Variance Reduction Methods for Stochastic Optimization

1 Introduction

We consider the following stochastic optimization problem with finite sum structure

$$\min_{w \in C} \phi(w), \quad \phi(w) = f(w), \qquad f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w). \tag{1}$$

We assume that $f(w)$ is L-smooth, and λ strongly convex.

In the previous lectures, we have shown that stochastic optimization converge faster than deterministic gradient methods in the beginning, when the gradient is relatively large compared to the variance.

In particular, we consider the SGD update rule with batch size of 1:

$$w^{(t)} = w^{(t-1)} - \eta_t \nabla f_i(w^{(t-1)}),$$

where i is sampled uniformly from D, which is the uniform distribution over $\{1, \ldots, n\}$.

Let

$$V = \frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(w) - \nabla f(w)\|_2^2,$$

then we can set the optimal learning rate as

$$\eta = \frac{\|\nabla f(w)\|_2^2}{L(\|\nabla f(w)\|_2^2 + V)},$$

and obtain the following one-step convergence result:

$$\mathbf{E}_{i \sim D} f(w - \eta \nabla f_i(w)) \leq f(w) - 0.5\eta \|\nabla f(w)\|_2^2.$$

As $\|\nabla f(w)\|_2 \to 0$, we have to set smaller and smaller learning rate proportional to $O(\|\nabla f(w)\|_2^2)$. Consequently, the convergence of SGD slows down, and we can only obtain sublinear convergence rate asymptotically.

However, the full gradient descent method does not suffer from this problem, and it converges linearly.

This lecture discuss variance reduction methods for stochastic gradient optimization, so that V converges to zero at the same rate of $\|\nabla f(w)\|_2^2$. It follows that we may use a constant learning rate, and achieve linear convergence speed.

2 Stochastic Variance Reduced Gradient

One practical issue for SGD is that in order to ensure convergence the learning rate η_t has to decay to zero. This leads to slower convergence. The need for a small learning rate is due to the variance of SGD (that is, SGD approximates the full gradient using a small batch of samples or even a single example, and this introduces variance). However, for the finite sum problem, we may design methods that reduce the variance. This section describes the SVRG method of [2].

At each time, we keep a version of estimated w as $\tilde{w}$ that is close to the optimal w. For example, we can keep a snapshot of $\tilde{w}$ after every m SGD iterations. Moreover, we maintain the average gradient

$$\tilde{\mu} = \nabla f(\tilde{w}) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{w}),$$

and its computation requires one pass over the data using $\tilde{w}$. Note that the expectation of $\nabla f_i(\tilde{w}) - \tilde{\mu}$ over i is zero. If we define the auxiliary function

$$\tilde{f}_i(w) = f_i(w) - (\nabla f_i(\tilde{w}) - \tilde{\mu})^\top w,$$

then

$$f(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) = \frac{1}{n} \sum_{i=1}^{n} \tilde{f}_i(w).$$

We can apply the SGD rule to the finite sum with respect to $\tilde{f}_i(w)$, and obtain the following update rule is generalized SGD:

$$w^{(t)} = w^{(t-1)} - \eta_t \nabla \tilde{f}_i(w^{(t-1)}), \quad \nabla \tilde{f}_i(w) = (\nabla f_i(w) - \nabla f_i(w) + \tilde{\mu}), \tag{2}$$

where we draw i randomly from D.

To see that the variance of the update rule (2) is reduced, we note that when both $\tilde{w}$ and $w^{(t)}$ converge to the same parameter w_*, then $\tilde{\mu} \to 0$. Therefore if $\nabla f_i(\tilde{w}) \to \nabla f_i(w_*)$, then

$$\nabla f_i(w^{(t-1)}) - \nabla f_i(\tilde{w}) + \tilde{\mu} \to \nabla f_i(w^{(t-1)}) - \nabla f_i(w_*) \to 0.$$

The idea described above can be formulated in Algorithm 1.

