - \mu)
\]

\[
= \frac{1}{4L} \frac{\mu n((4L + \frac{\mu}{n}) - \mu) + 4L_{\text{max}}(4L - \frac{\mu}{n})}{(4L + \frac{\mu}{n}) - \mu}
\]

Finally calculating the constant in Lyaponov constant in (30) we have

\[
\frac{\alpha}{2L_{\text{max}}^2} = \frac{\theta}{nL_{\text{max}}} \frac{\alpha}{2} = \frac{1}{(q(n - 1) + 1)L_{\text{max}}^2} \frac{\alpha}{2} \quad \square
\]

**B  Proof of Theorem 2**

Our proof of Theorem 2 is based on the more general convergence Theorem 3.6 in [34]. In the setting [34], the authors allow for different Euclidean norms based on a positive definite matrix \( W \in \mathbb{R}^{d \times d} \). Here we only consider the standard Euclidean norm (thus \( W = I \)). Let

\[
\mathcal{G} = \{ C : C \subset \{1, \ldots, n\}, |C| = \tau \},
\]

(22)

and consider the sampling given in (24). That is we either sample a coordinate \( j \) with probability \( (1 - q)/n \) or we sample a minibatch \( C \in \mathcal{G} \) with probability \( q/\binom{n}{\tau} \). Note that there are exactly \( \binom{n}{\tau} \)
sets in $G$. Furthermore, if we fix a coordinate $j$, there are exactly $c_1 \equiv \binom{n-1}{r-1}$ sets in $G$ that contain $j$, in other words
\[
c_1 \equiv \sum_{C \in G} 1 = \binom{n-1}{r-1}, \tag{23}\]
independent on our choice of $j$. We also define a projection matrix $\Pi_S$ based on this sampling in \cite{24}.
\[
\mathbf{S} = \begin{cases} 
\mathbf{I}_C \quad \text{with prob } \frac{n}{\binom{n}{r}}, & C \in G \\
\mathbf{e}_j \quad \text{with prob } \frac{n}{\binom{n}{r}}, & C \in G
\end{cases} \quad \Pi_S = \begin{cases} 
\mathbf{I}_C \mathbf{I}_C^\top \quad \text{with prob } \frac{n}{\binom{n}{r}}, & C \in G \\
\mathbf{e}_j \mathbf{e}_j^\top \quad \text{with prob } \frac{n}{\binom{n}{r}}
\end{cases} \tag{24}
\]
where $\mathbf{I}_C$ is a column submatrix of the identity matrix and $\mathbf{e}_j$ is the all zero vector except for the $j^{th}$ coordinate with a value of one. In this setting, we can now fully analyze our proposed algorithm as a direct implication of Theorem 3.6 in \cite{34} whose statement we repeat here for ease of reference.

**Theorem 4** (Convergence of JacSketch for General Sketches). Let $f$ be $\mu$–strongly convex and each $f_i$ be convex and $L_i$–smooth. Let $\mathbf{S}$ be defined by \cite{24} and let $\theta_S$ be the associated bias-correcting random variable that satisfies
\[
\mathbb{E} [\theta_S \Pi_S e] = e, \tag{25}
\]
where $e \in \mathbb{R}^n$ is the vector of all ones. Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be the expected smoothness constants that satisfy
\[
\frac{1}{n^2} \mathbb{E} \left[ \|\mathbf{R}\theta \Pi_S e\|_2^2 \right] \leq 2 \mathcal{L}_1 \left( f(x) - f(x^*) \right) \tag{26}
\]
\[
\mathbb{E} \left[ \|\mathbf{R}\Pi_S e\|_2^2 \right] \leq 2 \mathcal{L}_2 \left( f(x) - f(x^*) \right), \tag{27}
\]
where $\mathbf{R} = \nabla \mathbf{F}(x) - \nabla \mathbf{F}(x^*)$. Let
\[
\kappa = \lambda_{\min} \left( \mathbb{E} [\Pi_S] \right), \tag{28}
\]
be the stochastic condition number and assume that $\kappa > 0$. Let the sketch residual $\rho$ be defined as
\[
\rho \overset{\text{def}}{=} \lambda_{\max} \left( \mathbb{E} [\theta_S^2 \Pi_S e e^\top \Pi_S] - ee^\top \right) \geq 0. \tag{29}
\]
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 in \cite{24}. Consider the Lyapunov function
\[
\Psi^k \overset{\text{def}}{=} \|x^k - x^*\|_2^2 + \frac{\alpha}{2\mathcal{L}_2} \|\mathbf{J}^k - \nabla \mathbf{F}(x^*)\|_2^2. \tag{30}
\]
If the stepsize satisfies
\[
0 \leq \alpha \leq \min \left\{ \frac{1}{4\mathcal{L}_1}, \frac{\kappa}{4\mathcal{L}_2 \mu/\sqrt{n^2 + \mu}} \right\}, \tag{31}
\]
then
\[
\mathbb{E} [\Psi^k] \leq (1 - \mu \alpha)^k \cdot \Psi^0, \tag{32}
\]
If we choose $\alpha$ to be equal to the upper bound in \cite{31}, then
\[
k \geq \max \left\{ \frac{4\mathcal{L}_1}{\mu}, \frac{1}{\kappa} + \frac{4\rho \mathcal{L}_2}{\kappa \mu/\sqrt{n^2}} \right\} \log \left( \frac{1}{\epsilon} \right) \Rightarrow \mathbb{E} [\Psi^k] \leq \epsilon \Psi^0. \tag{33}
\]
Now all we need to do to apply this theorem is to compute the constants $\theta_S, \kappa, \mathcal{L}_1, \mathcal{L}_2, \rho$ under this new sampling $\mathbf{S}$.

