Noname manuscript No.
(will be inserted by the editor)

A Class of Stochastic Variance Reduced Methods with an Adaptive Stepsize

Yan Liu · Congying Han* · Tiande Guo

Received: date / Accepted: date

Abstract Stochastic variance reduced methods have recently surged into prominence for solving large scale optimization problems in the context of machine learning. Tan, Ma and Dai et al. first proposed the new stochastic variance reduced gradient (SVRG) method with the Barzilai-Borwein (BB) method to compute step sizes automatically, which performs well in practice. On this basis, we propose a class of stochastic variance reduced methods with an adaptive stepsize which is based on local estimation of Lipschitz constant. Specifically, we adapt this stepsize to SVRG and stochastic recursive gradient algorithm (SARAH), which leads to two algorithms: SVRG-AS and SARAH-AS. We prove that both SVRG-AS and SARAH-AS converge linearly for strongly convex objective function. Numerical experiments on standard datasets indicate that our algorithms are effective and robust. The performance of SVRG-AS is better than SVRG-BB, and SARAH-AS is comparable to SARAH with best-tuned stepsizes. And our proposed stepsize is suitable for some other stochastic variance reduced methods.

This work was supported by the National Natural Science Foundation of China (11731013,11571014,11331012)

Corresponding author:Congying Han
School of Mathematical Sciences, Key Laboratory of Big Data Mining and Knowledge Management, UCAS, Beijing, 100049, China
Tel.: +86-010-88256908
E-mail: hancy@ucas.ac.cn

Yan Liu
School of Mathematical Sciences, University of Chinese Academy of Sciences (UCAS), Beijing, 100049, China
E-mail: liuyan23ucas@outlook.com

Tiande Guo
School of Mathematical Sciences, Key Laboratory of Big Data Mining and Knowledge Management, UCAS, Beijing, 100049, China
E-mail: tdguo@ucas.ac.cn

Keywords stochastic variance reduced methods · Lipschitz constant · adaptive stepsize

1 Introduction

In large scale machine learning, it is common to encounter the following optimization problems. Given a sequence of loss function $f_i(w) : \mathbb{R}^d \to \mathbb{R}$, we seek to minimize the sum of cost functions over samples. This can be stated as

$$\min_{w\in\mathbb{R}^d} F(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w) \tag{1.1}$$

where n is the sample size, and $d << n$. We assume that each f_i is convex and differentiable, and the function F is strongly convex in this paper.

The predominant methodology to solve the above problem advocates the use of stochastic gradient descent (SGD) methods[1]. It is imperative to employ stochastic approximation (SA) algorithms to solve large scale optimization problems, which can be traced back to the seminal work by [2]. The classical SGD mimics the steepest gradient descent method using a stochastic gradient, i.e., it updates the k-th iteration via

$$w_{_{k+1}} = w_k - \eta_k \nabla f_{i_k}(w_k) \tag{1.2}$$

where $\nabla f_{i_k}(w_k)$ denotes the gradient of the i_k-th component function at w_k, $\eta_k > 0$ is the stepsize, it is usually assumed that $\nabla f_{i_k}(w_k)$ is an unbiased estimate of the gradient of F at w_k, namely, $E[\nabla f_{i_k}(w_k)|w_k] = \nabla F(w_k)$.

SGD has been extensively employed in machine learning[3][4][5]. The performance of SGD relies heavily on the gradient approximation and the choice of stepsize. Because SGD suffers from the adverse effect of noisy gradient estimates, methods endowed with variance reduction capabilities have been developed to address this limitation, such as SAG[7][8], SAGA[9], SVRG[10], SARAH[11]. These stochastic variance reduced methods are able to converge linearly for strongly convex objective function. Meanwhile, the variance reduced gradients have been applying to develop the second order methods for solving (1.1), such as VITE[12], Stochastic L-BFGS[13], and Stochastic Block BFGS[14]. These methods all compute a full gradient in the outer loops and update the variance reduced stochastic gradient and the approximation of Hessian or the inverse Hessian in the inner loops.

Another key obstacle to SGD is the necessity of selecting appropriate stepsizes. In deterministic optimization, line search is employed to select a steplength that make the objective function descrease sufficiently. But stochastic line search is computationally prohibited because they must make decisions based on noisy approximation of true objective F. In many algorithms, a "sufficiently small" constant or a diminishing stepsize[5] is always used. And a fixed stepsize η_k can be typically used in variance-reduced gradient methods we mentioned above, but this stepsize must be determined in experiments. The

following result was pointed out by the seminal work of Robbins and Monro[2], that the stepsize should satisfy,

$$\sum_{k=1}^{\infty} \eta_k = \infty \quad and \quad \sum_{k=1}^{\infty} \eta_k^2 < \infty.$$

There are some recent works about the choice of stepsize in SGD. AdaGrad[15] and Adam[16] adaptively select the stepsize for every component based on the sum of the squares of the past gradients. SVRG-BB[17] uses the Barzilai-Borwein (BB) method[6][18][19] to automatically compute stepsize. Big Batch SGD[22] derives the optimal stepsize on each iteration for a quadratic approximation, they estimate the curvature information using the BB least-squares rule. As a stochastic quasi-Newton algorithm for self-concordant functions, SA-BFGS[20] use a curvature-adaptive step size which based on local curvature information[21] and can be computed analytically. In this paper, we propose an adaptive stepsize based on local estimation of Lipschitz constant for stochastic variance reduced methods.

Our contributions in this paper are in several aspects.

1) We propose to use the local estimation of the Lipschitz constant[23][24] to compute the stepsize for SVRG[10] (experiments suggest that our proposed stepsize can also apply to SVRG's variants) and SARAH[11]. The two new methods are named as SVRG-AS and SARAH-AS, respectively. We use a moving average of previous local estimation of the Lipschitz constant, this make our stepsize more reliable and more robust.
2) We prove the linear convergence of SVRG-AS and SARAH-AS for strongly convex objective functions. As a by product, we show the linear convergence of SARAH with option I (SARAH-I).
3) We conduct numerical experiments for SVRG-AS and SARAH-AS on solving logistic regression problems. The numerical results indicate that SVRG-AS is better than SVRG-BB, and SARAH-AS is comparable to and sometimes even better than SARAH with best-tuned stepsizes.

