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
Département d'Informatique de l'École Normale Supérieure
Paris, France

July 6, 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. In a machine learning context, numerical experiments indicate that the new algorithm can dramatically outperform standard algorithms, both in terms of optimizing the training objective and reducing the testing objective quickly.

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, i.e.,

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

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, in the case of ℓ_2-regularized logistic regression, we have $f_i(x) := \frac{\lambda}{2}\|x\|^2 + \log(1 + \exp(-b_i a_i^T x))$, where $a_i \in \mathbb{R}^p$ and $b_i \in \{-1, 1\}$ are the training examples associated with a bi-

nary classification problem and λ is a regularization parameter. More generally, any ℓ_2-regularized empirical risk minimization problem of the form

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

falls in the framework of (1) if 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.

For optimizing (1), the standard *full gradient* (FG) method, which dates back to Cauchy (1847), uses iterations of the form

$$x^{k+1} = x^k - \alpha_k g'(x^k) = x^k - \frac{\alpha_k}{n}\sum_{i=1}^{n} f_i'(x^k). \tag{3}$$

Using x^* to denote the unique minimizer of g, the FG method 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 (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 fast convergence rate of the FG method, it can be unappealing when n is large because its iteration cost scales linearly in n. SG methods, on the other hand, have an iteration cost which is *independent* of n, making them suited for that setting. The basic SG method for optimizing (1) uses iterations of the form

$$x^{k+1} = x^k - \alpha_k f_{i_k}'(x^k), \tag{4}$$

where α_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 estimate of the true training 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

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

where the expectation is taken with respect to the selection of the i_k variables. Under certain assumptions this convergence rate is *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); Nemirovski et al. (2009); Agarwal et al. (2010)). Thus, we cannot hope to obtain a better convergence rate if the algorithm only relies on unbiased gradient measurements. Nevertheless, by using the stronger assumption that the functions are sampled from a finite dataset, in this paper we show that we can achieve an exponential convergence rate while preserving the iteration complexity of SG methods.

The primary contribution of this work is the analysis of 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. SAG uses iterations of the form

$$x^{k+1} = x^k - \frac{\alpha_k}{n} \sum_{i=1}^{n} y_i^k, \tag{5}$$

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, we show in this paper that *the SAG iterations have a linear 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 a machine learning context where $g(x)$ is a *training objective* associated with a predictor parameterized by x, we are often ultimately interested in the *testing objective*, i.e., the expected loss on unseen data points. Note that a linear convergence rate for the training objective does not translate into a similar rate for the testing objective, and an appealing propertly of SG methods is that they achieve the optimal $O(1/k)$ rate for the *testing objective* as long as every datapoint is seen *only once*. However, as is common in machine learning, we assume that we are only given a finite training data set and thus that datapoints are revisited multiple times. In this context, the analysis of SG methods only applies to the training objective and, although our analysis also focuses on the training objective, in our experiments the SAG method typically reached the optimal testing objective faster than both FG and SG 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, despite 60 years of extensive research on SG methods, with a significant portion of the applications focusing on finite datasets, we are not aware of any other SG method that achieves a linear 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. Section 5 presents a numerical comparison of an implementation based on SAG to SG and FG methods, indicating that the method may be very useful for problems where we can only afford to do a few passes through a data set.

2 Related Work

There is 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 β, and in this case we can rewrite 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 (5) in a similar form as

$$x^{k+1} = x^k - \sum_{j=1}^{k} \alpha_k S(j, i_{1:k}) f'_{i_j}(x^j), \tag{6}$$

where the selection function $S(j, i_{1:k})$ is equal to $1/n$ if j corresponds to the last 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,

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

which is similar to the SAG iteration in the form (5) but where *all* previous gradients are used. This approach is used in the dual averaging method (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 perform 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 (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; Xiao, 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; Xiao, 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) showed 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 and 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 (without additional assumptions).

