# Nonconvex Variance Reduced Optimization with Arbitrary Sampling

**Samuel Horváth** [1] **Peter Richtárik** [1 2 3]

## Abstract

We provide the first importance sampling variants of variance-reduced algorithms for empirical risk minimization with non-convex loss functions. In particular, we analyze non-convex versions of SVRG, SAGA and SARAH. Our methods have the capacity to speed up the training process by an order of magnitude compared to the state of the art on real datasets. Moreover, we also improve upon current mini-batch analysis of these methods by proposing importance sampling for minibatches in this setting. Ours are the first optimal samplings for minibatches in the literature on stochastic optimization. Surprisingly, our approach can in some regimes lead to superlinear speedup with respect to the minibatch size, which is not usually present in stochastic optimization. All the above results follow from a general analysis of the methods which works with *arbitrary sampling*, i.e., fully general randomized strategy for the selection of subsets of examples to be sampled in each iteration. Finally, we also perform a novel importance sampling analysis of SARAH in the convex setting.

## 1. Introduction

Empirical risk minimization (ERM) is a key problem in machine learning as it plays a key role in training supervised learning models, including classification and regression problems, such as support vector machine, logistic regression and deep learning. A generic ERM problem has the finite-sum form

$$\min_{x\in\mathbb{R}^d} f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where $x$ corresponds to the parameters defining a model, $f_i(x)$ is the loss of the model $x$ associated with data point $i$, and $f$ is the average (empirical) loss across the entire training dataset. In this paper we focus on the case when the functions $f_i$ are $L_i$–smooth but *non-convex*. We assume the problem has a solution $x^*$.

One of the most popular algorithms for solving (1) is stochastic gradient descent (SGD) (Nemirovsky and Yudin, 1983; Nemirovski et al., 2009). In recent years, tremendous effort was exerted to improve its performance, leading to various enhancements which use acceleration (Allen-Zhu, 2016), momentum (Loizou and Richtárik, 2017), minibatching (Takáč et al., 2013), distributed implementation (Ma et al., 2015; 2017), importance sampling (Zhao and Zhang, 2015; Csiba and Richtárik, 2018; Qu et al., 2015; Chambolle et al., 2018), higher-order information (Qu et al., 2016; Gower et al., 2016), and a number of other techniques.

### 1.1. Variance-reduced methods

A particularly important recent advance has to do with the design of *variance-reduced (VR)* stochastic gradient methods, such as SAG (Roux et al., 2012), SDCA (Shalev-Shwartz and Zhang, 2013; Richtárik and Takáč, 2014), SVRG (Johnson and Zhang, 2013), S2GD (Konečný and Richtárik, 2017), SAGA (Defazio et al., 2014a), MISO (Mairal, 2015), FINITO (Defazio et al., 2014b) and SARAH (Nguyen et al., 2017a), which operate by modifying the classical stochastic gradient direction in each step of the training process in various clever ways so as to progressively reduce its variance as an estimator of the true gradient. We note that SAG and SARAH, historically the oldest and one the newest VR methods in the list, respectively, use a biased estimator of the gradient. In theory, all these methods enjoy linear convergence rates on smooth and strongly convex functions, which is in contrast with slow sublinear rate of SGD. VR methods are also easier to implement as they do not rely on a decreasing learning rate schedule. VR methods were recently extended to work with (non-strongly) convex losses by Konečný and Richtárik (2017), and more recently also to non-convex losses by Reddi et al. (2016b;a); Allen-Zhu and Hazan (2016); Nguyen et al. (2017b) - in all cases leading to best current rates for (1) in a given function class.

[1] King Abdullah University of Science and Technology, Saudi Arabia [2] Moscow Institute of Physics and Technology, Russia [3] University of Edinburgh, United Kingdom. Correspondence to: Samuel Horváth <samuel.horvath@kaust.edu.sa>, Peter Richtárik <peter.richtarik@kaust.edu.sa>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

### 1.2. Importance sampling, minibatching and non-convex models

In the context of problem (1), importance sampling refers to the technique of assigning carefully designed *non-uniform* probabilities $\{p_i\}$ to the $n$ functions $\{f_i\}$, and using these, as opposed to uniform probabilities, to sample the next data point (stochastic gradient) during the training process.

Despite the huge theoretical and practical success of VR methods, there are still considerable gaps in our understanding. For instance, an importance sampling variant of the popular SAGA method, with the "correct" convergence rate, was only designed very recently by Gower et al. (2018); and the analysis applies to strongly convex $f$ only. A coordinate descent variant of SVRG with importance sampling, also in the strongly convex case, was analyzed by Konečný et al. (2017). However, the method does not seem to admit a fast implementation. For dual methods based on coordinate descent, importance sampling is relatively well understood (Nesterov, 2012; Richtárik and Takáč, 2014; Qu and Richtárik, 2016; Qu et al., 2015; Allen-Zhu et al., 2016).

The territory is completely unmapped in the non-convex case, however. To the best of our knowledge, *no importance sampling* VR methods have been designed nor analyzed in the popular case when the functions $\{f_i\}$ are *non-convex*. An exception to this is dfSDCA (Csiba and Richtárik, 2015); however, this method applies to an explicitly regularized version of (1), and while the individual functions are allowed to be non-convex, the average $f$ is assumed to be convex. Given the dominance of stochastic gradient type methods in training large non-convex models such as deep neural networks, theoretical investigation of VR methods that can benefit from importance sampling is much needed.

