Starting Small – Learning with Adaptive Sample Sizes

Hadi Daneshmand HADI.DANESHMAND@INF.ETHZ.CH
Aurelien Lucchi AURELIEN.LUCCHI@INF.ETHZ.CH
Thomas Hofmann THOMAS.HOFMANN@INF.ETHZ.CH
Department of Computer Science, ETH Zurich, Switzerland

Abstract

For many machine learning problems, data is abundant and it may be prohibitive to make multiple passes through the full training set. In this context, we investigate strategies for dynamically increasing the effective sample size, when using iterative methods such as stochastic gradient descent. Our interest is motivated by the rise of variance-reduced methods, which achieve linear convergence rates that scale favorably for smaller sample sizes. Exploiting this feature, we show – theoretically and empirically – how to obtain significant speed-ups with a novel algorithm that reaches statistical accuracy on an n-sample in $2n$, instead of $n \log n$ steps.

1. Introduction

In empirical risk minimization (ERM) (Vapnik, 1998) the training set $\mathcal{S}$ is used to define a sample risk $\mathcal{R}_{\mathcal{S}}$, which is then minimized with regard to a pre-defined function class. One effectively equates learning algorithms with optimization algorithms. However, for all practical purposes an approximate solution of $\mathcal{R}_{\mathcal{S}}$ will be sufficient, as long as the optimization error is small relative to the statistical accuracy at sample size $n := |\mathcal{S}|$. This is important for massive data sets, where optimization to numerical precision is infeasible. Instead of performing early stopping on black-box optimization, one ought to understand the trade-offs between statistical and computational accuracy, cf. (Chandrasekaran & Jordan, 2013). In this paper, we investigate a much neglected facet of this topic, namely how to dynamically control the effective sample size in optimization.

Many large-scale optimization algorithms are iterative: they use sampled or aggregated data to perform a sequence of update steps. This includes the popular family of gradient descent methods. Often, the computational complexity increases with the size of the training sample, e.g. in steepest-descent, where the cost of a gradient computation scales with n. Does one really need a highly accurate gradient though, in particular in the early phase of optimization? Why not use subsets $\mathcal{T}_t \subseteq \mathcal{S}$ which are increased in size with the iteration count t, matching-up statistical accuracy with optimization accuracy in a dynamic manner? This is the general program we pursue in this paper. In order to make this idea concrete and to reach competitive results, we focus on a recent variant of stochastic gradient descent (SGD), which is known as SAGA (Defazio et al., 2014). As we will show, this algorithm has a particularly interesting property in how its convergence rate depends on n.

1.1. Empirical Risk Minimization

Formally, we assume that training examples $\boldsymbol{x} \in \mathcal{S} \subseteq \mathcal{X}$ have been drawn i.i.d. from some underlying, but unknown probability distribution $\mathcal{P}$. We fix a function class $\mathcal{F}$ parametrized by weight vectors $\boldsymbol{w} \in \mathbb{R}^d$ and define the expected risk as $\mathcal{R}(\boldsymbol{w}) := \mathbf{E} f_{\boldsymbol{x}}(\boldsymbol{w})$, where f is an $\boldsymbol{x}$-indexed family of loss functions, often convex. We denote the minimum and the minimizer of $\mathcal{R}(\boldsymbol{w})$ over $\mathcal{F}$ by $\mathcal{R}^*$ and $\boldsymbol{w}^*$, respectively. Given that $\mathcal{P}$ is unknown, ERM suggests to rely on the empirical (or sample) risk with regard to $\mathcal{S}$

$$\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}) := \frac{1}{n} \sum_{\boldsymbol{x} \in \mathcal{S}} f_{\boldsymbol{x}}(\boldsymbol{w}), \quad \boldsymbol{w}^*_{\mathcal{S}} := \operatorname*{arg\,min}_{\boldsymbol{w} \in \mathcal{F}} \mathcal{R}_{\mathcal{S}}(\boldsymbol{w}) \,. \quad (1)$$

Note that one may absorb a regularizer in the definition of the loss $f_{\boldsymbol{x}}$.

1.2. Generalization bounds

The relation between $\boldsymbol{w}^*$ and $\boldsymbol{w}^*_{\mathcal{S}}$ has been widely studied in the literature on learning theory. It is usually analysed with the help of uniform convergence bounds that take the generic form (Boucheron et al., 2005)

$$\mathbf{E}_{\mathcal{S}} \left[\sup_{\boldsymbol{w} \in \mathcal{F}} |\mathcal{R}(\boldsymbol{w}) - \mathcal{R}_{\mathcal{S}}(\boldsymbol{w})| \right] \leq \mathcal{H}(n) \,, \quad (2)$$

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

where the expectation is over a random n-sample $\mathcal{S}$. Here $\mathcal{H}$ is a bound that depends on n, usually through a ratio n/d, where d is the capacity of $\mathcal{F}$ (e.g. VC dimension). This *fast* convergence rate has been shown to hold for a class of strictly convex loss functions such as quadratic, and logistic loss (Bartlett et al., 2006; 2005). In the realizable case, we may be able to observe a favorable $\mathcal{H}(n) \propto d/n$, whereas in the pessimistic case, we may only be able to establish weaker bounds such as $\mathcal{H}(n) \propto \sqrt{d/n}$ (e.g. for linear function classes); see also (Bousquet & Bottou, 2008). We ignore additional $\log$ factors that can be eliminated using the "chaining" technique (Bousquet, 2002; Bousquet & Bottou, 2008).

