The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Accelerated Method for Stochastic Composition Optimization with Nonsmooth Regularization

Zhouyuan Huo,[1] Bin Gu,[1] Ji Liu,[2] Heng Huang[1*]
[1]Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA
[2]Department of Computer Science, University of Rochester, Rochester, NY, 14627, USA
zhouyuan.huo@pitt.edu, big10@pitt.edu, jliu@cs.rochester.edu, heng.huang@pitt.edu

Abstract

Stochastic composition optimization draws much attention recently and has been successful in many emerging applications of machine learning, statistical analysis, and reinforcement learning. In this paper, we focus on the composition problem with nonsmooth regularization penalty. Previous works either have slow convergence rate, or do not provide complete convergence analysis for the general problem. In this paper, we tackle these two issues by proposing a new stochastic composition optimization method for composition problem with nonsmooth regularization penalty. In our method, we apply variance reduction technique to accelerate the speed of convergence. To the best of our knowledge, our method admits the fastest convergence rate for stochastic composition optimization: for strongly convex composition problem, our algorithm is proved to admit linear convergence; for general composition problem, our algorithm significantly improves the state-of-the-art convergence rate from $O(T^{-1/2})$ to $O((n_1+n_2)^{2/3}T^{-1})$. Finally, we apply our proposed algorithm to portfolio management and policy evaluation in reinforcement learning. Experimental results verify our theoretical analysis.

Introduction

Stochastic composition optimization draws much attention recently and has been successful in addressing many emerging applications of different areas, such as reinforcement learning (Dai et al. 2016; Wang and Liu 2016), statistical learning (Wang, Fang, and Liu 2014) and risk management (Dentcheva, Penev, and Ruszczyński 2016). The authors in (Wang, Fang, and Liu 2014; Wang and Liu 2016) proposed composition problem, which is the composition of two expected-value functions:

$$\min_{x\in\mathbb{R}^N} \underbrace{\mathbb{E}_i F_i(\mathbb{E}_j G_j(x))}_{f(x)} + h(x), \qquad (1)$$

where $G_j(x):\mathbb{R}^N \mapsto \mathbb{R}^M$ are inner component functions, $F_i(y):\mathbb{R}^M \mapsto \mathbb{R}$ are outer component functions. The regularization penalty $h(x)$ is a closed convex function but not necessarily smooth. In reality, we usually solve the finite-sum scenario for composition problem (1), and it can be represented as follows:

$$\min_{x\in\mathbb{R}^N} H(x) = \min_{x\in\mathbb{R}^N} \underbrace{\frac{1}{n_1}\sum_{i=1}^{n_1} F_i\left(\frac{1}{n_2}\sum_{j=1}^{n_2} G_j(x)\right)}_{f(x)} + h(x), \qquad (2)$$

where it is defined that $F(y) = \frac{1}{n_1}\sum\limits_{i=1}^{n_1} F_i(y)$ and $G(x) = \frac{1}{n_2}\sum\limits_{j=1}^{n_1} G_j(x)$. Throughout this paper, we mainly focus on the case that F_i and G_j are smooth. However, we do not require that F_i and G_j have to be convex.

Minimizing the composition of expected-value functions (1) or finite-sum functions (2) is challenging. Classical stochastic gradient method (SGD) and its variants are well suited for minimizing traditional finite-sum functions (Bottou, Curtis, and Nocedal 2016). However, they are not directly applicable to the composition problem. To apply SGD, we need to compute the unbiased sampling gradient $(\nabla G_j(x))^T \nabla F_i(G(x))$ of problem (2), which is time-consuming when $G(x)$ is unknown. Evaluating $G(x)$ requires traversing all inner component functions, which is unacceptable to compute in each iteration if n_2 is a large number.

In (Wang, Fang, and Liu 2014), the authors considered the problem (1) with $h(x) = 0$ and proposed stochastic compositional gradient descent algorithm (SCGD) which is the first stochastic method for composition problem. In their paper, they proved that the convergence rate of SCGD for strongly convex composition problem is $O(T^{-2/3})$, and for general problem is $O(T^{-1/4})$. They also proposed accelerated SCGD by using Nesterov smoothing technique (Nesterov 1983) which is proved to admit faster convergence rate. SCGD has constant query complexity per iteration, however, their convergence rate is far worse than full gradient method because of the noise induced by sampling gradients. Recently, variance reduction technique (Johnson and Zhang 2013) was applied to accelerate the convergence of stochastic composition optimization. (Lian, Wang, and Liu 2016) first utilized the variance reduction technique and proposed two variance reduced stochastic compositional gradient descent methods (Compositional-SVRG-1 and Compositional-SVRG-2). Both methods are proved to admit linear convergence rate. However, the methods proposed in (Wang, Fang, and Liu 2014)

[*] To whom all correspondence should be addressed.

Table 1: The table shows the comparisons of SCGD, Accelerated SCGD, ASC-PG, Compositional-SVRG-1, Compositional-SVRG-2, com-SVR-ADMM and our VRSC-PG in terms of convergence. For fair comparison, we consider query complexity in the convergence rate. We define that one query of Sampling Oracle ($\mathcal{SO}$) has three cases: (1) Given $x \in R^N$ and $j \in \{1, 2, ..., n_2\}$, $\mathcal{SO}$ returns $G_j(x) \in \mathbb{R}^M$; (2) Given $x \in R^N$ and $j \in \{1, 2, ..., n_2\}$, $\mathcal{SO}$ returns $\nabla G_j(x) \in \mathbb{R}^{M \times N}$; (3) Given $y \in \mathbb{R}^M$ and $i \in \{1, 2, ..., n_1\}$, $\mathcal{SO}$ returns $\nabla F_i(y) \in \mathbb{R}^M$. T denotes the total number of iterations and κ denotes condition number and $0 < \rho < 1$.

