Analysis of the Variance Reduction in SVRG and a New Acceleration Method

Abstract

Stochastic gradient descent (SGD) with variance reduction technique such as SVRG is efficient to train parameters of many machine learning models. Although many variants of SVRG have been proposed, the analysis of variance has not been thoroughly discussed. Besides, the variants of SVRG have to keep a snapshot of the full gradient in every epoch, which is computationally expensive. In this paper, we propose a framework EUI which is an abstraction of the existing variants of SVRG, and then provide a general and deep analysis of the variance from a new perspective. Moreover, a new variant of SGD with the variance reduction technique named SAMPLEVR is proposed. SAMPLEVR replaces the full gradient computation with an estimation, thus decreasing gradient complexity significantly. Both the theoretical analysis and the empirical studies show that SAMPLEVR makes training loss converge faster than its counterparts significantly.

Introduction

Many machine learning tasks such as classification and regression can be presented as solving the optimisation problem described by Equation 1. $F(\omega)$ denotes the training loss or the loss function which is the sum of a finite number of functions, i.e. $\nabla f_i(\omega)$ with $i \in \{1, 2, ..., n\}$. ω is the parameter of the machine learning model, and n represents the size of the training data. $R(\omega)$ is the regulariser which is used to prevent overfitting.

$$\min F(\omega), \qquad F(\omega) = \frac{1}{n}\sum_{i=1}^{n} f_i(\omega) + R(\omega) \tag{1}$$

Gradient descent (GD) is used to train the parameters for such underlying machine learning problems. Since GD computes a full gradient every iteration, it performs a large amount of gradient calculations. This would affect the performance significantly in the presence of a large amount of training data. The stochastic gradient descent (SGD) improves the time efficiency by using a stochastic gradient instead of the full gradient to train parameter. However, variance caused by the stochastic gradient usually impairs convergence of the training loss. Specifically, when the parameter is close to the optimum, it is increasingly difficult to make a further progress due to the variance. Conventional studies show that a decaying learning rate can be used to decrease the variance. But the training loss converges slowly when the learning rate is small.

Recently, Johnson & Zhang improve SGD with the variance reduction technique named SVRG which uses a constant learning rate to train the parameter (Johnson and Zhang 2013). Based on the variance reduction technique adopted by SVRG, many variants of SVRG such as S2GD (Konečnỳ and Richtárik 2013), mS2GD (Konečnỳ et al. 2016), EMGD (Zhang, Mahdavi, and Jin 2013), SVR-GHT (Li et al. 2016), Prox-SVRG (Xiao and Zhang 2014), SVRG++ (Allen-Zhu and Yuan 2016), and Katyusha (Allen-Zhu 2016) have been proposed. However, the analysis of the variance, which is essential to understand and exploit the variance reduction technique, lacks enough discussion. Although Allen-Zhu & Hazan present some impressive analysis of the variance, such technical analysis is obtained under the specific settings of an algorithm, which lacks the generality. Moreover, the analysis only provides an upper bound of the variance (Allen-Zhu and Hazan 2016), which lacks analysis of the lower bound. Besides, those existing algorithms are organised by epochs. One epoch consists of some iterations. SVRG and its variants have to keep a snapshot of the full gradient for every epoch, leading to a large amount of gradient calculations. When the size of the training data is huge, the snapshot of the full gradient is extremely time-consuming. In short, it is meaningful to give a quantitative analysis of the variance in general settings, including the upper and lower bounds. Additionally, if the snapshot of the full gradient in an epoch can be avoided, SVRG and its variants will be accelerated a lot, which results in a better performance of the convergence.

As an important improvement of SVRG and its variants, we provide a thorough analysis about the variance from a new prospective. To present the analysis, we propose a general framework named EUI, and then perform the analysis under the framework. The update rule of the parameter in the variance reduction technique can be divided into three parts, including the variance source, the variance reducer and the progressive direction. The variance reducer is the real reason to reduce the variance. More specifically, we provide both lower and upper bounds of the variance, and then analyse improvement of the variance reduction in the existing

variants of SVRG. Moreover, a new variant of SGD with the variance reduction technique denoted by SAMPLEVR is proposed, which replaces the snapshot of the full gradient by using an unbiased estimation, and accelerates the convergence of the training loss. The contributions of the paper are outlined as follows:

- The variance caused by the stochastic gradient is analysed in a framework, EUI. Both the lower and upper bounds of the variance are presented in general settings.
- A new variant of SGD with the variance reduction technique, i.e. SAMPLEVR is proposed, which achieves a linear convergence rate with low gradient complexity.
- Extensive evaluation tests show that SAMPLEVR outperforms other previous work significantly.