In our upcoming computations, we will often make use of the following type of equality
\[
\sum_{C \in G} \sum_{i \in C} a_{i,C} = \sum_{i=1}^n \sum_{C \in G : i \in C} a_{i,C}, \tag{34}
\]
which holds for any $a_{i,C} \in \mathbb{R}$ and can be proved using a double counting argument.
**The Bias Correcting Term:** $\theta_S$

It turns out that a constant bias correcting variable $\theta_S \equiv \theta$ is sufficient to guarantee (25) since

$$
E[\theta_S \Pi_S e] = \sum_i^n \theta e_i e_i^\top \frac{1-q}{n} e + \sum_{C \in G} \theta I_C I_C^\top e \frac{q}{(n\tau)} \quad (35)
$$

$$
= \theta \left(1 - \frac{q}{n} \mathbf{1} + \frac{q}{n} (n-1) \left(\frac{1}{\tau}-1\right)\right) e = e, \quad (36)
$$

where we used in the second equality that

$$
\sum_{C \in G} I_C I_C^\top = \left(\frac{n-1}{\tau}-1\right) \mathbf{I}. \quad (37)
$$

From (36) and the definition (25) we have that

$$
\theta = \frac{n}{q(\tau-1) + 1}. \quad (38)
$$

**The Stochastic Condition Number:** $\kappa$

To calculate $\kappa$ we first compute

$$
E[\Pi_S] = \sum_i^n \frac{1-q}{n} e_i e_i^\top + \sum_{C \in G} I_C I_C^\top \frac{q}{(n\tau)} \quad (37)
$$

$$
= \left(1 - \frac{q}{n} \mathbf{1} + \frac{q}{n} (n-1) \left(\frac{1}{\tau}-1\right)\right) \mathbf{I}
$$

$$
= \frac{q(\tau-1) + 1}{n} \mathbf{I} \quad (38)
$$

Consequently

$$
\kappa = \frac{1}{\theta}. \quad (39)
$$

**Expected Smoothness Constant of the Stochastic Gradient:** $L_1$

Here we deduce the formula (9) from the definition of expected smoothness in (26).

Since each $f_i(x)$ is $L_i$-smooth and convex, then for $C \subseteq [n]$, $f_C(x) = \frac{1}{|C|} \sum_{i \in C} f_i(x)$ is $L_C$ smooth where $L_C = \frac{1}{|C|} \sum_{i \in C} L_i$. Therefore, the following is equivalent to the smoothness assumption (See 2.1.7 in [38]):

$$
\|\nabla f_C(x) - \nabla f_C(x^*)\|_2^2 \leq 2L_C (f_C(x) - f_C(x^*)) - \langle \nabla f_C(x^*), x - x^* \rangle, \quad (41)
$$

and by an analogous argument and summing up we have

$$
\sum_{i=1}^n \|\nabla f_i(x) - \nabla f_i(x^*)\|_2^2 \leq 2n L_{\max} (f(x) - f(x^*)). \quad (42)
$$
It now follows that

\[
\frac{1}{n^2} \mathbb{E} \left[ \| R \theta \Pi_S \epsilon \|_2^2 \right] = \frac{\theta^2}{n^2} \mathbb{E} \left[ \| R \Pi_S \epsilon \|_2^2 \right] = \frac{\theta^2}{n^2} \sum_{C \in \mathcal{G}} \| R \Pi_s \mathbf{C}_e \|_2^2 \frac{q}{n} + \frac{\theta^2}{n^2} \sum_{i=1}^{n} \| R \epsilon_i \|_2^2 \frac{1-q}{n}
\]

\[
= \frac{\theta^2}{n^2} \left( \frac{q}{n} \sum_{C \in \mathcal{G}} \| R \epsilon_C \|_2^2 + \frac{1-q}{n} \sum_{i=1}^{n} \| R \epsilon_i \|_2^2 \right)
\]