The situation is worse still when one asks for *importance sampling of minibatches*. To the best of our knowledge, there are only a handful of papers on this topic (Richtárik and Takáč, 2016a; Csiba and Richtárik, 2018; Hanzely and Richtárik, 2019), none of which apply to the non-convex setting considered here, nor to the methods we will analyze, and the problem is open. This is despite the fact that minibatch methods are de-facto the norm for training deep nets. In practice, typically relatively small ($\mathcal{O}(1)$ or $\mathcal{O}(\log n)$) minibatch sizes are used.

### 1.3. Contributions

The main contributions of this paper are:

**Arbitrary sampling.** We peform a general analysis of three popular VR methods—SVRG (Johnson and Zhang, 2013), SAGA (Defazio et al., 2014a) and SARAH (Nguyen et al., 2017a)—in the *arbitrary sampling* paradigm (Richtárik and Takáč, 2016a; Qu and Richtárik, 2016; Qu and Richtárik, 2016; Qu et al., 2015; Chambolle et al., 2018). That is, we prove general complexity results which hold for an *arbitrary random set valued mapping* (aka arbitrary sampling) generating the minibatches of examples used by the algorithms in each iteration.

**Optimal sampling.** Starting from our general complexity results which hold for arbitrary sampling (see the second column in Table 1), we are able calculate the *optimal sampling out of all samplings of a given minibatch size* (see Lemma 2 and also the last column in Table 1). *This is the first time an optimal minibatch sampling was computed (from the class of all samplings) in the literature for any stochastic optimization method we know, including all variants of* SGD *and coordinate descent.* Indeed, while the results in (Richtárik and Takáč, 2016a; Csiba and Richtárik, 2018; Hanzely and Richtárik, 2019) and other works on this topic construct importance sampling for minibatches, these are not shown nor believed to be optimal.

**Improved rates.** Our iteration complexity bounds improve upon the best current rates for these methods even in the non-minibatch case. For SVRG and SAGA, this is true even when $L_i = L_j$ for all $i, j$, which is counter-intuitive as classical importance sampling is proportional to the constants $L_i$, which in this case would lead to uniform probabilities. Our importance sampling can be faster by up to the factor of $n$ compared to the current state of the art (see Table 1 and Appendix C). Our methods can enjoy *linear speedup* or even for some specific samplings *superlinear speedup* in minibatch size. That is, the number of iterations needed to output a solution of a given accuracy drops by a factor equal or greater to the minibatch size used. This is of utmost relevance to the practice of training neural nets with minibatch stochastic methods as our results predict that this is to be expected. We design importance sampling and *approximate importance sampling for minibatches* which in our experiments vastly outperform the standard uniform minibatch strategies.

**Best rates for SARAH under convexity.** Lastly, we also perform an analysis of importance sampling variant of SARAH in the convex and strongly convex case (Appendix I). These are the currently fastest rates for SARAH.

## 2. Importance Sampling for Minibatches

As mentioned in the introduction, we assume throughout that $f_i : \mathbb{R}^d \mapsto \mathbb{R}$ are smooth, but not necessarily convex. In particular, we assume that $f_i$ is $L_i$–smooth; that is, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i\|x - y\|$ for all $x, y \in \mathbb{R}^d$, where $\|x\| := (\sum_i x_i^2)^{1/2}$ is the standard Euclidean norm. Let us define $\bar{L} := \frac{1}{n}\sum_{i=1}^n L_i$. Without loss of generality assume that $L_1 \leq L_2 \leq \cdots \leq L_n$.

In this work, our aim is to find an $\epsilon$–accurate solution in expectation. A stochastic iterative algorithm for solving (1)

| Algorithm | Uniform sampling | Arbitrary sampling [NEW] | $S^*$ sampling [NEW] |
|---|---|---|---|
| SVRG | $\max\left\{n, \frac{(1+4/3)L_{\max}c_1 n^{2/3}}{\epsilon}\right\}$ | $\max\left\{n, \frac{(1+4\alpha/3)\bar{L}c_1 n^{2/3}}{\epsilon}\right\}$ | $\max\left\{n, \frac{\left(1+\frac{4(n-b)}{3n}\right)\bar{L}c_1 n^{2/3}}{\epsilon}\right\}$ |
| SAGA | $n + \frac{2L_{\max}c_2 n^{2/3}}{\epsilon}$ | $n + \frac{(1+\alpha)\bar{L}c_2 n^{2/3}}{\epsilon}$ | $n + \frac{(1+\frac{n-b}{n})\bar{L}c_2 n^{2/3}}{\epsilon}$ |
| SARAH | $n + \frac{\frac{n-b}{n-1}L_{\max}^2 c_3}{\epsilon^2}$ | $n + \frac{\alpha \bar{L}^2 c_3}{\epsilon^2}$ | $n + \frac{\frac{n-b}{n}\bar{L}^2 c_3}{\epsilon^2}$ |

**Table 1:** Stochastic gradient evaluation complexity for achieving $\mathrm{E}\left[\|\nabla f(x)\|^2\right] \leq \epsilon$ for two variants of SVRG, SAGA and SARAH for minimizing the average of smooth non-convex functions. Constants: $c_1, c_2, c_3$ are universal constant, $L_{\max} = \max_i L_i$; $\bar{L} = \frac{1}{n}\sum_i L_i$; $b$ = (average) minibatch size (hidden in $\alpha$); $\alpha$ can be for specific samplings smaller than 1 and decreasing with increasing $b$, which can lead to superlinear speedup in $b$. For SARAH this guarantee holds for one outer loop with minibatch size, where we assume $16\bar{L}^2(f(x^0) - f(x^*)^2)/(\epsilon b)^2 \gg 0$, in other words, minibatch size is not too big comparing to the required precision.

is said to achieve $\epsilon$–accurate solution if the random output $x_a$ of this algorithm satisfies $\mathrm{E}\left[\|\nabla f(x_a)\|^2\right] \leq \epsilon$.