To keep the paper concise, a list of symbols used in the paper and the proofs are given in support materials. The paper is organised as follows. First, we review recent related work about SGD with the variance reduction technique. Second, we present the general framework, i.e. EUI, and then provide the analysis of the variance reduction under the framework. After that, we present a new reduced variance SGD, i.e. SAMPLEVR, and provide the theoretical analysis of both the performance of convergence and the gradient complexity. Furthermore, we demonstrate the extensive performance evaluations to verify the theoretical analysis. Finally, we conclude this paper.

Related work

Various variants of SGD with the variance reduction technique have been proposed, including SAG (Schmidt, Roux, and Bach 2016), SAGA (Defazio, Bach, and Lacoste-Julien 2014), SDCA (Shalev-Shwartz 2016), and SVRG (Johnson and Zhang 2013) and its variants and so on. It is noting that SAG, SAGA and SDCA adopt different variance reduction techniques from SVRG and its variants. Although variance reduction techniques used in SAG, SAGA and SDCA are competitive with that of SVRG and its variants, this paper focuses on the variance reduction technique used in SVRG and its variants. To the best of our knowledge, the variants of SVRG at least includes S2GD (Konečnỳ and Richtárik 2013), mS2GD (Konečnỳ et al. 2016), EMGD (Zhang, Mahdavi, and Jin 2013), SVR-GHT (Li et al. 2016), Prox-SVRG (Xiao and Zhang 2014), SVRG++ (Allen-Zhu and Yuan 2016), and CHEAPSVRG (Shah et al. 2016).

Allen-Zhu & Hazan have presented an upper bound of the variance caused by the stochastic gradients when using the variance reduction technique (Allen-Zhu and Hazan 2016). However, such analysis is obtained for a specific algorithm, and lacks the lower bound of the variance. Our analysis about the variance is a complement to that work, which is obtained in general settings of algorithms, and provides the lower bound of the variance. Shah et al. have proposed CHEAPSVRG which uses sampled instances to estimate the full gradient (Shah et al. 2016). The number of those sampled instances is a parameter which needs to be identified before running of CHEAPSVRG. Comparing with CHEAPSVRG, our proposed algorithm, i.e. SAMPLEVR does not need to tune extra parameters. Additionally, although both CHEAPSVRG and SAMPLEVR can achieve a linear convergence rate, the analysis of CHEAPSVRG is obtained with two extra strong assumptions, which are not required in SAMPLEVR. Furthermore, extensive empirical studies show that SAMPLEVR outperforms CHEAPSVRG significantly.

Algorithm 1 EUI: the general framework of reduced variance SGD

Require: $\omega_0{=}\tilde{\omega}_0{=}\mathbf{0}$. $\forall i_t{\in}\{1,2,...,n\}$ where t is a non-negative integer.

Epoch: identify the sequence of epoch size $\{m_0, m_1, ..., m_S\}{\leftarrow}\mathcal{E}(s)$ with $s \in \{0, 1, ..., S\}$;
for $s = 0, 1, 2, ..., S-1$ **do**
  $\omega_0 = \tilde{\omega}_s$;
  $g = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{\omega}_s)$;
  for $t = 0, ..., m_s{-}1$ **do**
    pick an instance $<x_{i_t}, y_{i_t}>$ randomly;
    $\gamma_t = \nabla f_{i_t}(\omega_{i_t}) - \nabla f_{i_t}(\tilde{\omega}_s) + g$;
    Update: $\omega_{t+1} = \mathcal{U}(\eta, \omega_t, \gamma_t)$;
  Identify: $\tilde{\omega}_{s+1}{\leftarrow}\mathcal{I}(\omega_j)$ with $j{\in}\{0, 1, ..., m_s\}$;
return $\tilde{\omega}_S$.

Analysis of variance in a general framework

Framework

As illustrated in Algorithm 1, we present a general framework named EUI which contains a loop of epochs. Every epoch consists of a number of iterations. The epoch size needs to be identified, as calculated by the function $\mathcal{E}$. After that, a snapshot of the full gradient is computed at the beginning of an epoch. During the training of the parameter in an epoch, EUI randomly picks an instance x_{i_t} from the training data first. The parameter is then trained by the update rule, as denoted by the function $\mathcal{U}$. At the end of an epoch, the global parameter $\tilde{\omega}_s$ will be identified by the local parameter ω_t with $t \in \{0, 1, ..., m_s\}$, which is shown by the function $\mathcal{I}$.

Table 1 illustrates that SVRG and its variants can be unified by EUI when the functions, i.e. $\mathcal{E}$, $\mathcal{U}$, and $\mathcal{I}$ are implemented. For example, the function $\mathcal{E}$ in SVRG is implemented by a constant, that is, $m_s{=}m$ with $s{=}\{0, 1, ..., S{-}1\}$. The function $\mathcal{U}$ in SVRG is implemented by the steepest descent, and the function $\mathcal{I}$ is implemented by either any of the local parameters, i.e. ω_j with $j \in \{0, 1, ..., m_s{-}1\}$ or ω_{m_s}. Compared to SVRG, its variants implement those functions by using different strategies. Those strategies are explained with the analysis of the variance reduction in the following part.

