SVRG++ with Non-uniform Sampling

Tamás Kern **András György**

Department of Electrical and Electronic Engineering
Imperial College London, London, UK, SW7 2BT
{tamas.kern15,a.gyorgy}@imperial.ac.uk

Abstract

SVRG++ is a recent randomized optimization algorithm designed to solve non-strongly convex smooth composite optimization problems in the large data regime. In this paper we combine SVRG++ with non-uniform sampling of the data points (already present in the original SVRG algorithm), leading to an algorithm with the best sample complexity to date and state-of-the art empirical performance. While the combination and the analysis of the algorithm is admittedly straightforward, our experimental results show significant improvement over the original SVRG++ method with the new method outperforming all competitors on datasets where the smoothness of the components varies. This demonstrates that, despite its simplicity and limited novelty, this extension is important in practice.

1 Introduction

Minimizing the composite convex objective function

$$F(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) + \Psi(x), \tag{1}$$

over $x \in \mathbb{R}^d$, where $f_i : \mathbb{R}^d \to \mathbb{R}, 1 \le i \le n$, are smooth, convex functions and $\Psi : \mathbb{R}^d \to \mathbb{R}$ is a convex, but not necessarily smooth function (sometimes called the proximal function), has received significant attention recently. The popularity of this problem is due to the fact that empirical risk minimization (ERM), a widely used method in machine learning, often leads to this kind of optimization problems. For example, in several supervised machine learning tasks we are given n training samples $(w_i, y_i), 1 \le i \le n$, where $w_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \mathbb{R}$ is the label, and the goal is to estimate the label given a feature vector. ERM selects a predictor that minimizes the estimation error (plus possibly some penalty term) over the training sample in a class of prediction functions parametrized by $x \in \mathbb{R}^d$. Widely used special cases include

- Ridge regression: $f_i(x) = \frac{1}{2}(\langle w_i, x\rangle - y_i)^2 + \frac{\sigma}{2}\|x\|^2$ and $\Psi(x) \equiv 0$,
- Lasso: $f_i(x) = \frac{1}{2}(\langle w_i, x\rangle - y_i)^2$ and $\Psi(x) = \sigma\|x\|_1$,
- Elastic net: $f_i(x) = \frac{1}{2}(\langle w_i, x\rangle - y_i)^2 + \frac{\sigma}{2}\|x\|^2$ and $\Psi(x) = \sigma'\|x\|_1$,
- L_1-penalized logistic regression: $f_i(x) = \log(1 + \exp(-y_i \langle w_i, x\rangle))$ and $\Psi(x) = \sigma\|x\|_1$,

to mention a few.[1]

Let x^* denote the minimum of F over $\mathbb{R}^d$.[2] Since exact minimization of F is usually not possible, the goal of an optimization algorithms is to find an ε-optimal solution x_ε satisfying $F(x_\varepsilon) - F(x^*) \le \varepsilon$.

[1] Here and throughout the paper, $\|\cdot\|$ denotes the Euclidean norm.

[2] Throughout we assume that the minimum exists.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

The standard approach for minimizing F by gradient methods would involve computing all the component gradients ∇f_i for all i in every iteration of such an algorithm, whose $O(n)$ computational complexity becomes prohibitive if n is large, which is the case in many machine learning problems.

To overcome this issue, several stochastic gradient methods have been developed that utilize the special form of F in order to reduce the computational complexity of a single iteration to $O(1)$ (Zhang, 2004; Shalev-Shwartz and Zhang, 2012, 2013; Schmidt et al., 2013; Defazio et al., 2014; Xiao and Zhang, 2014; Allen-Zhu and Yuan, 2016). In essence, in every iteration of a gradient descent method, these algorithms select an index i uniformly at random and then estimate the gradient of $f(x) = \frac{1}{n}\sum_{j=1}^{n} f_j(x)$ using $\nabla f_i(x)$ only (i.e., not computing $\nabla f_j(x)$ for $j \neq i$). In the simplest stochastic gradient descent (SGD) case, this translates to using the estimate $\xi = \nabla f_i(x)$. While state-of-the-art methods like SAGA (Defazio et al., 2014), SVRG (Xiao and Zhang, 2014) or SVRG++ (Allen-Zhu and Yuan, 2016) use more refined techniques to estimate $\nabla f(x)$, another line of work, mostly for the related coordinate descent methods, showed that non-uniform sampling combined with importance weighted estimates also reduces the variance (see, e.g., Nesterov, 2012; Afkanpour et al., 2012). This idea of importance sampling was applied by Zhao and Zhang (2014) for SGD to find an ε-optimal solution to minimizing F, and Xiao and Zhang (2014) also used it in the original version of SVRG. On the other hand, non-uniform sampling was not applied in SVRG++, which is currently the best algorithm for non-strongly convex objective functions, and dealing with the importance-sampling idea is often left out from new algorithms to simplify notation and concentrate on more novel parts of the methods (see, e.g., Allen-Zhu and Yuan, 2016; Allen-Zhu, 2016).