1.3. Statistical efficiency

Assume now that we have some approximate optimization algorithm, which given $\mathcal{S}$ produces solutions $\boldsymbol{w}_{\mathcal{S}}$ that are on average $\epsilon(n)$ optimal, i.e. $\mathbf{E}_{\mathcal{S}}\left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}_{\mathcal{S}}) - \mathcal{R}^*_{\mathcal{S}}\right] \leq \epsilon(n)$. One can then provide the following quality guarantee in expectation over sample sets $\mathcal{S}$ (Bousquet & Bottou, 2008)

$$\mathbf{E}_{\mathcal{S}}\mathcal{R}(\boldsymbol{w}_{\mathcal{S}}) - \mathcal{R}^* \leq \mathcal{H}(n) + \epsilon(n)\,, \tag{3}$$

which is an additive decomposition of the expected solution suboptimality into an estimation (or statistical) error $\mathcal{H}(n)$ and an optimization (or computational) error $\epsilon(n)$. For a given computational budget, one typically finds that $\epsilon(n)$ is increasing with n, whereas $\mathcal{H}(n)$ is always decreasing. This hints at a trade-off, which may suggest to chose a sample size $m < n$. Intuitively speaking, concentrating the computational budget on fewer data may be better than spreading computations too thinly.

1.4. Stochastic Gradient Optimization

For large scale problems, stochastic gradient descent is a method of choice in order to optimize problems of the form given in Eq. (1). Yet, while SGD update directions equal the true (negative) gradient direction in expectation, high variance typically leads to sub-linear convergence. This is where variance-reducing methods for ERM such as SAG (Roux et al., 2012), SVRG (Johnson & Zhang, 2013), and SAGA (Defazio et al., 2014) come into play. We focus on the latter here, where one can establish the following result on the convergence rate (see appendix).

Lemma 1. *Let all $f_{\boldsymbol{x}}$ be convex with L-Lipschitz continuous gradients and assume that $\mathcal{R}_{\mathcal{S}}$ is μ-strongly convex. Then the suboptimality of the SAGA iterate $\boldsymbol{w}^t$ after t steps is over a randomly sampled $\mathcal{S}$ bounded by*

$$\mathbf{E}_{\mathcal{A}}\left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}^t) - \mathcal{R}^*_{\mathcal{S}}\right] \leq \rho_n^t C_{\mathcal{S}}, \;\; \rho_n = 1 - \min\left(\frac{1}{n}, \frac{\mu}{L}\right),$$

where the expectation is over the algorithmic randomness.

This highlights two different regimes: For small n, the condition number $\kappa := \frac{L}{\mu}$ dictates how fast the optimization algorithm converges. On the other hand, for large n, the convergence rate of SAGA becomes $\rho_n = 1 - \frac{1}{n}$.

1.5. Contributions

Our main question is: can we obtain faster convergence to a statistically accurate solution by running SAGA on an initially smaller sample, whose size is then gradually increased? Motivated by a simple, yet succinct analysis, we present a novel algorithm, called DYNASAGA that implements this idea and achieves $\epsilon(n) \leq \mathcal{H}(n)$ after only $2n$ iterations.

2. Related Work

Stochastic approximation is a powerful tool for minimizing objective Eq. (1) for convex loss functions. The pioneering work of (Robbins & Monro, 1951) is essentially a streaming SGD method where each observation is used only once. Another major milestones has been the idea of iterate averaging (Polyak & Juditsky, 1992). A thorough theoretical analysis of asymptotic convergence of SGD can be found in (Kushner & Yin, 2003), whereas some non-asymptotic results have been presented in (Moulines & Bach, 2011).

A line of recent work known as variance-reduced SGD, e.g. (Roux et al., 2012; Shalev-Shwartz & Zhang, 2013; Johnson & Zhang, 2013; Defazio et al., 2014; 2015; Konečnỳ & Richtárik, 2013; Zhang et al., 2013), has exploited the finite sum structure of the empirical risk to establish linear convergence for strongly convex objectives and also a a better convergence rate for purely convex objectives (Mahdavi et al., 2013). There is also evidence of slightly improved statistical efficiency (Babanezhad et al., 2015). (Frostig et al., 2015) provides a non-asymptotic analysis of a streaming SVRG algorithm (SSVRG), for which a convergence rate approaching that of the ERM is established.

There have also been related data-adaptive sampling approaches, e.g. in the context of unsupervised learning (Lucic et al., 2015) or for non-uniform sampling of data points (Schmidt et al., 2013; He & Takác, 2015) with the goal of sampling important data points more often. This direction is largely orthogonal to our dynamic sizing of the sample, which is purely based on random subsampling. Our sampling strategy is instead based on revisiting samples which has also been explored in (Wang et al., 2016) to empirically improve the convergence of certain variance-reduced methods.

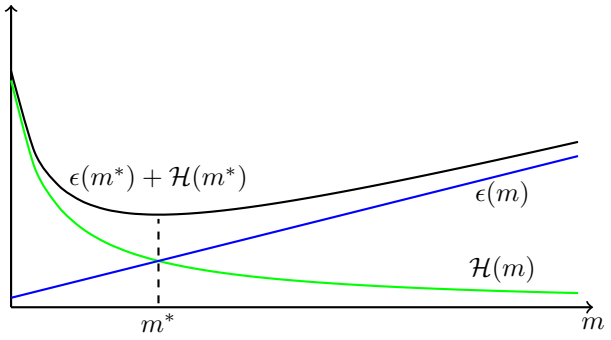

Figure 1. Tradeoff between sample statistical accuracy term $\mathcal{H}(m)$ and optimization suboptimality $\epsilon(m)$ using sample size $m < n$. Note that $\epsilon(m)$ is drawn by taking the first order approximation of the upper bound $Ce^{-\frac{n}{m}}$. Here, $m^* = O(n/\log n)$ yields the best balance between these two terms.

3. Methodology

3.1. Setting and Assumptions

We work under the assumptions made in Lemma 1 and focus on the large data regime, where $n \geq \kappa$ and the geometric rate of convergence of SAGA depends on n through $\rho_n = 1 - 1/n$. This is an interesting regime as the guaranteed progress per update is larger for smaller samples.