### 2.1. Samplings

Let $S$ be a random set-valued mapping ("sampling") with values in $2^{[n]}$, where $[n] := \{1, 2, \ldots, n\}$. A sampling[1] is uniquely defined by assigning probabilities to all $2^n$ subsets of $[n]$. With each sampling we associate a *probability matrix* $\mathbf{P} \in \mathbb{R}^{n \times n}$ defined by

$$\mathbf{P}_{ij} := \mathrm{Prob}(\{i, j\} \subseteq S).$$

The *probability vector* associated with $S$ is the vector composed of the diagonal entries of $\mathbf{P}$: $p = (p_1, \ldots, p_n) \in \mathbb{R}^n$, where $p_i := \mathrm{Prob}(i \in S)$. We say that $S$ is *proper* if $p_i > 0$ for all $i$. It is easy to show that

$$b := \mathrm{E}\left[|S|\right] = \mathrm{Trace}\left(\mathbf{P}\right) = \sum_{i=1}^{n} p_i. \tag{2}$$

From now on, we will refer to $b$ as the *minibatch size* of sampling $S$. It is known that $\mathbf{P} - pp^\top \succeq 0$ (Richtárik and Takáč, 2016b). Let us without loss of generality assume that $p_1 \leq p_2 \leq \cdots \leq p_n$ and define constant $k = k(S) := |\{i \in [n] : p_i < 1\}| = \max\{i : p_i < 1\}$ to be the number of $p_i$'s not equal to one.

While our complexity results are general in the sense that they hold for any proper sampling, we shall now consider three special samplings; all with expected minibatch size $b \in (0, n]$:

**1. Standard uniform minibatch sampling ($S = S^u$).** $S$ is chosen uniformly at random from all subsets of $[n]$ of cardinality $b$. Clearly, $|S| = b$ with probability 1. The probability matrix is given by

$$\mathbf{P}_{ij} = \begin{cases} \frac{b}{n} & i = j, \\ \frac{b(b-1)}{n(n-1)} & i \neq j. \end{cases} \tag{3}$$

In the literature, this is known as the $b$–nice sampling (Richtárik and Takáč, 2016b; Qu and Richtárik, 2016).

**2. Independent sampling** $(S = S^*)$**.** For each $i \in [n]$ we independently flip a coin, and with probability $p_i > 0$ include element $i$ into $S$. Hence, by construction, $p_i = \mathrm{Prob}(i \in S)$ and $\mathrm{E}\left[|S|\right] \overset{(2)}{=} \sum_i p_i = b$. The probability matrix of $S$ is

$$\mathbf{P}_{ij} = \begin{cases} p_i & i = j, \\ p_i p_j & i \neq j. \end{cases}$$

**3. Approximate independent sampling** $(S = S^a)$**.** Independent sampling has the disadvantage that $k = k(S)$ coin tosses need to be performed in order to generate the random set. However, we would like to sample at the cost $\mathcal{O}(b + k - n)$ coin tosses instead. We now design a sampling which has this property and which in a certain precise sense, as we shall see later, approximates the independent sampling. In particular, given an independent sampling with parameters $p_i$ for $i \in [n]$, let $a = \lceil k \max_{i \leq k} p_i \rceil$. Since $\max_{i \leq k} p_i \geq \frac{b+k-n}{k}$, it follows that $a \geq b + k - n$. On the other hand, if $\max_{i \leq k} p_i = \mathcal{O}((b + k - n)/k)$, then

[1] Note that With-Replacement Sampling (WRS) does not arise as a special case of our definition of a sampling. Indeed, WRC allows a single element to be selected multiple times, which would not result in a subset of $[n]$. We analyzed our methods for WRS as well, using the tools we developed in this work. However, we found that WRS does not lead to any improvements in the rates, and hence decided to omit this sampling strategy and focus on the notion of arbitrary sampling, as defined here, for uniformity and simplicity of exposition.

$a = \mathcal{O}(b+k-n)$. We now sample a single set $S'$ of cardinality $a$ using the standard uniform minibatch sampling (just for $i \leq k$). Subsequently, we apply an independent sampling to select elements of $S'$, with selection probabilities $p_i' = kp_i/a$. The resulting set is $S$. Since

$$\mathrm{Prob}(i \in S) = \frac{\binom{k-1}{a-1}}{\binom{k}{a}} \frac{kp_i}{a} = p_i,$$

$$\mathrm{Prob}(\{i,j\} \subseteq S) = \frac{\binom{k-2}{a-2}}{\binom{k}{a}} \frac{kp_i}{a}\frac{kp_j}{a} = \frac{(a-1)k}{a(k-1)} p_i p_j$$

for $i, j \leq k$, the probability matrix of $S$ is given by

$$\mathbf{P}_{ij} = \begin{cases} p_i & i = j, \\ \frac{(a-1)k}{a(k-1)} p_i p_j & i \neq j; i,j \leq k, \\ p_i p_j & \text{otherwise.} \end{cases}$$

Since $\frac{(a-1)k}{a(k-1)} \approx 1$, the probability matrix of the approximate independent sampling approximates that of the independent sampling. Note that $S$ includes both the standard uniform minibatch sampling and the independent sampling as special cases. Indeed, the former is obtained by choosing $p_i = b/n$ for all $i$ (whence $a = b$ and $p_i' = 1$ for all $i$), and the latter is obtained by choosing $a = n$ instead of $a = \lceil k \max_{i \leq k} p_i \rceil$.