This paper is organized as follows. In Section 2, we briefly introduce some backgrounds. In Section 3, we propose the algorithms SVRG-AS and SARAH-AS with an adaptive stepsize, and we prove the linear convergence of these methods for strongly convex objective functions. Numerical experiments are then presented in Section 4. Finally, we draw some conclusions in Section 5.

2 Background

We learn from the SDAS-2[24] which adapts stepsize using local estimation of the Lipschitz constant, and inspired by SVRG-BB method to develop our adaptive stepsize for stochastic variance reduced methods, SVRG and SARAH.

2.1 Local Estimation of the Lipschitz Constant

The value of the stepsize has been related to the value of the Lipschitz constant L, the steepest descent algorithm[23] states that the sequence $\{w_k\}_{k=0}^{\infty}$, defined by

$$w_{k+1} = w_k - \frac{1}{2L}\nabla F(w_k), \quad k = 0, 1, 2 \ldots. \tag{2.1}$$

Consider that when the objective function is "steep", a small value for the stepsize is chosen to guarantee convergence. On the other hand, when F is "flat", a large stepsize can be chosen to accelerate the convergence. According to a pair of consecutive updates w_k, w_{k-1}, the local estimation of the Lipschitz constant can be calculated as follow,

$$\Lambda_k = \frac{\|\nabla F(w_k) - \nabla F(w_{k-1})\|}{\|w_k - w_{k-1}\|}. \tag{2.2}$$

In [24], the steepest descent algorithm with the stepsize (2.2) is following,

$$w_{k+1} = w_k - \frac{1}{2\Lambda_k}\nabla F(w_k), \quad k = 0, 1, 2 \ldots. \tag{2.3}$$

2.2 The SVRG Method

We describe the SVRG[10] as Algorithm 2.1 in the first place.

Algorithm 2.1 Stochastic Variance Reduced Gradient(SVRG) Method

Parameters: update frequency m, stepsize η, initial point $\tilde{w}_0$.

for $k = 0, 1, 2, \ldots$ **do**
  $v_k = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{w}_k)$
  Set $w_0 = \tilde{w}_k$
  for $t = 0, \ldots, m-1$ **do**
    Randomly pick $i_t \in \{1, \ldots, n\}$
    $w_{t+1} = w_t - \eta(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k)$
  end for
  option I: $\tilde{w}_{k+1} = w_m$
  option II: Set $\tilde{w}_{k+1} = w_i$, where i is selected uniformly at random from $\{1, 2, \ldots, m\}$
end for

SVRG has two loops. A full gradient v_k is computed in the outer loops (each outer iteration is called an *epoch*) and lower variance stochastic gradients computed in the inner loops. There have two options to choose $\tilde{w}$ for next outer loop as we described in Algorithm 2.1. We named SVRG and SARAH with option I as SVRG-I and SARAH-I, respectively, and the same to option II. It is well known that SVRG-I is better than SVRG-II in practice. Therefore, our proposed stepsize is applied to stochastic variance reduced methods with option I in this paper, but it is also applicable to algorithms with option II.

We now provide the convergence analysis of SVRG-I in [17] as Theorem 1.

Theorem 1 *In SVRG-I, let w_* be the optimal solution, if m and η are chosen such that*

$$\alpha_I := (1 - 2\eta\mu(1 - \eta L))^m + \frac{4\eta L^2}{\mu(1 - \eta L)} < 1, \tag{2.4}$$

then it converges linearly in expectation:

$$E \left\| \tilde{w}_k - w_* \right\|_2^2 < \alpha_I^k \left\| \tilde{w}_0 - w_* \right\|_2^2 .$$

Then we cite the convergence analysis of SVRG-II given in [10].

Theorem 2 *In SVRG-II, assume that m is sufficiently large so that*

$$\alpha_{II} := \frac{1}{\mu\eta(1 - 2L\eta)m} + \frac{2\eta L}{1 - 2\eta L} < 1, \tag{2.5}$$

then it is linear convergence in expectation:

$$E[F(\tilde{w}_k) - F(w_*)] \leqslant \alpha_{II}^k [F(\tilde{w}_0) - F(w_*)].$$

2.2.1 The SVRG-BB Method

To our knowledge, the SVRG-BB method is the first work which applied the BB method[18] to stochastic gradient algorithms. The performance of SVRG-BB is comparable to SVRG with best-tuned step sizes. The SVRG-BB method chooses Option I in Algorithm 2.1, and computes the stepsize η_k using the BB method in every outer loop except the first. The update rule in the inner loops is given as follow,

$$w_{t+1} = w_t - \frac{1}{m} \cdot \frac{\left\| s_k \right\|^2}{s_k^T y_k} (\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k)$$

or

$$w_{t+1} = w_t - \frac{1}{m} \cdot \frac{s_k^T y_k}{\left\| y_k \right\|^2} (\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k)$$

where $s_k = \tilde{w}_k - \tilde{w}_{k-1}$, $y_k = v_k - v_{k-1}$.

We now cite the convergence analysis of SVRG-BB given in [17] as follow,

Theorem 3 *Denote $\theta = (1 - e^{-2\mu/L})/2$. It is easy to see that $\theta \in (0, 1/2)$. Let w_* be the optimal solution to problem (1.1). If m in Theorem 1 is chosen such that*

$$m > \max \left\{ \frac{2}{\log(1 - 2\theta) + 2\mu/L}, \frac{4L^2}{\theta\mu^2} + \frac{L}{\mu} \right\},$$

then SVRG-BB converges linearly in expectation:

$$E \left\| \tilde{w}_k - w_* \right\|^2 < (1 - \theta)^k \left\| \tilde{w}_0 - w_* \right\|^2 .$$

2.3 The SARAH Method

SARAH[11], as a novel framework to the finite-sum minimization problems, is similar to SVRG, they both contain outer loops which require one full gradient evaluation after every m inner loops. But SARAH updates the stochastic step direction v_t recursively by adding and subtracting component gradients to and from the previous v_{t-1} ($t \geqslant 1$) in the inner loops. The key step of SARAH is that

$$v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}, \tag{2.6}$$

and then the iterate update rule is:

$$w_{t+1} = w_t - \eta v_t. \tag{2.7}$$

Algorithm 2.2 with option II, namely as SARAH, can reach the following convergence result, if each f_i is L-smooth and convex. Assume the choice of η and m satisfy

$$\sigma := \frac{1}{\mu\eta(m+1)} + \frac{\eta L}{2-\eta L} < 1, \tag{2.8}$$

then, it converges linearly in expectation,

$$E[\|\nabla F(\tilde{w}_k)\|^2] \leqslant \sigma^k \|\nabla F(\tilde{w}_0)\|^2 .$$

The pseudocode is outlined as Algorithm 2.2.