The analysis of the variance

As illustrated in Table 1, the variants of SVRG usually adopt the same update rule of the parameter with SVRG. The update rule is shown by Equ. 3. We take SVRG as an example to present the analysis of the variance, and the improvement of variance reduction in other existing algorithms.

Table 1: Variants of SVRG can be unified by the general framework EUI when the functions $\mathcal{E}, \mathcal{U}$ and $\mathcal{I}$ are implemented.

Name	Algorithms					
	SVRG	S2GD	EMGD	SVR-GHT	Prox-SVRG	SVRG++
$\mathcal{E}$	$m_s{=}m$	$P(m_s{=}t){=}\frac{\phi(t)}{\sum_{t=1}^{m}\phi(t)}$ †	$m_s{=}m$	$m_s{=}m$	$m_s{=}m$	$m_s{=}2^s m$
$\mathcal{U}$	$\omega_t{-}\eta\gamma_t$	$\omega_t{-}\eta\gamma_t$	$\omega_t{-}\mathbb{B}_{\Delta_s}(\eta\gamma_t)$	$\mathcal{H}_\kappa(\omega_t{-}\eta\gamma_t)$	$\omega_t{-}\eta\gamma_t$ ‡	$\omega_t{-}\eta\gamma_t$ ‡
$\mathcal{I}$	randomly pick any of ω_j with $j{\in}\{0,1,...,m_s{-}1\}$	ω_{m_s}	$\frac{1}{m_s+1}\sum_{i=0}^{m_s}\omega_i$	ω_{m_s}	$\frac{1}{m_s}\sum_{i=0}^{m_s-1}\omega_i$	$\frac{1}{m_s}\sum_{i=0}^{m_s-1}\omega_i$
	ω_{m_s}					

† $\phi(t){=}(1{-}\mu\eta)^{m-t}$. $P(m_s{=}t)$ means the probability of $m_s{=}t$, which shows that a large epoch size is used with a high probability.
‡ The update rules of Prox-SVRG and SVRG++ are presented in the setting of the differentiable optimisation objective.

As illustrated in Equ. 3, the first item of γ_t is the stochastic gradient which is denoted by the "variance source". The second item of γ_t is denoted by the "variance reducer" which is used to reduce the variance. The third item of γ_t is denoted by the "progressive direction" which keeps γ_t not too far away from the full gradient. The update rule of the parameter in SGD and GD are denoted by γ^{SGD} and γ^{GD}, respectively. It is noting that γ^{SGD} and γ^{GD} are used to show the reason of the variance, and potential of the variance reduction we have exploited. They cannot be expressed by the general framework. We refer the variance of SGD to the maximum, and the variance of GD to the minimum.

$$\gamma^{\text{SGD}} = \nabla f_{i_t}(\omega_{i_t}) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\omega}) + \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\omega}); \quad (2)$$

$$\gamma_t = \nabla f_{i_t}(\omega_{i_t}) - \nabla f_{i_t}(\tilde{\omega}) + \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\omega}); \quad (3)$$

$$\gamma^{\text{GD}} = \nabla f_{i_t}(\omega_{i_t}) - \nabla f_{i_t}(\omega_{i_t}) + \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\omega}); \quad (4)$$

It is obvious that the difference among the update rule of SGD, GD and SVRG is the variance reducer. SGD causes the maximal variance because its variance reducer is a constant, which does not help to reduce the variance. GD does not lead to variance because that its variance reducer decreases all the variance caused by the variance source. The variance reducer in SVRG is a tradeoff between those of SGD and GD. It does not reduce all the variance like that of GD. The reason is that its input parameter, i.e. $\tilde{\omega}$ becomes stale against ω_t during the iterations in an epoch. The variance due to the staleness will be accumulated with the iterative updates of the parameter ω_t. Such the staleness of the parameter can be measured by the distance d_t with $d_t = \| \omega_t{-}\tilde{\omega} \|^2$. $d_0 = \| \omega_0{-}\tilde{\omega} \|^2 = 0$ holds according to the framework. If γ_t is p-dimensional, and can be denoted by $\gamma_t = (a_{t1}, a_{t2}..., a_{tp})$, we obtain Theorem 1 as follows, which is of obvious significance to the analysis of the variance reduction technique.