Accelerated methods: Accelerated SG methods, which despite their name are not related to the aforementioned AFG method, take advantage of the fast convergence 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 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 (Bertsekas, 1997). However, the weighting is numerically unstable and the linear convergence rate presented treats full cycles as iterations. A related strategy is to group the f_i functions into 'batches' of increasing size and performing SG iterations on the batches (Friedlander and Schmidt, 2011). In both cases, the iterations that achieve the linear rate have a cost that is not independent of n, as opposed to SAG.

Incremental Aggregated Gradient: Finally, Blatt et al. (2008) presented the most closely-related algorithm, the IAG method. This method is identical to the SAG iteration (5), 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. Using a non-trivial extension of their analysis and a novel proof technique involving bounding the gradient and iterates simultaneously by a Lyapunov potential function, in this work *we give an explicit linear convergence rate for general strongly-convex functions using the SAG 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. We also emphasize that in our experiments IAG is not faster than regular full gradient descent, while SAG is, showing that the simple change (random selection vs. cycling) can dramatically improve optimization 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

$$\|f_i'(x) - f_i'(y)\| \leq L\|x - y\|. \tag{7}$$

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 $g = \frac{1}{n}\sum_{i=1}^{n} f_i$ 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 ℓ_2-regularization term as in Eq. (2) 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 convergence results assume that we initialize y_i^0 to a zero vector for all i. We denote the variance of the gradients at the optimum x^* by $\sigma^2 = \frac{1}{n}\sum_i \|f_i'(x^*)\|^2$. Finally, all our convergence results consider expectations with respect to the internal randomization of the algorithm, and not with respect to the data (which are assumed to be deterministic and fixed).

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 constant step size of $\alpha_k = \frac{1}{2nL}$, the SAG iterations satisfy for $k \geq 1$:*

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

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

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

albeit with a constant which is proportional to n. Despite this constant, they are advantageous over SG methods in later iterations because they obtain an exponential convergence rate as in FG methods. We also note that an exponential convergence rate is obtained for any constant step size smaller than $\frac{1}{2nL}$.

Empirically, with a step size of $\alpha_k = 1/2nL$, the SAG iterations behave in a similar way to the IAG iterations with the same step size or the 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/μ (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 μ or L.

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

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

The proof is given in the Appendix. 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 $k \geqslant n$. This also leads to an $O((\log n)/k)$ rate while if we used the SAG iterations from the beginning we can obtain the same provable convergence rate but the constant is again proportional to n. Note that this bound is obtained when initializing all y_i to zero after the stochastic gradient phase.[1] However, in our experiments we do not use this initialization but rather use a minor variant of SAG (discussed in the next section), which appears more difficult to analyze but which gives better empirical performance. Further, though n appears in the convergence rate, if we perform n iterations of SAG (i.e., one effective pass through the data), the error is multiplied by $(1 - 1/8n)^n \leq \exp(-1/8)$, which is independent of n. Thus, each pass through the data reduces the excess objective by a constant multiplicative factor that is independent of the problem, as long as $n \geqslant 8L/\mu$.

Since Proposition 2 is true for all the values of μ satisfying $\frac{\mu}{L} \geqslant \frac{8}{n}$, 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. 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

[1] 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.

of $\frac{1}{2\mu}$ may diverge, even under these assumptions. Thus, for certain problems the SAG iterations can tolerate 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 values, the rates obtained on iterates and function values are equivalent because, by the Lipschitz and strong-convexity assumptions, we have $\frac{\mu}{2}\|x^k - x^*\|^2 \leqslant g(x^k) - g(x^*) \leqslant \frac{L}{2}\|x^k - x^*\|^2$.

4 Implementation Details

In order to simplify the analysis, above we considered a basic canonical variant of the algorithm. In this section, we describe modifications that prove useful in practice.