Algorithm 2.2 The SARAH Method

Parameters: update frequency m, stepsize $\eta > 0$, initial point $\tilde{w}_0$.

for $k = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{k-1}$
  $v_0 = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  $w_1 = w_0 - \eta v_0$
  for $t = 1, \ldots, m-1$ **do**
    Randomly pick $i_t \in \{1, \ldots, n\}$
    $v_t = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}$
    $w_{t+1} = w_t - \eta v_t$
  end for
  option I: $\tilde{w}_k = w_m$
  option II: Set $\tilde{w}_k = w_t$ with t chosen uniformly at random from $\{1, 2, \ldots, m\}$
end for

3 The Adaptive Stepsize

In this section, we propose the adaptive stepsize based on local estimation of L for stochastic variance reduced methods, SVRG and SARAH. To proceed with the analysis of the proposed algorithm, we make the following common assumption.

Assumption 1 *We assume that the objective function $F(w)$ is μ-strongly convex, i.e.,*

$$F(w) \geqslant F(w') + \nabla F(w')^T (w - w') + \frac{\mu}{2} \|w - w'\|^2, \quad \forall w, w' \in \mathbb{R}^d.$$

We also assume that the gradient of each component function $f_i(w)$ is convex and L-Lipschitz continuous, i.e.,

$$\|\nabla f_i(w) - \nabla f_i(w')\|_2 \leqslant L\|w - w'\|_2, \quad \forall w, w' \in \mathbb{R}^d.$$

Under this assumption, it is easy to see that $\nabla F(w)$ is also L-Lipschitz continuous:

$$\|\nabla F(w) - \nabla F(w')\|_2 \leqslant L\|w - w'\|_2, \quad \forall w, w' \in \mathbb{R}^d.$$

The convergence analyses of SVRG and SARAH suggest that the choice of η depends on L. Motivated by SDAS-2[24] and SVRG-BB[17], we calculate local estimation of L in every outer loop except the first one. And we utilize the moving average for most recent c estimations $\{\Lambda_{k-c}, \Lambda_{k-c+1}, \ldots, \Lambda_k\}$ to make it more reliable and more robust,

$$L_k = \sum\nolimits_{i=k-c+1}^{k} \beta_i \Lambda_i$$

where $\beta_i > 0$ for $i \in \{k - c + 1, \ldots, k\}, k \geqslant c$ and $\sum_{i=k-c+1}^{k} \beta_i = 1$.

Algorithm 3.1 Compute Adaptive Stepsize

Input: Λ_i, $\beta_i > 0, i \in \{1, \ldots, c\}$, $c > 1$, δ_1, δ_2.
1: **if** $k = 1$ **then**
2: $L_1 = \Lambda_1$
3: **else if** $1 < k < c$ **then**
4: $L_k = \rho\Lambda_k + (1 - \rho)L_{k-1}$
5: **else if** $k \geqslant c$ **then**
6: $L_k = \sum_{i=k-c+1}^{k} \beta_i \Lambda_i$
7: **end if**
8: compute $\eta_t^k = \frac{1}{m^{h(\delta_1 k + \delta_2 t)} L_k}$
Ouput: η_t^k

Consider that in the early epoch, stepsize should be larger to accelerate convergence, we construct a function of the current update in the k-th outer loop and t-th inner loop for computing the stepsize as follow,

$$\eta_t^k := \frac{1}{m^{h(\delta_1 k + \delta_2 t)} \cdot L_k}$$

where $h(\delta_1 k + \delta_2 t)$ is a strictly monotone increasing function, and satisfy

$$h(\delta_1 k + \delta_2 t) \in [\frac{1}{2}, 1), \forall k, t \quad \textit{and} \quad |h(\delta_1 k + \delta_2 m) - m| < \varepsilon, \forall k.$$

In practice, we make $\delta_1, \delta_2 \in \{0, 1\}$ and $0 < \varepsilon < \frac{1}{m}$. Algorithm 3.1 shows that how to calculate the adaptive stepsize (AS).

3.1 The SVRG-AS Method

Now we present SVRG-AS as Algorithm 3.2. Because the AS is not calculable in the first outer loop, we provide η_0 for the first epoch. In fact, the performance of SVRG-AS is not sensitive to η_0. The AS can also naturally incorporated to other SVRG's variants, such as Batching SVRG[25], S2GD[26].

Algorithm 3.2 Stochastic Variance Reduced Gradient with Adaptive Stepsize (SVRG-AS) Method

Parameters: update frequency m, initial stepsize η_0, and $\tilde{w}_0$, a constant c, $\beta_i > 0, i \in \{1, \ldots, c\}$, δ_1, δ_2.

for $k = 0, 1, 2, \ldots$ **do**
  $v_k = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{w}_k)$
  if k > 0 **then**
    compute and store $\Lambda_k = \frac{\|v_k - v_{k-1}\|}{\|\tilde{w}_k - \tilde{w}_{k-1}\|}$
  end if
  Set $w_0 = \tilde{w}_k$
  for $t = 0, \ldots, m-1$ **do**
    compute η_t^k as Algorithm 3.1,
    Randomly pick $i_t \in \{1, \ldots, n\}$
    $w_{t+1} = w_t - \eta_t^k(\nabla f_{i_t}(w_t) - \nabla f_{i_t}(\tilde{w}_k) + v_k)$
  end for
  $\tilde{w}_{k+1} = w_m$
end for

Remark 1 In order to reduce the computational cost, we can make $\delta_2 = 0$ in Algorithm 3.2, and calculate our adaptive stepsize in outer loops every epoch.

3.1.1 Convergence Analysis of SVRG-AS

We first prove the linear convergence of SVRG-I which is different from the convergence analysis in [17]. And then we give the convergence analysis of SVRG-AS.