\[
\leq \frac{\theta^2}{n^2} \left( \frac{n}{\tau} \sum_{C \in \mathcal{G}} \| \nabla f_C(x) - \nabla f_C(x^*) \|_2^2 + \frac{1-q}{n} \sum_{i=1}^{n} \| \nabla f_i(x) - \nabla f_i(x^*) \|_2^2 \right)
\]

\[
\leq \frac{\theta^2}{n^2} \left( \frac{n}{\tau} \sum_{i=1}^{n} \sum_{C \in \mathcal{G} : i \in C} L_C \| (f_i(x) - f_i(x^*) - \langle \nabla f_i(x^*), x - x^* \rangle) + 2(1-q)L_{\max}(f(x) - f(x^*)) \right)
\]

\[
= 2 \frac{\theta^2}{n^2} \left( \frac{n}{\tau} \sum_{i=1}^{n} \sum_{C \in \mathcal{G} : i \in C} L_C \right) + (1-q)L_{\max}(f(x) - f(x^*)).
\]

Note that \( \epsilon_C \) is the vector of all zeros except for the indices selected by the sample C have a value of one. Note that since the sampling is \( \tau \)-uniform therefore \( |C| = \tau \); therefore, the forth equality follows from the fact that \( \frac{1}{|C|} \nabla F(x) \epsilon_C = \nabla f_C(x) \). The second part of the first inequality follows from first order optimality conditions where \( \sum_i \nabla f_i(x^*) = 0 \). It now follows from the definition (36) and the above that

\[
\mathcal{L}_1 = \frac{\theta^2}{n} \frac{q \tau}{(\tau)} \left( \max_{i=1, \ldots, n} \sum_{C \in \mathcal{G} : i \in C} L_C \right) + \frac{\theta^2(1-q)}{n^2} L_{\max}. \tag{43}
\]

We can bound the max term in (43) under the assumption that \( L_C = \frac{1}{\tau} \sum_{i \in C} L_i \), as follows. Using another double counting argument we have that

\[
\sum_{C \in \mathcal{G}} \sum_{i \in C} L_C = \frac{1}{\tau} \sum_{C \in \mathcal{G}} \sum_{j \in C} L_j
\]

\[
= \frac{1}{\tau} \sum_{j=1}^{n} \sum_{C \in \mathcal{G} : j \in C} L_j
\]

\[
= \frac{1}{\tau} \sum_{j \neq i} \sum_{C \in \mathcal{G} : i \in C, j \in C} L_j + \frac{1}{\tau} \sum_{C \in \mathcal{G}} \sum_{i \in C} L_i
\]

\[
= \frac{1}{\tau} \sum_{j \neq i} \left( \frac{n-2}{\tau-2} \right) L_j + \frac{1}{\tau} \left( \frac{n-1}{\tau-1} \right) L_i
\]

\[
= \frac{n}{\tau} \left( \frac{n-2}{\tau-2} \right) L + \frac{1}{\tau} \left( \frac{n-1}{\tau-1} - \left( \frac{n-2}{\tau-2} \right) \right) L_i
\]

\[
= \left( \frac{n-2}{\tau-2} \right) \left( \frac{n L}{\tau} + \frac{1}{\tau} \frac{n-\tau}{\tau-1} L_i \right), \tag{44}
\]

16
where $L \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} L_i$. Inserting the above into (43) gives
\[
L_1 = \frac{\theta^2 q \tau}{n} \left( \frac{n-2}{(\tau - 2)} \right) \left( \frac{nL}{\tau} + \frac{1}{\tau} \frac{n-\tau}{1} L_{\max} \right) + \frac{\theta^2 (1 - q)}{n^2} L_{\max} \\
= q \theta^2 \frac{\tau (\tau - 1)}{n(n-1)} \left( \frac{L}{\tau} + \frac{n-\tau}{n(\tau - 1)} L_{\max} \right) + \frac{\theta^2 (1 - q)}{n^2} L_{\max} \\
= \frac{\theta^2}{n^2} \left( q \frac{\tau (n-\tau)}{n-1} + 1 \right) L_{\max} + q \theta^2 \frac{\tau (\tau - 1)}{n(n-1)} L. \tag{45}
\]

**Expected Smoothness Constant of the Stochastic Jacobian:** $L_2$

It follows from (87) in [32] that
\[
\mathbb{E} \left[ \| \mathbf{R} \mathbf{I}_S \|_F^2 \right] = \operatorname{tr} \left( \mathbf{R}^\top \mathbb{E} \left[ \mathbf{I}_S \mathbf{I}_S^\top \right] \right) \leq 2n \lambda_{\max} \left( \mathbf{D}_L^2 \mathbb{E} \left[ \mathbf{I}_S \mathbf{D}_L^2 \right] \right) (f(x) - f(x^*)) \tag{46}
\]
where $\mathbf{D}_L = \text{Diag}(L_1, \ldots, L_n)$. Moreover from (39) we have that $\mathbb{E}[\mathbf{I}_S] = \frac{1}{\theta} \mathbf{I}$. Thus comparing (26) and (46) we have that
\[
L_2 = \frac{n}{\theta} \lambda_{\max} L_i = \frac{nL_{\max}}{\theta}. \tag{47}
\]

