Journal of Machine Learning Research 18 (2017) 1-23 Submitted 10/16; Revised 6/17; Published 10/17

Accelerating Stochastic Composition Optimization

Mengdi Wang mengdiw@princeton.edu
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Ji Liu ji.liu.uwisc@gmail.com
Department of Computer Science and Department of Electrical and Computer Engineering
University of Rochester
Rochester, NY 14627, USA

Ethan X. Fang xxf13@psu.edu
Department of Statistics and Department of Industrial and Manufacturing Engineering
Pennsyvania State University
University Park, PA 16802, USA

Editor: Inderjit Dhillon

Abstract

We consider the stochastic nested composition optimization problem where the objective is a composition of two expected-value functions. We propose a new stochastic first-order method, namely the accelerated stochastic compositional proximal gradient (ASC-PG) method. This algorithm updates the solution based on noisy gradient queries using a two-timescale iteration. The ASC-PG is the first proximal gradient method for the stochastic composition problem that can deal with nonsmooth regularization penalty. We show that the ASC-PG exhibits faster convergence than the best known algorithms, and that it achieves the optimal sample-error complexity in several important special cases. We demonstrate the application of ASC-PG to reinforcement learning and conduct numerical experiments.

Keywords: Large-scale optimization, stochastic gradient, composition optimization, sample complexity.

1. Introduction

The popular stochastic gradient methods are well suited for minimizing expected-value objective functions or the sum of a large number of loss functions. Stochastic gradient methods find wide applications in estimation, online learning, and training of deep neural networks. Despite their popularity, they do not apply to the minimization of a nonlinear function involving expected values or a composition between two expected-value functions.

In this paper, we consider the stochastic composition problem, given by

$$\min_{x \in X} \quad H(x) := \underbrace{E_v(f_v(E_w(g_w(x))))}_{=:F(x)} + R(x) \tag{1}$$

. Both authors contributed equally.

where $f(g(x)) = (f \circ g)(x)$ denotes the function composition, $g_w(\cdot) : <^n \to <^m$ and $f_v(\cdot) : <^m \to <$ are continuously differentiable functions, $v; w$ are random variables, and $R(x) : <^n \to < [f+1 g$ is an extended real-valued closed convex function, X is a convex and closed set. We assume throughout that there exists at least one optimal solution $x \in X$ to problem (1). We focus on the case where f_v and g_w are smooth, but we allow R to be a nonsmooth penalty such as the $'_1$-norm. We do no require either the outer function f_v or the inner function g_w to be convex or monotone. As a result, the composition problem cannot be reformulated into a saddle point problem in general.

Our algorithmic objective is to develop efficient algorithms for solving problem (1) based on random evaluations of f_v, g_w and their gradients. Our theoretical objective is to analyze the rate of convergence for the stochastic algorithm and to improve it when possible. In the online setting, the iteration complexity of our stochastic methods can be interpreted as a sample-error complexity upper bound for estimating the optimal solution of problem (1).

1.1 Motivating Examples

One motivating example is reinforcement learning (Sutton and Barto, 1998). Consider a controllable Markov chain with states $1; : : : ; S$. Estimating the value-per-state of a fixed control policy is known as on-policy learning. It can be casted into an $S \times S$ system of Bellman equations:

$$P V + r = V;$$

where $\in (0; 1)$ is a discount factor, P_{ss} is the transition probability from state s to state s, and r_s is the expected state transition reward at state s. The solution V to the Bellman equation is the value vector, with $V(s)$ being the total expected reward starting at state s. In the blackbox simulation environment, $P; r$ are unknown but can be sampled from a simulator. As a result, solving the Bellman equation becomes a special case of the stochastic composition optimization problem:

$$\min_{x \in X} \quad kE[A]x \quad E[b]k^2; \tag{2}$$

where $A; B; b$ are random matrices and random vectors such that $E[A] = I \quad P$ and $E[b] = r$. It can be viewed as the composition of the square norm function $f(\cdot) \quad f_v(\cdot) = kk$ and the expected linear function $g(x) = E[A]x \quad E[b]$. We will give more details on the reinforcement learning application in Section 4.