Lemma 1 *Define*

$$\alpha_k := (1 - 2\eta(\mu - \eta L^2))^m + \frac{\eta L^2}{\mu - \eta L^2},$$

for both SVRG-I and SVRG-AS, we have

$$E[F(\tilde{w}_{k+1}) - F(w_*)] < \alpha_k [F(\tilde{w}_k) - F(w_*)].$$

Proof Let $v_t = \nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(\tilde{w}_k) + \nabla F(\tilde{w}_k)$ for the k-th epoch of SVRG-I or SVRG-AS. From Lemma 3 in Appendix, we obtain

$$E\,\|v_t\|_2^2 \leqslant 4L[F(w_{t-1}) - F(w_*) + F(\tilde{w}_k) - F(w_*)]. \tag{3.1}$$

Now by noticing that $Ev_t = \nabla F(w_{t-1})$, this leads to

$$
\begin{aligned}
&E[F(w_t) - F(w_*)] \\
&\leqslant F(w_{t-1}) - F(w_*) - \eta \nabla F(w_{t-1})^T E[v_t] + \frac{\eta^2 L}{2} E \left\| v_t \right\|^2 \\
&= F(w_{t-1}) - F(w_*) - \eta \left\| \nabla F(w_{t-1}) \right\|_2^2 + \frac{\eta^2 L}{2} E \left\| v_t \right\|^2 \\
&\leqslant F(w_{t-1}) - F(w_*) - 2\mu\eta[F(w_{t-1}) - F(w_*)] + 2\eta^2 L^2 [F(w_{t-1}) - F(w_*) + F(\tilde{w}_k) - F(w_*)] \\
&= [1 - 2\eta(\mu - \eta L^2)][F(w_{t-1}) - F(w_*)] + 2\eta^2 L^2 [F(\tilde{w}_k) - F(w_*)],
\end{aligned}
$$

where the first inequality uses the Lipschitz continuity of ∇F, the second inequality uses the strong convexity of F and the inequation (3.1).

Noting that $\tilde{w}_k = w_0$ and $\tilde{w}_{k+1} = w_m$, by recursively applying the above inequality over t, we have

$$
\begin{aligned}
&E[F(\tilde{w}_{k+1}) - F(w_*)] \\
&\leqslant [1 - 2\eta(\mu - \eta L^2)]^m [F(\tilde{w}_k) - F(w_*)] + 2\eta^2 L^2 \sum_{j=0}^{m-1} [1 - 2\eta(\mu - \eta L^2)][F(\tilde{w}_k) - F(w_*)] \\
&< [(1 - 2\eta(\mu - \eta L^2))^m + \frac{\eta L^2}{\mu - \eta L^2}][F(\tilde{w}_k) - F(w_*)].
\end{aligned}
$$

Corollary 1 *In SVRG-I, if m and η are chosen such that*

$$
\alpha_: = (1 - 2\eta(\mu - \eta L^2))^m + \frac{\eta L^2}{\mu - \eta L^2} < 1, \tag{3.2}
$$

then SVRG-I converges linearly in expectation:

$$
E[F(\tilde{w}_k) - F(w^*)] < \alpha^k [F(\tilde{w}_0) - F(w^*)].
$$

Theorem 4 *Denote $\theta_1 \in (0, 1)$. In SVRG-AS, if m is chosen such that*

$$
m^h > \frac{L^2}{\mu^2} + \frac{L^2}{\mu^2 \theta_1} \quad \textit{and} \quad m^{1-h} > \frac{\theta_1 \mu L + 2L^2}{2\mu^2},
$$

then SVRG-AS has linear convergence in expectation:

$$
E[F(\tilde{w}_k) - F(w^*)] < \tilde{\alpha}^k [F(\tilde{w}_0) - F(w^*)].
$$

Proof Using the strong convexity of function $F(w)$, it is easy to obtain the upper bound for our proposed stepsize,

$$
L_k = \sum\nolimits_{i=k-c+1}^{k} \beta_i \frac{\left\| \nabla F(w_i) - \nabla F(w_{i-1}) \right\|}{\left\| w_i - w_{i-1} \right\|} \geqslant \sum\nolimits_{i=k-c+1}^{k} \beta_i \mu = \mu,
$$

$$
\eta_t^k = \frac{1}{m^h \cdot L_k} \leqslant \frac{1}{m^h \mu}.
$$

Similarly, by the Lipschitz continuity of $\nabla F(w)$, we can obtain that η_t^k is uniformly lower bounded by $1/m^h L$. Therefore, we have

$$\begin{aligned}\tilde{\alpha} &\leqslant (1-\frac{2}{m^h L}(\mu-\frac{L^2}{m^h\mu}))^m+\frac{L^2}{m^h\mu^2-L^2}\\ &\leqslant \exp\{-\frac{2m^{1-h}}{L}(\mu-\frac{L^2}{m^h\mu})\}+\frac{L^2}{m^h\mu^2-L^2}.\end{aligned}$$

Let $\frac{L^2}{m^h\mu^2-L^2}<\theta_1$, we have $m^h>\frac{L^2}{\mu^2}+\frac{L^2}{\mu^2\theta_1}$. Then we render $\exp\{-\frac{2m^{1-h}}{L}(\mu-\frac{L^2}{m^h\mu})\}<1-\theta_1$, that is

$$\theta_1<1-\exp\{-\frac{2m^{1-h}}{L}(\mu-\frac{L^2}{m^h\mu})\}<\frac{2m^{1-h}}{L}(\mu-\frac{L^2}{m^h\mu}).$$

If $h=\frac{1}{2}$, we have $m^h>\frac{\theta_1\mu L+2L^2}{2\mu^2}$. It is obvious that $\frac{\theta_1\mu L+2L^2}{2\mu^2}<\frac{L^2}{\mu^2}+\frac{L^2}{\mu^2\theta_1}$. If $h>\frac{1}{2}$, let $\theta_1<\frac{2m^{1-h}\mu}{L}-\frac{2L}{\mu}<\frac{2m^{1-h}}{L}(\mu-\frac{L^2}{m^h\mu})$, then $m^{1-h}>\frac{\theta_1\mu L+2L^2}{2\mu^2}$. And then, we have

$$\tilde{\alpha}\leqslant\exp\{-\frac{2m^{1-h}}{L}(\mu-\frac{L^2}{m^h\mu})\}+\frac{L^2}{m^h\mu^2-L^2}<1-\theta_1+\theta_1=1.$$

Remark 2 In Theorem 4, it is obvious that when θ_1 decreases to the neighborhood of 0, m^h increases while m^{1-h} decreases. We all know that when m is too large, the optimization process of SVRG will oscillate, so we choose $h(\delta_1 k+\delta_2 t)\geqslant\frac{1}{2}$, then we can obtain appropriate m.