Theorem 1. *After t iterations in an epoch, the distance d_t holds that $d_t{=}\eta^2\sum_{j=1}^{p}\left(\sum_{i=1}^{t}a_{ij}\right)^2$. Furthermore, d_t has an upper bound such that $d_t{\le}\eta^2t^2p\left(\frac{1}{tp}\sum_{i=1}^{t}\sum_{j=1}^{p}a_{ij}^2\right)$, and a lower bound such that $d_t{\ge}\eta^2t^2p\left(\frac{1}{tp}\sum_{i=1}^{t}\sum_{j=1}^{p}a_{ij}\right)^2$.*

First, the upper bound and the lower bound are obtained in general settings of the loss functions, including convex or non-convex cases. The results are suitable to the various machine learning models if those models are trained by using the algorithms expressed by the general framework. Although some previous results have made impressive achievements (Shalev-Shwartz 2016), (Garber and Hazan 2015), (Allen-Zhu and Hazan 2016), our analysis outperforms them because of generality and the concise analysis. Additionally, the lower bound of the variance is provided which is superior to the previous results.

Second, the result is effective to analyse the variance of the algorithms which are expressed by the framework. For example, as illustrated in Table 1, the epoch size, i.e. m_s is designed as a constant in EMGD, SVR-GHT and Prox-SVRG, and an ascending variable for S2GD and SVRG++. Considering EMGD, $\| \gamma_t \|^2 \le \frac{\Delta_0}{2^{s-1}}$ holds and $d_t{\le}\eta^2t^2p\left(\frac{1}{tp}\sum_{i=0}^{t}\frac{\Delta_0}{2^{s-1}}\right){\le}\eta^2t^2\frac{\Delta_0}{2^{s-1}}$, which is decreased with the number of epochs exponentially. The hard-thresholding mechanism in SVR-GHT keeps the κ-largest elements and sets others to be zero. Without loss of generality, suppose that the elements a_{tj} with $j{\in}\{1,2,...,\kappa\}$ is the κ-largest elements. Thus, $d_t{\le}\eta^2t^2p\left(\frac{1}{tp}\sum_{i=0}^{t}\sum_{j=1}^{\kappa}a_{ij}^2\right)$ holds, which is smaller than the variance in SVRG. Besides, taking the expectation of t in S2GD, we obtain $\mathbb{E}(t) = \frac{\phi(t)m}{\sum_{t=1}^{m}\phi(t)}{\le}m$ which is smaller than that of SVRG significantly. SVRG++ increases the epoch size exponentially, and thus the variance grows fast.

Third, the result provides a guide to design a new variant of SGD with the variance reduction technique. Based on the analysis, we can implement the functions $\mathcal{E}, \mathcal{U}$, and $\mathcal{I}$ by using a dynamic method. Such the flexibility is superior to the previous work. For instance, as demonstrated in Theorem 1, the variance becomes large with a large learning rate η, a high dimension p, and the iterative updates of parameters. Given a specific machine learning task, we can dynamically set the learning rate and the epoch size based on the variance we can tolerate. Besides, the variance in the current epoch will be passed to the next epoch via the identification

Algorithm 2 SAMPLEVR

Require: $\alpha = 0.01, \dot{g} = \mathbf{0}$, and $\tilde{\omega}_0 = \mathbf{0}$. $\forall i_t \in \{1, 2, ..., n\}$. ϵ is a positive real number.

for $s = 0, 1, 2, ..., S-1$ **do**
  $\omega_0 = \tilde{\omega}_s$;
  for $t = 0, 1, ..., m-1$ **do**
    pick an instance $<x_{i_t}, y_{i_t}>$ randomly;
    $\gamma_t = \nabla f_{i_t}(\omega_t) - \nabla f_{i_t}(\tilde{\omega}_s) + \dot{g}$;
    $\omega_{t+1} = \omega_t - \eta\gamma_t$;
  $\tilde{\omega}_{s+1}$ is identified by using any of ω_i with $i \in \{0, 2, ..., m-1\}$ randomly;
  $k = -\frac{s \log \frac{\alpha}{2}}{\epsilon}$;
  $\dot{g} = \frac{1}{k} \sum_{j=1}^{k} \nabla f_{i_j}(\tilde{\omega}_{s+1})$;
return $\tilde{\omega}_S$.

of the parameters. As illustrated in Table 1, the majority of previous work use ω_{m_s} as the initial parameter of the next epoch, which contains all the updates of the current epoch, but leads to much variance to the next epoch. EMGD, ProxSVRG and SVRG++ use the mean of the local parameters which leads to less variance, but discards some updates of the parameters. We can dynamically adjust those strategies based on the analysis. That is, when the variance is small, ω_{m_s} is used. Otherwise, we use the mean of the parameters to identified the parameters.