First, while Proposition 2 assumes that the first n iterations use an SG method and that we then set all y_i to 0, in our experiments we found that it was more effective to simply run the SAG algorithm from the very first iteration, but replacing the factor n in the objective function by m, the number of unique i_k values we have sampled so far (which converges to n). Although this modification appears more difficult to analyze, in our experiments this modified SAG algorithm performed better than using the SG method for the first iteration, and it has the advantage that we only need to estimate a single constant step size. Another important modification for ℓ_2-regularized objectives is that we can use the exact gradient of the regularizer, rather than building an approximation of it. With these two changes, the modified SAG iterations take the simple form

$$\begin{aligned} d &\leftarrow d - y_i \\ y_i &\leftarrow l_i'(x^k) \\ d &\leftarrow d + y_i \\ x &\leftarrow \big(1 - \alpha\lambda\big)x - \frac{\alpha}{m}d\,. \end{aligned}$$

For data sets with sparse loss gradients $l_i'(x^k)$ resulting from a sparse dependency on x^k, the sparsity pattern will lead to a reduced storage cost of the y_i variables but the iteration appears unappealing because the update of x is always a dense vector operation. However, by taking advantage of the simple form of sequences of updates for elements of d that do not change and by tracking the last time a variable changed, we can apply 'just in time' updates to compute all needed elements of x_k, which means that the iteration cost is proportional to the sparsity level of $l_i'(x_k)$.

In our experiments, we found that the SAG algorithm can allow a much larger step size than indicated in our analysis. Indeed, from extensive experimentation we conjecture that, when μ/L is small, a step size of $\alpha = 1/L$ gives a convergence rate of $(1 - \mu/L)$, the same as the FG method but with iterations that are n times cheaper, though we have been unable to prove this result. Further, we found that a step size of $2/(L+n\mu)$, which corresponds to the best fixed step size for the FG method in the special case of $n = 1$ (Nesterov, 2004, Theorem 2.1.15), yields even better performance. In cases where L is not known, we also experimented with a basic line-search, where we start with an initial estimate of L_0, and double this estimate whenever the instantiated Lipschitz inequality

$$[f_{i_k}(x^{k+1}) - f_{i_k}(x^k)] \leq f_{i_k}'(x^k)'(x^{k+1} - x^k) + \frac{L_k}{2}\|x^{k+1} - x^k\|^2,$$

is not satisfied but the quantities in the inequality are above numerical precision.

5 Experimental Results

In our experiments, we compared an extensive variety of competitive FG and SG methods, as well as the IAG method. First, we compared the train and test performance of algorithms which do not rely on dataset-dependent tuning parameters. Then, we focused on the optimization performance of a wider range of methods where the optimal parameters were chosen in hindsight.

5.1 Parameter-free methods

In this set of experiments, we compared a variety of methods which do not have dataset-dependent tuning parameters (with the exception of the AFG method):

- **Steepest**: The full gradient method described by iteration (3), with a line-search that uses cubic Hermite polynomial interpolation to find a step size satisfying the strong Wolfe conditions, and where the parameters of the line-search were tuned for the problems at hand.
- **AFG**: Nesterov's accelerated full gradient method (Nesterov, 1983), where iterations of (3) with a fixed step size are interleaved with an extrapolation step. We report the performance of the method using the best step-size among all powers of 2.
- **L-BFGS**: A publicly-available limited-memory quasi-Newton method that has been tuned for log-linear models.[2] This method is by far the most complicated method we considered.
- **Pegasos**: The state-of-the-art SG method described by iteration (4) with step size of $\alpha_k = 1/\mu k$ and a projection step onto a norm-ball known to contain the optimal solution (Shalev-Shwartz et al., 2007).
- **RDA**: The regularized dual averaging method of Xiao (2010), another recent state-of-the-art SG method.
- **SAG**: The proposed stochastic average gradient method described by iteration (5) using the modifications discussed in the previous section. We tested a constant step-size of $\alpha_k = 2/(L + n\mu)$, and with a step-size of $\alpha_k = 1/L_k$ using the line-search described in the previous section to estimate L (we set $L_0 = 1$).