3.2 The SARAH-AS Method

We describe the SARAH-AS as Algorithm 3.3, it is similiar to SVRG-BB, except we calculate η_0^k in outer loop to update once for obtaining w_1 every epoch.

Remark 3 We can get rid of the calculation of η_t^k in Algorithm 3.3 and only use η_0^k every epoch for simplicity.

3.2.1 Convergence Analysis of SARAH-AS

We analyze the linear convergence of SARAH-AS (Algorithm 3.3) in this section. Firstly, we prove the linear convergence of SARAH-I (namely, SARAH with option I). Then we give the convergence analysis of SARAH-AS.

Under Assumption 1, let us define the optimal solution of (1.1) as w_*, then strong convexity of F implies that

$$2\mu[F(w)-F(w_*)]\leqslant\|\nabla F(w)\|^2,\quad\forall w\in\mathbb{R}^d. \tag{3.3}$$

Algorithm 3.3 The SARAH-AS Method

Parameters: update frequency m, initial stepsize $\eta_1 > 0$, $\tilde{w}_0$, a constant c, $\beta_i > 0, i \in \{1, \ldots, c\}$, δ_1, δ_2.

for $k = 1, 2, \ldots$ **do**
  $w_0 = \tilde{w}_{k-1}$
  $v_0^k = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(w_0)$
  if k > 1 **then**
    compute and store $\Lambda_k = \frac{\left\|v_0^k - v_0^{k-1}\right\|}{\left\|\tilde{w}_k - \tilde{w}_{k-1}\right\|}$
  end if
  compute $\eta_0^k = \frac{1}{m^{h(\delta_1 k)} L_k}$
  $w_1 = w_0 - \eta_0^k v_0^k$
  for $t = 1, \ldots, m-1$ **do**
    compute η_t^k as Algorithm 3.1.
    Randomly pick $i_t \in \{1, \ldots, n\}$
    $v_t^k = \nabla f_{i_t}(w_t) - \nabla f_{i_t}(w_{t-1}) + v_{t-1}^k$
    $w_{t+1} = w_t - \eta_t^k v_t^k$
  end for
  $\tilde{w}_k = w_m$
end for

Lemma 2 *Define*

$$\sigma_k := (1 - \mu\eta)^m + \frac{\eta L^2}{\mu(2 - \eta L)}, \tag{3.4}$$

for both SARAH-I and SARAH-AS with $\eta \leqslant \frac{1}{L}$, we have the following inequality for the k-th epoch:

$$E[F(\tilde{w}_{k+1}) - F(w^*)] \leqslant \sigma_k [F(\tilde{w}_k) - F(w^*)].$$

Proof In the k-th epoch of SARAH-I or SARAH-AS, for the t-th inner loop, we have

$$\begin{aligned}
&E[F(w_t) - F(w_*)] \\
&\leqslant E[F(w_{t-1}) - F(w_*)] - \eta E[\nabla F(w_{t-1})^T v_t^k] + \frac{L\eta^2}{2} E[\left\|v_t^k\right\|^2] \\
&= E[F(w_{t-1}) - F(w_*)] - \frac{\eta}{2} E[\|\nabla F(w_{t-1})\|^2] + \frac{\eta}{2} E[\left\|\nabla F(w^t) - v_t^k\right\|^2] - (\frac{\eta}{2} - \frac{L\eta^2}{2}) E[\left\|v_t^k\right\|^2] \\
&\leqslant E[F(w_{t-1}) - F(w_*)] - \mu\eta E[F(w_{t-1}) - F(w_*)] + \frac{\eta}{2} \cdot \frac{\eta L}{2 - \eta L} E[\|\nabla F(w_0)\|^2] \\
&\leqslant E[F(w_{t-1}) - F(w_*)] - \mu\eta E[F(w_{t-1}) - F(w_*)] + \frac{\eta^2 L^2}{2 - \eta L} E[F(w_0) - F(w_*)] \\
&= (1 - \mu\eta)\, E[F(w_{t-1}) - F(w_*)] + \frac{\eta^2 L^2}{2 - \eta L} E[F(w_0) - F(w_*)],
\end{aligned}$$

where the first inequality and the fourth inequality we applied the Lipschitz continuity of ∇F. The second equality follows from the fact $a^T b = \frac{1}{2}[\|a\|^2 + \|b\|^2 - \|a - b\|^2]$. The third inequality uses the strong convexity of F as (3.3) and Lemma 5 in Appendix.

By recursively applying the above inequality over t, and noting that $\tilde{w}_k = w_0$ and $\tilde{w}_{k+1} = w_m$, we can obtain

$$\begin{aligned}
&E[F(\tilde{w}_{k+1}) - F(w^*)] \\
&\leqslant (1-\mu\eta)^m [F(\tilde{w}_k) - F(w^*)] + \frac{\eta^2 L^2}{2-\eta L} \sum_{j=0}^{m-1} (1-\mu\eta)^j [F(\tilde{w}_k) - F(w^*)] \\
&\leqslant \left((1-\mu\eta)^m + \frac{\eta L^2}{\mu(2-\eta L)} \right) [F(\tilde{w}_k) - F(w^*)] \\
&= \sigma_k [F(\tilde{w}_k) - F(w^*)].
\end{aligned}$$

Then we have the linear convergence of SARAH-I.

Corollary 2 *In SARAH-I, if m and $\eta \leqslant \frac{1}{L}$ are chosen such that*

$$\sigma := (1-\mu\eta)^m + \frac{\eta L^2}{\mu(2-\eta L)} < 1, \tag{3.5}$$

then SARAH-I converges linearly in expectation:

$$E[F(\tilde{w}_k) - F(w^*)] < \sigma^k [F(\tilde{w}_0) - F(w^*)].$$

.