**The Sketch Residual:** $\rho$

To compute the sketch residual we first note that
\[
\Theta \overset{\text{def}}{=} \theta^2 \mathbb{E} \left[ \mathbf{I}_S \mathbf{e}_C \mathbf{e}_C^\top \mathbf{I}_S \right] - e e^\top \\
= \theta^2 \left( \sum_{C \in \mathcal{V}} q \frac{\tau}{\tau - 1} e_C e_C^\top + \sum_i \frac{1}{n} q e_i e_i^\top \right) - e e^\top \\
= \theta^2 \left( \frac{q}{(\tau - 1)} c_2 (e e^\top - \mathbf{I}) + \frac{q}{(\tau - 1)} c_1 \mathbf{I} + \frac{1 - q}{n} \mathbf{I} \right) - e e^\top \\
= \mathbf{I} \theta^2 \left( \frac{1 - q}{n} + \frac{q}{(\tau - 1)} c_2 - \frac{q}{(\tau - 1)} c_2 \right) + e e^\top \left( \frac{q}{(\tau - 1)} \theta^2 - 1 \right). 
\]

Note that $c_1 = \binom{n}{\tau - 1} \frac{1}{n} \forall \tau \geq 1$ and $c_2 = \binom{n}{\tau - 1} \frac{1}{n} \forall \tau > 1$. Note that $c_2 = 0$ for $\tau = 1$. Since $\Theta$ is a circulant matrix with a first column being $[\alpha + \beta; \beta \mathbf{e}^{n-1}]$, where $\mathbf{e}^{n-1}$ is the vector of all ones of size $(n - 1)$, a well known result for calculating eigenvalues of circulant matrices [32] states that the eigenvalues of $\Theta$ are simply the 1-D Discrete Fourier transform of the first column, that is $(\text{DFT}[\alpha + \beta; \beta \mathbf{e}^{n-1}]) = [\alpha + n \beta, \alpha \mathbf{e}_{n-1}]$. Since $\Theta$ only has two distinct eigenvalues we have that $\rho \overset{\mathbb{C}}{=} \lambda_{\max}(\Theta) = \max \left( \text{DFT}[\alpha + \beta; \beta \mathbf{e}^{n-1}] \right) = \max \left( \alpha + n \beta, \alpha \right)$. Consequently
\[
\rho = \begin{cases} 
\alpha + n \beta & \beta \geq 0, \\
\alpha & \beta < 0.
\end{cases}
\]

Inserting these values for $\alpha$ and $\beta$ gives
\[
\rho = \begin{cases} 
\theta^2 \left( \frac{1 - q}{n} + \frac{q (c_1 - c_2)}{(\tau - 1)} \right) + n \left( \frac{\theta^2 q c_2}{(\tau - 1)} - 1 \right) \\
\theta^2 \left( \frac{1 - q}{n} + \frac{q (c_1 - c_2)}{(\tau - 1)} \right) \end{cases} \geq \begin{cases} 0, \\
0 \end{cases} \tag{48}
\]

After further simplifications we have that
\[
\rho = \begin{cases} 
\theta^2 \left( \frac{1 - q}{n} + \frac{q \tau n \tau - 1}{n - 1} \right) \\
\theta^2 \left( \frac{1 - q}{n} + \frac{q \tau n \tau - 1}{n - 1} \right) + n \left( \theta^2 q \tau \frac{n - 1}{n - 1} - 1 \right) \end{cases} \geq \begin{cases} 0, \\
0 \end{cases} \tag{48}
\]

By substituting all the previously computed constants (38), (40), (49), (47) and (48) for the sketch $S$ in Theorem 4, the proof of Theorem 2 is complete.
\[\square\]
C Proof of Lemma 1

First note that when \( L_i = L_{\text{max}} \) we have that (9) becomes
\[
L_1 = \frac{q(\tau^2 - 1) + 1}{(q(\tau - 1) + 1)^2} L_{\text{max}}. \tag{49}
\]

The rest of the proof is straightforward.