Algorithm	$h(x) \neq 0$	**Strongly Convex**	**General Problem**
SCGD (Wang, Fang, and Liu 2014)	✗	$O(T^{-2/3})$	$O(T^{-1/4})$
Accelerated SCGD (Wang, Fang, and Liu 2014)	✗	$O(T^{-4/5})$	$O(T^{-2/7})$
Compositional-SVRG-1 (Lian, Wang, and Liu 2016)	✗	$O\left(\rho^{\frac{T}{n_1+n_2+\kappa^4}}\right)$	-
Compositional-SVRG-2 (Lian, Wang, and Liu 2016)	✗	$O(\rho^{\frac{T}{n_1+n_2+\kappa^3}})$	-
ASC-PG (Wang and Liu 2016)	✓	$O(T^{-4/5})$	$O(T^{-4/9})$
ASC-PG (if $G_j(x)$ are linear) (Wang and Liu 2016)	✓	$O(T^{-1})$	$O(T^{-1/2})$
com-SVR-ADMM (Yu and Huang 2017)	✓	$O\left(\rho^{\frac{T}{n_1+n_2+\kappa^4}}\right)$ [1]	-
VRSC-PG (Our)	✓	$O\left(\rho^{\frac{T}{n_1+n_2+\kappa^3}}\right)$	$O((n_1+n_2)^{2/3}T^{-1})$

and (Lian, Wang, and Liu 2016) are not applicable to composition problem with nonsmooth regularization penalty.

Composition problem with nonsmooth regularization was then considered in (Wang and Liu 2016; Yu and Huang 2017). In (Wang and Liu 2016), the authors proposed accelerated stochastic compositional proximal gradient algorithm (ASC-PG). They proved that the optimal convergence rate of ASC-PG for strongly convex problem and general problem is $O(T^{-1})$ and $O(T^{-1/2})$ respectively. However, ASC-PG suffers from slow convergence because of the noise of the sampling gradients. (Yu and Huang 2017) proposed com-SVR-ADMM using variance reduction. Although com-SVR-ADMM admits linear convergence for strongly convex composition problem, it is not optimal. Besides, they did not analyze the convergence for general (nonconvex) composition problem either. We review the convergence rate of stochastic composition optimization in Table 1.

In this paper, we propose variance reduced stochastic compositional proximal gradient method (VRSC-PG) for composition problem with nonsmooth regularization penalty. Applying the variance reduction technique to composition problem is nontrivial because the optimization procedure and convergence analysis are essentially different. We investigate the convergence rate of our method: for strongly convex problem, we prove that VRSC-PG has linear convergence rate $O\left(\rho^{\frac{T}{n_1+n_2+\kappa^3}}\right)$, which is faster than com-SVR-ADMM; For general problem, sometimes nonconvex, VRSC-PG significantly improves the state-of-the-art convergence rate of ASC-PG from $O(T^{-1/2})$ to $O((n_1+n_2)^{2/3}T^{-1})$. To the best of our knowledge, our result is the new benchmark for stochastic composition optimization. We further evaluate our method by applying it to portfolio management and reinforcement learning. Experimental results verify our theoretical analysis.

Preliminary

In this section, we briefly review stochastic composition optimization and proximal stochastic variance reduced gradient.

Stochastic Composition Optimization

The objective function of the stochastic composition optimization is the composition of expected-value (1) or finite-sum (2) functions, which is much more complicated than traditional finite-sum problem. The full gradient of composition problem using chain rule is $\nabla f(x) = (\nabla G(x))^T \nabla F(G(x))$. Given x, applying the classical stochastic gradient descent method in constant queries to compute the unbiased sampling gradient $(\nabla G_j(x))^T \nabla F_i(G(x))$ is not available, when $G(x)$ is unknown yet. In problem (2), evaluating $G(x)$ is time-consuming which requires n_2 queries in each iteration. Therefore, classical SGD is not applicable to composition optimization. In (Wang, Fang, and Liu 2014), the authors proposed the first stochastic compositional gradient descent (SCGD) for minimizing the stochastic composition problem (1) with $h(x) = 0$. In their paper, they proposed to use an auxiliary variable y to approximate $G(x)$. In each iteration t, we store x_t and y_t in memory. SCGD are briefly described in Algorithm 1.

In the algorithm, α_t and β_t are learning rate. Both of them are decreasing to guarantee convergence because of the noise induced by sampling gradients. In their paper, they supposed that $x \in \mathcal{X}$. In each iteration, x is projected to $\mathcal{X}$ after step 4. Furthermore, the authors proposed Accelerated SCGD by applying Nesterov smoothing (Nesterov 1983), which is proved to converge faster than basic SCGD.

[1] In (Yu and Huang 2017) , their result is $O(\rho^{\frac{T}{n_1+n_2+Am}})$. We prove that to get linear convergence, it must be satisfied that A and m are proportional to κ^2, which is not included in their paper. Check Remark 1 in supplementary material.

Algorithm 1 SCGD

```
1: Initialize $x_0 \in \mathbb{R}^N$, $y_0 \in \mathbb{R}^M$;
2: for $t = 0, 1, 2, \ldots, T-1$ do
3:     Uniformly sample $j$ from $\{1, 2, ..., n_2\}$ with replacement and query $G_j(x_t)$ and $\nabla_j G(x_t)$;   ▷ 2 queries
4:     Update $y_{t+1}$ using:
           $y_{t+1} \leftarrow (1-\beta_t)y_t + \beta_t G_j(x_t);$   (3)
5:     Uniformly sample $i$ from $\{1, 2, ..., n_1\}$ with replacement and query $\nabla F_i(y_{t+1})$;   ▷ 1 query
6:     Update $x_{t+1}$ using:
           $x_{t+1} \leftarrow x_t - \alpha_t(\nabla G_j(x_t))^T \nabla F_i(y_{t+1});$   (4)
7: end for
```