Theorem 5 *Given $0 < \theta_2 < 1$. In SARAH-AS, if m is chosen such that*

$$m^h > \frac{L^2}{2\mu^2\theta_2} + \frac{L}{2\mu} \quad \textit{and} \quad m^{1-h} > \frac{\theta_2 L}{\mu},$$

then SARAH-AS (Algorithm 3.3) converges linearly in expectation:

$$E[F(\tilde{w}_k) - F(w^*)] < \tilde{\sigma}^k [F(\tilde{w}_0) - F(w^*)].$$

Proof We know that $\frac{1}{m^h L} \leqslant \eta \leqslant \frac{1}{m^h \mu}$, then we have,

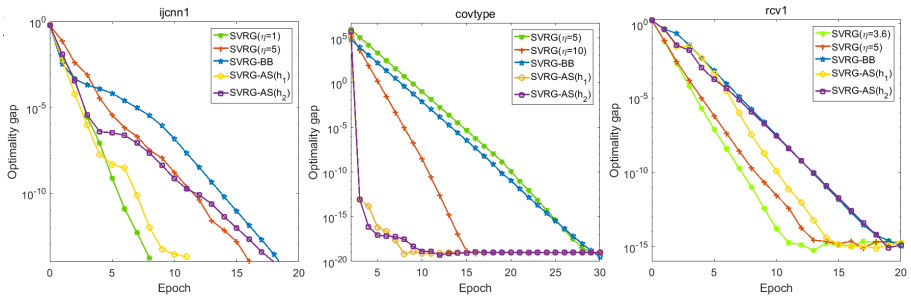
$$\begin{aligned}
\sigma &\leqslant \left(1 - \frac{\mu}{m^h L}\right)^m + \frac{L^2}{m^h\mu^2(2 - L/m^h\mu)} \\
&\leqslant \exp\{-\frac{m^{1-h}\mu}{L}\} + \frac{L^2}{m^h\mu^2(2 - L/m^h\mu)} \\
&= \exp\{-\frac{m^{1-h}\mu}{L}\} + \frac{L^2}{2m^h\mu^2 - L\mu}.
\end{aligned}$$

Let $\exp\{-\frac{m^{1-h}\mu}{L}\} < 1-\theta_2$, then $\theta_2 < 1 - \exp\{-\frac{m^{1-h}\mu}{L}\} < \frac{m^{1-h}\mu}{L}$, namely, $m^{1-h} > \frac{\theta_2 L}{\mu}$.

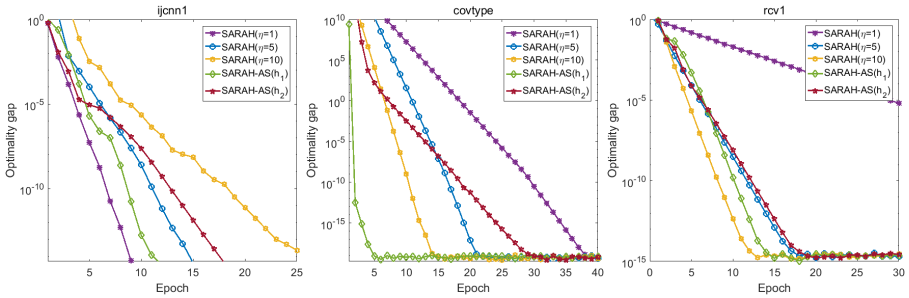
Let $\frac{L^2}{2m^h\mu^2 - L\mu} < \theta_2$, we have $m^h > \frac{L^2}{2\mu^2\theta_2} + \frac{L}{2\mu}$. So,

$$\sigma \leqslant \exp\{-\frac{m\mu}{m^h L}\} + \frac{L^2}{2m^h\mu^2 - L\mu} < 1 - \theta_2 + \theta_2 < 1.$$

Remark 4 It is observed from Theorem 5 that $m^h \sim O(\frac{L^2}{\mu^2})$ and $m^{1-h} \sim O(\frac{L}{\mu})$ when θ_2 is close to 1. To get the smaller m, this suggests that $h \geqslant 1-h$, namely, the value of function h should be no less than $\frac{1}{2}$.



(a) SVRG on ijcnn1 (b) SVRG on covtype (c) SVRG on rcv1



(d) SARAH on ijcnn1 (e) SARAH on covtype (f) SARAH on rcv1

Fig. 1: The upper is our adaptive stepsize for SVRG with different h (h_1 and h_2), denoted as SVRG-AS(h_1) or SVRG-AS(h_2), respectively, and SVRG-BB, SVRG with some constant stepsize. And the under is the same tests on SARAH, named SARAH-AS(h_1) or SARAH-AS(h_2).

4 Numerical Experiments

In this section, we present some numerical experiments to demonstrate the efficacy of SVRG-AS (Algorithm 3.2), SARAH-AS (Algorithm 3.3) and some SVRG's variants with our adaptive stepsize for ℓ_2-regularized logistic regression problem with

$$f_i(w) = \log(1 + \exp(-y_i x_i^T w)) + \frac{\lambda}{2}\|w\|^2$$

on datasets *covtype*, *ijcnn1*, *rcv1* as Table1[1].

Table 1: Data information

Dataset	*ijcnn1*	*covtype*	*rcv1*
n	49990	581012	20242
d	22	54	47236

The penalty parameter λ is set to $1/n$ as is common in practice[11][25]. We give some specific functions of h as follow

$$h_1(x) = \frac{x}{1+x} \quad or \quad h_2(x) = sigmoid(x-1) \quad x \in \mathbb{N}_+,$$