This form of ρ_n implies for the case of performing $t = n$ iterations, i.e. performing one pass[1]:

$$\mathbf{E}_{\mathcal{A}}\left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}^n) - \mathcal{R}^*_{\mathcal{S}}\right] \leq \left(1 - \frac{1}{n}\right)^n C_{\mathcal{S}} \leq \frac{C_{\mathcal{S}}}{e}\,. \qquad (4)$$

So we are guaranteed to improve the solution suboptimality on average by a factor $1/e$ per pass. This in turn implies that in order to get to a guaranteed accuracy $O(n^{-\alpha})$, we need $O(\alpha n \log n)$ update steps.

3.2. Sample Size Optimization

For illustrative purposes, let us use the above result to select a sample size for SAGA, which yields the best guarantees.

Proposition 2. *Assume $\mathcal{H}(m) = D/m$ and n is given. Define C to be an upper-bound on $C_{\mathcal{S}}, \forall \mathcal{S}$ (from Lemma 1), then for $m \geq \kappa$, $V(m) := \frac{D}{m} + Ce^{-\frac{n}{m}}$ provides a bound on the expected suboptimality of SAGA. It is minimized for the choice*

$$m^* = \max\left\{\kappa, \frac{n}{\log n + \log\frac{C}{D}}\right\}.$$

Proof. The first claim follows directly from the assumptions and Lemma 1. Moreover the tightest bound is obtained by differentiating V with regard to $1/m$ and solving for m (see Lemma 9 in appendix). □

[1]The SAGA analysis holds for i.i.d. sampling, so strictly speaking this is not a pass, but corresponds to n update steps.

The result implies that we will perform roughly $\log n + \log\frac{C}{D}$ epochs on the optimally sized sample. Also the value of the bound is (for simplicity, assuming $C = D$)

$$V(m^*) = \frac{\log n}{n} + \frac{1}{n} \leq V(n) = \frac{1}{n} + \frac{1}{e}\,, \qquad (5)$$

showing that the single pass approximation error on the full sample is too large (constant), relative to the statistical accuracy.

3.3. Dynamic Sample Growth

As we have seen, optimizing over a smaller sample can be beneficial (if we believe the significance of the bounds). But why chose a single sample size once and for all? A smaller sample set seems advantageous early on, but as an optimization algorithm approaches the empirical minimizer, it is hit by the statistical accuracy limit. This suggests that we should dynamically increment the size of the sample set. We illustrate this idea in Figure 2. In order to analyze such a dynamic sampling scheme, we need to relate the suboptimality on a sub-sample $\mathcal{T}$ to a suboptimality bound on $\mathcal{S}$. We establish a basic result in the following theorem.

Theorem 3. *Let $\boldsymbol{w}$ be an $(\epsilon, \mathcal{T})$-optimal solution, i.e. $\mathcal{R}_{\mathcal{T}}(\boldsymbol{w}) - \mathcal{R}^*_{\mathcal{T}} \leq \epsilon$, where $\mathcal{T} \subseteq \mathcal{S}$, $m := |\mathcal{T}|$, $n := |\mathcal{S}|$. Then the suboptimality of $\boldsymbol{w}$ for $\mathcal{R}_{\mathcal{S}}$ is bounded w.h.p. in the choice of $\mathcal{T}$ as:*

$$\mathbf{E}_{\mathcal{S}}\left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}) - \mathcal{R}^*_{\mathcal{S}}\right] \leq \epsilon + \frac{n-m}{n}\mathcal{H}(m)\,. \qquad (6)$$

Proof. Consider the following equality

$$\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}) - \mathcal{R}^*_{\mathcal{S}} = \mathcal{R}_{\mathcal{S}}(\boldsymbol{w}) \overset{(1)}{\mp} \mathcal{R}_{\mathcal{T}}(\boldsymbol{w}) \overset{(2)}{\mp} \mathcal{R}^*_{\mathcal{T}} \overset{(3)}{-} \mathcal{R}^*_{\mathcal{S}}$$

We bound the three involved differences (in expectation) as follows: (2): $\mathcal{R}_{\mathcal{T}}(\boldsymbol{w}) - \mathcal{R}^*_{\mathcal{T}} \leq \epsilon$ by assumption. (3): $\mathbf{E}_{\mathcal{S}}\left[\mathcal{R}_{\mathcal{T}}(\boldsymbol{w}^*_{\mathcal{T}}) - \mathcal{R}_{\mathcal{S}}(\boldsymbol{w}^*_{\mathcal{S}})\right] \leq 0$ as $\mathcal{T} \subseteq \mathcal{S}$. For (1) we apply the bound (see Lemma 10 in the appendix)

$$\mathbf{E}_{\mathcal{S}|\mathcal{T}}\left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}) - \mathcal{R}_{\mathcal{T}}(\boldsymbol{w})\right] \leq \frac{n-m}{n}\left|\mathcal{R}(\boldsymbol{w}) - \mathcal{R}_{\mathcal{T}}(\boldsymbol{w})\right|.$$

Moreover

$$\mathbf{E}_{\mathcal{T}}\left[\mathcal{R}(\boldsymbol{w}) - \mathcal{R}_{\mathcal{T}}(\boldsymbol{w})\right] \leq \sup_{\boldsymbol{w}'}\left|\mathcal{R}(\boldsymbol{w}') - \mathcal{R}_{\mathcal{T}}(\boldsymbol{w}')\right| \leq \mathcal{H}(m)$$

by Eq. (2), which concludes the proof. □

In plain English, this result suggests the following: If we have optimized $\boldsymbol{w}$ to $(\epsilon, \mathcal{T})$ accuracy on a sub-sample $\mathcal{T}$ and we want to continue optimizing on a larger sample $\mathcal{S} \supseteq \mathcal{T}$, then we can bound the suboptimality on $\mathcal{R}_{\mathcal{S}}$ by the same ϵ plus an additional "switching cost" of $(n-m)/n \cdot \mathcal{H}(m)$.