Another motivating example is risk-averse learning. For example, consider the mean-variance minimization problem

$$\min_{x \in X} \quad E_{a;b}[h(x; a; b)] + \mathrm{Var}_{a;b}[h(x; a; b)];$$

where $h(x; a; b)$ is some loss function parameterized by random variables a and b, and > 0 is a regularization parameter. In particular, the mean-variance minimization takes the composition form where $g(x) = (E[x]; E[h(x; a; b)])$, and $f(y_1; y_2) = E[h(y_1; a; b)] + E\ h(y_1; a; b) \quad y_2\ ^2$. Its batch version takes the form

$$\min_{x \in X} \quad \frac{1}{N} \sum_{i=1}^{N} h(x; a_i; b_i) + \frac{}{N} \sum_{i=1}^{N} h(x; a_i; b_i) \quad \frac{1}{N} \sum_{i=1}^{N} h(x; a_i; b_i) \Bigg)^2 :$$

Here the variance term is the composition of the mean square function and an expected loss function.

Although the stochastic composition problem (1) was barely studied, it actually finds a broad spectrum of emerging applications in estimation and machine learning (see Wang et al. (2017) for a list of applications). Fast optimization algorithms with theoretical guarantees will lead to new computation tools and online learning methods for a broader problem class, no longer limited to the expectation minimization problem.

1.2 Related Works and Contributions

Contrary to the expectation minimization problem, "unbiased" gradient samples are no longer available for the stochastic composition problem (1). The objective is nonlinear in the joint probability distribution of $(w; v)$, which substantially complicates the problem. In a recent work by Dentcheva et al. (2016), a special case of the stochastic composition problem, i.e., risk-averse optimization, has been studied. A central limit theorem has been established, showing that the K-sample batch problem converges to the true problem at the rate of $O(1=\sqrt{K})$ in a proper sense. For the case where $R(x) = 0$, Wang et al. (2017) has proposed and analyzed a class of stochastic compositional gradient/subgradient methods (SCGD). The SCGD involves two iterations of different time scales, one for estimating x by a stochastic quasi-gradient iteration, the other for maintaining a running estimate of $g(x)$. Wang and Liu (2016) studies the SCGD in the setting where samples are corrupted with Markov noises (instead of i.i.d. zero-mean noises). Both works establish almost sure convergence of the algorithm and several convergence rate results, which are the best-known convergence rate prior to the current paper.

The idea of using two-timescale quasi-gradient traced back to the earlier work Ermoliev (1976). It proposed for the first time a stochastic iteration that updates the solution and an auxiliary variable using two different stepsizes. The incremental treatment of proximal gradient iteration has been studied extensively for the expectation minimization problem, see for examples Nedić and Bertsekas (2001); Nemirovski et al. (2009); Wang et al. (2015); Wang and Bertsekas (2016) and references therein. However, except for Wang et al. (2017) and Wang and Liu (2016), all of these works focus on variants of the expectation minimization problem and do not apply to the stochastic composition problem (1). Another work partially related to this paper is by Dai et al. (2017). They consider a special case of problem (1) arising in kernel estimation, where they assume that all functions f_v's are convex and their conjugate functions f_v^*'s can be easily obtained/sampled. Under these additional assumptions, they essentially rewrite the problem into a saddle point optimization involving functional variables.

In this paper, we propose a new accelerated stochastic compositional proximal gradient (ASC-PG) method that applies to the penalized problem (1), which is a more general problem than the one considered in Wang et al. (2017). We use a coupled martingale stochastic analysis to show that ASC-PG achieves significantly better sample-error complexity in various cases. We also show that ASC-PG exhibits optimal sample-error complexity in two important special cases: the case where the outer function is linear and the case where the inner function is linear. Note that the current work has a preliminary conference version (Wang et al., 2016).