Proximal Stochastic Variance Reduced Gradient

Stochastic variance reduced gradient (SVRG) (Johnson and Zhang 2013) was proposed to minimize finite-sum functions:

$$\min_{x \in \mathbb{R}^N} \frac{1}{n_1} \sum_{i=1}^{n_1} f_i(x), \tag{5}$$

where component functions $f_i(x) : \mathbb{R}^N \to \mathbb{R}$. In large-scale optimization, SGD and its variants use unbiased sampling gradient $\nabla f_i(x)$ as the approximation of the full gradient, which only requires one query in each iteration. However, the variance induced by sampling gradients forces us to decease learning rate to make the algorithm converge. Suppose x^* is the optimal solution to problem (5), full gradient $\frac{1}{n_1} \sum\limits_{i=1}^{n_1} \nabla f_i(x^*) = 0$, while sampling gradient $\nabla f_i(x^*) \neq 0$. We should decease learning rate, otherwise the convergence of the objective function value can not be guaranteed. However, the decreasing learning rate makes SGD converge very slow at the same time. For example, if problem (5) is strongly convex, gradient descent method (GD) converges with linear rate, while SGD converges with a learning rate at $O(T^{-1})$. Reducing the variance is one of the most important ways to accelerate SGD, and it has been widely applied to large-scale optimization (Bottou, Curtis, and Nocedal 2016; Defazio, Bach, and Lacoste-Julien 2014; Gu, Huo, and Huang 2016b; Allen-Zhu and Yuan 2016; Huo and Huang 2017; Gu, Huo, and Huang 2016a). In (Xiao and Zhang 2014), the authors considered the nonsmooth regularization penalty $h(x) \neq 0$ and proposed proximal stochastic variance reduced gradient (Proximal SVRG). Proximal SVRG is briefly described in Algorithm 2. In their paper, they used v_t as the approximation of full gradient, where $\mathbb{E} v_t = 0$. It was also proved that the variance of v_t converges to zero: $\lim\limits_{t \to \infty} \mathbb{E}\|v_t - \frac{1}{n_1} \sum\limits_{i=1}^{n_1} \nabla f_i(x_t)\|_2^2 \to 0$. Therefore, we can keep learning rate η constant in the procedure. In step 7, $\text{Prox}_{\eta h(.)}(x)$ denotes proximal operator. With the definition of proximal mapping, we have:

$$\text{Prox}_{\eta h(.)}(x) = \arg\min_{x'} (h(x') + \frac{1}{\eta}\|x' - x\|^2), \tag{6}$$

Convergence analysis and experimental results confirmed that Proximal SVRG admits linear convergence in expectation for strongly convex optimization. In (Reddi et al. 2016b), the authors proved that Proximal SVRG has sublinear convergence rate of $O(n_1^{2/3} T^{-1})$ when $f_i(x)$ is nonconvex.

Algorithm 2 Proximal SVRG

```
1: Initialize $\tilde{x}^0 \in \mathbb{R}^N$;
2: for $s = 0, 1, 2, \ldots S-1$ do
3:     $x_0^{s+1} \leftarrow \tilde{x}^s$;
4:     $f' \leftarrow \frac{1}{n_1} \sum_{i=1}^{n_1} \nabla f_i(\tilde{x}^s)$;   ▷ $n_1$ queries
5:     for $t = 0, 1, 2, \ldots, m-1$ do
6:         Uniformly sample $i$ from $\{1, 2, ..., n_1\}$ with replacement and query $\nabla f_i(x_t^{s+1})$ and $\nabla f_i(\tilde{x}^s)$;   ▷ 2 queries
7:         Update $v_t^{s+1}$ using:
               $v_t^{s+1} \leftarrow \nabla f_i(x_t^{s+1}) - \nabla f_i(\tilde{x}^s) + f';$   (7)
8:         Update model $x_{t+1}^{s+1}$ using:
               $x_{t+1}^{s+1} \leftarrow \text{Prox}_{\eta h(.)}(x_t^{s+1} - \eta v_t^{s+1});$   (8)
9:     end for
10:    $\tilde{x}^{s+1} \leftarrow x_m^{s+1}$;
11: end for
```

Variance Reduced Stochastic Compositional Proximal Gradient

In this section, we propose variance reduced stochastic compositional proximal gradient method (VRSC-PG) for solving the finite-sum composition problem with nonsmooth regularization penalty (2).

The description of VRSC-PG is presented in Algorithm 3. Similar to the framework of Proximal SVRG (Xiao and Zhang 2014), our VRSC-PG also has two-layer loops. At the beginning of the outer loop s, we keep a snapshot of the current model $\tilde{x}^s$ in memory and compute the full gradient:

$$\nabla f(\tilde{x}^s) = \frac{1}{n_2} \sum_{j=1}^{n_2} (\nabla G_j(\tilde{x}^s))^T \frac{1}{n_1} \sum_{i=1}^{n_1} \nabla F_i(G^s), \tag{9}$$

where $G^s = \frac{1}{n_2} \sum_{j=1}^{n_2} G_j(\tilde{x}^s)$ denotes the value of the inner functions and $\nabla G(\tilde{x}^s) = \frac{1}{n_2} \sum_{j=1}^{n_2} \nabla G_j(\tilde{x}^s)$ denotes the gradient of inner functions. Computing the full gradient of $f(x)$ in problem (2) requires $(n_1 + 2n_2)$ queries.

To make the number of queries in each inner iteration irrelevant to n_2, we need to keep $\widehat{G}_t^{s+1}$ and $\nabla \widehat{G}_t^{s+1}$ in memory to work as the estimates of $G(x_t^{s+1})$ and $\nabla G(x_t^{s+1})$ respectively. In our algorithm, we query $G_{A_t}(x_t^{s+1})$ and $G_{A_t}(\tilde{x}^s)$, then $\widehat{G}_t^{s+1}$ is evaluated as follows:

$$\widehat{G}_t^{s+1} = G^s - \frac{1}{A}\sum_{1\le j\le A}\left(G_{A_t[j]}(\tilde{x}^s) - G_{A_t[j]}(x_t^{s+1})\right), \tag{10}$$

where $A_t[j]$ denotes element j in the set A_t and $|A_t| = A$. The elements of A_t are uniformly sampled from $\{1, 2, ..., n_2\}$ with replacement. In (10), we reduce the variance of $G_{A_t}(x_t^{s+1})$ by using G^s and $G_{A_t}(\tilde{x}^s)$. Similarly, we sample B_t with size B from $\{1, 2, ..., n_2\}$ uniformly with replacement, and query $\nabla G_{B_t}(x_t^{s+1})$ and $\nabla G_{B_t}(\tilde{x}^s)$. The estimation of $\nabla G(x_t^{s+1})$ is evaluated as follows:

$$\begin{aligned}&\nabla\widehat{G}_t^{s+1} = \nabla G(\tilde{x}^s)\\ &-\tfrac{1}{B}\sum_{1\le j\le B}\left(\nabla G_{B_t[j]}(\tilde{x}^s) - \nabla G_{B_t[j]}(x_t^{s+1})\right)\end{aligned} \tag{11}$$

where $B_t[j]$ denotes element j in the set B_t and $|B_t| = B$. It is important to note that A_t and B_t are independent. Computing $\widehat{G}_t^{s+1}$ and $\nabla\widehat{G}_t^{s+1}$ requires $(2A + 2B)$ queries in each inner iteration.