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 hundred), 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 cannot afford to do enough passes to warrant using FG algorithms like AFG or L-BFGS. 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. The *quantum* ($p = 50000$, $p = 78$) and *protein* ($n = 145751$, $p = 74$) data set was obtained from the KDD Cup 2004 website,[3] the *sido* data sets were obtained from the Causality Workbench website,[4] while the *rcv1* ($n = 20242$, $p = 47236$) and *covertype* ($n = 581012$, $p = 54$) data sets were obtained from the LIBSVM Data website.[5] Although our method can be applied to all strongly-

[2] `http://www.di.ens.fr/~mschmidt/Software/minFunc.html`
[3] `http://osmot.cs.cornell.edu/kddcup`
[4] `http://www.causality.inf.ethz.ch/home.php`
[5] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`

convex functions, on these data sets we focus on an ℓ_2-regularized logistic regression problem, with $\lambda = 1/n$. We split each dataset in two, training on one half and testing on the other half. We added a (regularized) bias term to all data sets, and for dense features we standardized so that they would have a mean of zero and a variance of one. In all the experiments, we measure the training and testing objectives as a function of the number of effective passes through the data. These results are thus independent of the practical implementation of the algorithms. We plot the training and testing objectives of the different methods for 30 effective passes through the data in Figure 1.

We can observe several trends across the experiments:

- **FG vs. SG**: Although the performance of SG methods can be catastrophic if the step size is not chosen carefully (e.g., the *covertype* data), with a carefully-chosen step size the SG methods do substantially better than FG methods on the first few passes through the data (e.g., the *rcv1* data). In contrast, FG methods are not sensitive to the step size and because of their steady progress we also see that FG methods slowly catch up to the SG methods and eventually (or will eventually) pass them (e.g., the *protein* data).
- **(FG and SG) vs. SAG**: The SAG iterations seem to achieve the best of both worlds. They start out substantially better than FG methods, but continue to make steady (linear) progress which leads to better performance than SG methods. In some cases (*protein* and *covertype*), the significant speed-up observed for SAG in reaching low training objectives also translates to reaching the optimal testing objective more quickly than the other methods.

5.2 Searching for best step-sizes

In this series of experiments, we sought to test whether SG methods with a carefully chosen step size would be competitive with the SAG iterations. In particular, we compared the following variety of basic FG and SG methods.

1. **FG**: The full gradient method described by iteration (3).
2. **AFG**: The accelerated full gradient method of Nesterov, where iterations of (3) are interleaved with an extrapolation step.
3. **peg**: The pegasos algorithm (Shalev-Shwartz et al., 2007), but where we multiply the step size by a constant.
4. **ASG**: The stochastic gradient method described by iteration (4), using a either a constant step size or using $\alpha_k = O(1/p^{2/3})$ where p is number of effective passes, and averaging the iterates (which we found gave better results with these large step sizes).
5. **IAG**: The incremental aggregated gradient method (Blatt et al., 2008) described by iteration (5) but with a cyclic choice of i_k. We used the modifications discussed in Section 4, which we found gave better performance.
6. **SAG**: The proposed stochastic average gradient method described by iteration (5) with the modifications discussed in Section 4.

For all of the above methods, we chose the step size that gave the best performance among powers of 10. In Figure 2, we compared these methods to each other using the selected step sizes, as well as the L-BFGS and SAG algorithms from the previous experiment. In this experiment we see that using a constant or nearly constant step size within SG methods and using averaging tends to perform much better than the basic SG method implemented by pegasos. This makes sense because SG methods

