

# A Stochastic Gradient Method with an Exponential Convergence Rate for Strongly-Convex Optimization with Finite Training Sets

Nicolas Le Roux  
nicolas@le-roux.name

Mark Schmidt  
mark.schmidt@inria.fr

Francis Bach  
francis.bach@ens.fr

INRIA - SIERRA Project - Team  
Laboratoire d'Informatique de l'École Normale Supérieure  
Paris, France

February 28, 2012

## Abstract

We propose a new stochastic gradient method for optimizing the sum of a finite set of smooth functions, where the sum is strongly convex. While standard stochastic gradient methods converge at sublinear rates for this problem, the proposed method incorporates a memory of previous gradient values in order to achieve a linear convergence rate. Numerical experiments indicate that the new algorithm can dramatically outperform standard algorithms.

## 1 Introduction

A plethora of the problems arising in machine learning involve computing an approximate minimizer of a function that sums a loss function over a large number of training examples, where there is a large amount of redundancy between examples. The most wildly successful class of algorithms for taking advantage of this type of problem structure are *stochastic gradient* (SG) methods [Robbins and Monro, 1951, Bottou and LeCun, 2003]. Although the theory behind SG methods allows them to be applied more generally, in the context of machine learning SG methods are typically used to solve the problem of optimizing a sample average over a finite training set,

$$\underset{x \in \mathbb{R}^p}{\text{minimize}} \quad g(x) := \frac{1}{n} \sum_{i=1}^n f_i(x). \quad (1)$$

In this work, we focus on such *finite training data* problems where each  $f_i$  is *smooth* and the average function  $g$  is *strongly-convex*.

As an example, consider the case of  $\ell_2$ -regularized logistic regression,

$$\underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \frac{\lambda}{2} \|x\|^2 + \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-b_i a_i^T x)), \quad (2)$$

where  $a_i \in \mathbb{R}^p$  and  $b_i \in \{-1, 1\}$  are the training examples associated with a binary classification problem and  $\lambda$  is a regularization parameter. This problem can be put in the framework of (1) by

using the choice

$$f_i(x) := \frac{\lambda}{2} \|x\|^2 + \log(1 + \exp(-b_i a_i^T x)).$$

More generally, any  $\ell_2$ -regularized empirical risk minimization problem of the form

$$\underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \frac{\lambda}{2} \|x\|^2 + \frac{1}{n} \sum_{i=1}^n l_i(x), \quad (3)$$

falls in the framework of (1) provided that the loss functions  $l_i$  are convex and smooth. An extensive list of convex loss functions used in machine learning is given by Teo et al. [2007], and we can even include non-smooth loss functions (or regularizers) by using smooth approximations. While many authors use sparsity-inducing regularizers such as the  $\ell_1$ -norm which do not automatically yield a strongly-convex problem when combined with a convex loss, as discussed by Zou and Hastie [2005] it may be beneficial to use an additional  $\ell_2$ -regularization term even when considering sparsity-inducing regularizers.

When the number of training examples  $n$  is large, SG methods are appealing because their iteration costs and convergence rates are *independent* of  $n$ . In particular, the basic SG method uses iterations of the form

$$x^{k+1} = x^k - \alpha_k f'_{i_k}(x^k), \quad (4)$$

where  $\alpha_k$  is a step-size and a training example  $i_k$  is selected uniformly among the set  $\{1, \dots, n\}$ . The randomly chosen gradient  $f'_{i_k}(x^k)$  yields an unbiased estimator of the true gradient  $g'(x^k)$ , and one can show under standard assumptions that for a suitably chosen decreasing step-size sequence  $\{\alpha_k\}$  the SG iterations achieve the sublinear convergence rate

$$\mathbb{E}[g(x^k)] - g(x^*) = O(1/k),$$

see Nemirovski et al. [2009, §2.1] and Bach and Moulines [2011, §3.1]. Further, under certain assumptions, this rate can be shown to be *optimal* for strongly-convex optimization in a model of computation where the algorithm only accesses the function through unbiased measurements of its objective and gradient [see Nemirovski and Yudin, 1983, Agarwal et al., 2010]. Thus, we cannot hope to obtain a better convergence rate if the algorithm only has access to unbiased measurements of the gradient. However, in the case of a *finite* training set where we may choose a training example more than once, the first-order stochastic oracle model is too general and better rates are possible.

For example, the standard *full gradient* (FG) method uses iterations of the form

$$x^{k+1} = x^k - \alpha_k g'(x^k) = x^k - \frac{\alpha_k}{n} \sum_{i=1}^n f'_i(x^k), \quad (5)$$

and with a constant step size achieves a *linear* convergence rate

$$g(x^k) - g(x^*) = O(\rho^k),$$

for some  $\rho < 1$  which depends on the condition number of  $g$  [see Nesterov, 2004, Theorem 2.1.15]. Linear convergence is also known as *geometric* or *exponential* convergence, because the error is cut by a fixed fraction on each iteration. Despite the faster convergence rate of the FG method, it can be unappealing when  $n$  is large because its iterations are  $n$  times more expensive than SG iterations.

In this work we analyze a new algorithm that we call the stochastic average gradient (SAG) method, a randomized variant of the incremental aggregated gradient (IAG) method of Blatt et al. [2008], which combines the low iteration cost of SG methods with a linear convergence rate as in FG methods. It uses iterations of the form

$$x^{k+1} = x^k - \frac{\alpha_k}{n} \sum_{i=1}^n y_i^k, \quad (6)$$

where at each iteration a random training example  $i_k$  is selected and we set

$$y_i^k = \begin{cases} f'_i(x^k) & \text{if } i = i_k, \\ y_i^{k-1} & \text{otherwise.} \end{cases}$$

That is, like the FG method, the step incorporates a gradient with respect to each training example. But, like the SG method, each iteration only computes the gradient with respect to a single training example and the cost of the iterations is independent of  $n$ . Despite the low cost of the SAG iterations, in this paper we show that *the SAG iterations have an exponential convergence rate*, like the FG method. That is, by having access to  $i_k$  and by keeping a *memory* of the most recent gradient value computed for each training example  $i$ , this iteration achieves a faster convergence rate than is possible for standard stochastic gradient methods.

In the next section, we review several closely-related algorithms from the literature, including previous attempts to combine the appealing aspects of FG and SG methods. However, note that we are not aware of any other SG method that achieves an exponential convergence rate while preserving the iteration cost of standard SG methods. Section 3 states the (standard) assumptions underlying our analysis and gives the main technical results, while Section 4 discusses practical implementation issues and Section 5 presents a numerical comparison of the new method to SG and FG methods.

## 2 Related Work

There are a large variety of approaches available to accelerate the convergence of SG methods, and a full review of this immense literature would be outside the scope of this work. Below, we comment on the relationships between the new method and several of the most closely-related ideas.

**Momentum:** SG methods that incorporate a momentum term use iterations of the form

$$x^{k+1} = x^k - \alpha_k f'_{i_k}(x^k) + \beta_k(x^k - x^{k-1}),$$

see Tseng [1998]. It is common to set all  $\beta_k = \beta$  for some constant  $\beta$ , and in this case we can re-write the SG with momentum method as

$$x^{k+1} = x^k - \sum_{j=1}^k \alpha_j \beta^{k-j} f'_{i_j}(x^j).$$

We can re-write the SAG updates (6) in a similar form as

$$x^{k+1} = x^k - \sum_{j=1}^k \alpha_k S(j, i_{1:k}) f'_{i_j}(x^j), \tag{7}$$

where the selection function  $S(j, i_{1:k})$  is set to  $1/n$  if  $j$  is the highest iteration where  $j = i_k$  and is set to 0 otherwise. Thus, momentum uses a *geometric weighting* of previous gradients while the SAG iterations *select* the most recent evaluation of each previous gradient. While momentum can lead to improved practical performance, it still requires the use of a decreasing sequence of step sizes and is not known to lead to a faster convergence rate.

**Gradient Averaging:** Closely related to momentum is using the sample average of all previous gradients,

$$x^{k+1} = x^k - \frac{\alpha_k}{k} \sum_{j=1}^k f'_{i_j}(x_j),$$

which is similar to the SAG iteration in the form (6) but where *all* previous gradients are used. This approach is used in the dual averaging method of Nesterov [2009], and while this averaging procedure leads to convergence for a constant step size and can improve the constants in the convergence rate [Xiao, 2010], it does not improve on the  $O(1/k)$  rate.

**Iterate Averaging:** Rather than averaging the gradients, some authors propose to use the basic SG iteration but use the average of the  $x^k$  over all  $k$  as the final estimator. With a suitable choice of step-sizes, this gives the same asymptotic efficiency as Newton-like second-order SG methods and also leads to increased robustness of the convergence rate to the exact sequence of step sizes [Polyak and Juditsky, 1992]. Baher’s method [see Kushner and Yin, 2003, §1.3.4] combines gradient averaging with online iterate averaging, and also displays appealing asymptotic properties. However, the convergence rates of these averaging methods remain sublinear.

**Stochastic versions of FG methods:** Various options are available to accelerate the convergence of the FG method for smooth functions, such as the accelerated full gradient (AFG) method of Nesterov [1983], as well as classical techniques based on quadratic approximations such as non-linear conjugate gradient, quasi-Newton, and Hessian-free Newton methods. Several authors have presented stochastic variants of these algorithms [Sunehag et al., 2009, Ghadimi and Lan, 2010, Martens, 2010]. Under certain conditions these variants are convergent and improve on the constant in the  $O(1/k)$  rate [Sunehag et al., 2009]. Alternately, if we split the convergence rate into a deterministic and stochastic part, it improves the convergence rate of the deterministic part [Ghadimi and Lan, 2010]. However, as with all other methods we have discussed thus far in this section, we are not aware of any existing method of this flavor that improves on the  $O(1/k)$  rate.

**Constant step size:** If the SG iterations are used with a *constant* step size (rather than a decreasing sequence), then Nedic and Bertsekas [2000, Proposition 2.4] show that the convergence rate of the method can be split into two parts. The first part depends on  $k$  and converges linearly to 0. The second part is independent of  $k$  but does not converge to 0. Thus, with a constant step size the SG iterations have a linear convergence rate up to some tolerance, and in general after this point the iterations do not make further progress. Indeed, convergence of the basic SG method with a constant step size has only been shown under extremely strong assumptions about the relationship between the functions  $f_i$  [Solodov, 1998]. This contrasts with the method we present in this work which converges to the optimal solution using a constant step size *and* does so with a linear rate.

**Accelerated methods:** Accelerated SG methods, which despite their name are not related to the aforementioned AFG method, take advantage of the fast convergent rate of SG methods with a constant step size. In particular, accelerated SG methods use a constant step size by default, and only decrease the step size on iterations where the inner-product between successive gradient estimates is negative Kesten [1958], Delyon and Juditsky [1993]. This leads to convergence of the method, and allows it to potentially achieve periods of linear convergence where the step size stays constant. However, the overall convergence rate of the method remains sublinear.

**Hybrid Methods:** Some authors have proposed variants of the SG method for problems of the form (1) that seek to gradually transform the iterates into the FG method in order to achieve a linear convergence rate. Bertsekas [1997] proposes to go through the data cyclically with a specialized weighting that allows the method to achieve a linear convergence rate for strongly-convex quadratic functions. However, the weighting is numerically unstable and the linear convergence rate presented treats full cycles as iterations. Friedlander and Schmidt [2011] consider grouping the  $f_i$  functions into ‘batches’ of increasing size and performing SG iterations on the batches. In both cases, the iterations that achieve the linear rate have a cost that is not independent of  $n$ .

**Incremental Aggregated Gradient:** Finally, Blatt et al. [2008] present the most closely-related algorithm, the IAG method. This method is identical to the SAG iteration (6), but uses a cyclic choice of  $i_k$  rather than sampling the  $i_k$  values. This distinction has several important consequences.

In particular, Blatt et al. are only able to show that the convergence rate is linear for strongly-convex quadratic functions (without deriving an explicit rate), and their analysis treats full passes through the data as iterations. In contrast, in this work we give an explicit linear convergence rate for the SAG iterations for general strongly-convex functions using iterations that only examine a single training example. Further, as our analysis and experiments show, when the number of training examples is sufficiently large the SAG iterations achieve a linear convergence rate under a much larger set of step sizes than the IAG method. This leads to more robustness to the selection of the step size and also, if suitably chosen, leads to a faster convergence rate and improved practical performance.