Now, we are able to compute the estimate of $\nabla f(x_t^{s+1})$ in inner iteration t as follows:

$$\begin{aligned} v_t^{s+1} &= \frac{1}{b_1}\sum_{i_t\in I_t}\Bigg(\left(\nabla\widehat{G}_t^{s+1}\right)^T\nabla F_{i_t}(\widehat{G}_t^{s+1})\\ &\quad - \left(\nabla G(\tilde{x}^s)\right)^T\nabla F_{i_t}(G^s)\Bigg) + \nabla f(\tilde{x}^s),\end{aligned} \tag{12}$$

where I_t is a set of indexes uniformly sampled from $\{1, 2, ..., n_1\}$ and $|I_t| = b_1$. As per (12), we need to query $\nabla F_{I_t}(\hat{G}_t^{s+1})$ and $\nabla F_{I_t}(G^s)$, and it requires $2b_1$ queries. Finally, we update the model with proximal operator:

$$x_{t+1}^{s+1} = \text{Prox}_{\eta h(\cdot)}\left(x_t^{s+1} - \eta v_t^{s+1}\right), \tag{13}$$

where η is the learning rate.

Convergence Analysis

In this section, we prove that (1) VRSC-PG admits linear convergence rate for the strongly convex problem; (2) VRSC-PG admits sublinear convergence rate $O((n_1 + n_2)^{2/3}T^{-1})$ for the general problem. To the best of our knowledge, both of them are the best results so far. Following are the assumptions commonly used for stochastic composition optimization (Wang, Fang, and Liu 2014; Wang and Liu 2016; Lian, Wang, and Liu 2016).

Strongly convex: To analyze the convergence of VRSC-PG for the strongly convex composition problem, we assume that the function f is μ-strongly convex.

Assumption 1 *The function $f(x)$ is μ-strongly convex. Therefore $\forall x$ and $\forall y$, we have:*

$$\|\nabla f(x) - \nabla f(y)\| \quad \ge \quad \mu\|x - y\|. \tag{15}$$

Equivalently, μ-strongly convexity can also be written as follows:

$$f(x) \quad \ge \quad f(y) + \langle\nabla f(y), x - y\rangle + \frac{\mu}{2}\|x - y\|^2 . \tag{16}$$

Algorithm 3 VRSC-PG

```
Input: The total number of iterations in the inner loop $m$,
    the total number of iterations in the outer loop $S$, the size
    of the mini-batch sets $A$,$B$ and $b_1$, learning rate $\eta$.
Initialize $\tilde{x}^0 \in \mathbb{R}^N$;
for $s = 0, 1, 2, \cdots, S - 1$ do
    $x_0^{s+1} \leftarrow \tilde{x}^s$;
    $G^s \leftarrow \frac{1}{n_2}\sum_{j=1}^{n_2} G_i(\tilde{x}^s)$;   ▷ $n_2$ queries
    $\nabla G(\tilde{x}^s) \leftarrow \frac{1}{n_2}\sum_{j=1}^{n_2} \nabla G_j(\tilde{x}^s)$;   ▷ $n_2$ queries
    Compute the full gradient $\nabla f(\tilde{x}^s)$ using (9) ;   ▷ $n_1$ queries
    for $t = 0, 1, 2, \cdots, m - 1$ do
        Uniformly sample $A_t$ from $\{1, 2, ..., n_2\}$ with replacement and $|A_t| = A$ ;
        Update $\widehat{G}_t^{s+1}$ using (10) ;   ▷ $2A$ queries
        Uniformly sample $B_t$ from $\{1, 2, ..., n_2\}$ with replacement and $|B_t| = B$;
        Update $\nabla\widehat{G}_t^{s+1}$ using (11);   ▷ $2B$ queries
        Uniformly sample $I_t$ from $\{1, 2, ..., n_1\}$ with replacement;
        Compute $v_t^{s+1}$ using (12):   ▷ $2b_1$ queries
        Update model $x_{t+1}^{s+1}$ using:
            $x_{t+1}^{s+1} \leftarrow \text{Prox}_{\eta h(\cdot)}\left(x_t^{s+1} - \eta v_t^{s+1}\right)$   (14)
    end for
    $\tilde{x}^{s+1} \leftarrow x_m^{s+1}$;
end for
```

Lipschitz Gradient: We assume that there exist Lipschitz constants L_F, L_G and L_f for $\nabla F_i(x)$, $\nabla G_j(x)$ and $\nabla f(x)$ respectively.

Assumption 2 *There exist constants L_F, L_G and L_f for $\nabla F_i(x)$, $\nabla G_j(x)$ and $\nabla f(x)$ satisfying that $\forall x$, $\forall y$, $\forall i \in \{1, \cdots, n_1\}$, $\forall j \in \{1, \cdots, n_2\}$:*

$$\|\nabla F_i(x) - \nabla F_i(y)\| \quad \le \quad L_F\|x - y\|, \tag{17}$$

$$\|\nabla G_j(x) - \nabla G_j(y)\| \quad \le \quad L_G\|x - y\|, \tag{18}$$

$$\begin{aligned}\|\left(\nabla G_j(x)\right)^T\nabla F_i(G(x)) \quad &- \quad \left(\nabla G_j(y)\right)^T\nabla F_i(G(y))\|\\ &\le \quad L_f\|x - y\|.\end{aligned} \tag{19}$$

As proved in (Lian, Wang, and Liu 2016), according to (19), we have:

$$\|\nabla f(x) - \nabla f(y)\| \quad \le \quad L_f\|x - y\|, \forall x, \forall y. \tag{20}$$

Equivalently, (20) can also be written as follows: $\forall x, \forall y$, we have

$$f(x) \le f(y) + \langle\nabla f(y), x - y\rangle + \frac{L_f}{2}\|x - y\|^2, \tag{21}$$

Bounded gradients: We assume that the gradients $\nabla F_i(x)$ and $\nabla G_j(x)$ are upper bounded.