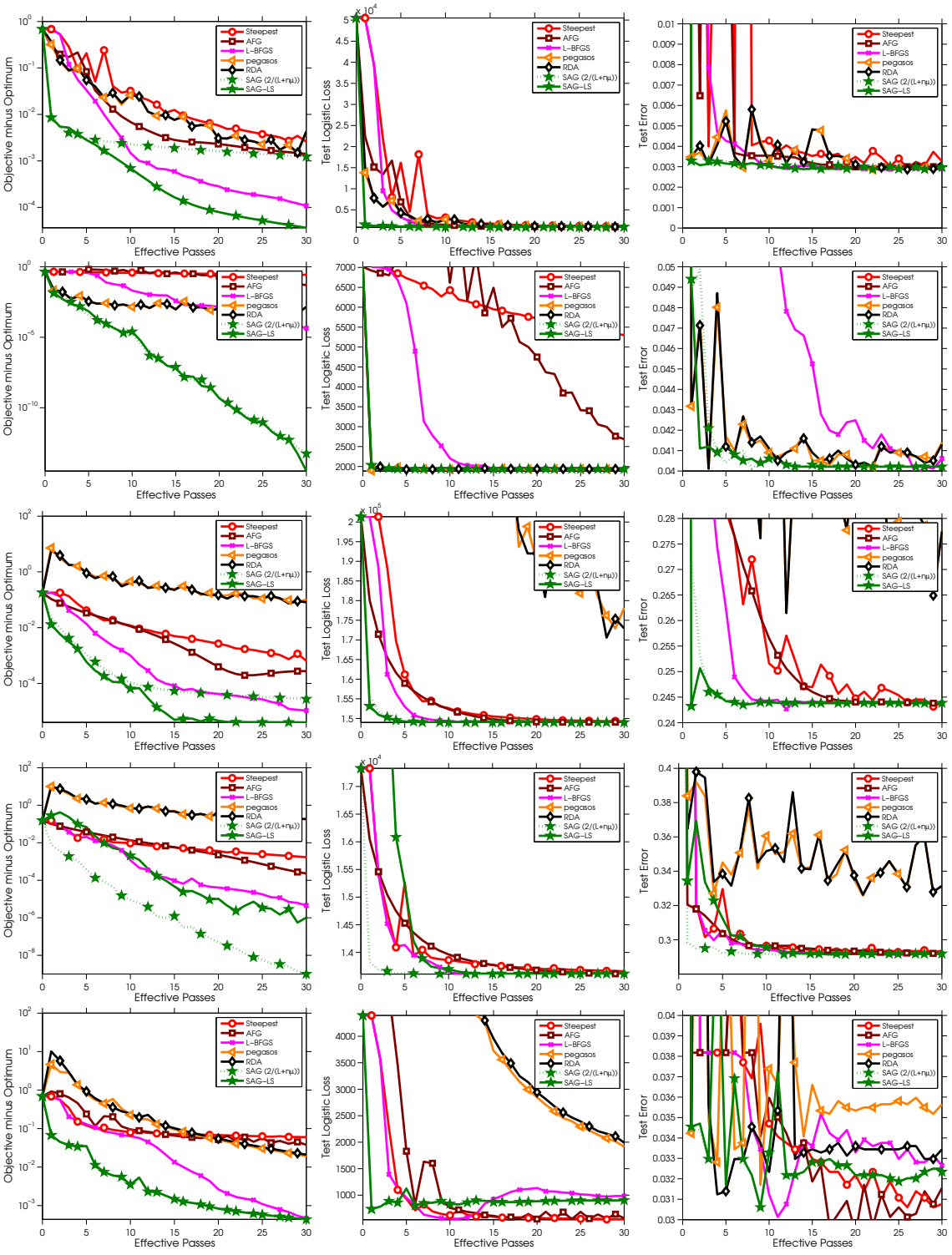

Figure 1: Comparison of optimization strategies for ℓ_2-regularized logistic regression. Left: training excess objective. Middle: testing objective. Right: test errors. From top to bottom are the results on the *protein*, *rcv1*, *covertype*, *quantum* and *sido* data sets. This figure is best viewed in colour.