SAMPLEVR: a new reduced variance SGD

Although the variance reduction technique is effective for decreasing the variance caused by the stochastic gradient, it has to keep a snapshot of the full gradient every epoch. Unfortunately, the computation of the full gradient requires extensive gradient calculations which are extremely time-consuming. We design a new reduced variance SGD which uses an estimation of the full gradient to replace the real computation of it. As illustrated in Algorithm 2, the new variant of SGD with the variance reduction technique is denoted by SAMPLEVR. SAMPLEVR estimates the full gradient by using stochastic gradients which are computed by using k sampled instances from the training data. The mean of the stochastic gradients denoted by $\dot{g}$ is used as the progressive direction in the next epoch. Since $\mathbb{E}(\dot{g}) = \mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\nabla f_i(\tilde{\omega})\right) = \mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\tilde{\omega})\right) = \nabla F(\tilde{\omega})$ holds, the new update gradient, i.e. $\dot{\gamma}_t$ is the same with γ_t in probability, that is, $\mathbb{E}(\dot{\gamma}_t) = \mathbb{E}(\gamma_t) = \nabla F(\omega_t)$.

We illustrate Assumption 1 and Assumption 2 for analysis, which are basic and used in SVRG and its variants. Although the estimation of the full gradient, i.e. $\dot{g}$ is unbiased, the variance between $\dot{g}$ and g impedes the convergence of the training loss, especially when the parameter gets close to the optimum. This problem is mitigated in SAMPLEVR by increasing the number of the sampled instances over epochs linearly. In specific, according to Assumption 1 and Assumption 2, every stochastic gradient $\nabla f_i(\omega)$ with $i \in \{1, 2, ..., n\}$ is bounded by a positive constant denoted by L, that is, $\| \nabla f_i(\omega) \| \leq L$. $\mathbb{E}\left(\frac{1}{k}\sum_{i=1}^{k}\| \nabla f_i(\tilde{\omega}_{s+1}) \|\right) = \frac{1}{n}\sum_{i=1}^{n}\| \nabla f_i(\tilde{\omega}_{s+1}) \|$. Suppose $\chi = \frac{1}{k}\sum_{i=1}^{k}\| \nabla f_i(\tilde{\omega}_{s+1}) \| - \frac{1}{n}\sum_{i=1}^{n}\| \nabla f_i(\tilde{\omega}_{s+1}) \|$, we obtain $P(|\chi| \geq \rho) \leq 2e^{\frac{-2k\rho^2}{L^2}}$, according to Hoeffding's inequality. Let $\alpha = P(|\chi| \geq \rho) \leq 2e^{\frac{-2k\rho^2}{L^2}}$. Such the probability represents the level of significance, i.e. α for a confidence interval around the expectation of size 2ρ. Let $\epsilon = \frac{2\rho^2}{L^2}$. If we require at least k instances to acquire $(1-\alpha)$-confidence interval $[-\rho, \rho]$, k should satisfy

$$k \geq -L^2\frac{\log\frac{\alpha}{2}}{2\rho^2} = -\frac{\log\frac{\alpha}{2}}{2\rho^2/L^2} = -\frac{\log\frac{\alpha}{2}}{\epsilon} \tag{5}$$

. Therefore, k is increased with the decrease of ϵ, and thus the variance caused by the estimation is reduced. However, k is not trivial to be identified. On one hand, a large k leads to much computation cost to obtain the estimation of the full gradient. On the other hand, a small k causes much variance between the estimation and the full gradient. SAMPLEVR increases k linearly, which achieves a good tradeoff between the time efficiency and the variance. The theoretical and extensive empirical studies have shown that SAMPLEVR decreases much gradient complexity and thus outperforms its counterparts.

Assumption 1. *Each differentiable function f_{i_t} with $i_t \in \{1, 2, ..., n\}$ in Equation 1 is L-Liptchiz continuous, that is, $\| f_{i_t}(\omega_i) - f_{i_t}(\omega_j) \| \leq L \| \omega_i - \omega_j \|$ holds for any two parameters ω_i and ω_j. Equivalently, we obtain*

$$f_{i_t}(\omega_i) \leq f_{i_t}(\omega_j) + \nabla f_{i_t}(\omega_j)^{\mathrm{T}}(\omega_i - \omega_j) + \frac{L}{2}\| \omega_i - \omega_j \|^2$$

.

Assumption 2. *The function F in Equation 1 is μ-strongly convex. That is, for any two parameters ω_i and ω_j, we obtain*

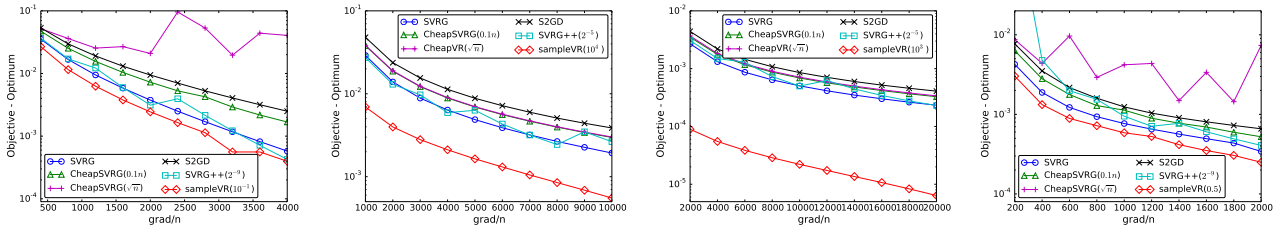
$$F(\omega_i) \geq F(\omega_j) + \nabla F(\omega_j)^{\mathrm{T}}(\omega_i - \omega_j) + \frac{\mu}{2}\| \omega_i - \omega_j \|^2$$

.