D Proof of Lemma 2

Using (49) we have
\[
g_1 = 4 \frac{4L_1}{\mu} (q(\tau - 1) + 1) \frac{4L_{\text{max}}}{\mu} \left( \frac{q\tau^2}{(q(\tau - 1) + 1)^2} + \frac{1-q}{(q(\tau - 1) + 1)^2} \right) (q(\tau - 1) + 1)
\]
and thus the derivative with respect to \( q \) is:
\[
\frac{\partial g_1}{\partial q} = 4 \frac{4L_{\text{max}}}{\mu} \left( \frac{\tau^2 - 1}{q(\tau - 1) + 1} - \frac{(\tau - 1)(1 + q(\tau^2 - 1))}{(1 + q(\tau - 1))^2} \right)
\]

Then it is sufficient to show that if \((\tau^2 - 1)(q(\tau - 1) + 1) - (\tau - 1)(q(\tau^2 - 1) + 1) \geq 0\) then \( g_1 \) is monotonically. Note that
\[
(\tau^2 - 1)(q(\tau - 1) + 1) - (\tau - 1)(q(\tau^2 - 1) + 1)
= \tau^2 - 1 - (\tau - 1) = \tau(\tau - 1) \geq 0 \quad \tau \geq 1
\]
Therefore \( g_1 \) is monotonically increasing in \( q \).

E Proof of Lemma 3

We will denote \( f \sim_D g \) to indicate that \( f \) and \( g \) have the same monotonicity on the domain \( D \), i.e., (if \( \frac{\partial f}{\partial q} \geq 0 \) then \( \frac{\partial g}{\partial q} \geq 0 \) on the domain \( D \) and vice verse.) Moreover, \( f \simeq g \) indicates that both functions have the same concavity direction, i.e, (if \( \frac{\partial^2 f}{\partial q^2} \geq 0 \) then \( \frac{\partial^2 g}{\partial q^2} \geq 0 \) on the domain \( D \) and vice verse). For instance, consider the domain of \( q \) to be the real numbers, that is \( D = \mathbb{R} \) and note that:
\[
g_2 = \left( \theta + \frac{4\rho L_{\text{max}}}{\mu n} \right) (q(\tau - 1) + 1)
\]
\[
\leq \left( n + \frac{4\rho L_{\text{max}}}{\mu n} (q(\tau - 1) + 1) \right) \sim \rho (q(\tau - 1) + 1).
\]
First we will start by showing that \( g_2 \) is a decreasing function in \( q \) in the interval \( q \in [0, q_-] \cup [q_+, 1] \). Therefore, consider the domain of \( q \) where \( D = [0, q_-] \cup [q_+, 1] \). Then \( \rho = \theta^2 \left( \frac{1-q}{n} + q \frac{n-\tau}{n-\tau} \right) \) and thus:
Now, with a similar argument but considering the second derivative with the domain of \( q \), note that since
\[
D \left( \frac{\tau}{n} \right) = \frac{n}{n-1} \frac{q}{(q-1) + 1} = \frac{n-1 - qn + q\tau n - q\tau^2}{(n-1)(q-1) + 1}
\]
which is decreasing in \( q \), i.e., \( \frac{\partial}{\partial q} \left( n - \tau - 1 + \frac{\tau}{q(n-1)+1} \right) \leq 0 \). Since \( g_2 \sim \left( n - \tau - 1 + \frac{\tau}{q(n-1)+1} \right) \), then \( \frac{\partial g_2}{\partial q} \leq 0 \); thus, \( g_2 \) is a decreasing function in \( q \) in the domain \( D = [0, q_-] \cup [q_+, 1] \).

Now, with a similar argument but considering the second derivative with the domain of \( q \) as \( D = [q_-, q_+] \). Note that in this domain \( \rho = \theta^2 \left( \frac{1-q}{n} + q \frac{n-q}{n-1} \right) + n \left( \theta^2 \frac{\tau^2}{n(n-1)} - 1 \right) \)

\[
g_2 = \frac{n(1-q)}{q(q-1) + 1} + \frac{nq\tau(n-\tau)}{(n-1)(q-1) + 1} + n(q-1) \left( \frac{nq\tau(q-1)}{(n-1)(q(q-1) + 1)^2} - 1 \right)
\]

\[
\sim \frac{n(1-q)}{q(q-1) + 1} + \frac{nq\tau(n-\tau)}{(n-1)(q-1) + 1} + n^2 \left( \frac{nq\tau(q-1)}{(n-1)(q(q-1) + 1)^2} - 1 \right)
\]

\[
\sim \frac{n^2 - n - n^2q + nq + n^2q\tau - nq\tau^2 + n^2q\tau^2 - n^2q\tau}{(n-1)(q(q-1) + 1)}
\]

\[
\sim \frac{q(-n^2 + n - n\tau^2 + n^2\tau^2) + n^2 - n}{q(q-1) + 1} = \frac{q(q^2 - 1)(n^2 - n) + n^2 - n}{q(q-1) + 1}
\]

\[
\sim \frac{q(q^2 - 1) + 1}{q(q-1) + 1} \approx \frac{(\tau + 1)(q(q-1) + 1) - \tau}{q(q-1) + 1} = \tau + 1 - \frac{\tau}{q(q-1) + 1}
\]

\[
\sim - \frac{\tau}{q(q-1) + 1}
\]