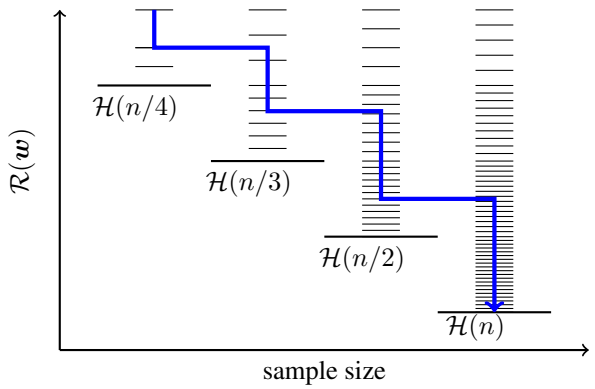

Figure 2. Illustration of an optimal progress path via sample size adjustment. The vertical black lines show the progress made at each step, thus illustrating the faster convergence for smaller sample size.

Table 1. Comparison of obtained bounds for different SAGA variants when performing $T \geq \kappa$ update steps.

METHOD	OPTIMIZATION ERROR	SAMPLES
SAGA (one pass)	const.	T
SAGA (optimal size)	$O(\log T \cdot \mathcal{H}(T))$	$T/\log T$
DYNASAGA	$O(\mathcal{H}(T))$	$T/2$

4. Algorithms & Analysis

4.1. Computational Limited Learning

The work of (Bottou, 2010) emphasized that for massive data sets the limiting factor of any learning algorithm will be its computational complexity T, rather than the number of samples n. For SGD this computational limit typically translates into the number of stochastic gradients evaluated by the algorithm, i.e. T becomes the number of update steps. One obvious strategy with abundant data is to sample a new data point in every iteration. There are asymptotic results establishing bounds for various SGD variants in (Bousquet & Bottou, 2008). However, SAGA and related algorithms rely on memorizing past stochastic gradients, cf. (Hofmann et al., 2015), which makes it beneficial to revisit data points, and which is at the root of results such as Lemma 1. This leads to a qualitatively different behavior and our findings indicate that indeed, the trade-offs for large scale learning need to be re-visited, cf. Table 1.

4.2. SAGA with Dynamic Sample Sizes

We suggest to modify SAGA to work with a dynamic sample size schedule. Let us define a *schedule* as a monotonic function $M : \mathbb{Z}_+ \to \mathbb{Z}_+$, where t is the iteration number and $M(t)$ the effective sample size used at t. We assume that a sequence of data points $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ drawn from P is given such that M induces a nested sequence of samples $\mathcal{T}_t := \{\boldsymbol{x}_i : 1 \leq i \leq M(t)\}$.

Algorithm 2 DYNASAGA

Input:
  training examples $\mathcal{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n)$, $\boldsymbol{x}_i \sim P$
  total number of iterations T (e.g. $T = 2n$)
  starting point $\boldsymbol{w}_0 \in \mathbb{R}^d$ (e.g $\boldsymbol{w}_0 = \mathbf{0}$)
  learning rate $\eta > 0$ (e.g. $\eta = \frac{1}{4L}$)
  sample schedule $M : [1:T] \to [1:n]$
$\boldsymbol{w} \leftarrow \boldsymbol{w}_0$
for $i = 1, \ldots, n$ **do**
  $\alpha_i \leftarrow \nabla f_{\boldsymbol{x}_i}(\boldsymbol{w}_0)$ {can also be done on the fly}
end for
for $t = 1, \ldots, T$ **do**
  sample $\boldsymbol{x}_i \sim \text{Uniform}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{M(t)})$
  $g \leftarrow \nabla f_{\boldsymbol{x}_i}(\boldsymbol{w}_{t-1})$
  $A \leftarrow \sum_{j=1}^{M(t)} \alpha_j / M(t)$ {can be done incrementally}
  $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} - \eta\,(g - \alpha_i + A)$
  $\alpha_i \leftarrow g$
end for

DYNASAGA generalizes SAGA (Defazio et al., 2014) in that it samples data points non-uniformly at each iteration. Specifically, for a given schedule M and iteration t, it samples uniformly from $\mathcal{T}_t$, but ignores $\mathcal{X} - \mathcal{T}_t$. The pseudocode for DYNASAGA is shown in Algorithm 1.

4.3. Upper Bound Recurrence

Assume we are given a stochastic optimization method that guarantees a geometrical decay at each iteration, i.e.

$$\mathbf{E}_{\mathcal{A}}\left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}^t) - \mathcal{R}^*_{\mathcal{S}}\right] \leq \rho_n \left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}^{t-1}) - \mathcal{R}^*_{\mathcal{S}}\right] \tag{7}$$

where $|\mathcal{S}| = n$ and expectation is over randomness of optimization process. [2] For acceleration, we pursue the strategy of using the basic inequalities obtained so far and to stitch them together in the form of a recurrence. At any iteration t we allow ourselves the choice to augment the current sample of size m by some increment $\triangle m \geq 0$. We define an upper bound function $\mathbf{U}$ as follows

$$\mathbf{U}(t,n) = \min \begin{cases} \rho_n \mathbf{U}(t-1,n) \\ \min\limits_{m<n} \left[\mathbf{U}(t,m) + \frac{n-m}{n}\mathcal{H}(m)\right], \end{cases} \tag{8}$$

such that $\mathbf{U}(0,m) = \xi$, where the initial error ξ is defined as:

$$\xi := \frac{4L}{\mu}\left[\mathcal{R}(\boldsymbol{w}^0) - \mathcal{R}(\boldsymbol{w}^*)\right]. \tag{9}$$

[2] Note that this assumption is slightly stronger than Lemma 1 but it leads to a much simpler proof technique.

We refer the reader to Lemma 8 in the Appendix for further details on how to derive the expression for ξ.