## 2.2. Key lemma

The following lemma, which we use as an upper bound for variance, plays a key role in our analysis.

**Lemma 1.** *Let $\zeta_1, \zeta_2, \ldots, \zeta_n$ be vectors in $\mathbb{R}^d$ and let $\bar{\zeta} := \frac{1}{n}\sum_{i=1}^n \zeta_i$ be their average. Let $S$ be a proper sampling (i.e., assume that $p_i = \mathrm{Prob}(i \in S) > 0$ for all $i$). Assume that there is $v \in \mathbb{R}^n$ such that*

$$\mathbf{P} - pp^\top \preceq \mathbf{Diag}\left(p_1 v_1, p_2 v_2, \ldots, p_n v_n\right). \tag{4}$$

*Then*

$$\mathrm{E}\left[\left\| \sum_{i \in S} \frac{\zeta_i}{np_i} - \bar{\zeta} \right\|^2\right] \leq \frac{1}{n^2}\sum_{i=1}^n \frac{v_i}{p_i}\|\zeta_i\|^2, \tag{5}$$

*where the expectation is taken over sampling $S$. Whenever (4) holds, it must be the case that*

$$v_i \geq 1 - p_i. \tag{6}$$

*Moreover, (4) is always satisfied for $v_i = n(1-p_i)$ for $i \leq k$ and $0$ otherwise. Further, if $|S| \leq d$ with probability 1 for some d, then (4) holds for $v_i = d$. The standard uniform minibatch sampling admits $v_i = \frac{n-b}{n-1}$, the independent sampling admits $v_i = 1 - p_i$, and the approximate independent sampling admits the choice*

$$v_i = 1 - p_i\left(1 - \tfrac{k-a}{a(k-1)}\right)$$

*if $i \leq k$, $v_i = 0$ otherwise.*

## 2.3. Optimal sampling

The following quantities play a key role in our general complexity results:

$$K := \frac{b}{n^2}\sum_{i=1}^n \frac{v_i L_i^2}{p_i}, \qquad \alpha := \frac{K}{\bar{L}^2}. \tag{7}$$

Above, $b = \mathrm{E}\left[|S|\right]$ is the minibatch size, $p_i = \mathrm{Prob}(i \in S)$, $\bar{L} := \frac{1}{n}\sum_i L_i$ and $\{v_i\}$ are defined in (4) in Lemma 1.

Our theory shows (see the 2nd column of Table 1 for a summary, and Section 3 for the full results) that in order to optimize the iteration complexity, we need to design sampling $S$ for which the value $\alpha$ is as small as possible. The following result sheds light on how $S$ should be chosen, from samplings of a given minibatch size $b$, to minimize $\alpha$.

**Lemma 2.** *Fix a minibatch size $b \in (0, n]$. Then the quantity $\alpha$, defined in (7), is minimized for the choice $S = S^*$ with the probabilities*

$$p_i := \begin{cases} (b+k-n)\frac{L_i}{\sum_{j=1}^k L_j}, & \text{if } i \leq k \\ 1, & \text{if } i > k \end{cases}, \tag{8}$$

*where $k$ is the largest integer satisfying $0 < b+k-n \leq \sum_{i=1}^k L_i / L_k$ (for instance, $k = n-b+1$ satisfies this). Usually, if $L_i$'s are not too much different, then $k = \mathcal{O}(n)$, for instance, if $bL_n \leq \sum_{i=1}^n L_i$ then $k = n$. If we choose $S = S^a$, then $\alpha$ is minimized for (8) with*

$$\alpha = \left(\frac{b\left(\sum_{i=1}^k L_i\right)^2}{(b+k-n)n^2} - \frac{bs}{n^2}\sum_{i=1}^k L_i^2\right) / \bar{L}^2, \tag{9}$$

*where $s = 1$ for $S^*$ and $s = 1 - \frac{k-a}{a(k-1)}$ for $S^a$. Moreover, if we assume[2] $bL_n \leq \sum_{i=1}^n L_i$, then $k = n$, thus*

$$\begin{aligned} \alpha_{S^*} &= 1 - b\frac{\sum_{i=1}^n L_i^2}{\left(\sum_{i=1}^n L_i\right)^2} \leq \frac{n-b}{n}, \\ \alpha_{S^u} &= (n-b)\frac{n}{n-1}\frac{\sum_{i=1}^n L_i^2}{\left(\sum_{i=1}^n L_i\right)^2}. \end{aligned}$$

*For the special case of all $L_i'$s are the same, one obtains*

$$\alpha_{S^*} = \frac{n-b}{n}, \quad \alpha_{S^u} = \frac{n-b}{n-1}.$$

From now on, let $S^*, S^a$ denote *Independent Sampling* and *Approximate Inpedendent Sampling*, respectively, with the probabilities defined in (8). Lemma 2 guarantees that the sampling $S^*$ is optimal (i.e., minimizes $\alpha$). Moreover, if we let $b_{\max} := \max\left\{b \mid bL_n \leq \sum_i L_i\right\}$, then we obtain *superlinear speedup in* $b$, up to $b_{\max}$ for all three algorithms.

[2] Note, that this can be always satisfied, if we uplift the smallest $L_i$'s, because if function is $L$-smooth, then it is also smooth with $L' \geq L$.