Note that since \( - \frac{\tau}{q(q-1)+1} \) is a concave function in \( q \) since \( \frac{\partial^2}{\partial q^2} \left( - \frac{\tau}{q(q-1)+1} \right) \leq 0 \) and since \( g_2 \sim - \frac{\tau}{q(q-1)+1} \) on the domain \( D = [q_-, q_+] \); thus, \( \frac{\partial^2 g_2}{\partial q^2} \leq 0 \) on the domain \( D = [q_-, q_+] \).
F Proof of Lemma 4

We first start by showing that \( q^- \) is a valid probability. First note that we are only interested in the case where \( \tau > 4^{n-1} \), or equivalently \( \tau \geq 4 \) in which \( q^- \) is real. Therefore,

\[
\frac{\partial q^-}{\partial \tau} = \frac{n - \frac{n\sqrt{n\tau}}{2\sqrt{4(1-n) + n^2\tau}} - \frac{n\sqrt{4(1-n) + n^2\tau}}{2\sqrt{n\tau}}}{2(n-1)(\tau-1)} - \frac{2 + n(\tau - 2) - \sqrt{n\tau}\sqrt{4(1-n) + n^2\tau}}{2(n-1)(\tau-1)^2}.
\]

Let \( D = \sqrt{4(1-n) + n^2\tau} \), then we need to show that \( \frac{\partial q^-}{\partial \tau} \leq 0 \forall \tau \) and thus \( q^- \) is a decreasing function in \( \tau \). Thus we need to establish the following:

\[
\begin{align*}
\lim_{n \to \infty} n(\tau - 1) - n(\tau - 1)\sqrt{n\tau} & \leq \frac{n(\tau - 1)D}{2D} - 2 - n(\tau - 2) + \sqrt{n\tau}D \leq 0 \\
\iff n + \sqrt{n\tau}D & \leq \frac{n(\tau - 1)\sqrt{n\tau}}{2D} + \frac{n(\tau - 1)D}{2\sqrt{n\tau}} + 2 \\
\iff 2Dn\sqrt{n\tau} + 2D^2(\tau\sqrt{n\tau}) & \leq n(\tau - 1)(\tau\sqrt{n\tau}) + D^2n(\tau - 1) + 4D\sqrt{n\tau} \\
\iff 2Dn \left[ \sqrt{n\tau} + D\tau \right] & \leq 2Dn \left[ 2\sqrt{\frac{\tau}{n}} + \frac{1}{2}D(\tau - 1) \right] + n(\tau - 1)(\tau\sqrt{n\tau}) \\
\iff n(\tau - 1)n\tau & \geq 2Dn \left[ \sqrt{n\tau} - 2\sqrt{\frac{\tau}{n}} + \frac{1}{2}D(\tau - 1) \right] \\
\iff n(\tau - 1)n\tau & \geq D^2n(\tau + 1) + 2Dn\sqrt{n\tau} - 4D\sqrt{n\tau} \\
\iff n^2\tau^2 - n^2\tau & \geq (4 - 4n + n\tau)(\tau + 1)n + 2Dn\sqrt{n\tau} - 4D\sqrt{n\tau} \\
\iff n^2\tau^2 - n^2\tau & \geq (4n\tau + 4n - 3n^2\tau - 4n^2 + n^2\tau^2) + 2Dn\sqrt{n\tau} - 4D\sqrt{n\tau} \\
\iff -n^2\tau & \geq (4n\tau + 4n - 3n^2\tau - 4n^2) + 2Dn\sqrt{n\tau} - 4D\sqrt{n\tau} \\
\iff 2n^2\tau + 4n^2 & \geq 4n(\tau + 1) + 2D\sqrt{n\tau} (n - 2) \\
\iff n^2(\tau + 2) & \geq 2n(\tau + 1) + D\sqrt{n\tau} (n - 2)
\end{align*}
\]

Note that since:

\[
D = \sqrt{4(1-n) + n^2\tau} \leq \sqrt{n\tau}
\]

Then:

\[
2n(\tau + 1) + D\sqrt{n\tau} (n - 2) \leq 2n(\tau + 1) + n\tau(n - 2) = n(n\tau + 2) \leq n^2(\tau + 2)
\]

Thus the last inequality always holds which shows that \( q^- \) is a decreasing function of \( \tau \). Furthermore, \( \lim_{\tau \to 4} q^- = \frac{n+1}{3(n-1)} \) and that \( \lim_{\tau \to \infty} q^- = \frac{1}{(n-1)^2} \). Therefore, since \( \frac{1}{(n-1)^2} \leq q^- \leq \frac{n+1-2\sqrt{n\tau}}{3(n-1)} \), \( q^- \) is a valid probability. A similar argument can be used to show that \( q^+ \) is increasing in \( \tau \) and that it is a valid probability.
G  Proof of Lemma 5

To find the intersection points between $g_1$ and $g_2$, we consider the two cases of $\rho$ separately. For a simplification, note that the intersection point between $g_1$ and $g_2$ is the same as for $\frac{g_1}{q(\tau - 1) + 1}$.