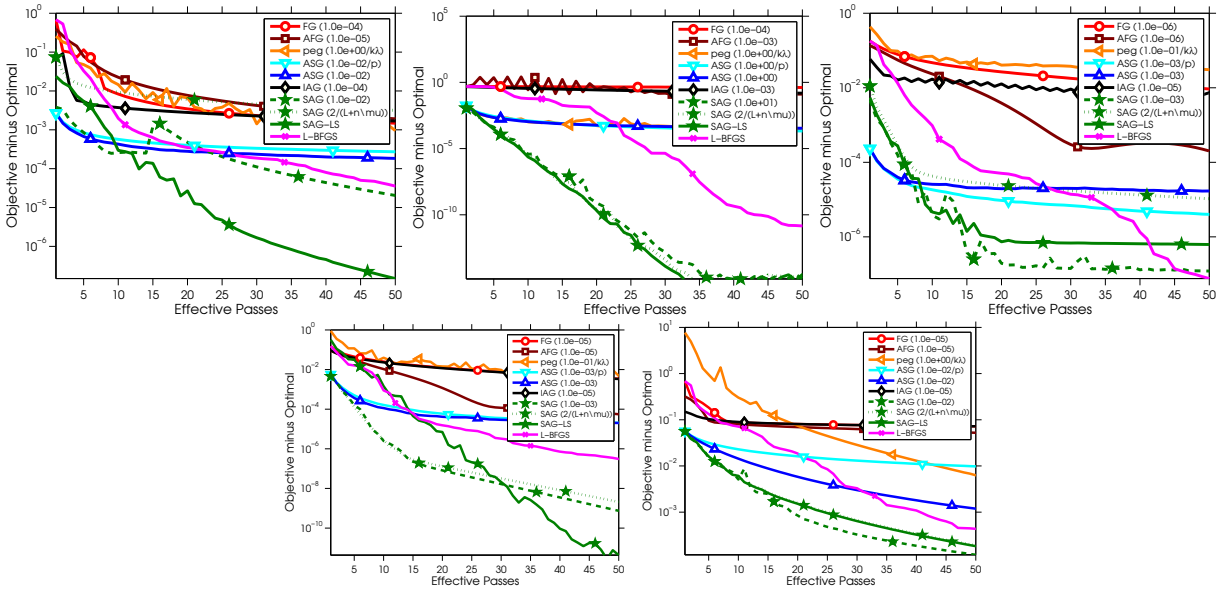

Figure 2: Comparison of optimization strategies that choose the best step-size in hindsight.

with a constant step size have a linear convergence rate when far from the solution.[6] However, the performance is still typically not comparable to that of the SAG iterations, which achieve a linear convergence rate even when close to the solution. Finally, we observe an unexpected trend across these experiments:

– **IAG vs. SAG**: Our experiments show that the IAG method does not improve over regular FG methods, but they also show the surprising result that the randomized SAG method outperforms the closely-related deterministic IAG method by a very large margin. This is due to the larger step sizes used by the SAG iterations, which would cause IAG to diverge.

6 Discussion

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 μ 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. Mini-batches also lessen the storage requirements of the algorithm, since we only need to store a vector y_i for each mini-batch rather than each training example.

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 in a linear model, L is equal to the largest value of the loss second derivative (1 for the

[6] We have also compared to a variety of other SG methods, including SG with momentum, SG with gradient averaging, regularized dual averaging, the accelerated SG method, and SG methods where averaging is delayed until after the first iteration. However, among all the SG methods we tried, the ASG methods above gave the best performance so we only plot these to keep the plots simple.

square loss, 1/4 for the logistic loss) times the uniform bound R on the norm of each data point. Thus, the constraint $\frac{\mu}{L} \geqslant \frac{8}{n}$ is satisfied when $\lambda \geqslant \frac{8R}{n}$. 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 λ 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.