Theorem 2. *Given $\delta = \frac{1+4L\mu m\eta^2}{\mu m\eta(1-2\eta L)} < 1$ holds with $\frac{1}{12L}\left(1 - \sqrt{\frac{\mu m - 24L}{\mu m}}\right) < \eta < \frac{1}{12L}\left(1 + \sqrt{\frac{\mu m - 24L}{\mu m}}\right)$, SAMPLEVR makes the training loss converge as $\mathbb{E}[F(\tilde{\omega}_{s+1}) - F(\omega_*)] \leq \delta\mathbb{E}[F(\tilde{\omega}_s) - F(\omega_*)] + \frac{8(\epsilon n + s\log\frac{\alpha}{2})^2L^2\eta}{\epsilon^2 n^2(1-2\eta L)}$.*

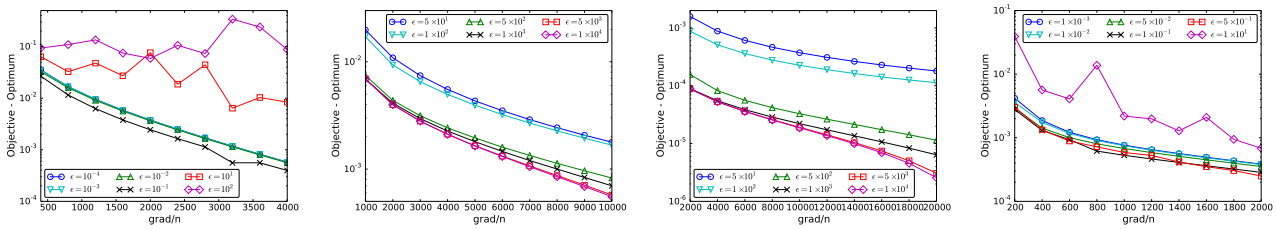
Theorem 3. *Let α be small enough, so that $\frac{8(\epsilon n + \log\frac{\alpha}{2})^2L^2\eta}{\epsilon^2 n^2(1-2\eta L)} \leq F(\tilde{\omega}_0) - F(\omega_*)$ holds,* SAMPLEVR *requires $O(\ln^2\frac{1}{\epsilon})$ atomic gradient calculations to achieve $\mathbb{E}[F(\tilde{\omega}_s) - F(\omega_*)] \leq \zeta$.*

As illustrated in Theorem 2, SAMPLEVR makes the optimisation objective converge at a linear rate. To be honest, the convergence performance is not the best when comparing with SVRG because that the ratio of the learning rate of



(a) ijcnn1 (b) colon-cancer (c) duke-breast-cancer (d) a9a

Figure 1: SAMPLEVR makes the training loss of the $l2$-regularised logistic regression tasks converge faster than the other existing algorithms.



(a) ijcnn1 (b) colon-cancer (c) duke-breast-cancer (d) a9a

Figure 2: Generally, SAMPLEVR with a large ϵ has a better performance for the the $l2$-regularised logistic regression tasks. However, the increase of the variance caused by an extremely large ϵ makes the convergence of the training loss slow down.

SAMPLEVR against that of SVRG is

$$\frac{\delta^{\text{SAMPLEVR}}}{\delta^{\text{SVRG}}} = \frac{1+4L\mu m\eta^2}{\mu m\eta(1-2\eta L)} \times \frac{1}{\frac{1}{\mu\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta}} = \frac{1+4L\mu m\eta^2}{1+2L\mu m\eta^2} < 2 \quad (6)$$

. However, SAMPLEVR has a significant advantage on the gradient complexity according to Theorem 3. For example, when the optimisation objective is strongly convex, and achieves $\mathbb{E}[F(\tilde{\omega}_s) - F(\omega_*)] \leq \zeta$ with $\zeta = \frac{1}{n}$, the gradient complexity of SVRG and its variants is $O(n \ln n)$ (Allen-Zhu and Yuan 2015). But, the gradient complexity of SAMPLEVR is $O(\ln^2 n)$. Considering $\ln n \ll n$, SAMPLEVR outperforms SVRG and its variants on the gradient complexity obviously.