Case I: $q\theta^2 \leq \frac{n - 1}{\tau - 1}$

\[
\frac{g_2 - g_1}{q(\tau - 1) + 1} = \theta \left( 1 + \frac{4L_{\text{max}}}{\mu n} \left( \frac{1 - q}{n} + \frac{\tau n - \tau}{n n - 1} \right) \right) - \frac{4\theta L_{\text{max}} q(\tau^2 - 1) + 1}{q(\tau - 1) + 1} = 0
\]

\[
\begin{align*}
&\frac{n}{q(\tau - 1) + 1} + \frac{4L_{\text{max}}}{q(\tau - 1) + 1} \frac{1}{(n - 1)(q(\tau - 1) + 1)} \left( nq(1 - \tau) \right) = 0 \\
&\frac{4L_{\text{max}}}{\mu} \frac{q(\tau - 1)}{(n - 1)(q(\tau - 1) + 1)} = 1 \\
&q_{i_1} = \frac{1 - n}{(\tau - 1)(n - 1) - \tau(\tau - 1)\frac{4L_{\text{max}}}{\mu}} = \frac{n - 1}{(\tau - 1)(\frac{4L_{\text{max}}}{\mu} + 1 - n)}
\end{align*}
\]

Now that we have the intersection point, we can next find the ranges of $\tau$ in which $q_{i_1}$ occur. In particular, the range of $\tau$ in which the following is satisfied $q_{i_1}\theta^2 \leq \frac{n - 1}{\tau - 1}$. Note that for $q_{i_1}$, we have $\theta = \frac{n(-n + 1 + \frac{4L_{\text{max}}}{\mu})}{\tau \frac{4L_{\text{max}}}{\mu}}$. Therefore, it is easy now to show that for

(a): if $n \leq \frac{4L_{\text{max}}}{\mu}$ then $q_{i_1} \geq 0$. However, for $q_{i_1} \leq 1$ then we have:

\[
\begin{align*}
n - 1 &\leq (\tau - 1) \left( \frac{4L_{\text{max}}}{\mu} + 1 - n \right) \\
\tau &\left( \frac{4L_{\text{max}}}{\mu} - \frac{4L_{\text{max}}}{\mu} - n + 1 \right) \geq 0
\end{align*}
\]

Therefore for $0 \leq q_{i_1} \leq 1$ where $n \leq \frac{4L_{\text{max}}}{\mu}$ then:

\[
\tau \geq \tau_{\text{min}} = \frac{n}{\frac{4L_{\text{max}}}{\mu}} + 1 - \frac{\mu}{4L_{\text{max}}}
\]

(b): Similarly, if $n > \frac{4L_{\text{max}}}{\mu}$ then for $0 \leq q_{i_1} \leq 1$ we need:

\[
\tau \leq \tau_{\text{max}} = \frac{n(n - 1)\mu}{(n - \frac{4L_{\text{max}}}{\mu})4L_{\text{max}}}
\]

Thus the range of

\[
\tau \in \left[ \max \left( 4, \tau_{\text{min}} \right), \min \left( \tau_{\text{max}}, n \right) \right]_{n > \frac{4L_{\text{max}}}{\mu}} + n_{n \leq \frac{4L_{\text{max}}}{\mu}}
\]
Case II: \( q\theta^2 \geq \frac{n(n-1)}{n-1} \)