### 3 Convergence Analysis

In our analysis we assume that each function  $f_i$  in (1) is differentiable and that each gradient  $f'_i$  is Lipschitz-continuous with constant  $L$ , meaning that for all  $x$  and  $y$  in  $\mathbb{R}^p$  we have

$$\|f'_i(x) - f'_i(y)\| \leq L\|x - y\|.$$

This is a fairly weak assumption on the  $f_i$  functions, and in cases where the  $f_i$  are twice-differentiable it is equivalent to saying that the eigenvalues of the Hessians of each  $f_i$  are bounded above by  $L$ . In addition, we also assume that the average function  $g = \frac{1}{n} \sum_i f_i$  is strongly-convex with constant  $\mu > 0$ , meaning that the function  $x \mapsto g(x) - \frac{\mu}{2}\|x\|^2$  is convex. This is a stronger assumption and is not satisfied by all machine learning models. However, note that in machine learning we are typically free to choose the regularizer, and we can always add an  $\ell_2$ -regularization term as in (3) to transform any convex problem into a strongly-convex problem (in this case we have  $\mu \geq \lambda$ ). Note that strong-convexity implies that the problem is solvable, meaning that there exists some unique  $x^*$  that achieves the optimal function value. Our results assume that we initialize  $y_i^0$  to a zero vector for all  $i$ , and they depend on the quantity  $\sigma^2 = \frac{1}{n} \sum_i \|f'_i(x^*)\|^2$ , which is the variance of the gradients at the optimum  $x^*$ .

We first consider the convergence rate of the method when using a constant step size of  $\alpha_k = \frac{1}{2nL}$ , which is similar to the step size needed for convergence of the IAG method in practice.

**Proposition 1** *With a step size of  $\alpha_k = \frac{1}{2nL}$ , the SAG iterations satisfy for  $k \geq 1$  that*

$$\mathbb{E} [\|x^k - x^*\|^2] \leq \left(1 - \frac{\mu}{8Ln}\right)^k \left[3\|x_0 - x^*\|^2 + \frac{9\sigma^2}{4L^2}\right].$$

The proof is given in the Appendix. Note that the SAG iterations also obtain the  $O(1/k)$  rate of SG methods, since

$$\left(1 - \frac{\mu}{8Ln}\right)^k \leq \exp\left(-\frac{k\mu}{8Ln}\right) = O(1/k),$$

albeit with a constant which is proportional to  $n$ . But they are advantageous over SG methods in later iterations because they obtain a linear convergence rate as in FG methods. We also note that a linear convergence rate is obtained for any  $\alpha \leq \frac{1}{2nL}$ .

With a step size of  $\alpha_k = \frac{1}{2nL}$ , the SAG iterations behave in a similar way to the IAG iterations proposed by Blatt et al. [2008], while  $n$  SAG iterations using this step size behave in a similar way to an FG method with a step size of  $1/2L$ . Thus, with this small step size, there is not a large difference between these three methods. However, our next result shows that, if the number of training examples is slightly larger than  $L/\mu$  (which will often be the case, as discussed in Section 6), then the SAG iterations can use a much larger step size and obtain a better convergence rate that depends on the number of training examples but not on  $\mu$  or  $L$ .

Data set	Training Examples	Variables	Reference
<b>quantum</b>	50 000	78	[Caruana et al., 2004]
<b>protein</b>	145 751	74	[Caruana et al., 2004]
<b>sido</b>	12 678	4 932	[Guyon, 2008]
<b>rcv1</b>	20 242	47 236	[Lewis et al., 2004]
<b>coverttype</b>	581 012	54	Blackard, Jock, and Dean [Frank and Asuncion, 2010]

Table 1: Binary data sets used in experiments.

**Proposition 2** *If  $\frac{\mu}{L} \geq \frac{8}{n}$ , with a step size of  $\alpha_k = \frac{1}{2n\mu}$  the SAG iterations satisfy for  $k \geq n$  that*

$$\mathbb{E} [g(x^k) - g(x^*)] \leq C \left(1 - \frac{1}{8n}\right)^k,$$

$$\text{with } C = \left[ \frac{16L}{3n} \|x^0 - x^*\|^2 + \frac{4\sigma^2}{3n\mu} \left(8 \log \left(1 + \frac{\mu n}{4L}\right) + 1\right) \right].$$

In this result we assume that the first  $n$  iterations of the algorithm use stochastic gradient descent and that we initialize the subsequent SAG iterations with the average of the iterates, which is why we state the result for  $k \geq n$ . This leads to an  $O(1/k)$  rate with a constant that is now only proportional to  $\log(n)$ , while if we used the SAG iterations from the beginning we can obtain the same convergence rate but the constant is again proportional to  $n$ . Also, note that this bound is obtained when initializing all  $y_i$  to zero after the stochastic gradient phase.<sup>1</sup>

Since Proposition 2 is true for all the values of  $\mu$  satisfying  $\frac{\mu}{L} \geq \frac{8}{n}$ , we can choose  $\mu = \frac{8L}{n}$ , and thus a step size as large as  $\alpha_k = \frac{1}{16L}$ , and still get the same convergence rate.<sup>2</sup> Note that we have observed in practice that the IAG method with a step size of  $\alpha_k = \frac{1}{2n\mu}$  and the FG method with a step size of  $\frac{1}{2\mu}$  may diverge, even under these assumptions. Thus, for certain problems the SAG iterations can use a much larger step size, which leads to increased robustness to the selection of the step size. Further, as our analysis and experiments indicate, the ability to use a large step size leads to improved performance of the SAG iterations.

While we have stated Proposition 1 in terms of the iterates and Proposition 2 in terms of the function value, note that the rates obtained on iterates and function values are equivalent because, by the Lipschitz and strong-convexity assumptions, we have

$$\frac{\mu}{2} \|x^k - x^*\|^2 \leq g(x^k) - g(x^*) \leq \frac{L}{2} \|x^k - x^*\|^2.$$

## 4 Implementation and First Pass

The bound in Proposition 2 assumes that the initialization of the SAG algorithm depends on the iterates obtained using  $n$  iterations of SG. In practice, this is not ideal as it requires the setting of two step sizes, one for the SG algorithm and one for the SAG algorithm. Thus, in our experiments we instead run the SAG algorithm from the beginning but we replace the factor  $n$  in the step size by  $m$ , the number of unique  $i_k$  values we have sampled (which converges to  $n$ ). This gives the algorithm a decreasing sequence of step sizes until all  $n$  training examples have been seen, making the initial

<sup>1</sup>While it may appear suboptimal to not use the gradients computed during the  $n$  iterations of stochastic gradient descent, using them only improves the bound by a constant.

<sup>2</sup>A more precise analysis, however, gives a rate of  $\left(1 - \frac{1}{3n} + \frac{2L}{n^2\mu}\right)^k$  for any  $\mu$  greater than  $\frac{6L}{n}$ . In this setting, increasing the step size decreases the convergence rate.

iterations very similar to SG iterations. In our experiments this modified SAG algorithm performed slightly better than using the SG method for the first iteration.

A natural way to implement the SAG iteration (6) is to have a current iterate  $x$ , a set of  $n$  variables  $y_i$  (initially zero), a search direction  $d$ , and to use the pseudo-code

$$\begin{aligned}
 d &\leftarrow d - y_i, \\
 y_i &\leftarrow f'_i(x^k), \\
 d &\leftarrow d + y_i, \\
 x &\leftarrow x - \frac{\alpha}{m}d.
 \end{aligned}
 \tag{8}$$

Other recursions may be more appealing depending on the structure of the problem. For example, for  $\ell_2$ -regularized objectives like (3), it may be more appealing to have the  $y_i$  variables only represent the contribution of the loss  $l_i$  and use

$$\begin{aligned}
 d &\leftarrow d - y_i, \\
 y_i &\leftarrow l'_i(x^k), \\
 d &\leftarrow d + y_i, \\
 x &\leftarrow \left(1 - \frac{\alpha\lambda}{m}\right)x - \frac{\alpha}{m}d.
 \end{aligned}
 \tag{9}$$

## 5 Experimental Results

In our experiments we compared the following methods:

1. **FG**: The full gradient method described by iteration (5).
2. **AFG**: The accelerated full gradient method of Nesterov [1983], where iterations of (5) are interleaved with an extrapolation step.
3. **IAG**: The incremental aggregated gradient method of Blatt et al. [2008] described by iteration (6) with a cyclic choice of  $i_k$ .
4. **SG**: The stochastic gradient method described by iteration (4), we tested both a constant step size and a decreasing sequence of step sizes.
5. **SAG**: The proposed stochastic average gradient method described by iteration (6).

The theoretical convergence rates suggest the following strategies for deciding on whether to use an FG or an SG method:

1. If we can only afford one pass through the data, then the SG method should be used.
2. If we can afford to do many passes through the data (say, several hundreds), then an FG method should be used.

We expect that the SAG iterations will be most useful between these two extremes, where we can afford to do more than one pass through the data but can not afford to do enough passes to warrant using FG algorithms. To test whether this is indeed the case on real data sets, we performed experiments on the set of freely available benchmark binary classification data sets listed in Table 1.

The *quantum* and *protein* data sets were obtained from the KDD Cup 2004 website,<sup>3</sup> the *sido* data set was obtained from the Causality Workbench website,<sup>4</sup> while the *rcv1* and *covertime* data sets were obtained from the LIBSVM Data website.<sup>5</sup> We plot in Figure 1 the results obtained when using the  $\ell_2$ -regularized logistic regression problem (2), and in Figure 2 the results obtained when using the Huberized support vector machine of Rosset and Zhu [2007] with a parameter of 0.5, which is

$$\underset{x \in \mathbb{R}^p}{\text{minimize}} \quad \frac{\lambda}{2} \|x\|^2 + \frac{1}{n} \sum_{i=1}^n l_i(x),$$

with the choice

$$l_i(x) = \begin{cases} 0 & \text{for } b_i a_i^T x \geq 1, \\ 0.75 - b_i a_i^T x & \text{for } b_i a_i^T x < 0.5, \\ (1 - b_i a_i^T x)^2 & \text{otherwise.} \end{cases}$$

This is a smooth objective function that is very similar to the objective used in support vector machines.

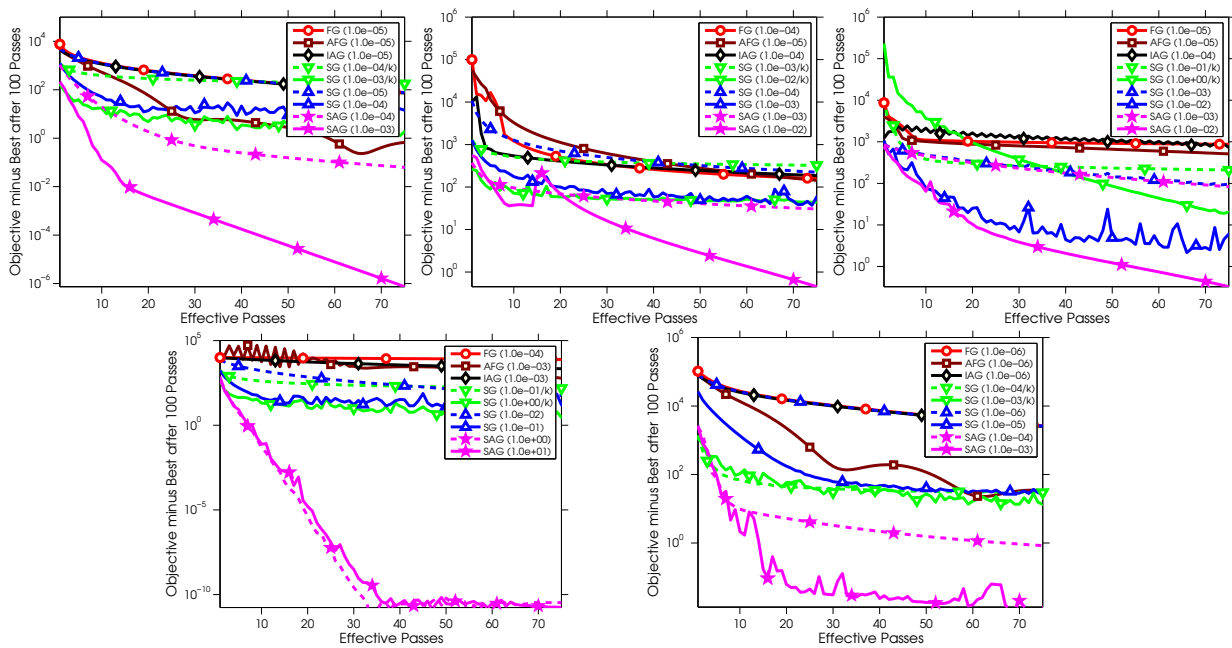


Figure 1: Comparison of optimization strategies for  $\ell_2$ -regularized logistic regression. The top row gives results on the *quantum* (left), *protein* (center), and *sido* (right) data sets. The bottom row gives results on the *rcv1* (left) and *covertime* (right) data sets. This figure is best viewed in colour.