The construction of Eq. (8) is motivated by the following result:

Proposition 4. *W.h.p. over the random n-sample $\mathcal{X}$, the iterate sequence $\boldsymbol{w}^t$ generated by* DYNASAGA *fulfils*

$$\mathbf{E}_{\mathcal{X}}\left[\mathcal{R}_{\mathcal{T}_n}(\boldsymbol{w}^t) - \mathcal{R}^*_{\mathcal{T}_n}\right] \leq \mathbf{U}(t,n)\,.$$

Proof. By induction over t. The result for $t = 0$ follows directly from Lemma 8. The first case in Eq. (8) for the induction step (fixed sample size) follows from Eq. (7). The second case holds by virtue of Theorem 3 for any m, hence also for the minimum. □

Although the $\mathbf{U}$-recursion can be solved for small n using dynamic programming (assuming knowledge of all constants), we analyse a much simpler heuristics and its $n \to \infty$ behavior. This leads to interesting insights, while being very practical. In particular, our algorithm is an *anytime* algorithm, which does not require knowledge of the total number of iterations T ahead of time.

4.4. Sample Schedules

In this section, we present and analyse two adaptive sample-size schemes for DYNASAGA.

LINEAR We start with sample size κ and perform 2κ steps. From then on, we add a new sample every other iteration. The effective sample size is thus

$$M_{\text{LIN}}(t) = \max\left\{2\kappa, \left\lceil \tfrac{t}{2} \right\rceil\right\} \tag{10}$$

Note that this strategy defines an upper bound on $\mathbf{U}(2t,t)$ and $\mathbf{U}(2t+1,t)$.

ALTERNATING We have also implemented a variant where we perform updates in alternation: every other iteration we sample a new data point, which is added to the set. However, we also *force* an update on this fresh sample. In alternation, we simply re-sample an existing data point uniformly at random. We do not provide a theoretical analysis for this scheme but show experimentally that it slightly outperforms the LINEAR strategy (see results in the appendix). We thus report results for the ALTERNATING strategy in the experimental section.

4.5. Analysis

We now provide an analysis that establishes the convergence rate of the LINEAR strategy.

Lemma 5. *For $\mathcal{H}(n) = Dn^{-\alpha}, 0 < \alpha \leq 1$, the* LINEAR *strategy obtains the following suboptimality*

$$\mathbf{U}(2n,n) \leq \mathcal{H}(n) + \frac{\xi}{2}\left(\frac{\kappa}{n}\right)^2 \tag{11}$$

Proof. By induction over n. The base case follows from $C_m \leq \xi$. Using Eq. (8) and (11) for the inductive case, we get

$$\begin{aligned}\mathbf{U}\left(2(n+1),n+1\right) &\overset{(8)}{\leq} \rho^2_{n+1}\left[\mathbf{U}\left(2n,n\right) + \frac{1}{n+1}\mathcal{H}(n)\right] \\ &\overset{(11)}{\leq} \frac{\xi}{2}\left(\frac{\kappa}{n+1}\right)^2 + \frac{n^2\left(n+2\right)}{\left(n+1\right)^3}\mathcal{H}(n)\end{aligned}$$

Note that by definition of the logarithmic function, $\log\left[n(n+2)\right] < 2\log(n+1)$, and moreover

$$\frac{n}{n+1}\frac{\mathcal{H}(n)}{\mathcal{H}(n+1)} = \frac{n^{1-\alpha}}{(n+1)^{1-\alpha}} \leq 1\,,$$

which completes the proof. □

This means that for large enough n the LINEAR strategy is able to approach the statistical accuracy with $2n$ iterations, i.e. two "passes" over the data. Note the very significant improvement relative to the $\log n$ factor inherent to the optimal fixed sample size choice (see Table 1 for a comparison of these two bounds).

What does that imply for the $T = n$ case that we have been emphasizing? It is simple to state an answer as a corollary.

Corollary 6. *Under the same assumptions as Lemma 5, it holds for even n*

$$\mathbf{U}(n,n) \leq \left(3 \cdot 2^{\alpha-1}\right)\mathcal{H}\left(n\right) + 2\xi\left(\frac{\kappa}{n}\right)^2$$

Proof. Note that with Eq. (8) (a) and Lemma 5 (b) we get

$$\mathbf{U}(2n,2n) \overset{(a)}{\leq} \mathbf{U}(2n,n) + \frac{1}{2}\mathcal{H}(n) \overset{(b)}{\leq} \frac{3}{2}\mathcal{H}(n) + 2\xi\left(\frac{\kappa}{2n}\right)^2$$

The fact that $\mathcal{H}(n) = 2^{\alpha}\mathcal{H}(2n)$ completes the proof. □

The proof of the above corollary suggests to only use $n = T/2$ samples, when performing T steps and to simply ignore the other half (that potentially could have been sampled). One might wonder if a better strategy than the LINEAR one could be defined, e.g. by iterating more than twice on each newly added sample or by increasing the sample size by more than one. The next lemma answers this question and proves that the LINEAR strategy is optimal for large-scale datasets as long as $\mathcal{H}(n) \propto 1/n$.

Lemma 7. *Assume that $\mathcal{H}(n) \propto D/n$, then the* LINEAR *strategy is optimal for all sample size $n > \kappa$.*

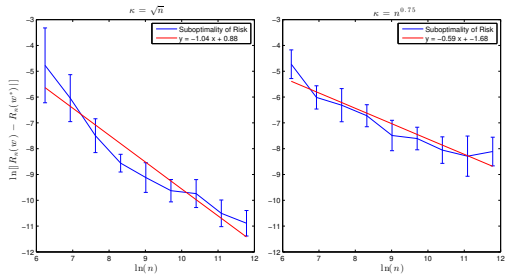

Figure 3. Results on synthetic dataset. (left) Since, the empirical suboptimality is $\propto 1/n$, we expect the slope measured on this plot to be close to one. (right) Since $\kappa = n^{0.75}$ slows down the convergence rate, the slope of this plot is less than one.

