Towards Closing the Gap between the Theory and Practice of SVRG

Othmane Sebbouh [1] Nidham Gazagnadou [1] Samy Jelassi [2] Francis Bach [3] Robert M. Gower [1]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris [2]ORFE Department, Princeton University
[3]INRIA, École Normale Supérieure, PSL Research University

Goal: Empirical Risk Minimization

Consider the optimization problem

$$x^* = \arg\min_{x\in\mathbb{R}^d} \left\{ f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x) \right\} , \qquad (1)$$

where

- f is L-smooth and μ-strongly convex
- each f_i is $L_{\max}$-smooth

Stochastic Variance Reduced Gradient

Algorithm 1 *SVRG [4]*

Parameters: inner loop size $\boldsymbol{m \gtrsim \frac{L_{\max}}{\mu}}$, step size α, $p_t := \frac{1}{m}$
Initialization: $w_0 = x_0^m \in \mathbb{R}^d$
for $s = 1, 2, \ldots$ **do**
  $\boldsymbol{x_s^0 = w_{s-1}}$
  for $t = 0, 1, \ldots, m-1$ **do**
    Sample i_t uniformly at random in $\{1, \ldots, n\}$
    $g_s^t = \nabla f_{i_t}(x_s^t) - \nabla f_{i_t}(w_{s-1}) + \nabla f(w_{s-1})$
    $x_s^{t+1} = x_s^t - \alpha g_s^t$
  end for
  $w_s = \sum_{t=0}^{m-1} p_t x_s^t$
end for

Problem: SVRG differs from practice

- Constraint on the size of the loop m
- First iterate reset to the average of past iterates
- No theoretical justification for benefits of mini-batching

Motivations

- Close gap between theory and practice of SVRG
- Offer theoretical convergence guarantees
- Demonstrate benefits from mini-batching

Stochastic Reformulation

Problem (1) can be reformulated as

$$x^* = \arg\min_{x\in\mathbb{R}^d} \mathbb{E}_{v\sim D}\left[\frac{1}{n}\sum_{i=1}^n v_i f_i(x)\right] =: \mathbb{E}_{v\sim D}\left[f_v(x)\right] , \qquad (2)$$

where $\mathbb{E}_{v\sim D}[v] = \mathbf{1}_n$. To solve (2), we can use SVRG:

$$x_s^{t+1} = x_s^t - \alpha\left(\nabla f_{v_t}(x_s^t) - \nabla f_{v_t}(w_{s-1}) + \nabla f(w_{s-1})\right) ,$$

where $v_t \sim \mathcal{D}$ is sampled at each iteration.
Arbitrary sampling includes all types of sampling.

Example: mini-batching without replacement

Let $S \subset \{1, \ldots, n\}$ be a random set such that

$$\mathbb{P}[S = B] = 1/\binom{n}{b} \quad \text{for all } B \subset \{1, \ldots, n\},\ |B| = b .$$

$$\text{Let} \qquad v_i = \begin{cases} n/b & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

Then, $f_v(x) = \frac{1}{b}\sum_{i\in S} f_i(x)$ and $\nabla f_v(x) = \frac{1}{b}\sum_{i\in S}\nabla f_i(x)$.

Proposed algortihm: Free-SVRG

Algorithm 2 *Free-SVRG* (or 1-SVRG [5])

Parameters: **Free inner loop length $\boldsymbol{m}$**, step size α, $p_t := (1-\alpha\mu)^{m-1-t} \Big/ \sum_{i=0}^{m-1}(1-\alpha\mu)^{m-1-i}$
Initialization: $w_0 = x_0^m \in \mathbb{R}^d$
for $s = 1, 2, \ldots$ **do**
  $\boldsymbol{x_s^0 = x_{s-1}^m}$
  for $t = 0, 1, \ldots, m-1$ **do**
    Sample $v_t \sim \mathcal{D}$
    $g_s^t = \nabla f_{v_t}(x_s^t) - \nabla f_{v_t}(w_{s-1}) + \nabla f(w_{s-1})$
    $x_s^{t+1} = x_s^t - \alpha g_s^t$
  end for
  $w_s = \sum_{t=0}^{m-1} p_t x_s^t$
end for

Solves several issues with SVRG

- Inner iterates (x_s^t) continuously updated (no resetting)
- Free choice of the inner loop size
- Much easier analysis

Algorithm analysis

An essential constant for the analysis:

Lemma: Expected smoothness

Let $v \sim \mathcal{D}$ be a sampling vector. There exists $\mathcal{L} \geq 0$ such that for all $x \in \mathbb{R}^d$,

$$\mathbb{E}_{v\sim D}\left[\|\nabla f_v(x) - \nabla f_v(x^*)\|_2^2\right] \leq 2\mathcal{L}\left(f(x) - f(x^*)\right) .$$

Example: **mini-batching without replacement** [1, 2]

$$\mathcal{L} = \mathcal{L}(\boldsymbol{b}) = \frac{1}{\boldsymbol{b}}\frac{n-\boldsymbol{b}}{n-1}L_{\max} + \frac{n}{\boldsymbol{b}}\frac{\boldsymbol{b}-1}{n-1}L .$$

In particular, $\mathcal{L}(\boldsymbol{1}) = L_{\max}$ and $\mathcal{L}(\boldsymbol{n}) = L$.

Lyapunov Convergence Theorem 1

Let $\phi_s := \|x_s^m - x^*\|_2^2 + 8\alpha^2\mathcal{L}S_m(f(w_s) - f(x^*))$,

where $S_m = \sum_{i=0}^{m-1}(1-\alpha\mu)^{m-1-i}$. If $\alpha \leq 1/6\mathcal{L}$, then the iterates of Algorithm 2 converge with

$$\mathbb{E}\left[\phi_s\right] \leq \beta^s\phi_0, \quad \text{where} \quad \beta = \max\left\{(1-\alpha\mu)^m, \tfrac{1}{2}\right\} .$$

Total complexity for mini-batching

The **total complexity** of finding an $\epsilon > 0$ approximate solution that satisfies $\mathbb{E}\left[\|x_s^m - x^*\|_2^2\right] \leq \epsilon\,\phi_0$ is

$$C_m(\boldsymbol{b}) \quad := \quad 2\left(\frac{n}{m} + 2\boldsymbol{b}\right)\max\left\{\frac{3\mathcal{L}(\boldsymbol{b})}{\mu}, m\right\}\log\left(\frac{1}{\epsilon}\right) .$$

And for **mini-batching** (dropping the log term):

$$C_m(\boldsymbol{b}) := 2\left(\frac{n}{m} + 2\boldsymbol{b}\right)\max\left\{\frac{3}{\boldsymbol{b}}\frac{n-\boldsymbol{b}}{n-1}\frac{L_{\max}}{\mu} + \frac{3n}{\boldsymbol{b}}\frac{\boldsymbol{b}-1}{n-1}\frac{L}{\mu}, m\right\} .$$

Alternative algorithm: L-SVRG-D

Problem: SVRG requires the strong convexity

- SVRG relies on knowing μ

Solution: [3] proposed a **loopless** version of SVRG.
Improvement: when the variance of the estimate of the gradient is high, **decrease the step size**.

Algorithm 3 *L-SVRG-D* (Loopless-SVRG-Decrease)

Parameters: step size α, $p \in (0, 1]$
Initialization: $w^0 = x^0 \in \mathbb{R}^d$, $\alpha_0 = \alpha$
for $k = 0, 1, 2, \ldots$ **do**
  Sample $v_k \sim \mathcal{D}$
  $g^k = \nabla f_{v_k}(x^k) - \nabla f_{v_k}(w^k) + \nabla f(w^k)$
  $x^{k+1} = x^k - \alpha_k g^k$
  $(w^{k+1}, \alpha_{k+1}) = \begin{cases} (x^k, \alpha) & \text{with prob. } p \\ (w^k, \boldsymbol{\sqrt{1-p}}\,\alpha_k) & \text{with prob. } 1-p \end{cases}$
end for

Lyapunov Convergence Theorem 2

Consider the iterates of Algorithm 3 and let

$$\phi^k := \|x^k - x^*\|_2^2 + \frac{8\alpha_k^2\mathcal{L}}{p(3-2p)}\left(f(w^k) - f(x^*)\right) .$$

If $p \approx \frac{1}{n}$ and $\alpha \lesssim 2/7\mathcal{L}$, then

$$\mathbb{E}\left[\phi^k\right] \leq \beta^k\phi^0, \quad \text{where} \quad \beta = \max\left\{1 - \frac{2}{3}\alpha\mu, 1 - \frac{p}{2}\right\} .$$

Benefits

- **Bigger step size** for the first iterations of the loop, when the **variance is low**
- **Smaller step size** for the last iterations of the loop, when the **variance is high**

Same total complexity and optimal parameter settings as *Free-SVRG* (up to constants).