Assumption 3 *The gradients $\nabla F_i(x)$ and $\nabla G_j(x)$ have upper bounds B_F and B_G respectively.*

$$\|\nabla F_i(x)\| \quad \le \quad B_F, \forall x, \forall i \in \{1, \cdots, n_1\} \tag{22}$$

$$\|\nabla G_j(x)\| \quad \le \quad B_G, \forall x, \forall j \in \{1, \cdots, n_2\} \tag{23}$$

Note that we do not need the strong convexity assumption when we analyze the convergence of VRSC-PG for the general problem.

Strongly Convex Problem

In this section, we prove that our VRSC-PG admits linear convergence rate for strongly convex finite-sum composition problem with nonsmooth penalty regularization (2). We need Assumptions 1, 2 and 3 in this section. Unlike Prox-SVRG in (Xiao and Zhang 2014), the estimated v_t^{s+1} is biased, i.e., $\mathbb{E}_{I_t,A_t,B_t}[v_t^{s+1}] \neq \nabla f(x_t^{s+1})$. It makes the theoretical analysis for proving the convergence rate of VRSC-PG more challenging than the analysis in (Xiao and Zhang 2014). In spite of this, we can demonstrate that $\mathbb{E}\|v_t^{s+1} - \nabla f(x_t^{s+1})\|^2$ is upper bounded as well.

Lemma 1 *Let x^* be the optimal solution to problem (2) $H(x)$ such that $x^* = \arg\min_{x\in\mathbb{R}^N} H(x)$. We define $\gamma = \left(\frac{64}{\mu}\left(\frac{B_F^2 L_G^2}{B} + \frac{B_G^4 L_F^2}{A}\right) + 8L_f\right)$. Supposing Assumptions 1, 2 and 3 hold, from the definition of v_t^{s+1} in (12), the following inequality holds that:*

$$\mathbb{E}\|v_t^{s+1} - \nabla f(x_t^{s+1})\|^2$$
$$\leq \gamma\bigg[H(x_t^{s+1}) - H(x^*) + H(\tilde{x}^s) - H(x^*)\bigg]. \quad (24)$$

Therefore, when x_t^{s+1} and $\tilde{x}^s$ converges to x^*, $\mathbb{E}\|v_t - \nabla f(x_t^{s+1})\|^2$ also converges to zero. Thus, we can keep learning rate constant, and obtain faster convergence.

Theorem 1 *Suppose Assumptions 1, 2 and 3 hold. We let the optimal solution $x^* = \arg\min_{x\in\mathbb{R}^N} H(x)$, if m, A, B and η are selected properly so that $\rho < 1$, where ρ is defined as follows:*

$$\rho = \frac{\frac{2}{\mu} + 2\eta\left(6\eta L_f + \rho_c\right)(m+1)}{2\eta\left(\frac{7}{8} - \left(6\eta L_f + \rho_c\right)\right)m} \quad (25)$$

$$\rho_c = \left(\frac{\eta}{2} + \frac{4}{\mu}\right)\frac{32}{\mu}\left(\frac{B_F^2 L_G^2}{B} + \frac{B_G^4 L_F^2}{A}\right) \quad (26)$$

we can prove that our VRSC-PG admits linear convergence rate:

$$\mathbb{E}H(\tilde{x}^S) - H(x^*) \leq \rho^S\bigg(\mathbb{E}H(\tilde{x}^0) - H(x^*)\bigg) \quad (27)$$

As per Theorem 1, we need to choose η, m, A and B properly to make $\rho < 1$. We provide an example to show how to select these parameters.

Corollary 1 *According to Theorem 1, we set η, m, A and B as follows:*

$$\eta = \frac{1}{96L_f} \quad (28)$$

$$m = 16\left(1 + \frac{96L_f}{\mu}\right) \quad (29)$$

$$A = \frac{2048 B_G^4 L_F^2}{\mu^2} \quad (30)$$

$$B = \frac{2048 B_F^2 L_G^2}{\mu^2} \quad (31)$$

we have the following linear convergence rate for VRSC-PG:

$$\mathbb{E}H(\tilde{x}^S) - H(x^*) \leq \left(\frac{2}{3}\right)^S\bigg(\mathbb{E}H(\tilde{x}^0) - H(x^*)\bigg) \quad (32)$$

Remark 1 *According to Theorem 1, to obtain*

$$\mathbb{E}H(\tilde{x}^s) - H(x^*) \leq \varepsilon \quad (33)$$

the number of stages S is required to satisfy:

$$S \geq \log\frac{\mathbb{E}H(\tilde{x}^0) - H(x^*)}{\varepsilon} / \log\frac{1}{\rho} \quad (34)$$

As per Algorithm 3 and the definition of Sampling Oracle in (Wang and Liu 2016), to make the objective value gap $\mathbb{E}H(\tilde{x}^s) - H(x^*) \leq \varepsilon$, the total query complexity we need to take is $O\bigg(\left(n_1+n_2+m(A+B+b_1)\right)\log(\frac{1}{\varepsilon})\bigg) = O\bigg((n_1+n_2+\kappa^3)\log(\frac{1}{\varepsilon})\bigg)$, where we let $\kappa = \max\left\{\frac{L_f}{\mu}, \frac{L_F}{\mu}, \frac{L_G}{\mu}\right\}$ and b_1 can be smaller than or proportional to κ^2. It is better than com-SVR-ADMM(Yu and Huang 2017) whose total query complexity is $O\bigg((n_1+n_2+\kappa^4)\log(\frac{1}{\varepsilon})\bigg)$.