Proof. Here, we briefly state a sketch of the proof . The details are presented in Appendix A.2. First, we reformulate the problem of the optimal sample size schedule in terms of number of iterations on each samples size. Given that this problem is convex, we can use the KKT conditions to prove the optimality of incrementing by one sample (see Lemma 12) and iterating twice on each sample size (see Lemma 13). □

5. Experimental Results

We present experimental results on synthetic as well as real-world data, which largely confirms the above analysis.

5.1. Baselines

We compare DYNASAGA (both the LINEAR and ALTERNATING strategy) to various optimization methods presented in Section 2. This includes SGD (with constant and decreasing step-size), SAGA, streaming SVRG (SSVRG) as well as the mixed SGD/SVRG approach presented in (Babanezhad et al., 2015).

5.2. Experiment on synthetic data

We consider linear regression, where inputs $\boldsymbol{a} \in \mathbb{R}^d$ are drawn from a Gaussian distribution $\mathcal{N}(0, \Sigma_{d\times d})$ and outputs are corrupted by additive noise $y = \langle \boldsymbol{x}, \boldsymbol{w}^* \rangle + \epsilon$, $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$. We are given n i.i.d observations of this model, $\mathcal{S} = \{(\boldsymbol{a}_i, y_i)\}_{i=1}^n$, from which we compute the least squares risk $\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n \left(\langle \boldsymbol{a}_i, \boldsymbol{w}\rangle - y_i\right)^2$.

By considering the matrix A_n to be a row-wise arrangement of the input vectors $\boldsymbol{a}_i$, we can write the Hessian matrix of $\mathcal{R}_n(\boldsymbol{w})$ as $\Sigma_n = \frac{1}{n}A_n^T A_n$. When $n \gg d$, the matrix Σ_n converges to Σ and we can therefore assume that $\mathcal{R}_n(\boldsymbol{w})$ is μ-strongly convex and L-Lipschitz where the constants μ and L are the smallest and largest eigenvalues of Σ. We experiment with two different values for the condition number κ.

Table 2. Details of the real datasets used in our experiments. All datasets were selected from the LIBSVM dataset collection.

DATASET	SIZE	NUMBER OF FEATURES
RCV1.BINARY	20242	47236
A9A	32561	123
W8A	49749	300
IJCNN1	49990	22
REAL-SIM	72309	20958
COVTYPE.BINARY	581012	54
SUSY	5000000	18

Case $\kappa = \sqrt{n}$: We use a diagonal Σ with elements decreasing from 1 to $\frac{1}{\sqrt{n}}$, hence $\kappa = \sqrt{n}$. In this particular case the analysis derived in Lemma 5 predicts an upper bound $\mathbf{U}(n, n) < O(\frac{1}{n})$ which is confirmed by the results shown in Figure 3.

Case $\kappa = n^{\frac{3}{4}}$: When $\kappa = n^{\frac{3}{4}}$, the term $\left(\frac{\kappa}{n}\right)^2$ is the dominating term in the proposed upper-bound. In this case, $\mathbf{U}(n, n)$ is thus upper-bounded by $O\left(\frac{1}{\sqrt{n}}\right)$, which is once again verified experimentally in Figure 3.

5.3. Experiments on Real Datasets

We also ran experiments on several real-world datasets in order to compare the performance of DYNASAGA to state-of-the-art methods. The details of the datasets are shown in Table 2. Throughout all the experiments we used the logistic loss with a regularizer $\lambda = \frac{1}{\sqrt{n}}$[3]. Figures 4, and 5 show the suboptimality on the empirical risk and expected risk after a single pass over the datasets. The various parameters used for the baseline methods are described in Table 3. A critical factor in the performance of most baselines, especially SGD, is the selection of the step-size. We picked the best-performing step-size within the common range guided by existing theoretical analyses, specifically $\eta = 1/L$ and $\eta = \frac{C}{C+\mu t}$ for various values of C. Overall, we can see that DYNASAGA performs very well, both as an optimization as well as a learning algorithm. SGD is also very competitive and typically achieves faster convergence than the other baselines, however, its behaviour is not stable throughout all the datasets. The SGD variant with decreasing step-size is typically very fast in the early stages but then slows down after a certain number of steps. The results on the RCV dataset are somehow surprising as SGD with constant step-size clearly outperforms all methods but we show in the appendix that its behaviour

[3] We also present some additional results for various regularizers of the form $\lambda = \frac{1}{n^p}, p < 1$ in the appendix