In our experiments we set the regularization parameter  $\lambda$  to  $1/n$ , which is the smallest that would typically be used in practice (as discussed in the next section) and which results in difficult optimization problems. We plot the results of the different methods for the first 75 effective passes through the data in Figures 1 and 2. For dense problems the runtime of all methods is  $O(p)$  per training example, though in practice the actual runtime will depend on the specific computational architecture (sparse problems are discussed in the next section). Similar to SG methods, setting the

<sup>3</sup><http://osmot.cs.cornell.edu/kddcup>

<sup>4</sup><http://www.causality.inf.ethz.ch/home.php>

<sup>5</sup><http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets>



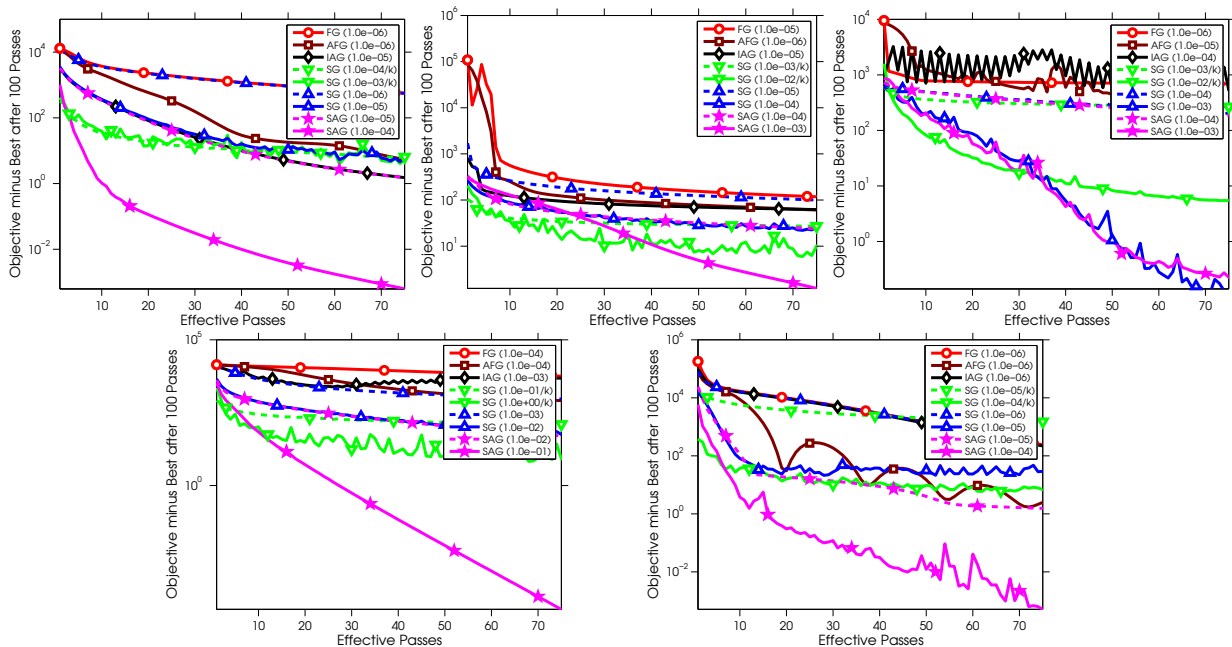


Figure 2: Comparison of optimization strategies for  $\ell_2$ -regularized Huberized support vector machine. The top row gives results on the quantum (left), protein (center), and sido (right) data sets. The bottom row gives results on the rcv1 (left) and coverytype (right) data sets. This figure is best viewed in colour.

step size in the IAG and SAG algorithms is a difficult issue so we only plot the step size that had the lowest function value after 75 passes (among powers of 10), and for the stochastic methods we also plot this step size divided by 10 to test the effect of choosing a poor step size. In the plots, we define the “best” function value as the lowest function value found across the methods and step sizes after 100 passes.

We can observe several trends across the experiments. First, we note that the SG methods usually do substantially better than the full gradient methods (FG and AFG) on the first few passes through the data. However, because of their steady progress we also see that the full gradient methods slowly catch up to the SG methods and eventually (or will eventually) pass them. The SAG iterations seem to achieve the best of both worlds. They start out substantially better than the full gradient methods, but continue to make steady (linear) progress which leads to better performance than SG methods. Indeed, after 25 passes through the data and beyond, the SAG iterations with a suitably chosen step size achieve the lowest error on every data set in the case of the  $\ell_2$ -regularized logistic regression and three out of five datasets in the case of the  $\ell_2$ -regularized Huberized SVM. Further, in the case of the  $\ell_2$ -regularized logistic regression, the SAG iterations even achieve a lower error than all other methods when not using the best step size in all but one case, indicating some robustness to this selection. This is in contrast to the SG methods where in many cases not using the best step size caused them to have similar or worse performance than the full gradient methods. Finally, we note that it is surprising that the randomized SAG method tends to outperform the closely-related deterministic IAG method by such a large margin, and we believe that this is due to the larger step sizes used by the SAG iterations; the step size that achieved the best performance for the SAG iteration was between 10 and 10000 times larger than for the IAG method.

## 6 Discussion

**Step-size selection and termination criteria:** The three major disadvantages of SG methods are: (i) the slow convergence rate, (ii) choosing the step size while running the algorithm, and (iii) deciding when to terminate the algorithm. Addressing the slow convergence rate was the focus of this paper, but as with previous work on SG methods we have not provided a fully satisfactory way to address choosing the step size. However, the SAG iterations achieve a linear convergence rate for any sufficiently small constant step size, which is in contrast to SG methods where misspecifying the step sizes has a disastrous effect on the convergence rate, see Nemirovski et al. [2009, §2.1] and Bach and Moulines [2011, §3.1]. We also note that the SAG iterations suggest a natural termination criterion; since the search direction converges to zero we can use  $\|(x^k - x^{k-1})/\alpha^k\|$  as an approximation of the optimality of  $x_k$ .

**Mini-batches:** Because of the use of vectorization and parallelism in modern architectures, practical SG implementations often group training examples into ‘mini-batches’ and perform SG iterations on the mini-batches. We can also use mini-batches within the SAG iterations, and our analysis even provides some guidance on the choice of mini-batch size. Specifically, in machine learning problems we can often obtain an estimate of  $L$  and  $\mu$  and we can use these to ensure that the number of mini-batches is large enough to allow using the larger step-size from Proposition 2. Alternately, in cases where  $n$  is smaller than  $\mu/L$ , our analysis suggests somewhat paradoxically that it may be advantageous to duplicate training examples in order to increase  $n$ . We have performed preliminary simulations using this technique, and found that in some cases duplicating training examples and then applying the SAG iterations with a large step-size can be beneficial.

**Sparse problems:** In their present form, the SAG iterations are unappealing for sparse problems because we need to store the  $y_i$  variables and the update from  $x^k$  to  $x^{k+1}$  is typically dense. However, both of these effects can be lessened by the use of mini-batches, and sparsity in the data will typically lead to reduced storage requirements. For example, in recursion (9) the sparsity pattern of the  $a_i$  variables exactly matches the one of the  $y_i$  variables.

**Optimal regularization strength:** One might wonder if the additional hypothesis in Proposition 2 is satisfied in practice. In a learning context, where each function  $f_i$  is the loss associated to a single data point, then  $L$  is equal to the largest value of the loss second derivative (1 for the square loss, 1/4 for the logistic loss) times the uniform bound  $R$  on the norm of each data point. Thus, the constraint  $\frac{\mu}{L} \geq \frac{\delta}{n}$  is satisfied when  $\lambda \geq \frac{8R}{n}$ . In low-dimensional settings, the optimal regularization parameter is of the form  $C/n$  [Liang et al., 2009] where  $C$  is a scalar constant, and may thus violate the constraint. However, the improvement with respect to regularization parameters of the form  $\lambda = C/\sqrt{n}$  is known to be asymptotically negligible, and in any case, in such low-dimensional settings, regular stochastic or batch gradient descent may be efficient enough in practice. In the more interesting high-dimensional settings where the dimension  $p$  of our covariates is not small compared to the sample size  $n$ , then all theoretical analyses we are aware of advocate settings of  $\lambda$  which satisfy this constraint. For example, Sridharan et al. [2008] consider parameters of the form  $\lambda = \frac{CR}{\sqrt{n}}$  in the parametric setting, while Eberts and Steinwart [2011] consider  $\lambda = \frac{CR}{n^\beta}$  with  $\beta < 1$  in a non-parametric setting.

**Algorithm extensions:** Our analysis and experiments focused on using a particular gradient approximation within the simplest possible gradient method. However, there are a variety of alternative gradient methods available. It would be interesting to explore SAG-like versions of AFG methods and other classical optimization methods. The latter methods typically achieve better performance by adaptively choosing the step size, and it is intriguing to consider whether it would be possible to use the previous function value computed for each  $f_i$  to adaptively set the step size in SAG-like algorithms. Other interesting directions of future work include using non-uniform sampling (such as

sampling proportional to the Lipschitz constant of each function), and exploring variants of the SAG iteration that, following Agarwal and Duchi [2011], work on large-scale distributed architectures but preserve the fast convergence rates.

## **Acknowledgements**

Nicolas Le Roux, Mark Schmidt and Francis Bach are supported by the European Research Council (SIERRA-ERC-239993).

## Appendix: Proofs of the propositions

We present here the proofs of Propositions 1 and 2.

### A.1 Problem set-up and notations

We use  $g = \frac{1}{n} \sum_{i=1}^n f_i$  to denote a  $\mu$ -strongly convex function, where the functions  $f_i$ ,  $i = 1, \dots, n$  are convex functions from  $\mathbb{R}^p$  to  $\mathbb{R}$  with  $L$ -Lipschitz continuous gradients. Let us denote by  $x^*$  the unique minimizer of  $g$ .

For  $k \geq 1$ , the stochastic average gradient algorithm performs the recursion

$$x^k = x^{k-1} - \frac{\alpha}{n} \sum_{i=1}^n y_i^k,$$

where an  $i_k$  is selected in  $\{1, \dots, n\}$  uniformly at random and we set

$$y_i^k = \begin{cases} f'_i(x^{k-1}) & \text{if } i = i_k, \\ y_i^{k-1} & \text{otherwise.} \end{cases}$$

Denoting  $z_i^k$  a random variable which takes the value  $1 - \frac{1}{n}$  with probability  $\frac{1}{n}$  and  $-\frac{1}{n}$  otherwise (thus with zero expectation), this is equivalent to

$$\begin{aligned} y_i^k &= \left(1 - \frac{1}{n}\right) y_i^{k-1} + \frac{1}{n} f'_i(x^{k-1}) + z_i^k [f'_i(x^{k-1}) - y_i^{k-1}] \\ x^k &= x^{k-1} - \frac{\alpha}{n} \sum_{i=1}^n \left[ \left(1 - \frac{1}{n}\right) y_i^{k-1} + \frac{1}{n} f'_i(x^{k-1}) + z_i^k [f'_i(x^{k-1}) - y_i^{k-1}] \right] \\ &= x^{k-1} - \frac{\alpha}{n} \left[ \left(1 - \frac{1}{n}\right) e^\top y^{k-1} + g'(x^{k-1}) + (z^k)^\top [f'(x^{k-1}) - y^{k-1}] \right], \end{aligned}$$

with

$$e = \begin{pmatrix} I \\ \vdots \\ I \end{pmatrix} \in \mathbb{R}^{np \times p}, \quad f'(x) = \begin{pmatrix} f'_1(x) \\ \vdots \\ f'_n(x) \end{pmatrix} \in \mathbb{R}^{np}, \quad z^k = \begin{pmatrix} z_1^k I \\ \vdots \\ z_n^k I \end{pmatrix} \in \mathbb{R}^{np \times p}.$$

Using this definition of  $z^k$ , we have  $\mathbb{E}[(z^k)(z^k)^\top] = \frac{1}{n} I - \frac{1}{n} e e^\top$ .

We also use the notation

$$\theta^k = \begin{pmatrix} y_1^k \\ \vdots \\ y_n^k \\ x^k \end{pmatrix} \in \mathbb{R}^{(n+1)p}, \quad \theta^* = \begin{pmatrix} f'_1(x^*) \\ \vdots \\ f'_n(x^*) \\ x^* \end{pmatrix} \in \mathbb{R}^{(n+1)p}.$$

Finally, if  $M$  is a  $tp \times tp$  matrix and  $m$  is a  $tp \times p$  matrix, then:

- $\text{diag}(M)$  is the  $tp \times p$  matrix being the concatenation of the  $t$   $(p \times p)$ -blocks on the diagonal of  $M$ ;
- $\text{Diag}(m)$  is the  $tp \times tp$  block-diagonal matrix whose  $(p \times p)$ -blocks on the diagonal are equal to the  $(p \times p)$ -blocks of  $m$ .

## A.2 Outline of the proofs

Each Proposition will be proved in multiple steps.

1. We shall find a Lyapunov function  $Q$  from  $\mathbb{R}^{(n+1)p}$  to  $\mathbb{R}$  such that the sequence  $\mathbb{E}Q(\theta^k)$  decreases at a linear rate.
2. We shall prove that  $Q(\theta^k)$  dominates  $\|x^k - x^*\|^2$  (in the case of Proposition 1) or  $g(x^k) - g(x^*)$  (in the case of Proposition 2) by a constant for all  $k$ .
3. In the case of Proposition 2, we show how using one pass of stochastic gradient as the initialization provides the desired result.