Algorithm 1: Stochastic Variance Reduced Gradient (SVRG)

```
Input: $\phi(\cdot)$, $w_0$, $\eta$, update frequency $m$
Output: $\tilde{w}^{(S)}$
Let $\tilde{w}^{(0)} = w_0$
for $s = 1, 2, \ldots, S$ do
    Let $\tilde{w} = \tilde{w}^{(s-1)}$
    Let $\tilde{\mu} = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\tilde{w})$
    Let $w_0 = \tilde{w}$
    for $t = 1, \ldots, m$ do
        Select $i_t$ uniformly at random from $\{1, \ldots, n\}$
        Let $g_{i_t} = (\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(\tilde{w}) + \tilde{\mu}$
        Let $w_t = w_{t-1} - \eta g_{i_t}$
    option I: set $\tilde{w}^{(s)} = w_m$
    option II: set $\tilde{w}^{(s)} = w_t$ for randomly chosen $t \in \{0, \ldots, m-1\}$
Return: $\tilde{w}^{(S)}$
```

In practical implementations, it is natural to choose option I, or take $\tilde{w}_s$ to be the average of the past t iterates. However, our analysis depends on option II. Note that each stage s requires $2m+n$ gradient computations (for some convex problems, one may save the intermediate gradients $\nabla f_i(\tilde{w})$, and thus only $m+n$ gradient computations are needed). Therefore it is natural to choose m to be the same order of n but slightly larger (for example $m = 2n$ for convex problems and $m = 5n$ for nonconvex problems). In comparison, standard SGD requires only m gradient computations. Since gradient may be the computationally most intensive operation, for fair comparison, we compare SGD to SVRG based on the number of gradient computations.

For simplicity we will only consider the case that each $f_i(w)$ is convex and smooth, and $f(w)$ is strongly convex.

Theorem 1 *Consider the SVRG algorithm with option II. Assume that all $f_i(w)$ are convex and L-smooth, and $f(w)$ is λ strongly convex. Let $w_* = \arg\min_w f(w)$. Assume that m is sufficiently large so that*

$$\rho = \frac{1}{\lambda\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta} < 1,$$

then we have geometric convergence in expectation for SVRG:

$$\mathbf{E} f(\tilde{w}^{(s)}) \leq \mathbf{E} f(w_*) + \rho^s [f(\tilde{w}^{(0)}) - f(w_*)].$$

Proof Given any i, consider

$$g_i(w) = f_i(w) - f_i(w_*) - \nabla f_i(w_*)^\top (w - w_*).$$

We know that $g_i(w_*) = \min_w g_i(w)$ since $\nabla g_i(w_*) = 0$. Therefore

$$\begin{aligned} 0 = g_i(w_*) \leq & \min_\eta [g_i(w - \eta \nabla g_i(w))] \\ & \leq \min_\eta [g_i(w) - \eta \|\nabla g_i(w)\|_2^2 + 0.5 L \eta^2 \|\nabla g_i(w)\|_2^2] = g_i(w) - \frac{1}{2L}\|\nabla g_i(w)\|_2^2. \end{aligned}$$

That is,

$$\|\nabla f_i(w) - \nabla f_i(w_*)\|_2^2 \leq 2L[f_i(w) - f_i(w_*) - \nabla f_i(w_*)^\top (w - w_*)].$$

By summing the above inequality over $i = 1, \ldots, n$, and using the fact that $\nabla f(w_*) = 0$, we obtain

$$n^{-1} \sum_{i=1}^n \|\nabla f_i(w) - \nabla f_i(w_*)\|_2^2 \leq 2L[f(w) - f(w_*)]. \tag{3}$$

We can now proceed to prove the theorem. Let $v_t = \nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(\tilde{w}) + \tilde{\mu}$. Conditioned on w_{t-1}, we can take expectation with respect to i_t, and obtain:

$$\begin{aligned} & \mathbf{E}\,\|v_t\|_2^2 \\ \leq & 2\mathbf{E}\,\|\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(w_*)\|_2^2 + 2\mathbf{E}\,\|[\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w_*)] - \nabla f(\tilde{w})\|_2^2 \\ = & 2\mathbf{E}\,\|\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(w_*)\|_2^2 + 2\mathbf{E}\,\|[\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w_*)] - \mathbf{E}\,[\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w_*)]\|_2^2 \\ \leq & 2\mathbf{E}\,\|\nabla f_{i_t}(w_{t-1}) - \nabla f_{i_t}(w_*)\|_2^2 + 2\mathbf{E}\,\|\nabla f_{i_t}(\tilde{w}) - \nabla f_{i_t}(w_*)\|_2^2 \\ \leq & 4L[f(w_{t-1}) - f(w_*) + f(\tilde{w}) - f(w_*)]. \end{aligned}$$