General Problem

In this section, we prove that VRSC-PG admits a sublinear convergence rate $O(T^{-1})$ for the general finite-sum composition problem with nonsmooth regularization penalty. It is much better than the state-of-the-art method ASC-PG (Wang and Liu 2016) whose optimal convergence rate is $O(T^{-1/2})$. In this section, we only need Assumption 2 and 3. The unbiased v_t^{s+1} makes our analysis nontrivial and it is much different from previous analysis for finite-sum problem (Reddi et al. 2016a). In our proof, we define:

$$\mathcal{G}_\eta(x) = \frac{1}{\eta}\left(x - \text{Prox}_{\eta h(.)}(x - \nabla f(x))\right). \quad (35)$$

Theorem 2 *Suppose Assumptions 2 and 3 hold. Let x^* be the optimal solution to problem (2), we have $x^* = \arg\min_{x\in\mathbb{R}^N} H(x)$. If m, A, B, b_1 and η are selected properly such that:*

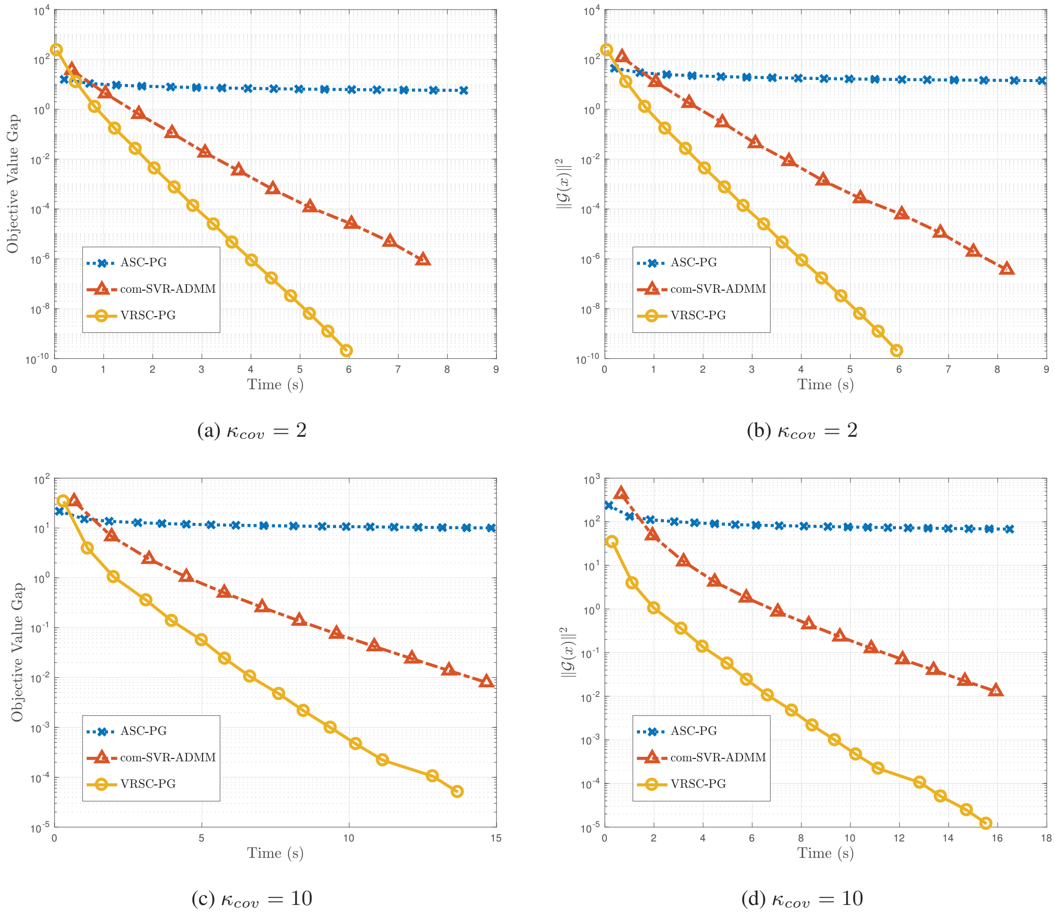
$$4\bigg(\frac{\eta m^2 L_f^2}{b_1} + \frac{2\eta m^2 B_G^4 L_F^2}{A} + \frac{2\eta m^2 B_F^2 L_G^2}{B}\bigg)$$
$$+\frac{L_f}{2} \leq \frac{1}{2\eta}, \quad (36)$$

then the following inequality holds that:

$$\mathbb{E}\|\mathcal{G}_\eta(x_a)\|^2 \leq \frac{2}{(1-2\eta L_f)\eta}\frac{H(\tilde{x}^0) - H(x^*)}{T} \quad (37)$$

where x_a is uniformly selected from $\{\{x_t^{s+1}\}_{t=0}^{m-1}\}_{t=0}^{S-1}$ and T is a multiple of m,

As per Theorem 2, we need to choose m, A, B, b_1 and η appropriately to make condition (36) satisfied. We provide an example to show how to select these parameters.



(a) $\kappa_{cov} = 2$

(b) $\kappa_{cov} = 2$

(c) $\kappa_{cov} = 10$

(d) $\kappa_{cov} = 10$

Figure 1: Experimental results for meaning-variance portfolio management on synthetic data. κ_{cov} is the conditional number of the covariance matrix of the corresponding Gaussian distribution which is used to generate reward. We use time as x axis, and it is proportional to the query complexity. In y axis, the objective value gap is defined as $H(x) - H(x^*)$, where x^* is obtained by running our methods for enough iterations until convergence. $\|\mathcal{G}(x)\|^2$ denotes the ℓ_2-norm of the full gradient, where $\mathcal{G}(x) = \nabla f(x) + \partial h(x)$.