How to set the inner loop size?

We found a **range of values** minimizing the total complexity.
If $m \in [\min(n, L_{\max}/\mu), \max(n, L_{\max}/\mu)]$, then

$$C_m(1) \quad = \quad \boldsymbol{O\left(\left(n + \frac{L_{\max}}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)\right)} .$$

⚠ Includes **the practical choice $\boldsymbol{m = n}$** ⚠

How to set the mini-batch size?

For any fixed inner loop size m

- the **total complexity** is a **convex function** of b
- the **step size** is an **increasing function** of b

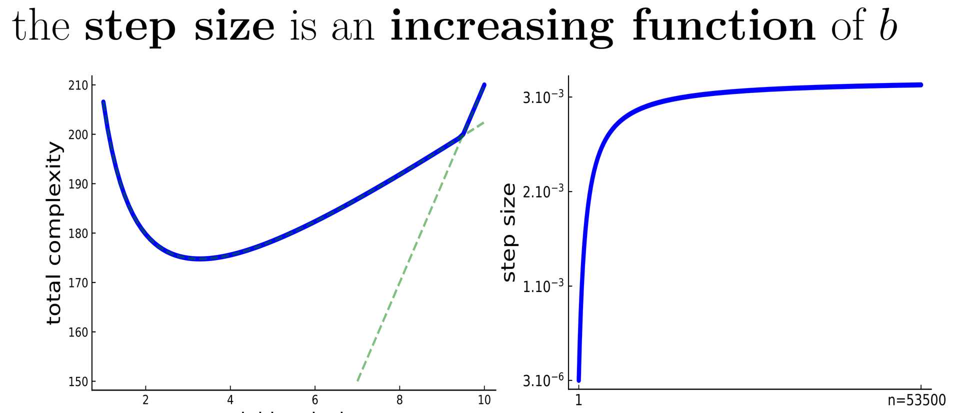

Figure: The total complexity (left) and the step size (right) as b increases.

We obtain **the optimal mini-batch size** for *Free-SVRG* (resp. *L-SVRG-D*) for the usual choice $m = n$ $\left(\text{resp. } p = \frac{1}{n}\right)$:

$$b^* = \begin{cases} 1 & \text{if } n \geq \frac{3L_{\max}}{\mu} \\ \left\lfloor \min(\tilde{b}, \hat{b}) \right\rfloor & \text{if } \frac{3L}{\mu} < n < \frac{3L_{\max}}{\mu} \\ \left\lfloor \hat{b} \right\rfloor & \text{otherwise, if } n \leq \frac{3L}{\mu} \end{cases}$$

$$\text{where } \hat{b} := \sqrt{\frac{n}{2}\frac{L_{\max}-L}{nL-L_{\max}}} \text{ and } \tilde{b} := \frac{3n(L_{\max}-L)}{n(n-1)\mu - 3(nL-L_{\max})} .$$

Experiments

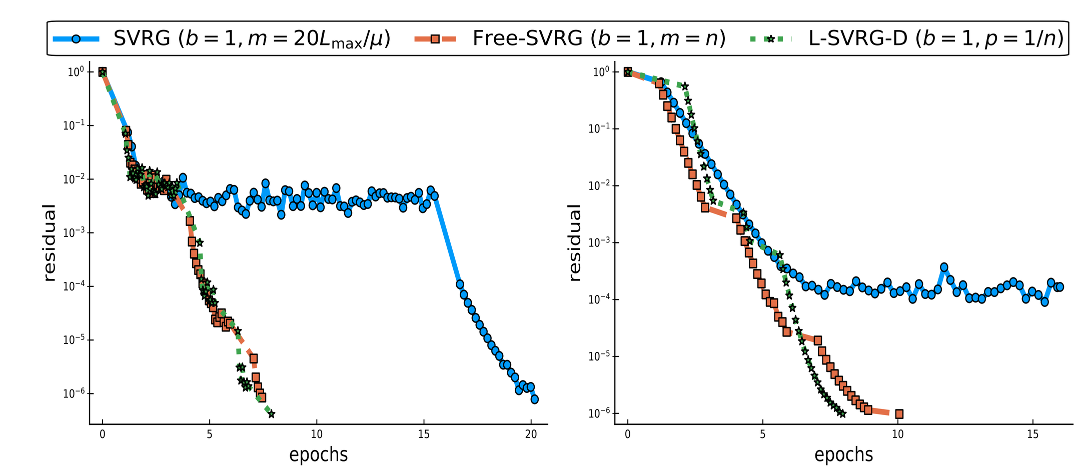

Figure: Theoretical settings for SVRG, *Free*-SVRG and *L-SVRG-D*.
Left: l_2–regularized logistic regression on *ijcnn1*.
Right: l_2–regularized ridge regression on *YearPredictionMSD*.