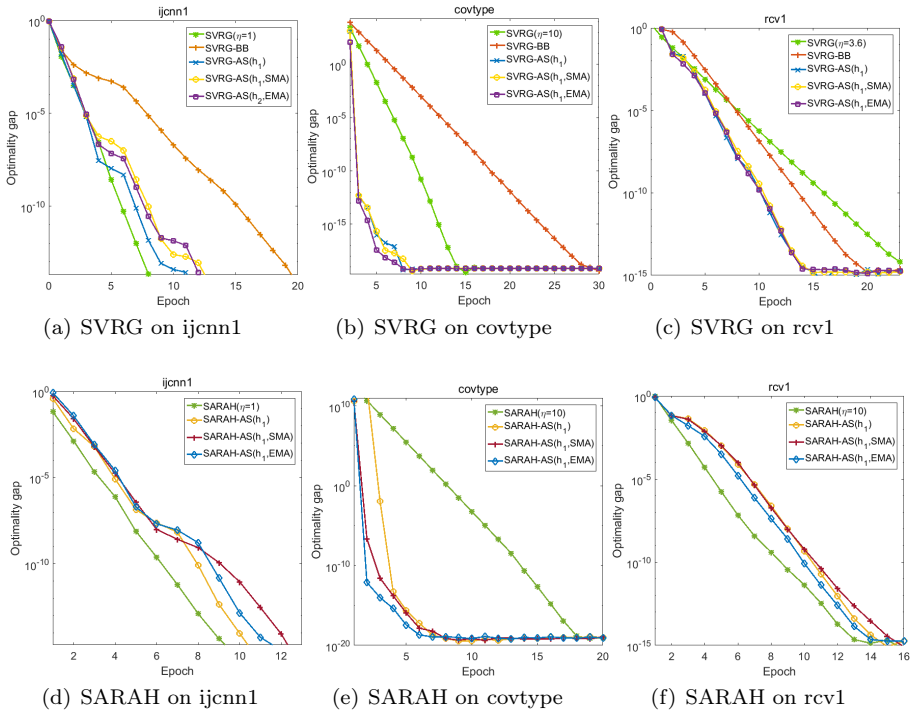
[1] All datasets are available at https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/

and we exhibit the concrete form of moving average, such as Simple Moving Average (SMA) and Exponential Moving Average (EMA).

$$\text{SMA}: L_{\text{k}} = \frac{1}{c}\sum_{i=k-c+1}^{k} \Lambda_i,$$

$$\text{EMA}: L_{\text{k}} = \rho[\Lambda_k + (1-\rho)\Lambda_{k-1} + \cdots + (1-\rho)^{c-1}\Lambda_{k-c+1}] + (1-\rho)^c \Lambda_k.$$

Our adaptive stepsize performs well even when only using the current local estimation of L in experiments.



(a) SVRG on ijcnn1 (b) SVRG on covtype (c) SVRG on rcv1

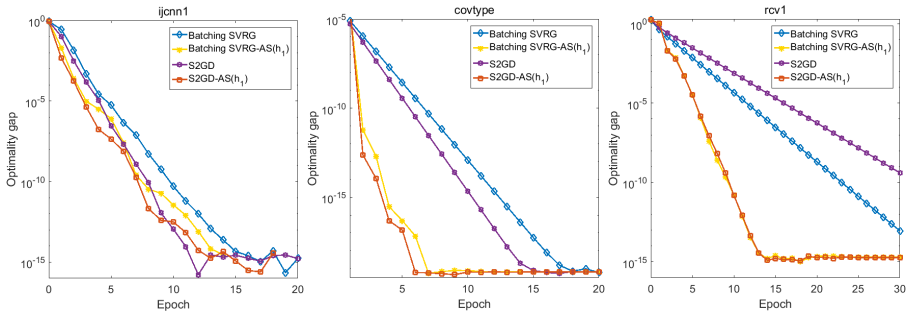
(d) SARAH on ijcnn1 (e) SARAH on covtype (f) SARAH on rcv1

Fig. 2: This figure show that different moving average for our adaptive stepsize. The upper shows that how SMA and EMA affect the performance of our proposed stepsize on SVRG, and the under is on SARAH.

Fig. 1, Fig. 2 and Fig. 3 show numerical results in terms of optimality gap $F(w) - F(w_*)$ on the datasets, the x-axis denotes the number of epochs. In Fig. 1, we test for different function h: h_1 and h_2. We choose $\eta = 1$ as the initial stepsize, and choose some constant stepsize for comparison. In all experiments $m = O(n)$, $\delta_1 = 1$, $\delta_2 = 0$. And we just use the current local estimation of L in Fig. 1. The results suggest that h_1 is better than h_2 both for SVRG and SARAH, and our proposed stepsize performs better than BB stepsize for SVRG, even better than best-tuned stepsizes for some dataset.

Fig. 2 shows that how different moving average strategies affect the performance of our adaptive stepsize. We choose the best constant stepsize and h_1 for comparing with SMA and EMA. We make $c = 2$ and the decay rate $\rho = 0.9$ for EMA. Experiments show that there is no major changes for different moving average, but EMA is better than SMA on most datasets.

In Fig. 3, we do some numerical experiments to demonstrate that our adaptive stepsize is also suitable for some variants of SVRG, such as Batching SVRG[25] and S2GD[26]. We set $\left|B^k\right| = 2^k$ for Batching SVRG. In S2GD, we just set the max inner loop as n, and $v = \frac{1}{n}$ for the geometric law. And we compute η in the outer loops for S2GD, namely $\delta_2 = 0$, because the calculation of the number of inner loop needs η. It is obviously that our adaptive stepsize is effective.



(a) ijcnn1 (b) covtype (c) rcv1

Fig. 3: Comparison of our adaptive stepsize for Batching SVRG and S2GD with Batching SVRG and S2GD with fixed stepsize on different datasets.

5 Conclusion

In this paper, we propose an adaptive stepsize for stochastic variance reduced methods, SVRG and SARAH. And this stepsize is based on the local estimation of Lipschitz constant. Numerical results indicate that our proposed stepsize works well in practice, and we prove the linear convergence of SVRG-AS and SARAH-AS, respectively. But we just give a general form of our stepsize, how to choose the most suitable parameter will be our future work. For example, the best choice of function of the number of outer and inner loop, and the best choice of moving average variations.