Corollary 2 *According to Theorem 2, we let* $m = \left\lfloor (n_1 + n_2)^{\frac{1}{3}} \right\rfloor$, $\eta = \frac{1}{4L_f}$, $b_1 = (n_1 + n_2)^{\frac{2}{3}}$ *and* T *be a multiple of* m*, it is easy to know that if* A *and* B *are lower bounded:*

$$A \geq \frac{8m^2 B_G^4 L_F^2}{L_f} \tag{38}$$

$$B \geq \frac{8m^2 B_F^2 L_G^2}{L_f} \tag{39}$$

we can obtain sublinear convergence rate for VRSC-PG:

$$\mathbb{E}\|\mathcal{G}_\eta(x_a)\|^2 \leq 16L_f \frac{H(\tilde{x}^0) - H(x^*)}{T} \tag{40}$$

Remark 2 *According to Theorem 2, to obtain*

$$\mathbb{E}\|\mathcal{G}_\eta(x_a)\|^2 \leq \varepsilon \tag{41}$$

the number of iterations T *is required to satisfy:*

$$T \geq 16L_f \frac{\mathbb{E}H(\tilde{x}^0) - H(x^*)}{\varepsilon} \tag{42}$$

As per Algorithm 3 and the definition of Sampling Oracle in (Wang and Liu 2016), to obtain ε-accurate solution, $\mathbb{E}\|\mathcal{G}_\eta(x_a)\|^2 \leq \varepsilon$, the total query complexity we need to take is $O(n_1 + n_2 + \frac{A+B+b_1}{\varepsilon}) = O\left(n_1 + n_2 + \frac{(n_1+n_2)^{2/3}}{\varepsilon}\right)$, where A, B and b_1 are proportional to $(n_1 + n_2)^{\frac{2}{3}}$. Therefore, our method improves the state-of-the-art convergence rate of stochastic composition optimization for general problem from $O(T^{-1/2})$ (Optimal convergence rate for ASC-PG) to $O\left((n_1 + n_2)^{2/3} T^{-1}\right)$.

Experimental Results

We conduct two experiments to evaluate our proposed method: (1) application to portfolio management; (2) application to policy evaluation in reinforcement learning.

In the experiments, there are three compared methods for stochastic composition optimization:

- Accelerated stochastic compositional proximal gradient (ASC-PG) (Wang and Liu 2016);
- Stochastic variance reduced ADMM for Stochastic composition optimization (com-SVR-ADMM) (Yu and Huang 2017);
- Variance Reduced Stochastic Compositional Proximal Gradient (VRSC-PG)(Our method).

In our experiments, learning rate η is tuned from $\{1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. We keep the learning rate constant for com-SVR-ADMM and VRSC-PG in the optimization. For ASC-PG, in order to guarantee convergence, learning rate is decreased as per $\frac{\eta}{1+t}$, where t denotes the number of iterations.

Application to Portfolio Management

Suppose there are N assets we can invest, $r_t \in \mathbb{R}^N$ denotes the rewards of N assets at time t. Our goal is to maximize the return of the investment and to minimize the risk of the investment at the same time. Portfolio management problem can be formulated as the mean-variance optimization as follows:

$$\min_{x\in\mathbb{R}^N} -\frac{1}{n}\sum_{t=1}^{n}\langle r_t, x\rangle + \frac{1}{n}\sum_{t=1}^{n}\left(\langle r_t, x\rangle - \frac{1}{n}\sum_{j=1}^{n}\langle r_j, x\rangle\right)^2 \quad (43)$$

where $x \in \mathbb{R}^N$ denotes the investment quantity vector in N assets. According to (Lian, Wang, and Liu 2016), problem (43) can also be viewed as the composition problem as (2). In our experiment, we also add a nonsmooth regularization penalty $h(x) = \lambda|x|$ in the mean-variance optimization problem (43).

Similar to the experimental settings in (Lian, Wang, and Liu 2016), we let $n = 2000$ and $N = 200$. Rewards r_t are generated in two steps: (1) Generate a Gaussian distribution on $\mathbb{R}^N$, where we define the condition number of its covariance matrix as κ_{cov}. Because κ_{cov} is proportional to κ, in our experiment, we will control κ_{cov} to change the value of κ; (2) Sample rewards r_t from the Gaussian distribution and make all elements positive to guarantee that this problem has a solution. In the experiment, we compared three methods on two synthetic datasets, which are generated through Gaussian distributions with $\kappa_{cov} = 2$ and $\kappa_{cov} = 10$ separately. We set $\lambda = 10^{-3}$ and $A = B = b_1 = 5$. We just select the values of A, B, b_1 casually, it is probable that we can get better results as long as we tune them carefully.

Figure 1 shows the convergence of compared methods regarding time. We suppose that the elapsed time is proportional to the query complexity. Objective value gap means $H(x_t) - H(x^*)$, where x^* is the optimal solution to $H(x)$. We compute $H(x^*)$ by running our method until convergence. Firstly, by observing the x and y axises in Figure 1, we can know that when $\kappa_{cov} = 10$, all compared methods need more time to minimize problem (43), which is consistent with our analysis. Increasing κ will increase the total query complexity. Secondly, we can also find out that com-SVR-ADMM and VRSC-PG admit linear convergence rate. ASC-PG runs faster at the beginning, because of their low query complexity in each iteration. However, their convergence slows down when the learning rate gets small. In four figures, our SVRC-PG always has the best performance compared to other compared methods.

Application to Reinforcement Learning

We then apply stochastic composition optimization to reinforcement learning and evaluate three compared methods in the task of policy evaluation. In reinforcement learning, let $V^\pi(s)$ be the value of state s under policy π. The value function $V^\pi(s)$ can be evaluated through Bellman equation as follows:

$$V^\pi(s_1) = \mathbb{E}[r_{s_1,s_2} + \gamma V^\pi(s_2)|s_1] \quad (44)$$

for all $s_1, s_2 \in \{1, 2, ..., S\}$, where S represents the number of total states. According to (Wang and Liu 2016), the Bellman equation (44) can also be written as a composition problem. In our experiment, we also add sparsity regularization $h(x) = \lambda|x|$ in the objective function.