Training error vs. testing objective: The theoretical contribution of this work is limited to the convergence rate of the training objective. Though there are several settings where this is the metric of interest (e.g., variational inference in graphical models), in many cases one will be interested in the convergence speed of the testing objective. Since the $O(1/k)$ convergence rate of the testing objective, achieved by SG with decreasing step sizes (assuming a single pass through the data), is provably optimal when the algorithm only accesses the function through unbiased measurements of the objective and its gradient, it is unlikely that one can obtain a linear convergence rate for the testing objective with the SAG iterations. However, as shown in our experiments, the testing objective often reaches its minimum quicker than existing SG methods, and we could expect to improve the constant in the $O(1/k)$ convergence rate, as is the case with online second-order methods (Bottou and Bousquet, 2008).

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. It is intriguing to consider whether better performance could be achieved by approximating second-order information about the function. Other interesting directions of future work include using non-uniform sampling (such as sampling proportional to the Lipschitz constant of each function), proximal-gradient variants of the method for constrained and non-smooth optimization, 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.

Step-size selection and termination criteria: The three major disadvantages of SG methods are: (i) the slow convergence rate, (ii) deciding when to terminate the algorithm, and (iii) choosing the step size while running the algorithm. This paper showed that the SAG iterations achieve a much faster convergence rate, but the SAG iterations may also be advantageous in terms of step sizes and termination criterion. In particular, the SAG iterations suggest a natural termination criterion; since the step size stays constant, we can use $\|(x^k - x^{k-1})/\alpha\|$ as an approximation of the optimality of x^k. Further, while SG methods require specifying a sequence of step sizes and mis-specifying this sequence can have a disastrous effect on the convergence rate (Nemirovski et al., 2009, §2.1), our theory shows that the SAG iterations iterations achieve a linear convergence rate for any sufficiently small constant step size and our experiments indicate that a simple line-search gives strong performance.

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, $i = 1, \ldots, n$ 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 $k \geqslant 1$, the stochastic average gradient algorithm performs the recursion

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

where an i_k is selected in $\{1, \ldots, n\}$ 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

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

with

$$e = \begin{pmatrix} I \\ \vdots \\ I \end{pmatrix} \in \mathbb{R}^{np \times p}, \qquad f'(x) = \begin{pmatrix} f_1'(x) \\ \vdots \\ f_n'(x) \end{pmatrix} \in \mathbb{R}^{np}, \qquad z^k = \begin{pmatrix} z_1^k I \\ \vdots \\ z_n^k I \end{pmatrix} \in \mathbb{R}^{np \times p}.$$

Using this definition of z^k, we have $\mathbb{E}[(z^k)(z^k)^\top] = \frac{1}{n} I - \frac{1}{n^2} e e^\top$. Note that, for a given k, the variables $z_1^k, \ldots, z_n^k$ are not independent.

We also use the notation

$$\theta^k = \begin{pmatrix} y_1^k \\ \vdots \\ y_n^k \\ x^k \end{pmatrix} \in \mathbb{R}^{(n+1)p}, \qquad \theta^* = \begin{pmatrix} f_1'(x^*) \\ \vdots \\ f_n'(x^*) \\ x^* \end{pmatrix} \in \mathbb{R}^{(n+1)p}.$$

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

- $\mathrm{diag}(M)$ is the $tp \times p$ matrix being the concatenation of the t $(p \times p)$-blocks on the diagonal of M;
- $\mathrm{Diag}(m)$ 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 σ-field of information up to (and including time k), i.e., $\mathcal{F}_k$ is the σ-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:

$$\tilde{x}^k = \tilde{x}^{k-1} - \gamma_k f_{i_k}'\left(\tilde{x}^{k-1}\right) .$$

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

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

Indeed, we have

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

By taking expectations, we get

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

Thus, if we take

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

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

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

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

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

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 θ^{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*

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

with

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

Note that for square $n \times n$ matrix, $\mathrm{diag}(M)$ denotes a vector of size n composed of the diagonal of M, while for a vector m of dimension n, $\mathrm{Diag}(m)$ is the $n \times n$ diagonal matrix with m on its diagonal. Thus $\mathrm{Diag}(\mathrm{diag}(M))$ is a diagonal matrix with the diagonal elements of M on its diagonal, and $\mathrm{diag}(\mathrm{Diag}(m)) = m$.