A Appendix

Lemma 3 *Let $v_t = \nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(\tilde{w}_k) + \nabla F(\tilde{w}_k)$. Conditioned on w_{t-1}, we can take expectation with respect to i_t, and obtain:*

$$\begin{aligned} E\left\|v_t\right\|^2 &= E\left\|\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(w_*) - (\nabla f_{i_t}(\tilde{w}_k) - \nabla f_{i_t}(w_*) - \nabla F(\tilde{w}_k))\right\|^2 \\ &\leqslant 2E\left\|\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(w_*)\right\|^2 + 2E\left\|\nabla f_{i_t}(\tilde{w}_k) - \nabla f_{i_t}(w_*) - \nabla F(\tilde{w}_k)\right\|^2 \\ &= 2E\left\|\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(w_*)\right\|^2 + 2E\left\|\nabla f_{i_t}(\tilde{w}_k) - \nabla f_{i_t}(w_*)\right\|^2 \\ &\leqslant 2E\left\|\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(w_*)\right\|^2 + 2E\left\|\nabla f_{i_t}(\tilde{w}_k) - \nabla f_{i_t}(w_*)\right\|^2 \\ &\leqslant 4L[F(w_{t-1}) - F(w_*) + F(\tilde{w}_k) - F(w_*)] \end{aligned}$$

Lemma 4 *Suppose that f is convex and its gradient is L-Lipschitz continuous. Then, for any $w, w' \in \mathbb{R}^d$,*

$$f(w) \leqslant f(w') + \nabla f(w')^T(w - w') + \frac{L}{2}\left\|w - w'\right\|^2$$

$$f(w) \geqslant f(w') + \nabla f(w')^T(w - w') + \frac{1}{2L}\left\|\nabla f(w) - \nabla f(w')\right\|^2$$

Lemma 5 *If v_t defined as 2.6 in SARAH (Algorithm 2.2) with $\eta < \frac{2}{L}$. Then we have that for any $t > 1$,*

$$E[\left\|\nabla F(w^t) - v_t^k\right\|^2] \leqslant \frac{\eta L}{2 - \eta L}[E[\left\|v_0^k\right\|^2] - E[\left\|v_t^k\right\|^2]] \leqslant \frac{\eta L}{2 - \eta L}E[\left\|v_0^k\right\|^2]$$

The proof of Lemma 5 can be found in [11].

References

1. Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Le, Q.V., Ng, A.Y.: On optimization methods for deep learning. In: International Conference on Machine Learning, pp. 265-272 (2011)
2. Robbins, H., Monro, S.: A stochastic approximation method. Annals of Mathematical Statistics, 22(3), 400-407 (1951)
3. Bousquet, O., Bottou, L.: The tradeoffs of large scale learning. In: Neural Information Processing Systems, pp. 161-168 (2007)
4. Bottou, L.: Stochastic gradient learning in neural networks. In: Neuro Nimes, (2007)
5. Bottou, L., Curtis, F. E., Nocedal, J.: Optimization methods for large-scale machine learning. SIAM Review, 60(2), 223-311 (2018)
6. Dai, Y. H., Liao, L. Z.: R-linear convergence of the Barzilai and Borwein gradient method. IMA Journal of Numerical Analysis, 22(1), 1-10 (2002)
7. Roux, N. L., Schmidt, M., Bach, F. R.: A stochastic gradient method with an exponential convergence rate for finite training sets. In: Neural Information Processing Systems, pp. 2663-2671 (2013)
8. Schmidt, M.W., Roux, N.L., Bach, F.: Minimizing finite sums with the stochastic average gradient. Mathematical Programming, 162(1), 83-112 (2017)
9. Defazio, A., Bach, F., Lacoste-Julien, S.: SAGA: a fast incremental gradient method with support for non-strongly convex composite objectives. In: Neural Information Processing Systems, pp. 1646-1654 (2014)
10. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Neural Information Processing Systems, pp. 315-323 (2013)
11. Nguyen, L.M., Liu, J., Scheinberg, K., Takac, M.: SARAH: a novel method for machine learning problems using stochastic recursive gradient. In: Neural Information Processing Systems, pp. 2613-2621 (2017).
12. Lucchi, A., McWilliams, B., Hofmann, T.: A variance reduced stochastic Newton method. arXiv: 1503.08316. (2015)

13. Moritz, P., Nishihara, R., Jordan, M.: A linearly-convergent stochastic L-BFGS algorithm. In: Artificial Intelligence and Statistics, pp. 249-258 (2016)
14. Gower, R., Goldfarb, D., Richtárik, P.: Stochastic block BFGS: squeezing more curvature out of data. In: International Conference on Machine Learning, pp. 1869-1878 (2016)
15. Duchi, J.C., Hazan, E., Singer, Y.J.: Adaptive subgradient methods for online learning and stochastic optimization, In: Journal of Machine Learning Research, pp. 2121-2159, (2011)
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations, pp. 1-13 (2015)
17. Tan, C., Ma, S., Dai, Y., Qian, Y.: Barzilai-Borwein step size for stochastic gradient descent. In: Neural Information Processing Systems, pp. 685-693 (2016)
18. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA Journal of Numerical Analysis, 8(1), 141-148 (1988)
19. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM Journal on Optimization 7(1), 26–33 (1997).
20. Zhou, C., Gao, W., Goldfarb, D.: Stochastic adaptive quasi-Newton methods for minimizing expected values. In: International Conference on Machine Learning, pp. 4150-4159 (2017)
21. Gao, W., Goldfarb, D.: Quasi-newton methods: superlinear convergence without line searches for self-concordant functions. Optimization Methods and Software, 1-24 (2018)
22. De, S., Yadav, A.K., Jacobs, D.W., Goldstein, T.: Automated inference with adaptive batches. In: International Conference on Artificial Intelligence and Statistics, pp. 1504-1513 (2017)
23. Armijo, L.: Minimization of functions having lipschitz conditions first partial derivatives, Pacific J Math, 16(1), 1-3 (1966)
24. Vrahatis, M.N., Androulakis, G.S., Lambrinos, J.N., Magoulas, G.D.: A class of gradient unconstrained minimization algorithms with adaptive stepsize. Journal of Computational & Applied Mathematics, 114(2), 367-386 (2000)
25. Babanezhad, R., Ahmed, M.O., Virani, A., Schmidt, M.W., Konecný, J., Sallinen, S.: Stop wasting my gradients: practical SVRG. In: Neural Information Processing Systems, pp. 2251-2259 (2015)
26. Konecný, J., Richtarik, P.: Semi-stochastic gradient descent methods. Frontiers in Applied Mathematics and Statistics, 3, 9 (2017)