Throughout the proofs,  $\mathcal{F}_k$  will denote the  $\sigma$ -field of information up to (and including time  $k$ ), i.e.,  $\mathcal{F}_k$  is the  $\sigma$ -field generated by  $z^1, \dots, z^k$ .

## A.3 Convergence results for stochastic gradient descent

The constant in both our bounds depends on the initialization chosen. While this does not affect the linear convergence of the algorithm, the bound we obtain for the first few passes through the data is the  $O(1/k)$  rate one would get using stochastic gradient descent, but with a constant proportional to  $n$ . This problem can be alleviated for the second bound by running stochastic gradient descent for a few iterations before running the SAG algorithm. In this section, we provide bounds for the stochastic gradient descent algorithm which will prove useful for the SAG algorithm.

The assumptions made in this section about the functions  $f_i$  and the function  $g$  are the same as the ones used for SAG. To get initial values for  $x^0$  and  $y^0$ , we will do one pass of standard stochastic gradient.

We denote by  $\sigma^2 = \frac{1}{n} \sum_{i=1}^n \|f'_i(x^*)\|^2$  the variance of the gradients at the optimum. We will use the following recursion:

$$\tilde{x}^k = \tilde{x}^{k-1} - \gamma_k f'_{i_k}(\tilde{x}^{k-1}) .$$

Denoting  $\delta_k = \mathbb{E}\|\tilde{x}^k - x^*\|^2$ , we have (following Bach and Moulines [2011])

$$\delta_k \leq \delta_{k-1} - 2\gamma_k(1 - \gamma_k L) \mathbb{E} [g'(\tilde{x}^{k-1})^\top (\tilde{x}^{k-1} - x^*)] + 2\gamma_k^2 \sigma^2 .$$

Indeed, we have

$$\begin{aligned} \|\tilde{x}^k - x^*\|^2 &= \|\tilde{x}^{k-1} - x^*\|^2 - 2\gamma_k f'_{i_k}(\tilde{x}^{k-1})^\top (\tilde{x}^{k-1} - x^*) + \gamma_k^2 \|f'_{i_k}(\tilde{x}^{k-1})\|^2 \\ &\leq \|\tilde{x}^{k-1} - x^*\|^2 - 2\gamma_k f'_{i_k}(\tilde{x}^{k-1})^\top (\tilde{x}^{k-1} - x^*) + 2\gamma_k^2 \|f'_{i_k}(x^*)\|^2 + 2\gamma_k^2 \|f'_{i_k}(\tilde{x}^{k-1}) - f'_{i_k}(x^*)\|^2 \\ &\leq \|\tilde{x}^{k-1} - x^*\|^2 - 2\gamma_k f'_{i_k}(\tilde{x}^{k-1})^\top (\tilde{x}^{k-1} - x^*) + 2\gamma_k^2 \|f'_{i_k}(x^*)\|^2 \\ &\quad + 2L\gamma_k^2 (f'_{i_k}(\tilde{x}^{k-1}) - f'_{i_k}(x^*))^\top (\tilde{x}^{k-1} - x^*) . \end{aligned}$$

By taking expectations, we get

$$\begin{aligned} \mathbb{E} [\|\tilde{x}^k - x^*\|^2 | \mathcal{F}_{k-1}] &\leq \|\tilde{x}^{k-1} - x^*\|^2 - 2\gamma_k g'(\tilde{x}^{k-1})^\top (\tilde{x}^{k-1} - x^*) + 2\gamma_k^2 \sigma^2 + 2L\gamma_k^2 g'(\tilde{x}^{k-1})^\top (\tilde{x}^{k-1} - x^*) \\ \mathbb{E} [\|\tilde{x}^k - x^*\|^2] &\leq \mathbb{E} [\|\tilde{x}^{k-1} - x^*\|^2] - 2\gamma_k(1 - \gamma_k L) \mathbb{E} [g'(\tilde{x}^{k-1})^\top (\tilde{x}^{k-1} - x^*)] + 2\gamma_k^2 \sigma^2 \end{aligned}$$

Thus, if we take

$$\gamma_k = \frac{1}{2L + \frac{\mu}{2}k} ,$$

we have  $\gamma_k \leq 2\gamma_k(1 - \gamma_k L)$  and

$$\begin{aligned}
\delta_k &\leq \delta_{k-1} - \gamma_k \mathbb{E} [g'(\tilde{x}^{k-1})^\top (x^{k-1} - x^*)] + 2\gamma_k^2 \sigma^2 \\
&\leq \delta_{k-1} - \gamma_k \left[ \mathbb{E} [g(x^{k-1}) - g(x^*)] + \frac{\mu}{2} \delta_{k-1} \right] + 2\gamma_k^2 \sigma^2 \text{ using the strong convexity of } g \\
\mathbb{E} g(x^{k-1}) - g(x^*) &\leq -\frac{1}{\gamma_k} \delta_k + \left( \frac{1}{\gamma_k} - \frac{\mu}{2} \right) \delta_{k-1} + 2\gamma_k \sigma^2 \\
&\leq -\left( 2L + \frac{\mu}{2} k \right) \delta_k + \left( 2L + \frac{\mu}{2} (k-1) \right) \delta_{k-1} + 2\gamma_k \sigma^2.
\end{aligned}$$

Averaging from  $i = 0$  to  $k-1$  and using the convexity of  $g$ , we have

$$\begin{aligned}
\frac{1}{k} \sum_{i=0}^{k-1} \mathbb{E} g(x^{k-1}) - g(x^*) &\leq \frac{2L}{k} \delta_0 + \frac{2\sigma^2}{k} \sum_{i=1}^k \gamma_i \\
\mathbb{E} g \left( \frac{1}{k} \sum_{i=0}^{k-1} x^i \right) - g(x^*) &\leq \frac{2L}{k} \delta_0 + \frac{2\sigma^2}{k} \sum_{i=1}^k \gamma_i \\
&\leq \frac{2L}{k} \|x^0 - x^*\|^2 + \frac{2\sigma^2}{k} \sum_{i=1}^k \frac{1}{2L + \frac{\mu}{2} i} \\
&\leq \frac{2L}{k} L \|x^0 - x^*\|^2 + \frac{2\sigma^2}{k} \int_0^k \frac{1}{2L + \frac{\mu}{2} t} dt \\
&\leq \frac{2L}{k} \|x^0 - x^*\|^2 + \frac{4\sigma^2}{k\mu} \log \left( 1 + \frac{\mu k}{4L} \right).
\end{aligned}$$

## A.4 Important lemma

In both proofs, our Lyapunov function contains a quadratic term  $R(\theta^k) = (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*)$  for some values of  $A$ ,  $b$  and  $c$ . The lemma below computes the value of  $R(\theta^k)$  in terms of elements of  $\theta^{k-1}$ .

**Lemma 1** *If  $P = \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix}$ , for  $A \in \mathbb{R}^{np \times np}$ ,  $b \in \mathbb{R}^{np \times p}$  and  $c \in \mathbb{R}^{p \times p}$ , then*

$$\begin{aligned}
&\mathbb{E} \left[ (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*) \middle| \mathcal{F}_{k-1} \right] \\
&= (y^{k-1} - f'(x^*))^\top \left[ \left( 1 - \frac{2}{n} \right) S + \frac{1}{n} \text{Diag}(\text{diag}(S)) \right] (y^{k-1} - f'(x^*)) \\
&+ \frac{1}{n} (f'(x^{k-1}) - f'(x^*))^\top \text{Diag}(\text{diag}(S)) (f'(x^{k-1}) - f'(x^*)) \\
&+ \frac{2}{n} (y^{k-1} - f'(x^*))^\top [S - \text{Diag}(\text{diag}(S))] (f'(x^{k-1}) - f'(x^*)) \\
&+ 2 \left( 1 - \frac{1}{n} \right) (y^{k-1} - f'(x^*))^\top \left[ b - \frac{\alpha}{n} ec \right] (x^{k-1} - x^*) \\
&+ \frac{2}{n} (f'(x^{k-1}) - f'(x^*))^\top \left[ b - \frac{\alpha}{n} ec \right] (x^{k-1} - x^*) \\
&+ (x^{k-1} - x^*)^\top c (x^{k-1} - x^*),
\end{aligned}$$

with

$$S = A - \frac{\alpha}{n} b e^\top - \frac{\alpha}{n} e b^\top + \frac{\alpha^2}{n^2} e c e^\top.$$

**Proof** We have

$$\begin{aligned} & \mathbb{E} \left[ (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*) \middle| \mathcal{F}_{k-1} \right] \\ &= E \left[ (y^k - f'(x^*))^\top A (y^k - f'(x^*)) + 2(y^k - f'(x^*))^\top b (x^k - x^*) + (x^k - x^*)^\top c (x^k - x^*) \middle| \mathcal{F}_{k-1} \right]. \end{aligned} \quad (10)$$

The first term (within the expectation) on the right-hand side of Eq. (10) is equal to

$$\begin{aligned} (y^k - f'(x^*))^\top A (y^k - f'(x^*)) &= \left(1 - \frac{1}{n}\right)^2 (y^{k-1} - f'(x^*))^\top A (y^{k-1} - f'(x^*)) \\ &+ \frac{1}{n^2} (f'(x^{k-1}) - f'(x^*))^\top A (f'(x^{k-1}) - f'(x^*)) \\ &+ [\text{diag}(z^k)(f'(x^{k-1}) - y^{k-1})]^\top A [\text{diag}(z^k)(f'(x^{k-1}) - y^{k-1})] \\ &+ \frac{2}{n} \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top A (f'(x^{k-1}) - f'(x^*)). \end{aligned}$$

The only random term (given  $\mathcal{F}_{k-1}$ ) is the third one whose expectation is equal to

$$\begin{aligned} & \mathbb{E} \left[ [\text{diag}(z^k)(f'(x^{k-1}) - y^{k-1})]^\top A [\text{diag}(z^k)(f'(x^{k-1}) - y^{k-1})] \middle| \mathcal{F}_{k-1} \right] \\ &= \frac{1}{n} (f'(x^{k-1}) - y^{k-1})^\top \left[ \text{Diag}(\text{diag}(A)) - \frac{1}{n} A \right] (f'(x^{k-1}) - y^{k-1}). \end{aligned}$$

The second term (within the expectation) on the right-hand side of Eq. (10) is equal to

$$\begin{aligned} (y^k - f'(x^*))^\top b (x^k - x^*) &= \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top b (x^{k-1} - x^*) \\ &+ \frac{1}{n} (f'(x^{k-1}) - f'(x^*))^\top b (x^{k-1} - x^*) \\ &- \frac{\alpha}{n} \left(1 - \frac{1}{n}\right)^2 (y^{k-1} - f'(x^*))^\top b e^\top (y^{k-1} - f'(x^*)) \\ &- \frac{\alpha}{n} \frac{1}{n} \left(1 - \frac{1}{n}\right) (f'(x^{k-1}) - f'(x^*))^\top b e^\top (y^{k-1} - f'(x^*)) \\ &- \frac{\alpha}{n} \frac{1}{n} \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top b e^\top (f'(x^{k-1}) - f'(x^*)) \\ &- \frac{\alpha}{n} \frac{1}{n^2} (f'(x^{k-1}) - f'(x^*))^\top b e^\top (f'(x^{k-1}) - f'(x^*)) \\ &- \frac{\alpha}{n} [\text{diag}(z^k)(f'(x^{k-1}) - y^{k-1})]^\top b (z^k)^\top [(f'(x^{k-1}) - y^{k-1})] \end{aligned}$$

The only random term (given  $\mathcal{F}_{k-1}$ ) is the last one whose expectation is equal to

$$\begin{aligned} & \mathbb{E} \left[ [\text{diag}(z^k)(f'(x^{k-1}) - y^{k-1})]^\top b (z^k)^\top [(f'(x^{k-1}) - y^{k-1})] \middle| \mathcal{F}_{k-1} \right] \\ &= \frac{1}{n} (f'(x^{k-1}) - y^{k-1})^\top \left( \text{Diag}(\text{diag}(b e^\top)) - \frac{1}{n} b e^\top \right) (f'(x^{k-1}) - y^{k-1}). \end{aligned}$$