Performance evaluation

Experimental settings

The existing variants of SGD with the variance reduction technique, including SVRG, S2GD, SVRG++, CHEAPSVRG have been used to conduct the performance evaluation with our proposed algorithm, i.e. SAMPLEVR. The number of sampled instances in CHEAPSVRG is identified as $0.1n$ and $\sqrt{n}$ where n represents the size of the training data. Those algorithms are evaluated on eight datasets, including ijcnn1, colon-cancer, duke-breast-cancer, a9a, mg, cpusmall, yearPredictionMSD, and space-ga. All of those datasets are public on the LibSVM website. [1]

[1] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

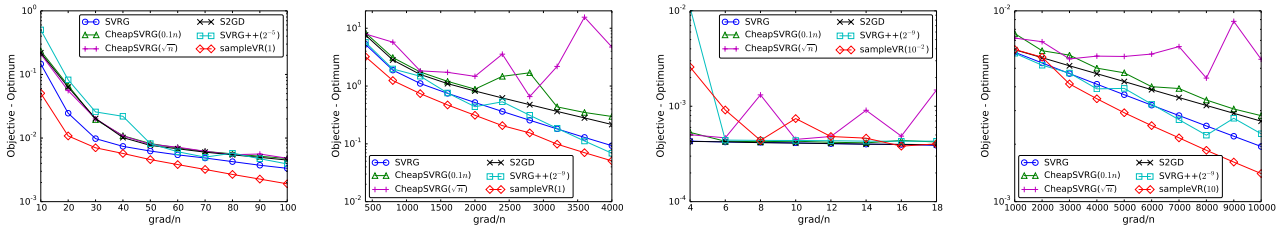
First, those algorithms are compared by conducting the $l2$-regularised logistic regression tasks on the datasets: ijcnn1, colon-cancer, duke-breast-cancer, and a9a. When the label of an instance is set to be 1 or -1, the loss function of the $l2$-regularised logistic regression tasks is:

$$\min_{\omega} \frac{1}{n} \sum_{i=1}^{n} \log(1 + e^{-y_i \omega^{\mathrm{T}} x_i}) + \lambda \parallel \omega \parallel^2$$

. Second, we compare those algorithms by conducting ridge regression tasks on the other four datasets, i.e. mg, cpusmall, yearPredictionMSD, and space-ga. The loss function of the ridge regression tasks is:

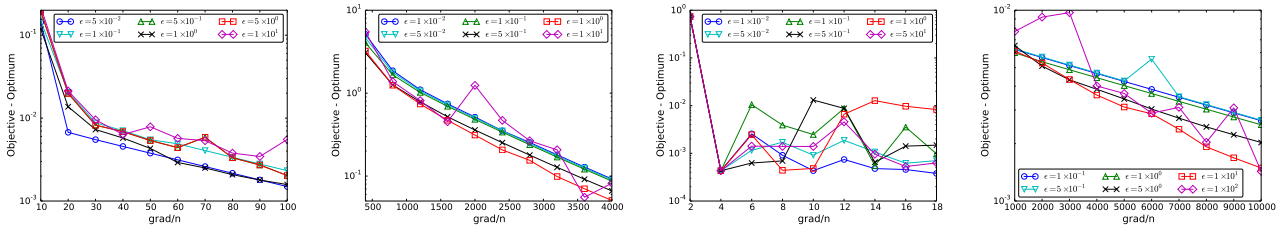
$$\min_{\omega} \frac{1}{n} \sum_{i=1}^{n} \left(\omega^{\mathrm{T}} x_i - y_i\right)^2 + \lambda \parallel \omega \parallel^2$$

. We set λ to be 10^{-5}, and the learning rate, i.e. η to be 10^{-4} for all evaluations. The epoch size m_s in SVRG and CHEAPSVRG is set to be the size of training data, i.e. $m_s{=}n$. The maximal epoch size in S2GD is set to be the size of training data, i.e. n. The x-axis in all figures represents the computational cost. The computational cost is measured by the number of gradient computations divided by the size of training data, i.e. n. The y-axis in all the figures denotes training loss residual which is the training loss minus the optimum. Here, the optimum is estimated by running the gradient descent for a long time. The value in the bracket of the legend of SVRG++ and SAMPLEVR represents the initial epoch size divided by the size of the training data, i.e. n and ϵ according to Algorithm 2, respectively.



(a) mg (b) cpusmall (c) yearPredictionMSD (d) space-ga

Figure 3: Generally, SAMPLEVR outperforms the other existing algorithms on the convergence of the training loss when conducting the ridge regression tasks.



(a) mg (b) cpusmall (c) yearPredictionMSD (d) space-ga

Figure 4: A large ϵ leads to the fast convergence of the training loss for SAMPLEVR when conducting the ridge regression tasks. However, the performance of SAMPLEVR is impaired due to the increase of variance when ϵ is set to be too large.