\[
\frac{\theta_2 - \theta_1}{\theta (\tau - 1) + 1} = \theta \left( 1 + \frac{4}{\mu n^2} n L_{\text{max}} \left( q^2 \left( \frac{1-q}{n} + q \frac{\tau - \tau}{n \tau - 1} \right) + n \left( \theta^2 q \frac{\tau - 1}{n \tau - 1} - 1 \right) \right) \right)
\]
\[
\Rightarrow -\frac{4\theta L_{\text{max}}}{\mu n} q(\tau^2 - 1) + 1 = 0
\]
\[
\Rightarrow \theta + \frac{4}{\mu n^2} n L_{\text{max}} \left( q^2 \left( \frac{1-q}{n} + q \frac{\tau - \tau}{n \tau - 1} \right) + n \left( \theta^2 q \frac{\tau - 1}{n \tau - 1} - 1 \right) \right)
\]
\[
- \frac{4\theta L_{\text{max}}}{\mu n} q(\tau^2 - 1) + 1 = 0
\]
\[
\Rightarrow \theta + \frac{4\theta L_{\text{max}}}{\mu n} \left( q \left( \frac{1-q}{n} + q \frac{\tau - \tau}{n \tau - 1} \right) + n \left( \theta q \frac{\tau - 1}{n \tau - 1} - 1 \right) - \frac{q(\tau^2 - 1) + 1}{q(\tau - 1) + 1} \right) = 0
\]
\[
\Rightarrow \theta + \frac{4\theta L_{\text{max}}}{\mu n} \left( q \frac{nq\tau(1-\tau)}{(n-1)(q(\tau - 1) + 1)} - n \left( \theta q \frac{\tau - 1}{n \tau - 1} - \frac{1}{\theta} \right) \right) = 0
\]
\[
\Rightarrow 1 = \frac{4L_{\text{max}}}{\mu n} \left( q \frac{nq\tau(\tau - 1)}{(n-1)(q(\tau - 1) + 1)} - n \left( \theta q \frac{\tau - 1}{n \tau - 1} - \frac{1}{\theta} \right) \right) = 0
\]
\[
\Rightarrow 1 = \frac{4L_{\text{max}}}{\mu n} \left( q \frac{nq\tau(\tau - 1)}{(n-1)(q(\tau - 1) + 1)} - n \left( \theta q \frac{\tau - 1}{n \tau - 1} - \frac{1}{\theta} \right) \right) = 0
\]
\[
\Rightarrow 1 = \frac{4L_{\text{max}}}{\mu n} \left( q \frac{nq\tau(\tau - 1)}{(n-1)(q(\tau - 1) + 1)} - n \left( \theta q \frac{\tau - 1}{n \tau - 1} - \frac{1}{\theta} \right) \right) = 0
\]
\[
\Rightarrow 1 = \frac{4L_{\text{max}}}{\mu n} (q(\tau - 1) + 1) = 0 \Rightarrow q_2 = \frac{1 - \frac{4L_{\text{max}}}{\mu n}}{\frac{4L_{\text{max}}}{\mu n} (\tau - 1)} = \frac{n - \frac{4L_{\text{max}}}{\mu}}{\frac{4L_{\text{max}}}{\mu} (\tau - 1)}
\]

and that \( \theta = \frac{4L_{\text{max}}}{\mu n} \). For \( q_2 \theta^2 \geq \frac{n(n-1)}{n-1} \) then

\[
\frac{4L_{\text{max}}}{\mu} \tau \left( n - \frac{4L_{\text{max}}}{\mu} \right) \geq n(n-1)
\]
\[
\tau \geq \frac{n(n-1)}{\left( n - \frac{4L_{\text{max}}}{\mu} \right) \frac{4L_{\text{max}}}{\mu}} = \tau_{\text{max}}
\]

Then the range of \( \tau \) that assures \( 0 \leq q_2 \leq 1 \) are as follows:

\( \tau \in [\tau_{\text{max}}, n] \)
H Experiments with logistic loss

In this section we conduct some further experiments on the logistic loss:

\[
f(x) = \frac{1}{2n} \sum_{i=1}^{n} \log \left( 1 + \exp \left( -y_i a_i^\top x \right) \right) + \frac{\lambda}{2} \|x\|_2^2
\]

The experiments are conducted on both synthetic and real datasets similarly to section 5. Figure 6 demonstrates time comparisons and epochs comparisons among SAGA, SAGD with \((q^*, \tau^*)\), SAGD with \((q^*, \frac{\tau^*}{2})\) and \((q^*, 2\tau^*)\) on three different datasets. SAGD with optimal \((q^*, \tau^*)\) pair achieve the best performance and suggest a linear speed up in minibatch size for a parallel implementation. The same conclusion can be drawn from the the second synthetic datasets that has a larger dimensional problem. Moreover we conduct some further experiments on real datasets namely jcn1, a9a and australian, demonstrating the effectiveness of SAGD.

Figure 6: Shows a comparison between SAGA and our method with optimal \((q^*, \tau^*)\) pair. We also compare them both against \((q^*, \frac{\tau^*}{2})\) and \((q^*, 2\tau^*)\). The first row shows a comparison in time while the second row shows the same experiment as compared in the total number of effective passes.
Figure 7: Shows a comparison between SAGA and our method with optimal \((q^*, \tau^*)\) pair. We also compare them both against \((q^*, \frac{\tau}{2})\) and \((q^*, 2\tau^*)\). The first row shows a comparison in time while the second row shows the same experiment as compared in the total number of effective passes.
Figure 8: Shows a similar comparison but on the real datasets ijcnn1, a9a and australian.