In this paper we argue that this is not the right approach, as non-uniform sampling can significantly improve the performance of the algorithms in practice. In particular, we combine SVRG++ with importance sampling, and show that, despite of its simplicity, this minor change can dramatically improve the algorithm's performance. Specifically, the new algorithm outperforms other state-of-the-art methods in the literature for smooth, non-strongly convex objectives (including, e.g., all aforementioned examples with L_1-penalty), if the smoothness of the individual f_is varies significantly.

2 SVRG++ with non-uniform sampling (SVRG++NUS)

In the rest of the paper we assume that for each $1 \leq i \leq n$, f_i is convex and L_i-smooth for some $L_i > 0$, that is, $\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i\|x - y\|$ for all $x, y \in \mathbb{R}^d$. We define $L = \max_i\ L_i$ and $\bar{L} = \frac{1}{n}\sum_{i=1}^{n} L_i$.

The main idea underlying first order optimization methods to minimize F of the form (1) is to repeatedly perform the optimization step

$$x_{t+1} = \operatorname*{argmin}_{y\in\mathbb{R}^d} \left\{ \frac{1}{2\eta}\|y - x_t\|^2 + \langle \xi_t, y\rangle + \Psi(y) \right\},$$

where η is a step size parameter and $\mathbb{E}[\xi_t|\mathcal{F}_{t-1}] = \nabla f(x_t)$ with $\mathcal{F}_{t-1}$ denoting the σ-field generated by all randomness before the beginning of iteration t. The addition of SVRG is that the full gradient of f is computed from time to time, and ξ_t is defined as

$$\xi_t = \nabla f_i(x_t) - \nabla f_i(\tilde{x}_t) + \nabla f(\tilde{x}_t), \tag{2}$$

where i is selected uniformly at random and $\tilde{x}_t$ is the last point where the full gradient was computed. This variance reduction idea implies that as long as the full gradient is sampled in every m step (m being some parameter) $\mathbb{E}[(\xi - \nabla f(x_t))^2|\mathcal{F}_{t-1}] \to 0$. SVRG++ changes the resulting algorithm by increasing m exponentially over time. In this work we propose to choose i in (2) non-uniformly, according to some distribution $\boldsymbol{p}$. The resulting method, which we call SVRG++NUS (SVRG++ with Non-Uniform Sampling), is given in Algorithm 1.

It is easy to check that the stochastic gradient estimate ξ_t^s is unbiased, that is, $\mathbb{E}[\xi_t^s|\mathcal{F}_{t-1}^s] = \nabla f(x_t^s)$.[3] Similarly to SVRG (Xiao and Zhang, 2014), we chose the distribution $\boldsymbol{p}$ according to the smoothness parameters, that is, $p_i = \frac{L_i}{\sum_{i=1}^{n} L_i}$.

Combining the convergence analysis of SVRG (Xiao and Zhang, 2014) and SVRG++ (Allen-Zhu and Yuan, 2016), it is straightforward to show the following convergence guarantee about SVRG++NUS:

[3] $\mathcal{F}_{t-1}^s$ denotes the σ-field capturing all randomness before the tth iteration of epoch s.

Algorithm 1 SVRG++ with non-uniform sampling (SVRG++NUS) (x^ϕ, m_0, S, η)

$\tilde{x}^0 \leftarrow x^\phi$, $x_0^1 \leftarrow x^\phi$
for $s \leftarrow 1, \ldots, S$ **do**
  $\tilde{\mu}_{s-1} \leftarrow \nabla f(\tilde{x}^{s-1})$
  $m_s \leftarrow 2^s m_0$
  for $t \leftarrow 0, \ldots, m_s - 1$ **do**
    Pick $i \in \{1, \ldots, n\}$ with distribution $\boldsymbol{p}$
    $\xi_t^s \leftarrow (\nabla f_i(x_t^s) - \nabla f_i(\tilde{x}^{s-1}))/(np_i) + \tilde{\mu}_{s-1}$
    $x_{t+1}^s = \operatorname{argmin}_{y \in \mathbb{R}^d} \left\{ \frac{1}{2\eta}\|y - x_t^s\|^2 + \langle \xi_t^s, y \rangle + \Psi(y) \right\}$
  end for
  $\tilde{x}^s \leftarrow \frac{1}{m_s} \sum_{t=1}^{m_s} x_t^s$
  $x_0^{s+1} \leftarrow x_m^s$
end for
return $\tilde{x}^S$

Dataset	n	d	τ_{ridge}	$\tau_{lasso\ \&\ log}$
adult	32561	123	1.0094	1.0094
ijcnn1	49990	22	2.6151	2.6152
skin_nonskin	245057	3	3.1924	3.1924
w8a	49749	300	9.7852	9.7852

Table 1: Dimensions and τ values of the datasets.