$l2$-regularised logistic regression

As illustrated in Figure 1, we compare the performance of all the algorithms by conducting $l2$-regularised logistic regression tasks. It is obvious that our proposed algorithm, i.e. SAMPLEVR makes the training loss converge linearly, and outperforms other existing algorithms. The main reason is that SAMPLEVR replaces the full gradient with an estimation, thus getting rid of the time-consuming calculations of the gradient. Although the estimation of the full gradient is used in CHEAPSVRG, the number of sampled instances in CHEAPSVRG is set before the running of the algorithm and then keep a constant such as $\sqrt{n}$. The constant number of sampled instances leads to much computation cost at first and much variance in the end during the training of the parameters. Instead, SAMPLEVR increases the number of sampled instances linearly, and thus achieves a good trade-off between time efficiency and variance. Additionally, the comparison of the performance of SAMPLEVR by varying ϵ is shown in Figure 2. SAMPLEVR generally obtains a better performance with a large ϵ. It is because that the number of the sampled instances becomes small with a large ϵ according to Equation 5, which reduces the computational cost significantly. However, as illustrated in Figure 2(a) and 2(d), if ϵ is set to be too large, the number of the sampled instances becomes extremely small. Thus, the large variance makes the training loss converge slowly.

Ridge regression

As illustrated in Figure 3, we report the comparison of the performance by using all the algorithms to conduct ridge regression tasks. SAMPLEVR has a better performance for the datasets mg, cpusmall, and space-ga than the existing algorithms significantly. The main reason is that SAMPLEVR uses an unbiased estimation of the full gradient, instead of costing much time to compute it. Although SAMPLEVR does not outperform other algorithms for the dataset yearPredictionMSD at the beginning of the train process, its performance is comparable to the other algorithms, and finally shows the advantage over most of the existing algorithms. As illustrated in Figure 4, the performance of SAMPLEVR has been compared by varying ϵ. It is significant that the variance becomes noticeable with the increase of ϵ. It is because that a large ϵ leads to few sampled instances according to Equ. 5, incurring much variance in the estimation of the full gradient. Moreover, an extremely small ϵ impairs the performance of SAMPLEVR. The reason is that such small ϵ leads to a large number of the instances, thus incurring much calculations of gradients. It is noting that the best setting of ϵ in different datasets varies a lot, for example 10^{-2} in yearPredictionMSD and 10 in space-ga. The best method to tune ϵ will be studied in the future work. Generally, a practical method is to tune the value of the ϵ on a subset of the training data to obtain the best settings.

Conclusion

This paper first analyses the variance reduction technique from a new prospective in a general framework. Then, a new variant of SGD with the variance reduction technique, i.e. SAMPLEVR is proposed. The theoretical and empirical studies show the advantages of SAMPLEVR significantly.

References

Allen-Zhu, Z., and Hazan, E. 2016. Variance reduction for faster non-Convex optimization. In International Conference on Machine Learning.

Allen-Zhu, Z., and Yuan, Y. 2015. Univr: A universal variance reduction framework for proximal stochastic gradient method. arXiv preprint arXiv:1506.01972.

Allen-Zhu, Z., and Yuan, Y. 2016. Improved SVRG for non-strongly-convex or sum-of-non-convex objectives. In International Conference on Machine Learning.

Allen-Zhu, Z. 2016. Katyusha: accelerated rariance reduction for faster sgd. arXiv preprint arXiv:1603.05953.

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

Garber, D., and Hazan, E. 2015. Fast and Simple PCA via Convex Optimization. arXiv.org.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. Advances in Neural Information Processing Systems 315–323.

Konečnỳ, J., and Richtárik, P. 2013. Semi-stochastic gradient descent methods. arXiv preprint arXiv:1312.1666.

Konečnỳ, J.; Liu, J.; Richtárik, P.; and Takáč, M. 2016. Mini-batch semi-stochastic gradient descent in the proximal setting. IEEE Journal of Selected Topics in Signal Processing 10(2):242–255.

Li, X.; Zhao, T.; Arora, R.; Liu, H.; and Haupt, J. 2016. Stochastic variance reduced optimization for nonconvex sparse learning. In International Conference on Machine Learning.

Schmidt, M.; Roux, N. L.; and Bach, F. 2016. Minimizing finite sums with the stochastic average gradient. Mathematical Programming 1–30.

Shah, V.; Asteris, M.; Kyrillidis, A.; and Sanghavi, S. 2016. Trading-off variance and complexity in stochastic gradient descent. arXiv preprint arXiv:1603.06861.

Shalev-Shwartz, S. 2016. SDCA without duality, regularization, and individual convexity. In International Conference on Machine Learning.

Xiao, L., and Zhang, T. 2014. A proximal stochastic gradient method with progressive variance reduction. SIAM Journal on Optimization 24(4):2057–2075.

Zhang, L.; Mahdavi, M.; and Jin, R. 2013. Linear convergence with condition number independent access of full gradients. In Advances in Neural Information Processing Systems, 980–988.