Following (Dann, Neumann, and Peters 2014), we generate a Markov decision process (MDP). There are 400 states and 10 actions at each state. The transition probability is generated randomly from the uniform distribution in the range of $[0, 1]$. We then add 10^{-5} to each element of transition matrix to ensure the ergodicity of our MDP. The rewards $r(s, s')$ from state s to state s' are also sampled uniformly in the range of $[0, 1]$. In our experiment, we set $\lambda = 10^{-3}$ and $A = B = b_1 = 5$. We also select these values casually, better results can be obtained if we tune them carefully.

In Figure 2, we plot the convergence of the objective value and $\|\mathcal{G}(x)\|^2$ in terms of time. We can observe that VRSC-PG is much faster than ASC-PG, which has been reflected in the analysis of convergence rate already. It is also obvious that our VRSC-PG converges faster than com-SVR-ADMM. Experimental results on policy evaluation also verify our theoretical analysis.

Conclusion

In this paper, we propose variance reduced stochastic compositional proximal gradient method (VRSC-PG) for composition problem with nonsmooth regularization penalty. We also analyze the convergence rate of our method: (1) for strongly convex composition problem, VRSC-PG is proved to admit linear convergence; (2) for general composition problem, VRSC-PG significantly improves the state-of-the-art convergence rate from $O(T^{-1/2})$ to $O((n_1 + n_2)^{2/3}T^{-1})$. Both of our theoretical analysis, to the best of our knowledge, are the state-of-the-art results for stochastic composition optimization. Finally, we apply our method to two different applications, portfolio management and reinforcement learning. Experimental results show that our method always has the best performance in different cases and verify the conclusions of theoretical analysis.

Acknowledgement

Z. H., B. G., H. H. were partially supported by the following grants: NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, NSF-IIS 1619308, NSF-IIS 1633753, NIH R01 AG049371. J. L. was partially supported by NSF-CCF 1718513.

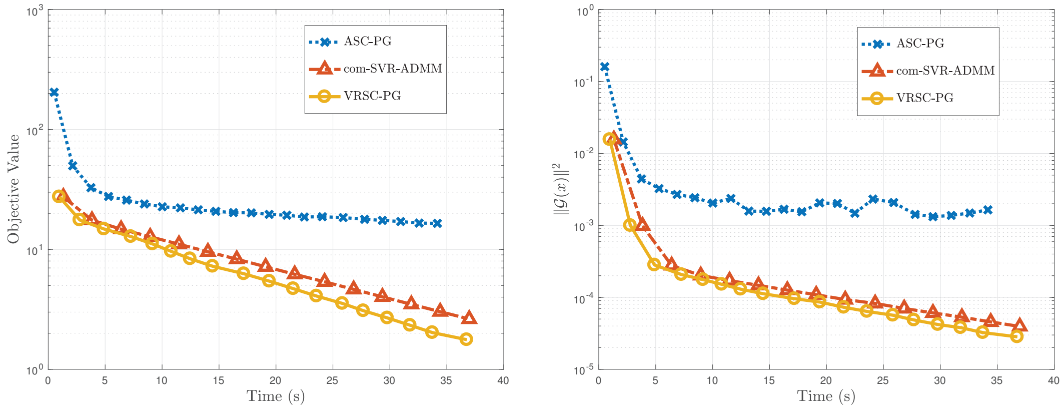


Figure 2: Figures show the experimental results of policy evaluation in reinforcement learning. We plot the convergence of objective value and the full gradient $\|\mathcal{G}(x)\|^2$ regarding time respectively. $\|\mathcal{G}(x)\|^2$ denotes the ℓ_2-norm of the full gradient, where $\mathcal{G}(x) = \nabla f(x) + \partial h(x)$.

References

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*, 1080–1089.

Bottou, L.; Curtis, F. E.; and Nocedal, J. 2016. Optimization methods for large-scale machine learning. *arXiv preprint arXiv:1606.04838*.

Dai, B.; He, N.; Pan, Y.; Boots, B.; and Song, L. 2016. Learning from conditional distributions via dual kernel embeddings. *arXiv preprint arXiv:1607.04579*.

Dann, C.; Neumann, G.; and Peters, J. 2014. Policy evaluation with temporal differences: a survey and comparison. *Journal of Machine Learning Research* 15(1):809–883.

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.

Dentcheva, D.; Penev, S.; and Ruszczyński, A. 2016. Statistical estimation of composite risk functionals and risk optimization problems. *Annals of the Institute of Statistical Mathematics* 1–24.

Gu, B.; Huo, Z.; and Huang, H. 2016a. Asynchronous stochastic block coordinate descent with variance reduction. *arXiv preprint arXiv:1610.09447*.

Gu, B.; Huo, Z.; and Huang, H. 2016b. Zeroth-order asynchronous doubly stochastic algorithm with variance reduction. *arXiv preprint arXiv:1612.01425*.

Huo, Z., and Huang, H. 2017. Asynchronous mini-batch gradient descent with variance reduction for non-convex optimization. In *AAAI*, 2043–2049.

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

Lian, X.; Wang, M.; and Liu, J. 2016. Finite-sum composition optimization via variance reduced gradient descent. *arXiv preprint arXiv:1610.04674*.

Nesterov, Y. 1983. A method for unconstrained convex minimization problem with the rate of convergence o (1/k2). In *Doklady an SSSR*, volume 269, 543–547.

Reddi, S. J.; Sra, S.; Poczos, B.; and Smola, A. 2016a. Fast stochastic methods for nonsmooth nonconvex optimization. *arXiv preprint arXiv:1605.06900*.

Reddi, S. J.; Sra, S.; Poczos, B.; and Smola, A. J. 2016b. Proximal stochastic methods for nonsmooth nonconvex finite-sum optimization. In *Advances in Neural Information Processing Systems*, 1145–1153.

Wang, M., and Liu, J. 2016. Accelerating stochastic composition optimization. In *Advances In Neural Information Processing Systems*, 1714–1722.

Wang, M.; Fang, E. X.; and Liu, H. 2014. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *arXiv preprint arXiv:1411.3803*.

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

Yu, Y., and Huang, L. 2017. Fast stochastic variance reduced admm for stochastic composition optimization. *arXiv preprint arXiv:1705.04138*.