The last term on the right-hand side of Eq. (10) is equal to

$$\begin{aligned}
(x^k - x^*)^\top c(x^k - x^*) &= (x^{k-1} - x^*)^\top c(x^{k-1} - x^*) \\
&+ \frac{\alpha^2}{n^2} \left(1 - \frac{1}{n}\right)^2 (y^{k-1} - f'(x^*))^\top ece^\top (y^{k-1} - f'(x^*)) \\
&+ \frac{\alpha^2}{n^2} \frac{1}{n^2} (f'(x^{k-1}) - f'(x^*))^\top ece^\top (f'(x^{k-1}) - f'(x^*)) \\
&- \frac{2\alpha}{n} \left(1 - \frac{1}{n}\right) (x^{k-1} - x^*)^\top ce^\top (y^{k-1} - f'(x^*)) \\
&- \frac{2\alpha}{n} \frac{1}{n} (x^{k-1} - x^*)^\top ce^\top (f'(x^{k-1}) - f'(x^*)) \\
&+ \frac{2\alpha^2}{n^2} \frac{1}{n} \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top ece^\top (f'(x^{k-1}) - f'(x^*)) \\
&+ \frac{\alpha^2}{n^2} [(z^k)^\top (f'(x^{k-1}) - y^{k-1})]^\top c [(z^k)^\top (f'(x^{k-1}) - y^{k-1})] .
\end{aligned}$$

The only random term (given  $\mathcal{F}_{k-1}$ ) is the last one whose expectation is equal to

$$\begin{aligned}
&\mathbb{E} \left[ [(z^k)^\top (f'(x^{k-1}) - y^{k-1})]^\top c [(z^k)^\top (f'(x^{k-1}) - y^{k-1})] \mid \mathcal{F}_{k-1} \right] \\
&= \frac{1}{n} (f'(x^{k-1}) - y^{k-1})^\top \left[ \text{Diag}(\text{diag}(ece^\top)) - \frac{1}{n} ece^\top \right] (f'(x^{k-1}) - y^{k-1}) .
\end{aligned}$$

Summing all these terms together, we get the following result:

$$\begin{aligned}
&\mathbb{E} \left[ (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*) \mid \mathcal{F}_{k-1} \right] \\
&= \left(1 - \frac{1}{n}\right)^2 (y^{k-1} - f'(x^*))^\top S (y^{k-1} - f'(x^*)) \\
&+ \frac{1}{n^2} (f'(x^{k-1}) - f'(x^*))^\top S (f'(x^{k-1}) - f'(x^*)) \\
&+ \frac{1}{n} (f'(x^{k-1}) - y^{k-1})^\top \left[ \text{Diag}(\text{diag}(S)) - \frac{1}{n} S \right] (f'(x^{k-1}) - y^{k-1}) \\
&+ \frac{2}{n} \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top S (f'(x^{k-1}) - f'(x^*)) \\
&+ 2 \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top \left[ b - \frac{\alpha}{n} ec \right] (x^{k-1} - x^*) \\
&+ \frac{2}{n} (f'(x^{k-1}) - f'(x^*))^\top \left[ b - \frac{\alpha}{n} ec \right] (x^{k-1} - x^*) \\
&+ (x^{k-1} - x^*)^\top c (x^{k-1} - x^*)
\end{aligned}$$

with  $S = A - \frac{\alpha}{n} be^\top - \frac{\alpha}{n} eb^\top + \frac{\alpha^2}{n^2} ece^\top = A - bc^{-1}b^\top + (b - \frac{\alpha}{n} ec)c^{-1}(b - \frac{\alpha}{n} ec)^\top$ .



Rewriting  $f'(x^{k-1}) - y^{k-1} = (f'(x^{k-1}) - f'(x^*)) - (y^{k-1} - f'(x^*))$ , we have

$$\begin{aligned}
& f'(x^{k-1}) - y^{k-1} \left[ \text{Diag}(\text{diag}(S)) - \frac{1}{n}S \right] (f'(x^{k-1}) - y^{k-1}) \\
&= (f'(x^{k-1}) - f'(x^*))^\top \left[ \text{Diag}(\text{diag}(S)) - \frac{1}{n}S \right] (f'(x^{k-1}) - f'(x^*)) \\
&+ (y^{k-1} - f'(x^*))^\top \left[ \text{Diag}(\text{diag}(S)) - \frac{1}{n}S \right] (y^{k-1} - f'(x^*)) \\
&- 2(y^{k-1} - f'(x^*))^\top \left[ \text{Diag}(\text{diag}(S)) - \frac{1}{n}S \right] (f'(x^{k-1}) - f'(x^*)).
\end{aligned}$$

Hence, the sum may be rewritten as

$$\begin{aligned}
& \mathbb{E} \left[ (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*) \middle| \mathcal{F}_{k-1} \right] \\
&= (y^{k-1} - f'(x^*))^\top \left[ \left(1 - \frac{2}{n}\right)S + \frac{1}{n} \text{Diag}(\text{diag}(S)) \right] (y^{k-1} - f'(x^*)) \\
&+ \frac{1}{n} (f'(x^{k-1}) - f'(x^*))^\top \text{Diag}(\text{diag}(S)) (f'(x^{k-1}) - f'(x^*)) \\
&+ \frac{2}{n} (y^{k-1} - f'(x^*))^\top [S - \text{Diag}(\text{diag}(S))] (f'(x^{k-1}) - f'(x^*)) \\
&+ 2 \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top \left[ b - \frac{\alpha}{n}ec \right] (x^{k-1} - x^*) \\
&+ \frac{2}{n} (f'(x^{k-1}) - f'(x^*))^\top \left[ b - \frac{\alpha}{n}ec \right] (x^{k-1} - x^*) \\
&+ (x^{k-1} - x^*)^\top c (x^{k-1} - x^*)
\end{aligned}$$

This concludes the proof. ■

## A.5 Analysis for $\alpha = \frac{1}{2nL}$

We now prove Proposition 1, providing a bound for the convergence rate of the SAG algorithm in the case of a small step size,  $\alpha = \frac{1}{2nL}$ .

**Proof**

### Step 1 - Linear convergence of the Lyapunov function

In this case, our Lyapunov function is quadratic, i.e.,

$$Q(\theta^k) = (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*).$$

We consider

$$\begin{aligned}
A &= 3n\alpha^2 I + \frac{\alpha^2}{n} \left(\frac{1}{n} - 2\right) ee^\top \\
b &= -\alpha \left(1 - \frac{1}{n}\right) e \\
c &= I \\
S &= 3n\alpha^2 I \\
b - \frac{\alpha}{n} ec &= -\alpha e.
\end{aligned}$$

The goal will be to prove that  $\mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1})$  is negative for some  $\delta > 0$ . This will be achieved by bounding all the terms by a term depending on  $g'(x^{k-1})^\top(x^{k-1} - x^*)$  whose positivity is guaranteed by the convexity of  $g$ .

We have, with our definition of  $A$ ,  $b$  and  $c$ :

$$\begin{aligned}
S - \text{Diag}(\text{diag}(S)) &= 3n\alpha^2 I - 3n\alpha^2 I = 0 \\
e^\top (f'(x^{k-1}) - f'(x^*)) &= n[g'(x^{k-1}) - g'(x^*)] = ng'(x^{k-1}).
\end{aligned}$$

This leads to (using the lemma of the previous section):

$$\begin{aligned}
\mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] &= \mathbb{E}\left[(\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*) \middle| \mathcal{F}_{k-1}\right] \\
&= \left(1 - \frac{1}{n}\right) 3n\alpha^2 (y^{k-1} - f'(x^*))^\top (y^{k-1} - f'(x^*)) \\
&\quad + (x^{k-1} - x^*)^\top (x^{k-1} - x^*) - \frac{2\alpha}{n} (x^{k-1} - x^*)^\top e^\top (f'(x^{k-1}) - f'(x^*)) \\
&\quad + 3\alpha^2 (f'(x^{k-1}) - f'(x^*))^\top (f'(x^{k-1}) - f'(x^*)) \\
&\quad - 2\alpha \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top e (x^{k-1} - x^*) \\
&= \left(1 - \frac{1}{n}\right) 3n\alpha^2 (y^{k-1} - f'(x^*))^\top (y^{k-1} - f'(x^*)) \\
&\quad + (x^{k-1} - x^*)^\top (x^{k-1} - x^*) - 2\alpha (x^{k-1} - x^*)^\top g'(x^{k-1}) \\
&\quad + 3\alpha^2 (f'(x^{k-1}) - f'(x^*))^\top (f'(x^{k-1}) - f'(x^*)) \\
&\quad - 2\alpha \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top e (x^{k-1} - x^*) \\
&\leq \left(1 - \frac{1}{n}\right) 3n\alpha^2 (y^{k-1} - f'(x^*))^\top (y^{k-1} - f'(x^*)) \\
&\quad + (x^{k-1} - x^*)^\top (x^{k-1} - x^*) - 2\alpha (x^{k-1} - x^*)^\top g'(x^{k-1}) \\
&\quad + 3\alpha^2 nL (x^{k-1} - x^*)^\top g'(x^{k-1}) \\
&\quad - 2\alpha \left(1 - \frac{1}{n}\right) (y^{k-1} - f'(x^*))^\top e (x^{k-1} - x^*).
\end{aligned}$$

The third line is obtained using the Lipschitz property of the gradient, that is

$$\begin{aligned}
(f'(x^{k-1}) - f'(x^*))^\top (f'(x^{k-1}) - f'(x^*)) &= \sum_{i=1}^n \|f'_i(x^{k-1}) - f'_i(x^*)\|^2 \\
&\leq \sum_{i=1}^n L(f'_i(x^{k-1}) - f'_i(x^*))^\top (x^{k-1} - x^*) \\
&= nL(g'(x^{k-1}) - g'(x^*))^\top (x^{k-1} - x^*).
\end{aligned}$$

We have

$$\begin{aligned}
(1 - \delta)Q(\theta^{k-1}) &= (1 - \delta)(\theta^{k-1} - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^{k-1} - \theta^*) \\
&= (1 - \delta)(y^{k-1} - f'(x^*))^\top \left[ 3n\alpha^2 I + \frac{\alpha^2}{n} \left( \frac{1}{n} - 2 \right) ee^\top \right] (y^{k-1} - f'(x^*)) \\
&\quad + (1 - \delta)(x^{k-1} - x^*)^\top (x^{k-1} - x^*) \\
&\quad - 2\alpha(1 - \delta) \left( 1 - \frac{1}{n} \right) (y^{k-1} - f'(x^*))^\top e(x^{k-1} - x^*).
\end{aligned}$$

The difference is then:

$$\begin{aligned}
&\mathbb{E}[Q(\theta^k) | \mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1}) \\
&\leq (y^{k-1} - f'(x^*))^\top \left[ 3n\alpha^2 \left( \delta - \frac{1}{n} \right) I + (1 - \delta) \frac{\alpha^2}{n} \left( 2 - \frac{1}{n} \right) ee^\top \right] (y^{k-1} - f'(x^*)) \\
&\quad + \delta(x^{k-1} - x^*)^\top (x^{k-1} - x^*) \\
&\quad - (2\alpha - 3\alpha^2 nL)(x^{k-1} - x^*)^\top g'(x^{k-1}) \\
&\quad - 2\alpha\delta \left( 1 - \frac{1}{n} \right) (y^{k-1} - f'(x^*))^\top e(x^{k-1} - x^*).
\end{aligned}$$

Using the fact that, for any negative definite matrix  $M$  and for any vectors  $z$  and  $t$ , we have

$$z^\top M z + z^\top t \leq -\frac{1}{4} t^\top M^{-1} t,$$

we have, with

$$\begin{aligned}
M &= \left[ 3n\alpha^2 \left( \delta - \frac{1}{n} \right) I + (1 - \delta) \frac{\alpha^2}{n} \left( 2 - \frac{1}{n} \right) ee^\top \right] \\
&= \left[ 3n\alpha^2 \left( \delta - \frac{1}{n} \right) \left( I - \frac{ee^\top}{n} \right) + \alpha^2 \left[ 3n\delta - 1 - 2\delta + \frac{\delta - 1}{n} \right] \frac{ee^\top}{n} \right] \\
z &= y^{k-1} - f'(x^*) \\
t &= -2\alpha\delta \left( 1 - \frac{1}{n} \right) e(x^{k-1} - x^*),
\end{aligned}$$

$$\begin{aligned}
& (y^{k-1} - f'(x^*))^\top \left[ 3n\alpha^2 \left( \delta - \frac{1}{n} \right) I + (1-\delta) \frac{\alpha^2}{n} \left( 2 - \frac{1}{n} \right) ee^\top \right] (y^{k-1} - f'(x^*)) \\
& \quad - 2\alpha\delta \left( 1 - \frac{1}{n} \right) (y^{k-1} - f'(x^*))^\top e (x^{k-1} - x^*) \\
& \leq -\alpha^2\delta^2 \left( 1 - \frac{1}{n} \right)^2 (x^{k-1} - x^*)^\top e^\top \left[ 3n\alpha^2 \left( \delta - \frac{1}{n} \right) \left( I - \frac{ee^\top}{n} \right) + \alpha^2 \left[ 3n\delta - 1 - 2\delta + \frac{\delta-1}{n} \right] \frac{ee^\top}{n} \right]^{-1} e (x^{k-1} - x^*) \\
& = -\frac{\alpha^2\delta^2 \left( 1 - \frac{1}{n} \right)^2 n}{\alpha^2 [3n\delta - 1 - 2\delta + \frac{\delta-1}{n}]} \|x^{k-1} - x^*\|^2 \\
& = -\frac{\delta^2 \left( 1 - \frac{1}{n} \right)^2 n}{[3n\delta - 1 - 2\delta + \frac{\delta-1}{n}]} \|x^{k-1} - x^*\|^2.
\end{aligned}$$

A sufficient condition for  $M$  to be negative definite is to have  $\delta \leq \frac{1}{3n}$ .