**Algorithm 1** SVRG with arb. sampling $\left(x^0, m, T, \eta, S\right)$

```
$\tilde{x}^0 = x_m^0 = x^0$, $M = \lceil T/m \rceil$
for $s = 0$ to $M - 1$ do
    $x_0^{s+1} = x_m^s$
    $g^{s+1} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{x}^s)$
    for $t = 0$ to $m - 1$ do
        Draw a random subset (minibatch) $S_t \sim S$
        $v_t^{s+1} = \sum_{i_t \in S_t} \frac{\nabla f_{i_t}(x_t^{s+1}) - \nabla f_{i_t}(\tilde{x}^s)}{np_{i_t}} + g^{s+1}$
        $x_{t+1}^{s+1} = x_t^{s+1} - \eta v_t^{s+1}$
    end for
    $\tilde{x}^{s+1} = x_m^{s+1}$
end for
Output: Iterate $x_a$ chosen uniformly at random from $\{\{x_t^{s+1}\}_{t=0}^m\}_{s=0}^M$.
```

**Algorithm 2** SAGA with arbitrary sampling $\left(x^0, d, T, \eta, S\right)$

```
$\alpha_i^0 = x^0$ for $i \in [n]$,
$g^0 = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\alpha_i^0)$
for $t = 0$ to $T - 1$ do
    Draw a random subset (minibatch) $S_t \sim S$
    Pick random subset $J_t \subset [n]$ s.t. Prob$(j \in J_t) = \frac{d}{n}$
    $v^t = \sum_{i \in S_t} \frac{\nabla f_i(x^t) - \nabla f_i(\alpha_{i_t}^t)}{np_{i_t}} + g^t$
    $x^{t+1} = x^t - \eta v^t$
    $\alpha_j^{t+1} = x^t$ for $j \in J_t$ and $\alpha_j^{t+1} = \alpha_j^t$ for $j \notin J_t$
    $g^{t+1} = g^t - \frac{1}{n}\sum_{j \in J_t}(\nabla f_j(\alpha_j^t) - \nabla f_j(\alpha_j^{t+1}))$
end for
Output: Iterate $x_a$ chosen uniformly at random from $\{x^t\}_{t=0}^T$.
```

**Algorithm 3** SARAH with arb. sampling $\left(x^0, m, T, \eta, S\right)$

```
$x_0^0 = x^0$
for $s = 1$ to $M - 1$ do
    $v_s^0 = \frac{1}{n}\sum_{i=1}^n \nabla f_i(x_s^0)$
    $x^1 = x^0 - \eta v^0$
    for $t = 1$ to $m - 1$ do
        Draw a random subset (minibatch) $S_t \sim S$
        $v^t = \sum_{i \in S_t} \frac{1}{np_i}(\nabla f_i(x^t) - \nabla f_i(x^{t-1})) + v^{t-1}$
        $x^{t+1} = x^t - \eta v^t$
    end for
    $x_{s+1}^0$ chosen uniformly at randomly from $\{x_s^t\}_{t=0}^m$
end for
Output: Iterate $x_a = x_M^0$
```

## 3. SVRG, SAGA and SARAH

In all of the results of this section we assume that $S$ is an *arbitrary proper sampling*. Let $b = \mathrm{E}\left[|S|\right]$ be the (average) minibatch size. We assume that $v$ satisfies (4) and that $\alpha$ (which depends on $v$) is defined as in (7). All complexity results will depend on $\alpha$ and $b$.

We propose three methods, Algorithm 1, 2 and 3, which are generalizations of original SVRG (Reddi et al., 2016a), SAGA (Reddi et al., 2016b) and SARAH (Nguyen et al., 2017b) to the arbitrary sampling setting, respectively. The original non-minibatch methods arise as special cases for the sampling $S = \{i\}$ with probability $1/n$, and the original minibatch methods arise as a special case for the sampling $S^u$ (described in Section 2.1).

Our general result for SVRG follows.

**Theorem 3** (Complexity of SVRG with arbitrary sampling)**.** *There exist universal constants $\mu_2 > 0$, $0 < \nu_2 < 1$ such that the output of Alg. 1 with mini-batch size $b \le \alpha n^{2/3}$, step size $\eta = \mu_2 b/(\alpha \bar{L} n^{2/3})$, and parameters $\beta = \bar{L}/n^{1/3}$, $m = \lfloor n\alpha/(3b\mu_2) \rfloor$ and $T$ (multiple of $m$) satisfies:*

$$\mathrm{E}\left[\|\nabla f(x_a)\|^2\right] \le \frac{\alpha \bar{L} n^{2/3}[f(x^0) - f(x^*)]}{bT\nu_2}.$$

*Thus in terms of stochastic gradient evaluations to obtain $\epsilon$-accurate solution, one needs following number of iterations*

$$\max\left\{n, \frac{\mu_2 \bar{L} n^{(2/3)}(f(x^0) - f(x^*))}{\epsilon \nu_2}\left(1 + \frac{\alpha}{3\mu_2}\right)\right\}.$$

In the next theorem we provide a generalization of the results by Reddi et al. (2016b).

**Theorem 4** (Complexity of SAGA with arbitrary sampling)**.** *There exist universal constants $\mu_3 > 0$, $0 < \nu_3 < 1$ such that the output of Alg. 2 with mini-batch size $b \le \alpha n^{2/3}$, step size $\eta = b/(\mu_3 \alpha \bar{L}^2 n^{2/3})$, and parameter $d = b/\alpha$ satisfies:*

$$\mathrm{E}\left[\|\nabla f(x_a)\|^2\right] \le \frac{\alpha \bar{L} n^{2/3}[f(x^0) - f(x^*)]}{bT\nu_3}.$$

*Thus, in terms of stochastic gradient evaluations, to obtain $\epsilon$–accurate solution, one needs following number of iterations*

$$n + \frac{\bar{L} n^{(2/3)}(f(x^0) - f(x^*))}{\epsilon \nu_3}(1 + \alpha).$$

We now introduce Algorithm 3: a general form of the SARAH algorithm (Nguyen et al., 2017b).