The first inequality uses $\|a+b\|_2^2 \leq 2\|a\|_2^2 + 2\|b\|_2^2$ and $\tilde{\mu} = \nabla f(\tilde{w})$. The second inequality uses $\mathbf{E}\|\xi - \mathbf{E}\xi\|_2^2 = \mathbf{E}\|\xi\|_2^2 - \|\mathbf{E}\xi\|_2^2 \leq \mathbf{E}\|\xi\|_2^2$ for any random vector ξ. The third inequality uses (3).

Now by noticing that conditioned on w_{t-1}, we have $\mathbf{E}v_t = \nabla f(w_{t-1})$; and this leads to

$$\begin{aligned}
&\mathbf{E}\,\|w_t - w_*\|_2^2 \\
=&\|w_{t-1} - w_*\|_2^2 - 2\eta(w_{t-1} - w_*)^\top \mathbf{E}\,v_t + \eta^2 \mathbf{E}\,\|v_t\|_2^2 \\
\leq&\|w_{t-1} - w_*\|_2^2 - 2\eta(w_{t-1} - w_*)^\top \nabla f(w_{t-1}) + 4L\eta^2[f(w_{t-1}) - f(w_*) + f(\tilde{w}) - f(w_*)] \\
\leq&\|w_{t-1} - w_*\|_2^2 - 2\eta[f(w_{t-1}) - f(w_*)] + 4L\eta^2[f(w_{t-1}) - f(w_*) + f(\tilde{w}) - f(w_*)] \\
=&\|w_{t-1} - w_*\|_2^2 - 2\eta(1-2L\eta)[f(w_{t-1}) - f(w_*)] + 4L\eta^2[f(\tilde{w}) - f(w_*)].
\end{aligned}$$

The first inequality uses the previously obtained inequality for $\mathbf{E}\|v_t\|_2^2$, and the second inequality convexity of $f(w)$, which implies that $-(w_{t-1} - w_*)^\top \nabla f(w_{t-1}) \leq f(w_*) - f(w_{t-1})$.

We consider a fixed stage s, so that $\tilde{w} = \tilde{w}_{s-1}$ and $\tilde{w}_s$ is selected after all of the updates have completed. By summing the previous inequality over $t = 1, \ldots, m$, taking expectation with all the history, and using option II at stage s, we obtain

$$\begin{aligned}
&\mathbf{E}\,\|w_m - w_*\|_2^2 + 2\eta(1-2L\eta)m\mathbf{E}\,[f(\tilde{w}_s) - f(w_*)] \\
\leq&\mathbf{E}\|w_0 - w_*\|_2^2 + 4Lm\eta^2\mathbf{E}[f(\tilde{w}) - f(w_*)] \\
=&\mathbf{E}\|\tilde{w} - w_*\|_2^2 + 4Lm\eta^2\mathbf{E}[f(\tilde{w}) - f(w_*)] \\
\leq&\frac{2}{\lambda}\mathbf{E}[f(\tilde{w}) - f(w_*)] + 4Lm\eta^2\mathbf{E}[f(\tilde{w}) - f(w_*)] \\
=&2(\lambda^{-1} + 2Lm\eta^2)\mathbf{E}[f(\tilde{w}) - f(w_*)].
\end{aligned}$$

The second inequality uses the strong convexity. We thus obtain

$$\mathbf{E}\,[f(\tilde{w}^{(s)}) - f(w_*)] \leq \left[\frac{1}{\lambda\eta(1-2L\eta)m} + \frac{2L\eta}{1-2L\eta}\right]\mathbf{E}[f(\tilde{w}^{(s-1)}) - f(w_*)].$$

This implies that $\mathbf{E}\,[f(\tilde{w}^{(s)}) - f(w_*)] \leq \rho^s \mathbf{E}\,[f(\tilde{w}^{(0)}) - f(w_*)]$. The desired bound follows. ■

To interpret the bounds, we may consider for simplicity the most indicative case where the condition number $L/\lambda = n$. Due to the poor condition number, the standard batch gradient descent requires complexity of $n\ln(1/\epsilon)$ iterations over the data to achieve accuracy of ϵ, which means we have to process $n^2\ln(1/\epsilon)$ number of examples. In comparison, in our procedure we may take $\eta = 0.1/L$ and $m = O(n)$ to obtain a convergence rate of $\rho = 1/2$. Therefore to achieve an accuracy of ϵ, we need to process $n\ln(1/\epsilon)$ number of examples. This matches the results of SDCA in Lecture 17.