Theorem 1. *Assume f_i is convex and L_i-smooth with $L_i > 0$ for all $1 \leq i \leq n$. Then, if Algorithm 1 is run with step size is $\eta = 1/(7\bar{L})$, it holds that*

$$\mathbb{E}[F(\tilde{x}^S) - F(x^*)] = O\left(\frac{F(x^\phi) - F(x^*)}{2^S} + \frac{\bar{L}\|x^\phi - x^*\|^2}{2^S m_0}\right),$$

and the algorithm requires $O(Sn + 2^S m_0)$ component gradient evaluations.

Following Allen-Zhu and Yuan (2016) to set the parameters of the algorithm, if the initial point x^ϕ satisfies $\|x^\phi - x^*\|^2 \leq \Theta$ and $F(x^\phi) - F(x^*) \leq \Delta$ for some known $\Theta, \Delta \in \mathbb{R}$, setting $S = \log_2(\Delta/\varepsilon)$ and $m_0 = \bar{L}\Theta/\Delta$ guarantees that an ε-optimal solution is achieved by $O\left(n \log \frac{\Delta}{\varepsilon} + \frac{\bar{L}\Theta}{\varepsilon}\right) = O\left(n \log \frac{1}{\varepsilon} + \frac{\bar{L}}{\varepsilon}\right)$ component gradient evaluations. This improves upon the SVRG++ bound of Allen-Zhu and Yuan (2016), which depends on L instead of $\bar{L}$.

3 Experiments

In this section we compare the new method, SVRG++NUS, with SVRG++ (Allen-Zhu and Yuan, 2016), the original SVRG method with non-uniform sampling (Xiao and Zhang, 2014) and also with SAGA (Defazio et al., 2014).[4] We consider three objectives: ridge regression, lasso, and L_1-penalized logistic regression, described in Section 1, with regularization parameter $\sigma = 10^{-4}$ in all cases. We have run the experiments on four different datasets from Chang and Lin (2011). The advantage of non-uniform sampling is expected to appear when the ratio $\tau = L/\bar{L}$, measuring the variability of the individual smoothness parameters, is larger; similarly to Zhao and Zhang (2014), the datasets were chosen to cover different values of τ. Properties of the datasets are summarized in Table 1.

All algorithms started at the origin, and were used with their theoretically optimal step size and parameters suggested by their original authors.[5] The smoothness parameters L_i were obtained by bounding the largest eigenvalues of the Hessian of f_i.

[4] Experimental results of Allen-Zhu and Yuan (2016) suggest that these algorithms usually outperform SDCA (Shalev-Shwartz and Zhang, 2012), which therefore is not included in our experiments.

[5] The value of η was selected as $1/(7\bar{L}), 1/(7L), 1/(5\bar{L}), 1/(3L)$ for SVRG++NUS, SVRG++, SVRG-NUS, and SAGA, respectively. The epoch size for SVRG is $m = 2n$, and $m_0 = n/4$ for SVRG++ and SVRG++NUS.