**Theorem 5** (Complexity of SARAH with arbitrary sampling)**.** *Consider one outer loop of Alg. 3 with*

$$\eta \le \frac{2}{\bar{L}\left(\sqrt{1 + \frac{4\alpha m}{b}} + 1\right)}. \tag{10}$$

*Then the output $x_a$ satisfies:*

$$\mathrm{E}\left[\|\nabla f(x_a)\|^2\right] \le \frac{2}{\eta(m+1)}[f(x_s^0) - f(x^*)].$$

**Algorithm 4** GD-Algorithm$\big(x^0, T, \mathbb{A}\big)$

```
Input: $x^0 \in \mathbb{R}^d$, $T$, $\mathbb{A}$
for $k = 0$ to $K$ do
    $x^k$ = Non-convex algorithm$(x^{k-1}, T, \mathbb{A})$
end for
Output: $x^K$
```

*Thus, to obtain $\epsilon$–accurate solution, one needs*

$$n + \tfrac{16\alpha\bar{L}^2(f(x^0)-f(x^*))^2}{2\epsilon^2} + \tfrac{\sqrt{16^2\alpha^2\bar{L}^4(f(x^0)-f(x^*))^4+16\epsilon^2\bar{L}^2(f(x^0)-f(x^*))^2b^2}}{2\epsilon^2}$$

*stochastic gradient evaluations.*

If all $L_i$'s are the same and we choose $S$ to be $S^a$, thus uniform with mini-batch size $b$, we can get back original result from (Nguyen et al., 2017b). Taking $b = n$, we can restore gradient descent with the correct step size $^1/_{\bar{L}}$.

## 4. Additional Results

In this section we describe three additional results: linear convergence for SVRG, SAGA and SARAH for gradient dominated functions, the first importance sampling results for SARAH for convex functions, and an array of new rates (with slight improvements) for non-minibatch versions of the above three methods for non-convex problems.

### 4.1. Gradient dominated functions

**Definition 1.** We say that $f$ is $\tau$-gradient dominated if

$$f(x) - f(x^*) \leq \tau\|\nabla f(x)\|^2,$$

for all $x \in \mathbb{R}^d$, where $x^*$ is an optimal solution of (1).

Gradient dominance is a weaker version of strong convexity due to the fact that if function is $\mu$-strongly convex then it is $\tau$-gradient dominated, where $\tau = 1/(2\mu)$.

Any of the non-convex methods in this paper can be used as a subroutine of Algorithm 4, where $T$ is the number of steps of the subroutine and $\mathbb{A}$ is the set of optimal parameters for the subroutine. We set $T = \alpha n^{2/3}/(b\nu_2)$ for SVRG and $T = \alpha n^{2/3}/(b\nu_3)$ for SAGA. In the case of SARAH, $T$ is obtained by solving $m + 1 = 2/\eta$ in $m$ and setting $T \leftarrow m$. Using Theorems 3, 4, 5 and the above special choice of $T$, we get

$$\mathrm{E}\left[\|\nabla f(x^k)\|^2\right] \leq \frac{1}{2\tau}\left(\mathrm{E}\left[f(x^{k-1})\right] - f(x^*)\right).$$

Combined with Definition 1, this guarantees a linear convergence with the same constant terms consisting of $\alpha$, $\bar{L}$ and $b$ that we had before in our analysis.

### 4.2. Importance sampling for SARAH under convexity

In addition to the results presented in previous sections, we also establish importance sampling results for SARAH in convex and strongly convex cases (Appendix I) with similar improvements as for the non-convex algorithm. Ours are the best current rates for SARAH in these settings.

### 4.3. Better rates for non-minibatch methods for non-convex problems

Lastly, we also provide specialized non-minibatch versions of non-convex SAGA, SARAH and SVRG, which are special cases of their minibatch versions presented in the main part with slightly improved guarantees (see Theorems 17, 18, 19 and 20 in the Appendix).

## 5. Experiments

In this section, we perform experiments with regression for binary classification, where our loss function has the form

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}(1 - y_i\sigma(a_i^\top x))^2,$$

where $\sigma(z)$ is the sigmoid function. Hence, $f$ is smooth but non-convex. We chose this function because $L_i$'s can be easily computed (however, in many cases, even for much more complex problems, they can usually be estimated). We use four LIBSVM datasets[3]: *covtype, ijcnn1, splice, australian*.

Parameters of each algorithm are chosen as suggested by the theorems in Section 3, and $x^0 = 0$. For SARAH, we chose $m = \lceil n/b \rceil$. The $y$ axis in all plots displays the norm of the gradient ($\|\nabla f(x)\|^2$) or the function value $f(x)$, and the $x$ axis depicts either epochs (1 epoch = 1 pass over data) or iterations.

### 5.1. Importance vs uniform sampling

Here we provide comparison of the methods with uniform ($S^u$) and importance ($S^*$) sampling. Looking at Figure 1, one can see that importance sampling outperforms uniform sampling for all three methods, in some cases, even by *several orders of magnitude.* For instance, in the left plot (mini-batch size $b = 2$ and *australian* dataset) the improvement is as large as 4 orders of magnitude.