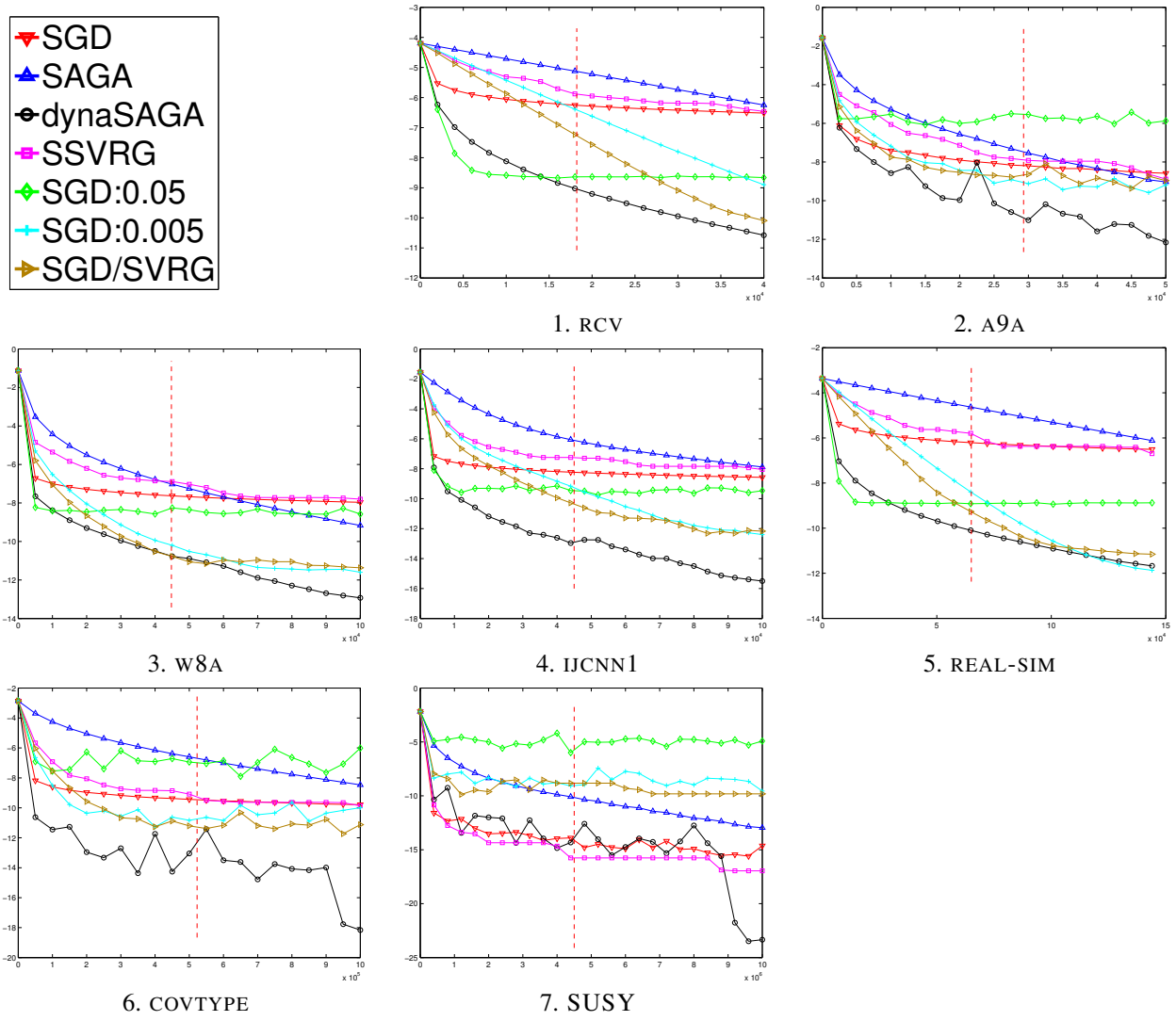

Figure 4. Suboptimality on the empirical risk. The vertical axis shows the suboptimality of the empirical risk, i.e. $\log_2 \mathbf{E}_{10}\left[\mathcal{R}_{\mathcal{T}}(\boldsymbol{w}^t) - \mathcal{R}^*_{\mathcal{T}}\right]$ where the expectation is taken over 10 independent runs. The training set includes 90% of the data. The vertical red dashed line is drawn after exactly one epoch over the data.

gets worse as we increase the condition number. As can be seen very clearly, DYNASAGA yields excellent solutions in terms of expected risk after one pass (see suboptimality values that intersect with the vertical red dashed lines).

6. Conclusion

We have presented a new methodology to exploit the trade-off between computational and statistical complexity, in order to achieve fast convergence to a statistically efficient solution. Specifically, we have focussed on a modification of SAGA and suggested a simple dynamic sampling schedule that adds one new data point every other update step. Our analysis shows competitive convergence rates both in term of suboptimality on the empirical risk as well as (more importantly) the expected risk in a one pass or a two pass setting. These results have been validated experimentally.

Our approach depends on the underlying optimization method only through its convergence rate for minimizing an empirical risk. We thus suspect that a similar sample size adaption is applicable to a much wider range of algorithms, including to non-convex optimization methods for deep learning.