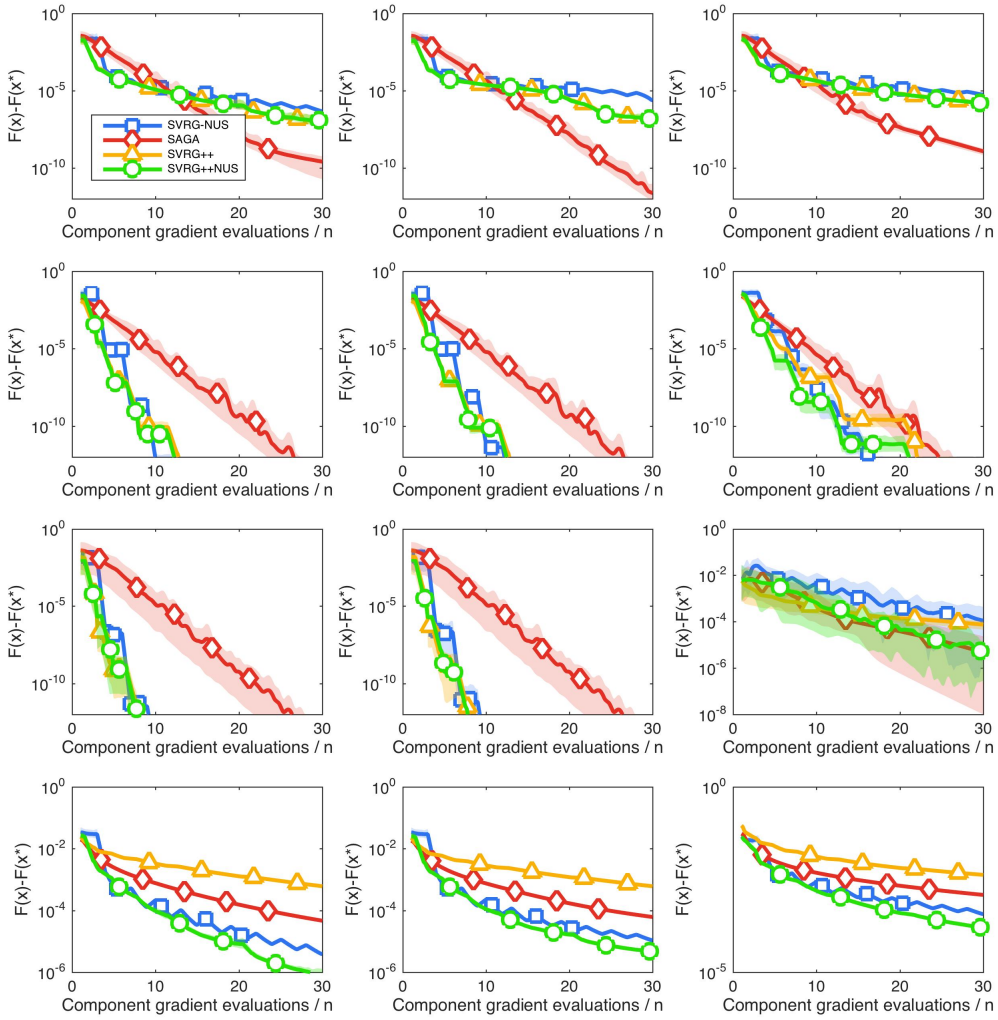

Figure 1: Performance of stochastic gradient methods on the datasets: adult (top row), ijcnn1(second row), skin_nonskin (third row), w8a (bottom row); with different objective functions: ridge regression (left column), lasso (middle column), logistic regression (right column). The legend for all plots is given in the top left plot.

The results of the experiments are shown in Figure 1. One can observe that for all datasets with a noticeable variability in the smoothness, SVRG++NUS is always among the best algorithms, and hence it should be the preferred choice for optimization for larger values of τ.

4 Conclusions

In this paper we presented a variant of SVRG++ (Allen-Zhu and Yuan, 2016) using non-uniform sampling. While the analysis of the algorithm is a straightforward combination of the original SVRG++ and SVRG proofs, the experimental results demonstrate that introducing non-uniform sampling yields a remarkable performance improvement for datasets with varying individual smoothness. This shows that, despite its theoretical simplicity, such non-uniform sampling could have significant effects in practice and should be considered in similar 'stochastic gradient'-type algorithms.

References

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 116–. 2004.

S. Shalev-Shwartz and T. Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *ArXiv e-prints*, 2012.

Shai Shalev-Shwartz and Tong Zhang. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. *ArXiv e-prints*, 2013.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing Finite Sums with the Stochastic Average Gradient. *ArXiv e-prints*, 2013.

Aaron Defazio, Francis R. Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. *CoRR*, abs/1407.0202, 2014.

Lin Xiao and Tong Zhang. A Proximal Stochastic Gradient Method with Progressive Variance Reduction. *ArXiv e-prints*, March 2014.

Zeyuan Allen-Zhu and Yang Yuan. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. volume abs/1506.01972. 2016.

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

Arash Afkanpour, András György, Csaba Szepesvári, and Michael H. Bowling. A randomized strategy for learning to combine many features. *CoRR*, abs/1205.0288, 2012.

Peilin Zhao and Tong Zhang. Stochastic Optimization with Importance Sampling. *ArXiv e-prints*, 2014.

Z. Allen-Zhu. Katyusha: The First Direct Acceleration of Stochastic Gradient Methods. *ArXiv e-prints*, 2016.

Chih-Chung Chang and Chih-Jen Lin. Libsvm : A library for support vector machines. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (2016/06), 2011. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27.