3 Proximal SVRG

The proximal extension of SVRG is given in [5], which considers the following finite sum problem with nonzero $g(w)$:

$$\min_{w \in C} \phi(w), \quad \phi(w) = f(w) + g(w), \qquad f(w) = \frac{1}{n}\sum_{i=1}^{n} f_i(w). \tag{4}$$

The corresponding method is given in Algorithm 2, where

$$\mathrm{prox}_{\eta g}(w) = \arg\min_z \left[\frac{1}{2}\|z - w\|_2^2 + g(z)\right].$$

Algorithm 2: Proximal Stochastic Variance Reduced Gradient (Prox-SVRG)

```
Input: $\phi(\cdot)$, $w_0$, $\eta$, update frequency $m$
Output: $\tilde{w}^{(S)}$
Let $\tilde{w}^{(0)} = w_0$
Let $D$ be the distribution on $\{1, \ldots, n\}$ according to probability $Q = \{q_1, \ldots, q_n\}$
for $s = 1, 2, \ldots, S$ do
    Let $\tilde{w} = \tilde{w}^{(s-1)}$
    Let $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^n \nabla f_i(\tilde{w})$
    Let $w_0 = \tilde{w}$
    for $t = 1, \ldots, m$ do
        Randomly pick $i \sim D$ and update weight
        Let $g_i = (\nabla f_i(w_{t-1}) - \nabla f_i(\tilde{w}))/(q_i n) + \tilde{\mu}$
        Let $w_t = \mathrm{prox}_{\eta g}(w_t - \eta g_i)$
    Let $\tilde{w}^{(s)} = \frac{1}{m}\sum_{t=1}^m w_t$
Return: $\tilde{w}^{(S)}$
```

Theorem 2 *Assume that each $f_i(w)$ is L_i-smooth, and $g(w)$ is λ strongly-convex. Let $w_* = \arg\min_w \phi(w)$ and $L_Q = \max_i L_i/(q_i n)$. In addition, assume that $0 < \eta < 1/(4L_Q)$ and m is sufficiently large so that*

$$\rho = \frac{1}{\lambda\eta(1 - 4L_Q\eta)m} + \frac{4L_Q\eta(m+1)}{(1 - 4L_Q\eta)m} < 1. \tag{5}$$

Then the Prox-SVRG method has geometric convergence in expectation:

$$\mathbf{E}f(\tilde{x}^{(s)}) - f(x_*) \leq \rho^s[f(\tilde{w}^{(0)}) - f(w_*)].$$

Note that Algorithm 2 uses importance sampling to further reduces variance. For importance sampling, the optimal Q is to pick $q_i \propto L_i$. With this choice, the convergence depends on the average smoothness $n^{-1}\sum_{i=1}^n L_i$ instead of the maximum smoothness $\max_i L_i$ as in the case of uniform sampling.

4 SDCA as Variance Reduction

In the following we present the variance reduction view of SDCA. In SDCA, we consider the following dual representation of (4) with $g(w) = \frac{\lambda}{2}\|w\|_2^2$:

$$w_* = \frac{1}{\lambda n}\sum_{i=1}^n \alpha_i^*, \quad \alpha_i^* = -\nabla f_i(w_*). \tag{6}$$

We may maintain a similar dual representation as follows,

$$w^{(t)} = \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_i^{(t)}, \tag{7}$$

and update the primal w using SGD as:

$$w^{(t)} = w^{(t-1)} - \lambda \eta_t w^{(t-1)} - \eta_t \nabla f_i(w^{(t-1)}). \tag{8}$$