Looking at Figure 2, one can see that there is an improvement not just in the norm of the gradient, but also in the function value. In the case of the *australian* dataset, the constants $L_i$'s are very non-uniform, and we can see that

[3] The LIBSVM dataset collection is available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

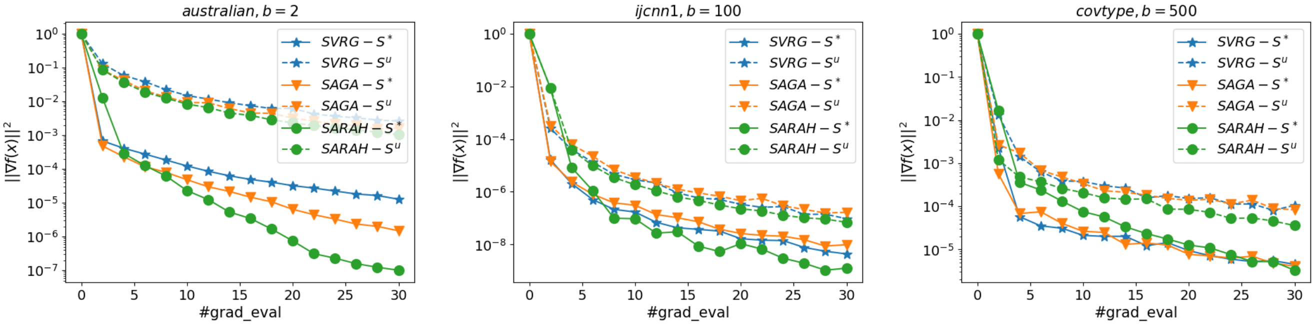

**Figure 1:** Comparison of all methods with uniform and importance sampling: gradient norm.

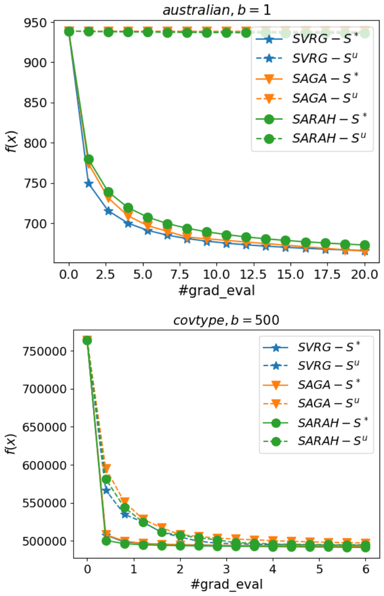

**Figure 2:** Comparison of all methods with uniform and importance sampling for function values.

the improvement is very significant.

### 5.2. Linear or superlinear speedup

Our theory suggests that linear or even superlinear speedup (in minibatch size $b$) can be obtained using the optimal independent $S^*$. Our experiments show that this is indeed the case in practice as well, and for all three algorithms.

Figure 3 confirms that linear, and sometimes even superlinear, speedup is present. For this dataset, such speedup is present up to the minibatch size of 250. The plots in the top row of Figure 3 depict convergence in a simulated multi-core setting, where the number of cores is the same as the minibatch size.

### 5.3. Independent vs approximate independent sampling

According to our theory, *Independent Sampling* $S^*$ is slightly better than *Approximate Independent Sampling* $S^a$. However, it is more expensive to use it in practice as generating samples from it involves more computational effort for large $n$.

The goal of our next experiment is to show that in practice $S^a$ yields comparable or even faster convergence. Hence, it is more reasonable to use this sampling for datasets where the number of data points $n$ is big. For an efficient implementation of $S^a$, we can almost get rid of dependence on $n$. Intuitively, $S^a$ works better because it has smaller variance in minibatch size than $S^*$.

Indeed, it can be seen from Figure 4 that $S^a$ can outperform $S^*$ in practice even though $S^*$ is optimal in theory. The difference can be small (the left and the middle plot), but also quite significant (right plot).

## References

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *STOC 2017: Symposium on Theory of Computing, 19-23*, 2016.

Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *The 33th International Conference on Machine Learning*, pages 699–707, 2016.

Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *The 33rd International Conference on Machine Learning*, pages 1110–1119, 2016.

Antonin Chambolle, Matthias J. Ehrhardt, Peter Richtárik, and Carola-Bibiane Schöenlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28 (4):2783–2808, 2018.