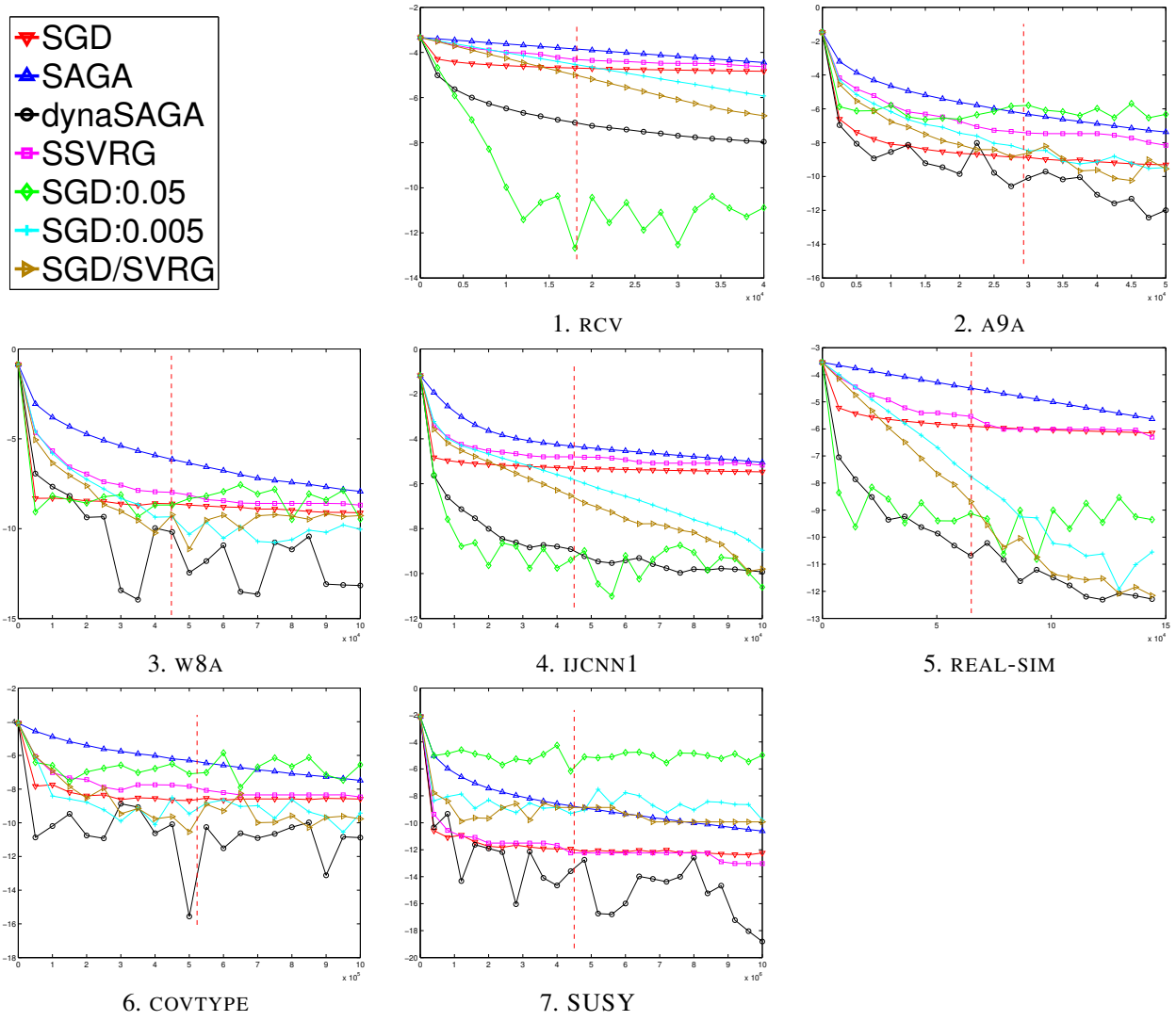

Figure 5. Suboptimality on the expected risk. The vertical axis shows the suboptimality of the expected risk, i.e. $\log_2 \mathbf{E}_{10}\left[\mathcal{R}_{\mathcal{S}}(\boldsymbol{w}^t) - \mathcal{R}_{\mathcal{S}}(\boldsymbol{w}^*_{\mathcal{T}})\right]$, where $\mathcal{S}$ is a test set which includes 10% of the data and $\boldsymbol{w}^*_{\mathcal{T}}$ is the optimum of the empirical risk on $\mathcal{T}$. The vertical red dashed line is drawn after exactly one epoch over the data.

References

Babanezhad, Reza, Ahmed, Mohamed Osama, Virani, Alim, Schmidt, Mark, Konečnỳ, Jakub, and Sallinen, Scott. Stop wasting my gradients: Practical svrg. *Advances in Neural Information Processing Systems*, 2015.

Bartlett, Peter L, Bousquet, Olivier, and Mendelson, Shahar. Local rademacher complexities. *Annals of Statistics*, pp. 1497–1537, 2005.

Bartlett, Peter L, Jordan, Michael I, and McAuliffe, Jon D. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Bottou, Léon. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pp. 177–186. Springer, 2010.

Boucheron, Stéphane, Bousquet, Olivier, and Lugosi, Gábor. Theory of classification: A survey of some recent advances. *ESAIM: probability and statistics*, 9:323–375, 2005.

Bousquet, Olivier. Concentration inequalities and empirical processes theory applied to the analysis of learning algorithms. *PhD thesis, Ecole Polytechnique*, 2002.

Bousquet, Olivier and Bottou, Léon. The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems*, pp. 161–168, 2008.

Boyd, Stephen and Vandenberghe, Lieven. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

Chandrasekaran, Venkat and Jordan, Michael I. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110 (13):E1181–E1190, 2013.

Defazio, Aaron, Bach, Francis, and Lacoste-Julien, Simon. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pp. 1646–1654, 2014.

Defazio, Aaron J, Caetano, Tibério S, and Domke, Justin. Finito: A faster, permutable incremental gradient method for big data problems. In *The international conference on Machine learning*, 2015.

Frostig, Roy, Ge, Rong, Kakade, Sham M., and Sidford, Aaron. Competing with the empirical risk minimizer in a single pass. In *The Conference on Learning Theory*, pp. 728–763, 2015.

He, Xi and Takác, Martin. Dual free SDCA for empirical risk minimization with adaptive probabilities. *CoRR*, abs/1510.06684, 2015.

Hofmann, Thomas, Lucchi, Aurelien, Lacoste-Julien, Simon, and McWilliams, Brian. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems 28*, pp. 2296–2304. Curran Associates, Inc., 2015.

Johnson, Rie and Zhang, Tong. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pp. 315–323, 2013.

Konečnỳ, Jakub and Richtárik, Peter. Semi-stochastic gradient descent methods. *arXiv preprint arXiv:1312.1666*, 2013.

Kushner, Harold J and Yin, George. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

Lucic, Mario, Ohannessian, Mesrob I, Karbasi, Amin, and Krause, Andreas. Tradeoffs for space, time, data and risk in unsupervised learning. In *AISTATS*, 2015.

Mahdavi, Mehrdad, Zhang, Lijun, and Jin, Rong. Mixed optimization for smooth functions. In *Advances in Neural Information Processing Systems*, pp. 674–682, 2013.

Moulines, Eric and Bach, Francis R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pp. 451–459, 2011.

Polyak, Boris T and Juditsky, Anatoli B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Roux, Nicolas L, Schmidt, Mark, and Bach, Francis R. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems*, pp. 2663–2671, 2012.

Schmidt, Mark, Roux, Nicolas Le, and Bach, Francis. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

Shalev-Shwartz, Shai and Zhang, Tong. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14:567–599, 2013.

Vapnik, Vlamimir. *Statistical learning theory*, volume 1. Wiley New York, 1998.

Wang, Jialei, Wang, Hai, and Srebro, Nathan. Reducing runtime by recycling samples. *arXiv preprint arXiv:1602.02136*, 2016.

Zhang, Lijun, Mahdavi, Mehrdad, and Jin, Rong. Linear convergence with condition number independent access of full gradients. In *Advances in Neural Information Processing Systems*, pp. 980–988, 2013.