Proof Throughout the proof, we will use the equality $g'(x) = e^\top f'(x)/n$. Moreover, all conditional expectations of linear functions of z^k will be equal to zero.

We have

$$\begin{aligned}
&\mathbb{E}\left[(\theta^k - \theta^*)^\top \begin{pmatrix} A & b \\ b^\top & c \end{pmatrix} (\theta^k - \theta^*) \middle| \mathcal{F}_{k-1} \right] \\
&\quad = E\left[(y^k - f'(x^*))^\top A (y^k - f'(x^*)) + 2(y^k - f'(x^*))^\top b (x^k - x^*) + (x^k - x^*)^\top c (x^k - x^*) | \mathcal{F}_{k-1} \right] \ .
\end{aligned} \tag{8}$$

The first term (within the expectation) on the right-hand side of Eq. (8) is equal to

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

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

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

The second term (within the expectation) on the right-hand side of Eq. (8) is equal to

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

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

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

The last term on the right-hand side of Eq. (8) is equal to

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

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

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

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

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

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

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

Hence, the sum may be rewritten as

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

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

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

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:

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

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

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

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

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

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:

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

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

$$s^\top M s + s^\top t \leqslant -\frac{1}{4} t^\top M^{-1} t\,,$$

we have, with

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

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

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

The bound then becomes

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

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

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

This yields the final bound

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

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 $\left(2\alpha - 3\alpha^2 nL + \frac{\delta^2\left(1 - \frac{1}{n}\right)^2}{\left[3n\delta - 1 - 2\delta + \frac{\delta-1}{n}\right]}\frac{n}{\mu} - \frac{\delta}{\mu}\right)$ is positive.

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

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

Hence,

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

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

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

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

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

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 $Q(\theta^k) \geqslant \frac{1}{3}\|x^k - x^*\|^2$.

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

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

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

This yields

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

We have

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

Initializing all the y_i^0 to 0, we get

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

and

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

■

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

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

This yields

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

We have

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

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

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

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

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

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

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

Using the convexity of g gives

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

and, consequently,

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

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

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

Let us rewrite this as

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

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

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

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

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

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

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

Finally, we have

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

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

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

Thus,

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

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

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

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

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

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

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 $g(x^k)-g(x^*)$.

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

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

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

We thus get

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

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

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

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

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

Since

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

we get

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

References

A. Agarwal and J. C. Duchi. Distributed delayed stochastic optimization. *Adv. NIPS*, 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. *Adv. NIPS*, 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 O. Bousquet. The tradeoffs of large scale learning. In *Adv. NIPS*, 2008.

L. Bottou and Y. LeCun. Large scale online learning. *Adv. NIPS*, 2003.

M. A. Cauchy. Méthode générale pour la résolution des systèmes déquations simultanées. *Comptes rendus des séances de l'Académie des sciences*, 25(1):536–538, 1847.

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. *Adv. NIPS*, 2011.

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*, July, 2010.

H. Kesten. Accelerated stochastic approximation. *Ann. Math. Stat.*, 29(1):41–59, 1958.

H. J. Kushner and G. Yin. *Stochastic approximation algorithms and applications*. Springer-Verlag, second edition, 2003.

P. Liang, F. Bach, and M. I. Jordan. Asymptotically optimal regularization in smooth parametric models. *Adv. NIPS*, 2009.

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. *Ann. Math. Stat.*, 22(3):400–407, 1951.

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proc. ICML*, 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. *Adv. NIPS*, 2008.

P. Sunehag, J. Trumpf, 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 S. V. N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. *Proc. SIGKDD Conference*, 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.