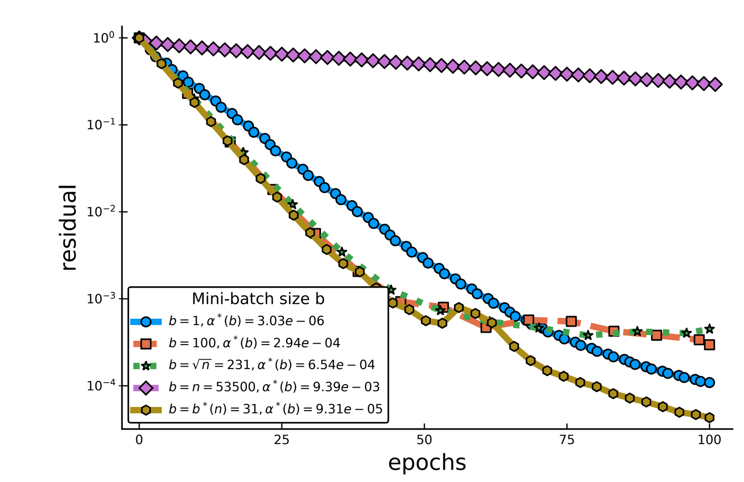

Figure: Different mini-batch sizes for *Free-SVRG* for a l_2–regularized ridge regression problem on the *slice* data set.

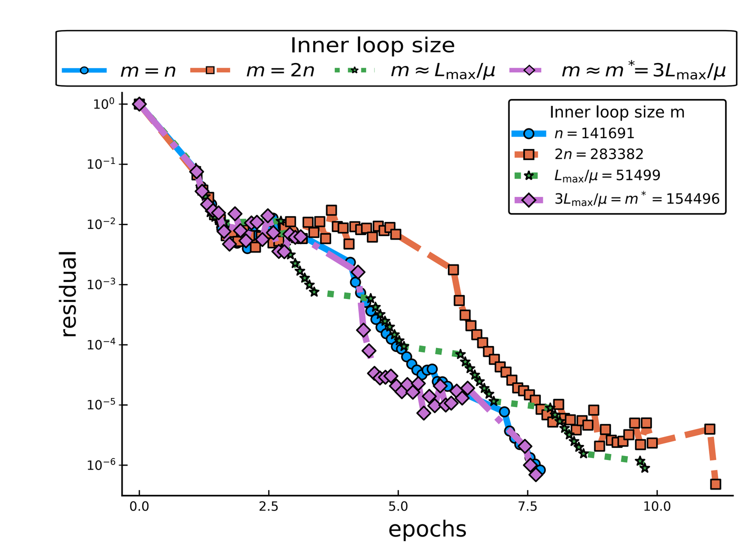

Figure: Different inner loop sizes for *Free-SVRG* for a l_2–regularized logistic regression problem on the *ijcnn1* data set.

References

[1] N. Gazagnadou, R. M. Gower, and J. Salmon.
Optimal Mini-Batch and Step Sizes for SAGA.
International Conference on Machine Learning, 2019.

[2] R. M. Gower, N. Loizou, X. Qian, A. Sailanbayev, E. Shulgin, and P. Richtárik.
SGD: General Analysis and Improved Rates.
International Conference on Machine Learning, 2019.

[3] T. Hofmann, A. Lucchi, S. Lacoste-Julien, and B. McWilliams.
Variance Reduced Stochastic Gradient Descent with Neighbors.
In *Advances in Neural Information Processing Systems*, 2015.

[4] R. Johnson and T. Zhang.
Accelerating Stochastic Gradient Descent using Predictive Variance Reduction.
In *Advances in Neural Information Processing Systems*, 2013.

[5] A. Raj and S. U. Stich.
k-SVRG: Variance Reduction for Large Scale Optimization.
arXiv:1805.00982, 2018.