The bound then becomes

$$\begin{aligned}
\mathbb{E}[Q(\theta^k) | \mathcal{F}_{k-1}] - (1-\delta)Q(\theta^{k-1}) & \leq -(2\alpha - 3\alpha^2 nL)(x^{k-1} - x^*)^\top g'(x^{k-1}) \\
& \quad + \left( \delta - \frac{\delta^2 \left( 1 - \frac{1}{n} \right)^2}{[3n\delta - 1 - 2\delta + \frac{\delta-1}{n}]} n \right) \|x^{k-1} - x^*\|^2.
\end{aligned}$$

We now use the strong convexity of  $g$  to get the inequality

$$\|x^{k-1} - x^*\|^2 \leq \frac{1}{\mu} (x^{k-1} - x^*)^\top g'(x^{k-1}).$$

This yields the final bound

$$\mathbb{E}[Q(\theta^k) | \mathcal{F}_{k-1}] - (1-\delta)Q(\theta^{k-1}) \leq - \left( 2\alpha - 3\alpha^2 nL + \frac{\delta^2 \left( 1 - \frac{1}{n} \right)^2}{[3n\delta - 1 - 2\delta + \frac{\delta-1}{n}]} \frac{n}{\mu} - \frac{\delta}{\mu} \right) (x^{k-1} - x^*)^\top g'(x^{k-1}).$$

Since we know that  $(x^{k-1} - x^*)^\top g'(x^{k-1})$  is positive, due to the convexity of  $g$ , we need to prove that  $\left( 2\alpha - 3\alpha^2 nL + \frac{\delta^2 \left( 1 - \frac{1}{n} \right)^2}{[3n\delta - 1 - 2\delta + \frac{\delta-1}{n}]} \frac{n}{\mu} - \frac{\delta}{\mu} \right)$  is positive.

Using  $\delta = \frac{\mu}{8nL}$  and  $\alpha = \frac{1}{2nL}$  gives

$$\begin{aligned}
2\alpha - 3\alpha^2 nL + \frac{\delta^2 \left( 1 - \frac{1}{n} \right)^2}{[3n\delta - 1 - 2\delta + \frac{\delta-1}{n}]} \frac{n}{\mu} - \frac{\delta}{\mu} & = \frac{1}{nL} - \frac{3}{4nL} - \frac{1}{8nL} - \frac{\delta^2 \left( 1 - \frac{1}{n} \right)^2 \frac{n}{\mu}}{1 - 3n\delta + 2\delta + \frac{1-\delta}{n}} \\
& \geq \frac{1}{8nL} - \frac{\delta^2 \frac{n}{\mu}}{1 - 3n\delta} \\
& = \frac{1}{8nL} - \frac{\frac{\mu}{64nL^2}}{1 - \frac{3\mu}{8L}} \\
& \geq \frac{1}{8nL} - \frac{\frac{\mu}{64nL^2}}{1 - \frac{3}{8}} \\
& = \frac{1}{8nL} - \frac{\mu}{40nL^2} \\
& = \frac{1}{8nL} - \frac{1}{40nL} \\
& \geq 0.
\end{aligned}$$

Hence,

$$\mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1}) \leq 0.$$

We can then take a full expectation on both sides to obtain:

$$\mathbb{E}Q(\theta^k) - (1 - \delta)\mathbb{E}Q(\theta^{k-1}) \leq 0.$$

Since  $Q$  is a non-negative function (we show below that it dominates a non-negative function), this results proves the linear convergence of the sequence  $\mathbb{E}Q(\theta^k)$  with rate  $1 - \delta$ . We have

$$\mathbb{E}Q(\theta^k) \leq \left(1 - \frac{\mu}{8nL}\right)^k Q(\theta^0).$$

### Step 2 - Domination of $\|x^k - x^*\|^2$ by $Q(\theta^k)$

We now need to prove that  $Q(\theta^k)$  dominates  $\|x^k - x^*\|^2$ . If  $P - \begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{3}I \end{pmatrix}$  is positive definite, then  $Q(\theta^k) \geq \frac{1}{3}\|x^k - x^*\|^2$ .

We shall use the Schur complement condition for positive definiteness. Since  $A$  is positive definite, the other condition to verify is  $\frac{2}{3}I - b^\top A^{-1}b \succ 0$ .

$$\begin{aligned} \frac{2}{3}I - \alpha^2 \left(1 - \frac{1}{n}\right)^2 e^\top \left[ \left(3n\alpha^2 + \frac{\alpha^2}{n} - 2\alpha^2\right) \frac{ee^\top}{n} \right]^{-1} e &= \frac{2}{3}I - \frac{n \left(1 - \frac{1}{n}\right)^2}{3n + \frac{1}{n} - 2} \frac{ee^\top}{n} \\ &\succ \frac{2}{3}I - \frac{n}{3n - 2} \frac{ee^\top}{n} \\ &\succ 0 \text{ for } n \geq 2, \end{aligned}$$

and so  $P$  dominates  $\begin{pmatrix} 0 & 0 \\ 0 & \frac{1}{3}I \end{pmatrix}$ .

This yields

$$\begin{aligned} \mathbb{E}\|x^k - x^*\|^2 &\leq 3\mathbb{E}Q(\theta^k) \\ &\leq 3 \left(1 - \frac{\mu}{8nL}\right)^k Q(\theta^0). \end{aligned}$$

We have

$$\begin{aligned} Q(\theta^0) &= 3n\alpha^2 \sum_i \|y_i^0 - f'_i(x^*)\|^2 + \frac{(1 - 2n)\alpha}{n^2} \left\| \sum_i y_i^0 \right\|^2 - 2\alpha \left(1 - \frac{1}{n}\right) (x^0 - x^*)^\top \left( \sum_i y_i^0 \right) + \|x^0 - x^*\|^2 \\ &= \frac{3}{4nL^2} \sum_i \|y_i^0 - f'_i(x^*)\|^2 + \frac{(1 - 2n)}{2n^3L} \left\| \sum_i y_i^0 \right\|^2 - \frac{n - 1}{n^2L} (x^0 - x^*)^\top \left( \sum_i y_i^0 \right) + \|x^0 - x^*\|^2. \end{aligned}$$

Initializing all the  $y_i^0$  to 0, we get

$$Q(\theta^0) = \frac{3\sigma^2}{4L^2} + \|x^0 - x^*\|^2,$$

and

$$\mathbb{E}\|x^k - x^*\|^2 \leq \left(1 - \frac{\mu}{8nL}\right)^k \left( \frac{9\sigma^2}{4L^2} + 3\|x^0 - x^*\|^2 \right).$$

■

## A.6 Analysis for $\alpha = \frac{1}{2n\mu}$

### Step 1 - Linear convergence of the Lyapunov function

We now prove Proposition 2, providing a bound for the convergence rate of the SAG algorithm in the case of a small step size,  $\alpha = \frac{1}{2n\mu}$ .

We shall use the following Lyapunov function:

$$Q(\theta^k) = 2g\left(x^k + \frac{\alpha}{n}e^\top y^k\right) - 2g(x^*) + (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*),$$

with

$$\begin{aligned} A &= \frac{\eta\alpha}{n}I + \frac{\alpha}{n}(1 - 2\nu)ee^\top \\ b &= -\nu e \\ c &= 0. \end{aligned}$$

This yields

$$\begin{aligned} S &= \frac{\eta\alpha}{n}I + \frac{\alpha}{n}ee^\top \\ \text{Diag}(\text{diag}(S)) &= \frac{(1 + \eta)\alpha}{n}I \\ S - \text{Diag}(\text{diag}(S)) &= \frac{\alpha}{n}(ee^\top - I) \\ \left(1 - \frac{2}{n}\right)S + \frac{1}{n}\text{Diag}(\text{diag}(S)) &= \left(1 - \frac{2}{n}\right)\left[\frac{\eta\alpha}{n}I + \frac{\alpha}{n}ee^\top\right] + \frac{1}{n}\frac{(1 + \eta)\alpha}{n}I = \left(1 - \frac{2}{n}\right)\frac{\alpha}{n}ee^\top + \left(\eta - \frac{\eta - 1}{n}\right)\frac{\alpha}{n}I. \end{aligned}$$

We have

$$\begin{aligned} &\mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1}) \\ &= 2g(x^{k-1}) - 2g(x^*) - 2(1 - \delta)g\left(x^{k-1} + \frac{\alpha}{n}e^\top y^{k-1}\right) + 2(1 - \delta)g(x^*) \\ &\quad + (y^{k-1} - f'(x^*))^\top \left[ \left(1 - \frac{2}{n}\right)\frac{\alpha}{n}ee^\top + \left(\eta - \frac{\eta - 1}{n}\right)\frac{\alpha}{n}I - (1 - \delta)\frac{\eta\alpha}{n}I - (1 - \delta)\frac{\alpha}{n}(1 - 2\nu)ee^\top \right] (y^{k-1} - f'(x^*)) \\ &\quad - \frac{2\nu}{n}(x^{k-1} - x^*)^\top e^\top (f'(x^{k-1}) - f'(x^*)) \\ &\quad + \frac{(1 + \eta)\alpha}{n^2}(f'(x^{k-1}) - f'(x^*))^\top (f'(x^{k-1}) - f'(x^*)) \\ &\quad + \frac{2\alpha}{n^2}(y^{k-1} - f'(x^*))^\top [ee^\top - I] (f'(x^{k-1}) - f'(x^*)) \\ &\quad + 2\left(\frac{1}{n} - \delta\right)\nu(y^{k-1} - f'(x^*))^\top e(x^{k-1} - x^*). \end{aligned}$$

Our goal will now be to express all the quantities in terms of  $(x^{k-1} - x^*)^\top g'(x^{k-1})$  whose positivity is guaranteed by the convexity of  $g$ .

Using the convexity of  $g$ , we have

$$-2(1-\delta)g\left(x^{k-1} + \frac{\alpha}{n}e^\top y^{k-1}\right) \leq -2(1-\delta)\left[g(x^{k-1}) + \frac{\alpha}{n}g'(x^{k-1})e^\top y^{k-1}\right].$$

Using the Lipschitz property of the gradients of  $f_i$ , we have

$$\begin{aligned} (f'(x^{k-1}) - f'(x^*))^\top (f'(x^{k-1}) - f'(x^*)) &= \sum_{i=1}^n \|f'_i(x^{k-1}) - f'_i(x^*)\|^2 \\ &\leq \sum_{i=1}^n L(f'_i(x^{k-1}) - f'_i(x^*))^\top (x^{k-1} - x^*) \\ &= nL(g'(x^{k-1}) - g'(x^*))^\top (x^{k-1} - x^*). \end{aligned}$$

Using  $e^\top f'(x^{k-1}) - f'(x^*) = ng'(x^{k-1})$ , we have

$$\begin{aligned} -\frac{2\nu}{n}(x^{k-1} - x^*)^\top e^\top (f'(x^{k-1}) - f'(x^*)) &= -2\nu(x^{k-1} - x^*)^\top g'(x^{k-1}) \\ \frac{2\alpha}{n^2}(y^{k-1} - f'(x^*))^\top ee^\top (f'(x^{k-1}) - f'(x^*)) &= \frac{2\alpha}{n}(y^{k-1} - f'(x^*))^\top eg'(x^{k-1}). \end{aligned}$$

Reassembling all the terms together, we get

$$\begin{aligned} \mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1-\delta)Q(\theta^{k-1}) &\leq 2\delta[g(x^{k-1}) - \delta g(x^*)] + \frac{2\delta\alpha}{n}g'(x^{k-1})e^\top y^{k-1} \\ &\quad + (y^{k-1} - f'(x^*))^\top \left[ \left(1 - \frac{2}{n}\right) \frac{\alpha}{n}ee^\top + \left(\eta - \frac{\eta-1}{n}\right) \frac{\alpha}{n}I - (1-\delta)\frac{\eta\alpha}{n}I - (1-\delta)\frac{\alpha}{n}(1-2\nu)ee^\top \right] (y^{k-1} - f'(x^*)) \\ &\quad - \left(2\nu - \frac{(1+\eta)\alpha L}{n}\right) (x^{k-1} - x^*)^\top g'(x^{k-1}) \\ &\quad - \frac{2\alpha}{n^2}(y^{k-1} - f'(x^*))^\top (f'(x^{k-1}) - f'(x^*)) \\ &\quad + 2\left(\frac{1}{n} - \delta\right) \nu(y^{k-1} - f'(x^*))^\top e(x^{k-1} - x^*). \end{aligned}$$