Now we may use the relationship

$$-\lambda \eta_t w^{(t-1)} = -\eta_t \frac{1}{n} \sum_{i=1}^{n} \alpha_i^{(t-1)},$$

to replace the gradient $-\lambda \eta_t w^{(t-1)}$ by stochastic gradient $-\eta_t \alpha_i^{(t-1)}$. This leads to the following update

$$w^{(t)} = w^{(t-1)} - \eta_t \alpha_i^{(t-1)} - \eta_t \nabla f_i(w^{(t-1)}). \tag{9}$$

The corresponding dual update is a version of the SDCA method:

$$\alpha_\ell^{(t)} = \begin{cases} \alpha_i^{(t-1)} - \eta_t(\nabla f_i(w^{(t-1)}) + \alpha_i^{(t-1)}) & \ell = i \\ \alpha_\ell^{(t-1)} & \ell \neq i \end{cases},$$

together with primal update

$$w^{(t)} = w^{(t-1)} + \frac{1}{\lambda n}(\alpha_i^{(t)} - \alpha_i^{(t-1)}).$$

The advantage of (9) over (8) is that we may take constant stepsize in (9) when $t \to \infty$. This is because according to (6), when the primal-dual parameters (w, α) converge to the optimal parameters (w_*, α_*), we have

$$(\nabla f_i(w) + \alpha_i) \to 0.$$

Therefore SDCA can also be regarded as a variance reduction method for SGD.

5 SAGA

There are a number of other variance reduction techniques. We will describe the SAGA method in Algorithm 3 of [1]. It can be seen that

$$\mathbf{E}_i g_i = \frac{1}{n} \sum_{i=1}^{n} -\alpha_i' = \nabla f(w^{(t-1)}).$$

Therefore SAGA is also a version of proximal SGD. Moreover, when $t \to \infty$, $\alpha_i' \to \alpha_i$. Therefore similar to SVRG, SAGA has variance reduction effect. However, unlike SVRG, which forms $\tilde{\mu}$ using a snap-shot $\tilde{w}$. In SAGA, $\tilde{\mu}$ is aggregated using solutions from different time.

Note that in the implementation, if $f_i(w)$ has a form of $\psi_i(X_i^\top w)$, where X_i is a $d \times k$ matrix with $k \ll d$, then

$$\nabla f_i(w) = X_i \nabla \psi_i(X_i^\top w).$$

It follows that we may store the variables

$$\beta_i = -\nabla \psi_i(X_i^\top w) \in \mathbb{R}^k,$$

and let $\alpha_i = X_i \beta_i$. This leads to savings of storage from nd for $\{\alpha_i\}$ to nk for $\{\beta_i\}$.

```
Algorithm 3: SAGA
Input: $\phi(\cdot)$, $w^{(0)}$, $\eta$, update frequency $m$
Output: $\tilde{w}^{(T)}$
Initialize $\alpha_1, \ldots, \alpha_n$
Let $\tilde{\mu} = \frac{1}{n} \sum_{i=1}^n (-\alpha_i)$
for $t = 1, 2, \ldots, T$ do
    Select $i$ uniformly at random from $\{1, \ldots, n\}$
    Let $\alpha_i' = -\nabla f_i(w^{(t-1)})$
    Let $g_i = (-\alpha_i') - (-\alpha_i) + \tilde{\mu}$
    Let $w_t = \text{prox}_{\eta g}(w^{(t-1)} - \eta g_i)$
    Let $\tilde{\mu} = \tilde{\mu} + \frac{1}{n}(\alpha_i - \alpha_i')$
    Let $\alpha_i = \alpha_i'$
Return: $w^{(T)}$
```

6 Minibatch

For smooth problems, the convergence rate of variance reduction methods will reduce when we use minibatch. This is because we have already set learning rate at the level $O(1/L)$, which is optimal for gradient descent. Therefore we cannot increase the learning rate by a factor of m to $O(m/L)$, as in SGD. From the last lecture we know that for SGD, we need to increase the learning rate for a larger batch size.

For this reason, when using minibatch optimization with variance reduction, one needs to use acceleration to keep the optimal convergence rate [4, 3]. We describe the accelerated version of SVRG in Algorithm 4. Similar methods can be derived for SAGA.