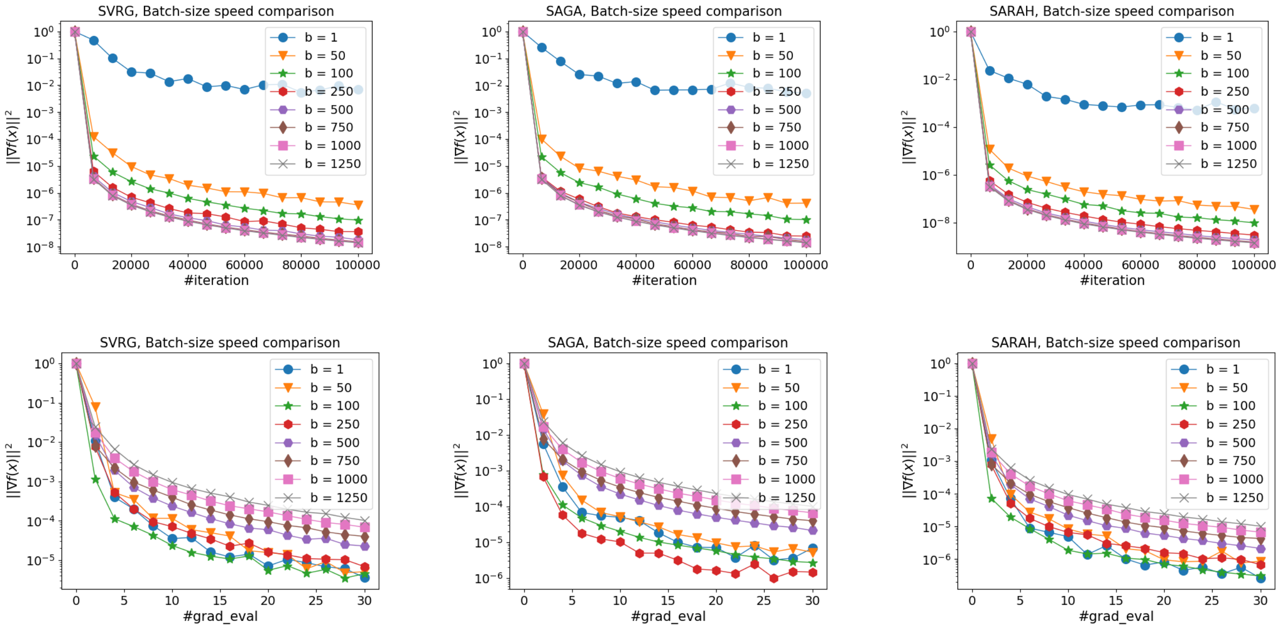

**Figure 3:** Minibatch speedup, *ijcnn1* dataset

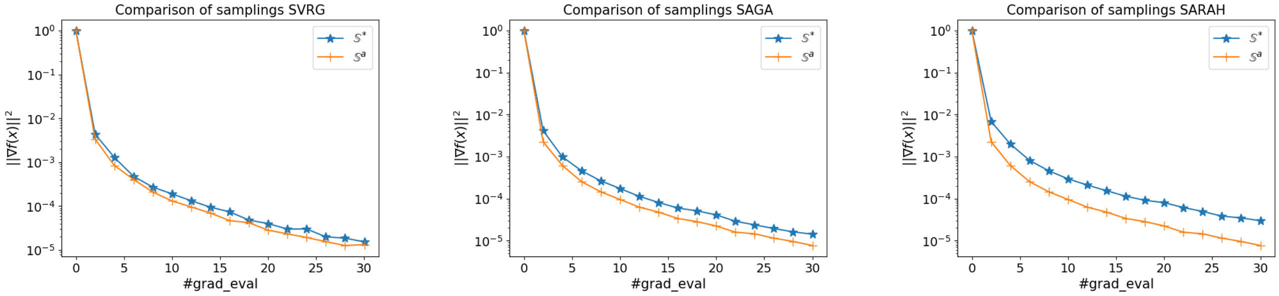

**Figure 4:** Performance of sampling $S^*$ vs. $S^a$, *splice* dataset.

Dominik Csiba and Peter Richtárik. Primal method for ERM with flexible mini-batching schemes and non-convex losses. *arXiv:1506.02227*, 2015.

Dominik Csiba and Peter Richtárik. Importance sampling for minibatches. *Journal of Machine Learning Research*, 19(27), 2018.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in Neural Information Processing Systems 27*, 2014a.

Aaron Defazio, Tiberio Caetano, and Justin Domke. Finito: A faster, permutable incremental gradient method for Big Data problems. *The 31st International Conference on Machine Learning*, 2014b.

Robert Mansel Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block BFGS: squeezing more curvature out of data. In *The 33rd International Conference on Machine Learning*, pages 1869–1878, 2016.

Robert Mansel Gower, Peter Richtárik, and Francis Bach. Stochastic quasi-gradient methods: variance reduction via Jacobian sketching. *arXiv:1805.02632*, 2018.

Filip Hanzely and Petert Richtárik. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

Jakub Konečný and Peter Richtárik. S2GD: Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, pages 1–14, 2017.

Jakub Konečný, Zheng Qu, and Peter Richtárik. S2CD: Semi-stochastic coordinate descent. *Optimization Methods and Software*, 32(5):993–1005, 2017.

Nicolas Loizou and Peter Richtárik. Momentum and stochastic momentum for stochastic gradient, Newton, proximal point and subspace descent methods. *arXiv:1712.09677*, 2017.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtárik, and Martin Takáč. Adding vs. averaging in distributed primal-dual optimization. In *The 32nd International Conference on Machine Learning*, pages 1973–1982, 2015.

Chenxin Ma, Jakub Konečný, Martin Jaggi, Virginia Smith, Michael I Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4):813–848, 2017.

Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

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.

Arkadi Nemirovsky and David B. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, New York, 1983.

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

Lam Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. *The 34th International Conference on Machine Learning*, 2017a.

Lam M Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. Stochastic recursive gradient algorithm for nonconvex optimization. *arXiv:1705.07261*, 2017b.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling I: algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

Zheng Qu, Peter Richtárik, and Tong Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in Neural Information Processing Systems 28*, pages 865–873, 2015.

Zheng Qu, Peter Richtárik, Martin Takáč, and Olivier Fercoq. SDNA: stochastic dual Newton ascent for empirical risk minimization. In *The 33rd International Conference on Machine Learning*, pages 1823–1832, 2016.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *The 33th International Conference on Machine Learning*, pages 314–323, 2016a.

Sashank J Reddi, Suvrit Sra, Barnabás Póczos, and Alex Smola. Fast incremental method for smooth nonconvex optimization. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 1971–1977. IEEE, 2016b.

Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10(6):1233–1243, 2016a.

Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016b.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(2):1–38, 2014.

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

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

Martin Takáč, Avleen Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. In *30th International Conference on Machine Learning*, 2013.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In *The 32rd International Conference on Machine Learning*, pages 1–9, 2015.