Using the convexity of  $g$  gives

$$2\delta[g(x^{k-1}) - \delta g(x^*)] \leq 2\delta[x^{k-1} - x^*]^\top g'(x^{k-1}),$$

and, consequently,

$$\begin{aligned} \mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1-\delta)Q(\theta^{k-1}) &\leq 2\delta[(x^{k-1}) - (x^*)]^\top g'(x^{k-1}) + \frac{2\delta\alpha}{n}g'(x^{k-1})e^\top y^{k-1} \\ &\quad + (y^{k-1} - f'(x^*))^\top \left[ \left(1 - \frac{2}{n}\right) \frac{\alpha}{n}ee^\top + \left(\eta - \frac{\eta-1}{n}\right) \frac{\alpha}{n}I - (1-\delta)\frac{\eta\alpha}{n}I - (1-\delta)\frac{\alpha}{n}(1-2\nu)ee^\top \right] (y^{k-1} - f'(x^*)) \\ &\quad - \left(2\nu - \frac{(1+\eta)\alpha L}{n}\right) (x^{k-1} - x^*)^\top g'(x^{k-1}) \\ &\quad - \frac{2\alpha}{n^2}(y^{k-1} - f'(x^*))^\top (f'(x^{k-1}) - f'(x^*)) \\ &\quad + 2\left(\frac{1}{n} - \delta\right) \nu(y^{k-1} - f'(x^*))^\top e(x^{k-1} - x^*). \end{aligned}$$

If we regroup all the terms in  $[(x^{k-1}) - (x^*)]^\top g'(x^{k-1})$  together, and all the terms in  $(y^{k-1} - f'(x^*))^\top$  together, we get

$$\begin{aligned} & \mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1}) \\ & \leq \frac{\alpha}{n}(y^{k-1} - f'(x^*))^\top \left[ \left( \delta\eta - \frac{\eta - 1}{n} \right) I + \left( \delta - \frac{2}{n} + 2\nu(1 - \delta) \right) ee^\top \right] (y^{k-1} - f'(x^*)) \\ & \quad - \left( 2\nu - 2\delta - \frac{(1 + \eta)\alpha L}{n} \right) (x^{k-1} - x^*)^\top g'(x^{k-1}) \\ & \quad + 2(y^{k-1} - f'(x^*))^\top \left[ -\frac{\alpha}{n^2}(f'(x^{k-1}) - f'(x^*)) + \left( \frac{1}{n} - \delta \right) \nu e(x^{k-1} - x^*) + \frac{\delta\alpha}{n} eg'(x^{k-1}) \right]. \end{aligned}$$

Let us rewrite this as

$$\begin{aligned} & \mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1}) \\ & \leq (y^{k-1} - f'(x^*))^\top \left( \tau_{y,I} I + \tau_{y,e} \frac{ee^\top}{n} \right) (y^{k-1} - f'(x^*)) \\ & \quad + \tau_{x,g} (x^{k-1} - x^*)^\top g'(x^{k-1}) \\ & \quad + (y^{k-1} - f'(x^*))^\top \left[ \tau_{y,f} (f'(x^{k-1}) - f'(x^*)) + \tau_{y,x} e(x^{k-1} - x^*) + \tau_{y,g} eg'(x^{k-1}) \right] \end{aligned}$$

with

$$\begin{aligned} \tau_{y,I} &= \frac{\alpha}{n} \left( \delta\eta - \frac{\eta - 1}{n} \right) \\ \tau_{y,e} &= \alpha \left( \delta - \frac{2}{n} + 2\nu(1 - \delta) \right) \\ \tau_{x,g} &= -(2\nu - 2\delta - \frac{(1 + \eta)\alpha L}{n}) \\ \tau_{y,f} &= -\frac{2\alpha}{n^2} \\ \tau_{y,x} &= 2 \left( \frac{1}{n} - \delta \right) \nu \\ \tau_{y,g} &= \frac{2\delta\alpha}{n}. \end{aligned}$$

Assuming that  $\tau_{y,I}$  and  $\tau_{y,e}$  are negative, we have by completing the square that

$$\begin{aligned} & (y^{k-1} - f'(x^*))^\top \left( \tau_{y,I} I + \tau_{y,e} \frac{ee^\top}{n} \right) (y^{k-1} - f'(x^*)) \\ & \quad + (y^{k-1} - f'(x^*))^\top \left( \tau_{y,f} (f'(x^{k-1}) - f'(x^*)) + \tau_{y,x} e(x^{k-1} - x^*) + \tau_{y,g} eg'(x^{k-1}) \right) \\ & \leq -\frac{1}{4} \left( \tau_{y,f} (f'(x^{k-1}) - f'(x^*)) + \tau_{y,x} e(x^{k-1} - x^*) + \tau_{y,g} eg'(x^{k-1}) \right)^\top \left( \frac{1}{\tau_{y,I}} \left( I - \frac{ee^\top}{n} \right) + \frac{1}{\tau_{y,I} + \tau_{y,e}} \frac{ee^\top}{n} \right) \\ & \quad \left( \tau_{y,f} (f'(x^{k-1}) - f'(x^*)) + \tau_{y,x} e(x^{k-1} - x^*) + \tau_{y,g} eg'(x^{k-1}) \right) \\ & = -\frac{1}{4} \frac{\tau_{y,f}^2}{\tau_{y,I}} \|f'(x^{k-1}) - f'(x^*)\|^2 - \frac{1}{4} \frac{\tau_{y,f}^2 n}{\tau_{y,I} + \tau_{y,e}} \|g'(x^{k-1})\|^2 \left( \frac{1}{\tau_{y,I} + \tau_{y,e}} - \frac{1}{\tau_{y,I}} \right) \\ & \quad - \frac{1}{4} \frac{\tau_{y,x}^2 n}{\tau_{y,I} + \tau_{y,e}} \|x^{k-1} - x^*\|^2 - \frac{1}{4} \frac{\tau_{y,g}^2 n}{\tau_{y,I} + \tau_{y,e}} \|g'(x^{k-1})\|^2 \\ & \quad - \frac{1}{2} \frac{\tau_{y,f} \tau_{y,x} n}{\tau_{y,I} + \tau_{y,e}} (x^{k-1} - x^*)^\top g'(x^{k-1}) - \frac{1}{2} \frac{\tau_{y,f} \tau_{y,g} n}{\tau_{y,I} + \tau_{y,e}} \|g'(x^{k-1})\|^2 - \frac{1}{2} \frac{\tau_{y,g} \tau_{y,x} n}{\tau_{y,I} + \tau_{y,e}} (x^{k-1} - x^*)^\top g'(x^{k-1}), \end{aligned}$$



where we used the fact that  $(f'(x^{k-1}) - f'(x^*))^\top e = g'(x^{k-1})$ . After reorganization of the terms, we obtain

$$\begin{aligned} \mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1}) &\leq \left[ \tau_{x,g} - \frac{n\tau_{y,x}}{2(\tau_{y,I} + \tau_{y,e})}(\tau_{y,f} + \tau_{y,g}) \right] (x^{k-1} - x^*)^\top g'(x^{k-1}) \\ &\quad - \left[ \frac{1}{4}\tau_{y,f}^2 n \left( \frac{1}{\tau_{y,I} + \tau_{y,e}} - \frac{1}{\tau_{y,I}} \right) + \frac{1}{4} \frac{\tau_{y,g}^2 n}{\tau_{y,I} + \tau_{y,e}} + \frac{1}{2} \frac{\tau_{y,f}\tau_{y,g}n}{\tau_{y,I} + \tau_{y,e}} \right] \|g'(x^{k-1})\|^2 \\ &\quad - \frac{1}{4} \frac{\tau_{y,f}^2}{\tau_{y,I}} \|f'(x^{k-1}) - f'(x^*)\|^2 - \frac{1}{4} \frac{\tau_{y,x}^2 n}{\tau_{y,I} + \tau_{y,e}} \|x^{k-1} - x^*\|^2. \end{aligned}$$

We now use the strong convexity of the function to get the following inequalities:

$$\begin{aligned} \|f'(x^{k-1}) - f'(x^*)\|^2 &\leq Ln(x^{k-1} - x^*)^\top g'(x^{k-1}) \\ \|x^{k-1} - x^*\|^2 &\leq \frac{1}{\mu}(x^{k-1} - x^*)^\top g'(x^{k-1}). \end{aligned}$$

Finally, we have

$$\begin{aligned} &\mathbb{E}[Q(\theta^k)|\mathcal{F}_{k-1}] - (1 - \delta)Q(\theta^{k-1}) \\ &\leq \left[ \tau_{x,g} - \frac{n\tau_{y,x}}{2(\tau_{y,I} + \tau_{y,e})}(\tau_{y,f} + \tau_{y,g}) - \frac{Ln}{4} \frac{\tau_{y,f}^2}{\tau_{y,I}} - \frac{1}{4\mu} \frac{\tau_{y,x}^2 n}{\tau_{y,I} + \tau_{y,e}} \right] (x^{k-1} - x^*)^\top g'(x^{k-1}) \\ &\quad - \left[ \frac{1}{4}\tau_{y,f}^2 n \left( \frac{1}{\tau_{y,I} + \tau_{y,e}} - \frac{1}{\tau_{y,I}} \right) + \frac{1}{4} \frac{\tau_{y,g}^2 n}{\tau_{y,I} + \tau_{y,e}} + \frac{1}{2} \frac{\tau_{y,f}\tau_{y,g}n}{\tau_{y,I} + \tau_{y,e}} \right] \|g'(x^{k-1})\|^2. \end{aligned}$$

If we choose  $\delta = \frac{\tilde{\delta}}{n}$  with  $\tilde{\delta} \leq \frac{1}{2}$ ,  $\nu = \frac{1}{2n}$ ,  $\eta = 2$  and  $\alpha = \frac{1}{2n\mu}$ , we get

$$\begin{aligned} \tau_{y,I} &= \frac{1}{2n^2\mu} \left( \frac{2\tilde{\delta}}{n} - \frac{1}{n} \right) = -\frac{1-2\tilde{\delta}}{2n^3\mu} \leq 0 \\ \tau_{y,e} &= \frac{1}{2n\mu} \left( \frac{\tilde{\delta}}{n} - \frac{2}{n} + \frac{1}{n} \left( 1 - \frac{\tilde{\delta}}{n} \right) \right) = -\frac{1}{2n^2\mu} \left( 1 - \tilde{\delta} + \frac{\tilde{\delta}}{n} \right) \leq 0 \\ \tau_{x,g} &= -\left( \frac{1}{n} - \frac{2\tilde{\delta}}{n} - \frac{3L}{2n^2\mu} \right) = \frac{3L}{2n^2\mu} - \frac{1-2\tilde{\delta}}{n} \\ \tau_{y,f} &= -\frac{1}{n^3\mu} \\ \tau_{y,x} &= \frac{1-\tilde{\delta}}{n^2} \\ \tau_{y,g} &= \frac{\tilde{\delta}}{n^3\mu}. \end{aligned}$$

Thus,

$$\begin{aligned}
\tau_{x,g} & - \frac{n\tau_{y,x}}{2(\tau_{y,I} + \tau_{y,e})}(\tau_{y,f} + \tau_{y,g}) - \frac{Ln}{4} \frac{\tau_{y,f}^2}{\tau_{y,I}} - \frac{1}{4\mu} \frac{\tau_{y,x}^2 n}{\tau_{y,I} + \tau_{y,e}} \\
& \leq \frac{3L}{2n^2\mu} - \frac{1-2\tilde{\delta}}{n} - \frac{\frac{1-\tilde{\delta}}{2n} \frac{2\tilde{\delta}-1}{n^3\mu}}{\tau_{y,I} + \tau_{y,e}} + \frac{Ln}{4} \frac{1}{n^6\mu^2} - \frac{1}{4\mu} \frac{(1-\tilde{\delta})^2}{n^3} \\
& = \frac{L}{n^2\mu} \left[ \frac{3}{2} + \frac{1}{2(1-2\tilde{\delta})} \right] - \frac{1-2\tilde{\delta}}{n} - \frac{1}{\mu n^3(\tau_{y,I} + \tau_{y,e})} \left[ \frac{(1-\tilde{\delta})^2}{4} + \frac{(1-\tilde{\delta})(2\tilde{\delta}-1)}{2n} \right] \\
& \leq \frac{L}{n^2\mu} \frac{2-3\tilde{\delta}}{1-2\tilde{\delta}} - \frac{1-2\tilde{\delta}}{n} + \frac{1}{\mu n^3 \left( \frac{1-2\tilde{\delta}}{2n^3\mu} + \frac{1}{2n^2\mu} \left( 1 - \tilde{\delta} + \frac{\tilde{\delta}}{n} \right) \right)} \frac{(1-\tilde{\delta})^2}{4} \\
& = \frac{L}{n^2\mu} \frac{2-3\tilde{\delta}}{1-2\tilde{\delta}} - \frac{1-2\tilde{\delta}}{n} + \frac{(1-\tilde{\delta})^2}{2-4\tilde{\delta}+2n-2n\tilde{\delta}+2\tilde{\delta}} \\
& = \frac{L}{n^2\mu} \frac{2-3\tilde{\delta}}{1-2\tilde{\delta}} - \frac{1-2\tilde{\delta}}{n} + \frac{1-\tilde{\delta}}{2(1+n)} \\
& \leq \frac{L}{n^2\mu} \frac{1-3\tilde{\delta}}{1-2\tilde{\delta}} - \frac{1-2\tilde{\delta}}{n} + \frac{1-\tilde{\delta}}{2n} \\
& = \frac{L}{n^2\mu} \frac{2-3\tilde{\delta}}{1-2\tilde{\delta}} - \frac{1-3\tilde{\delta}}{2n}.
\end{aligned}$$

This quantity is negative for  $\tilde{\delta} \leq \frac{1}{3}$  and  $\frac{\mu}{L} \geq \frac{4-6\tilde{\delta}}{n(1-2\tilde{\delta})(1-3\tilde{\delta})}$ . If we choose  $\tilde{\delta} = \frac{1}{8}$ , then it is sufficient to have  $\frac{n\mu}{L} \geq 8$ .