Algorithm 4: Minibatch Accelerated Prox-SVRG

```
Input: $\phi(\cdot)$, $w_0$, $\eta$, update frequency $m$
Output: $\tilde{w}^{(S)}$
Let $\tilde{w}^{(0)} = w_0$
Let $D$ be the distribution on $\{1, \ldots, n\}$ according to probability $Q = \{q_1, \ldots, q_n\}$
for $s = 1, 2, \ldots, S$ do
    Let $\tilde{w} = \tilde{w}^{(s-1)}$
    Let $\tilde{\mu} = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\tilde{w})$
    Let $w_0 = \tilde{w}$
    for $t = 1, \ldots, m$ do
        Randomly pick minibatch $B \sim D$ and update weight
        Let $u_t = w_{t-1} + \beta(w_{t-1} - w_{t-2})$
        Let $g_B = (\nabla f_B(u_t) - \nabla f_B(\tilde{w}))/(q_i n) + \tilde{\mu}$
        Let $w_t = \mathrm{prox}_{\eta g}(w_t - \eta g_B)$
    Let $\tilde{w}^{(s)} = \frac{1}{m}\sum_{t=1}^{m} w_t$
Return: $\tilde{w}^{(S)}$
```

7 Empirical Studies

We study the smoothed hinge loss function $\phi_\gamma(z)$ with $\gamma = 1$, and solves the following $L_1 - L_2$ regularization problem:

$$\min_w \left[\underbrace{\frac{1}{n}\sum_{i=1}^{n} \phi_\gamma(w^\top x_i y_i)}_{f(w)} + \underbrace{\frac{\lambda}{2}\|w\|_2^2 + \mu\|w\|_1}_{g(w)} \right].$$

We compare SVRG and SAGA, with a fixed learning rate η. We try different minibatch sizes and different β for acceleration. The sample size is $n = 5000$. Therefore when $\lambda \ll 1/n$, SDCA does not work well.

References

[1] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS' 14*, 2014.

[2] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS' 13*, 2013.

[3] Atsushi Nitanda. Stochastic proximal gradient descent with acceleration techniques. In *NIPS' 14*. 2014.

[4] Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS' 13*. 2013.

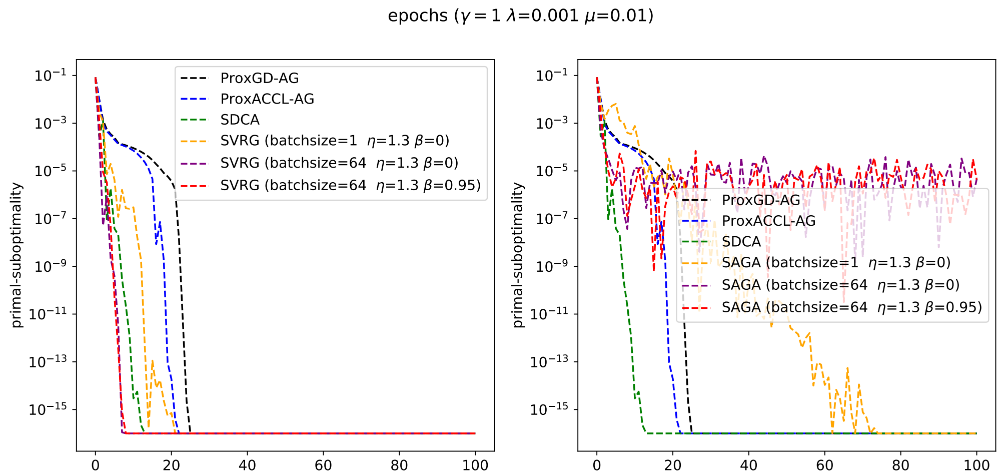

Figure 1: Comparisons of different stochastic algorithms (smooth and strongly convex)

[5] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24:2057–2075, 2014.