To finish the proof, we need to prove the positivity of the factor of  $\|g'(x^{k-1})\|^2$ .

$$\begin{aligned}
\frac{1}{4} \tau_{y,f}^2 n \left( \frac{1}{\tau_{y,I} + \tau_{y,e}} - \frac{1}{\tau_{y,I}} \right) + \frac{1}{4} \frac{\tau_{y,g}^2 n}{\tau_{y,I} + \tau_{y,e}} + \frac{1}{2} \frac{\tau_{y,f} \tau_{y,g} n}{\tau_{y,I} + \tau_{y,e}} & = \frac{n}{4} \frac{1}{\tau_{y,I} + \tau_{y,e}} (\tau_{y,f} + \tau_{y,g})^2 - \frac{n}{4} \frac{\tau_{y,f}^2}{\tau_{y,I}} \\
& \geq \frac{n}{4} \frac{(\tau_{y,f} + \tau_{y,g})^2}{\tau_{y,I}} - \frac{n}{4} \frac{\tau_{y,f}^2}{\tau_{y,I}} \\
& = \frac{n}{4\tau_{y,I}} \tau_{y,g} (2\tau_{y,f} + \tau_{y,g}) \\
& \geq 0.
\end{aligned}$$

Then, following the same argument as in the previous section, we have

$$\begin{aligned}
\mathbb{E}Q(\theta^k) & \leq \left( 1 - \frac{1}{8n} \right)^k Q(\theta^0) \\
& = \left( 1 - \frac{1}{8n} \right)^k \left[ 2(g(x^0) - g(x^*)) + \frac{\sigma^2}{n\mu} \right],
\end{aligned}$$

with  $\sigma^2 = \frac{1}{n} \sum_i \|f'_i(x^*)\|^2$  the variance of the gradients at the optimum.

## Step 2 - Domination of $g(x^k) - g(x^*)$ by $Q(\theta^k)$

We now need to prove that  $Q(\theta^k)$  dominates  $g(x^k) - g(x^*)$ .

$$\begin{aligned}
Q(\theta^k) &= 2g\left(x^k + \frac{\alpha}{n}e^\top y^k\right) - 2g(x^*) + (\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*) \\
&= 2g\left(x^k + \frac{\alpha}{n}e^\top y^k\right) - 2g(x^*) + \frac{1}{n^2\mu} \sum_i \|y_i^k - f'_i(x^*)\|^2 + \frac{n-1}{2n^3\mu} \|e^\top y\|^2 - \frac{1}{n}(x^k - x^*)^\top (e^\top y^k) \\
&\geq 2g(x^k) + \frac{2\alpha}{n}g'(x^k)^\top (e^\top y^k) - 2g(x^*) \\
&\quad + \frac{1}{n^2\mu} \sum_i \left\| \frac{1}{n}e^\top y^k + y_i^k - \frac{1}{n}e^\top y^k - f'_i(x^*) \right\|^2 + \frac{n-1}{2n^3\mu} \|e^\top y\|^2 - \frac{1}{n}(x^k - x^*)^\top (e^\top y^k) \\
&\text{using the convexity of } g \text{ and the fact that } \sum_i f'_i(x^*) = 0 \\
&= 2g(x^k) - 2g(x^*) + \left( \frac{2\alpha}{n}g'(x^k) - \frac{1}{n}(x^k - x^*) \right)^\top (e^\top y^k) \\
&\quad + \frac{1}{n^3\mu} \|e^\top y^k\|^2 + \frac{1}{n^2\mu} \sum_i \left\| y_i^k - \frac{1}{n}e^\top y^k - f'_i(x^*) \right\|^2 + \frac{n-1}{2n^3\mu} \|e^\top y\|^2 \\
&\geq 2g(x^k) - 2g(x^*) + \left( \frac{2\alpha}{n}g'(x^k) - \frac{1}{n}(x^k - x^*) \right)^\top (e^\top y^k) + \frac{n+1}{2n^3\mu} \|e^\top y\|^2 \\
&\text{by dropping some terms.}
\end{aligned}$$

The quantity on the right-hand side is minimized for  $e^\top y = \frac{n^3\mu}{n+1} \left( \frac{1}{n}(x^k - x^*) - \frac{2\alpha}{n}g'(x^k) \right)$ . Hence, we have

$$\begin{aligned}
Q(\theta^k) &\geq 2g(x^k) - 2g(x^*) - \frac{n^3\mu}{2(n+1)} \left\| \frac{1}{n}(x^k - x^*) - \frac{2\alpha}{n}g'(x^k) \right\|^2 \\
&= 2g(x^k) - 2g(x^*) - \frac{n^3\mu}{2(n+1)} \left( \frac{1}{n^2} \|x^k - x^*\|^2 + \frac{4\alpha^2}{n^2} \|g'(x^k)\|^2 - \frac{4\alpha}{n^2} (x^k - x^*)^\top g'(x^k) \right) \\
&\geq 2g(x^k) - 2g(x^*) - \frac{n^3\mu}{2(n+1)} \left( \frac{1}{n^2} \|x^k - x^*\|^2 + \frac{4\alpha^2}{n^2} \|g'(x^k)\|^2 \right) \\
&\text{using the convexity of } g \\
&\geq 2g(x^k) - 2g(x^*) - \frac{n\mu}{2(n+1)} \left( 1 + \frac{L^2}{\mu^2 n^2} \right) \|x^k - x^*\|^2 \\
&\text{using the Lipschitz continuity of } g' \\
&\geq 2g(x^k) - 2g(x^*) - \frac{n\mu}{2(n+1)} \frac{65}{64} \|x^k - x^*\|^2 \text{ since } \frac{\mu}{L} \geq \frac{8}{n} \\
&\geq 2g(x^k) - 2g(x^*) - \frac{n}{(n+1)} \frac{65}{64} (g(x^k) - g(x^*)) \\
&\geq \frac{63}{64} (g(x^k) - g(x^*)) \\
&\geq \frac{6}{7} (g(x^k) - g(x^*)).
\end{aligned}$$

We thus get

$$\begin{aligned}\mathbb{E} [g(x^k) - g(x^*)] &\leq 2\mathbb{E}Q(\theta^k) \\ &= \left(1 - \frac{1}{8n}\right)^k \left[ \frac{7}{3}(g(x^0) - g(x^*)) + \frac{7\sigma^2}{6n\mu} \right].\end{aligned}$$

### Step 3 - Initialization of $x^0$ using stochastic gradient descent

During the first few iterations, we obtain the  $O(1/k)$  rate obtained using stochastic gradient descent, but with a constant which is proportional to  $n$ . To circumvent this problem, we will first do  $n$  iterations of stochastic gradient descent to initialize  $x^0$ , which will be renamed  $x^n$  to truly reflect the number of iterations done.

Using the bound from section A.3, we have

$$\mathbb{E}g\left(\frac{1}{n}\sum_{i=0}^{n-1}\tilde{x}^i\right) - g(x^*) \leq \frac{2L}{n}\|x^0 - x^*\|^2 + \frac{4\sigma^2}{n\mu}\log\left(1 + \frac{\mu n}{4L}\right).$$

And so, using  $x^n = \frac{1}{n}\sum_{i=0}^{n-1}\tilde{x}^i$ , we have for  $k \geq n$

$$\mathbb{E} [g(x^k) - g(x^*)] \leq \left(1 - \frac{1}{8n}\right)^{k-n} \left[ \frac{14L}{3n}\|x^0 - x^*\|^2 + \frac{28\sigma^2}{3n\mu}\log\left(1 + \frac{\mu n}{4L}\right) + \frac{7\sigma^2}{6n\mu} \right].$$

Since

$$\left(1 - \frac{1}{8n}\right)^{-n} \leq \frac{8}{7},$$

we get

$$\mathbb{E} [g(x^k) - g(x^*)] \leq \left(1 - \frac{1}{8n}\right)^k \left[ \frac{16L}{3n}\|x^0 - x^*\|^2 + \frac{32\sigma^2}{3n\mu}\log\left(1 + \frac{\mu n}{4L}\right) + \frac{4\sigma^2}{3n\mu} \right].$$

## References

- A. Agarwal and J.C. Duchi. Distributed delayed stochastic optimization. *Neural Information Processing Systems*, 2011.
- A. Agarwal, P.L. Bartlett, P. Ravikumar, and M.J. Wainwright. Information-theoretic lower bounds on the oracle complexity of convex optimization. *arXiv:1009.0571*, 2010.
- F. Bach and E. Moulines. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Neural Information Processing Systems*, 2011.
- D.P. Bertsekas. A new class of incremental gradient methods for least squares problems. *SIAM Journal on Optimization*, 7(4):913–926, 1997.
- D. Blatt, A.O. Hero, and H. Gauchman. A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, 18(1):29–51, 2008.
- L. Bottou and Y. LeCun. Large scale online learning. *Neural Information Processing Systems*, 2003.

- R. Caruana, T. Joachims, and L. Backstrom. KDD-cup 2004: results and analysis. *ACM SIGKDD Newsletter*, 6(2):95–108, 2004.
- B. Delyon and A. Juditsky. Accelerated stochastic approximation. *SIAM Journal on Optimization*, 3:868–881, 1993.
- M. Eberts and I. Steinwart. Optimal learning rates for least squares SVMs using Gaussian kernels. *Neural Information Processing Systems*, 2011.
- A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL <http://archive.ics.uci.edu/ml>.
- M.P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *arXiv:1104.2373*, 2011.
- S. Ghadimi and G. Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization. *Optimization Online*, 2010.
- I. Guyon. Sido: A pharmacology dataset, 2008. URL <http://www.causality.inf.ethz.ch/data/SID0.html>.
- H. Kesten. Accelerated stochastic approximation. *The Annals of Mathematical Statistics*, 29(1):41–59, 1958.
- H.J. Kushner and G. Yin. *Stochastic approximation algorithms and applications*. Springer-Verlag, second edition, 2003.
- D.D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.
- P. Liang, F. Bach, and M. Jordan. Asymptotically optimal regularization in smooth parametric models. *Neural Information Processing Systems*, 2009.
- J. Martens. Deep learning via hessian-free optimization. *International Conference on Machine Learning*, 2010.
- A. Nedic and D. Bertsekas. Convergence rate of incremental subgradient algorithms. *Stochastic Optimization: Algorithms and Applications*, pages 263–304, 2000.
- A. Nemirovski and D.B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, 1983.
- A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.
- Y. Nesterov. A method for solving a convex programming problem with rate of convergence  $O(1/k^2)$ . *Soviet Math. Doklady*, 269(3):543–547, 1983.
- Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, 2004.
- Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009.
- B.T. Polyak and A.B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.
- H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

- S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, 35(3):1012–1030, 2007.
- M.V. Solodov. Incremental gradient algorithms with stepsizes bounded away from zero. *Computational Optimization and Applications*, 11(1):23–35, 1998.
- K. Sridharan, S. Shalev-Shwartz, and N. Srebro. Fast rates for regularized objectives. *Neural Information Processing Systems*, 2008.
- P. Sunehag, J. Trunpf, SVN Vishwanathan, and N. Schraudolph. Variable metric stochastic approximation theory. *International Conference on Artificial Intelligence and Statistics*, 2009.
- C.H. Teo, Q. Le, A.J. Smola, and SVN Vishwanathan. A scalable modular convex solver for regularized risk minimization. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2007.
- P. Tseng. An incremental gradient(-projection) method with momentum term and adaptive stepsize rule. *SIAM Journal on Optimization*, 8(2):506–531, 1998.
- L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.
- H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.