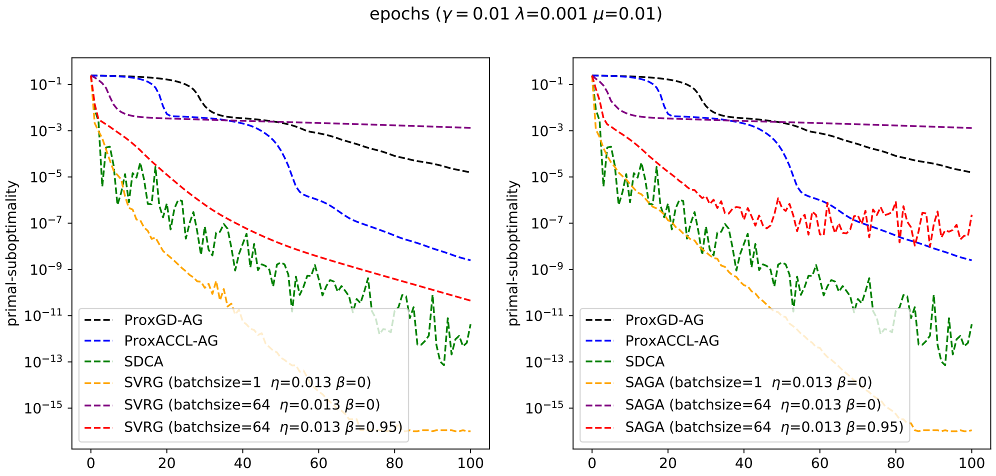

Figure 2: Comparisons of different stochastic algorithms (near non-smooth and strongly convex)

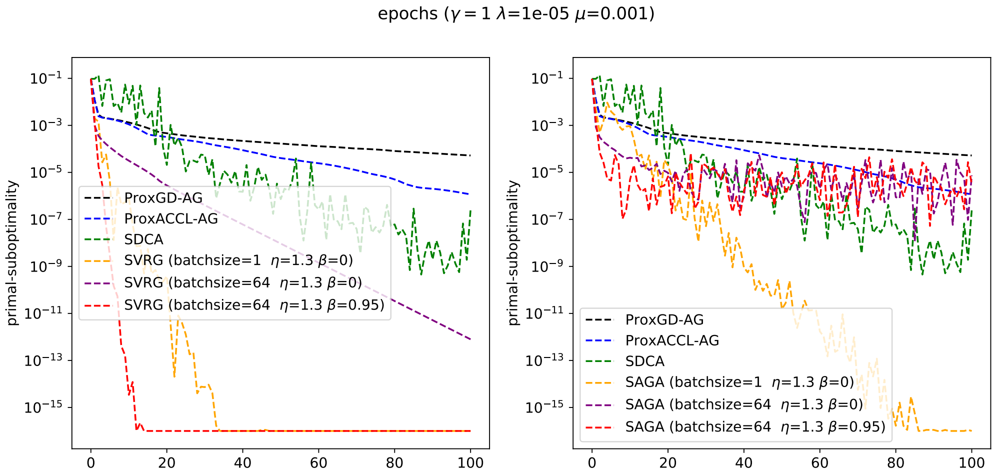

Figure 3: Comparisons of different stochastic algorithms (near non-smooth and near non-strongly convex)

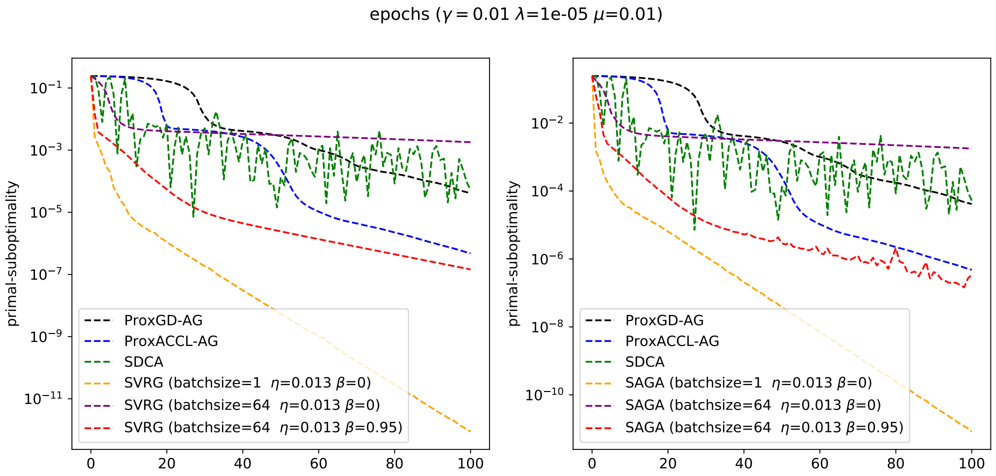

Figure 4: Comparisons of different stochastic algorithms (near non-smooth and near non-strongly convex)