The current journal version provides more complete technical details as well as important theoretical extensions (Theorem 3).

Our major contributions are summarized as follows:

1. We propose the first stochastic proximal-gradient method for the stochastic composition problem. This is the first algorithm that is able to address the nonsmooth regularization penalty $R()$ without deteriorating the convergence rate.

2. We obtain a convergence rate $O(K^{-4=9})$ for smooth optimization problems that are not necessarily convex, where K is the number of queries to the stochastic first-order oracle. This improves the best known convergence rate and provides a new benchmark for the stochastic composition problem.

3. We provide a comprehensive analysis and results that apply to various special cases. In particular, our results contain as special cases the known optimal rate results for the expectation minimization problem, i.e., $O(1=\sqrt{K})$ for general objectives and $O(1=K)$ for strongly convex objectives.

4. In the special case where the inner function $g()$ is a linear mapping, we show that it is sufficient to use one timescale to guarantee convergence. Our result achieves the non-improvable rate of convergence $O(1=K)$ for optimal strongly convex optimization and $O(1=\sqrt{K})$ for convex optimization and nonconvex smooth optimization. It implies that the inner linearity does not bring fundamental difficulty to the stochastic composition problem.

5. We show that the proposed method leads to a new on-policy reinforcement learning algorithm. The new learning algorithm achieves the optimal convergence rate $O(1=\sqrt{K})$ for solving Bellman equations based on K observations of state-to-state transitions.

In comparison with Wang et al. (2017), our analysis is more succinct and leads to stronger results. To the best of our knowledge, Theorems 1,2,3 in this paper provide the best-known rates for the stochastic composition problem.

Paper Organization. Section 2 states the sampling oracle and the accelerated stochastic compositional proximal gradient algorithm (ASC-PG). Section 3 states the convergence rate results in the case of general nonconvex objective and in the case of strongly convex objective, respectively. Section 4 describes an application of ASC-PG to reinforcement learning and gives numerical experiments.

Notations and Definitions. For $x \in \mathbb{R}^n$, we denote by x^0 its transpose, and by $\|x\|$ its Euclidean norm (i.e., $\|x\| = \sqrt{x^0 x}$). For two sequences $\{y_k\}$ and $\{z_k\}$, we write $y_k = O(z_k)$ if there exists a constant $c > 0$ such that $\|y_k\| \le c\|z_k\|$ for each k. We denote by $\mathbb{I}_{\text{condition}}^{\text{value}}$ the indicator function, which returns "value" if the "condition" is satisfied; otherwise 0. We denote by H the optimal objective function value of problem (1), denote by X the set of optimal solutions, and denote by $P_S(x)$ the Euclidean projection of x onto S for any convex set S. We also denote by short that $f(y) = E_v[f_v(y)]$ and $g(x) = E_w[g_w(x)]$.

2. Algorithm

We focus on the black-box sampling environment. Suppose that we have access to a stochastic first-order oracle, which returns random realizations of first-order information upon queries. This is a typical simulation oracle that is available in both online and batch learning. More specifically, assume that we are given a Sampling Oracle (SO) such that

Given some $x \in \mathbb{R}^n$, the SO returns a random vector $g_w(x)$ and a noisy subgradient $\nabla g_w(x)$:

Given some $y \in \mathbb{R}^m$, the SO returns a noisy gradient $\nabla f_v(y)$:

Now we propose the Accelerated Stochastic Compositional Proximal Gradient (ASC-PG) algorithm, see Algorithm 1. ASC-PG is a generalization of the SCGD proposed by Wang et al. (2017), in which a proximal step is used to replace the projection step.

Algorithm 1 Accelerated Stochastic Compositional Proximal Gradient (ASC-PG)

Require: $x_1 \in \mathbb{R}^n$, SO, K, stepsize sequences $\{\alpha_k\}_{k=1}^K$ and $\{\beta_k\}_{k=1}^K$
Ensure: $\{x_k\}_{k=1}^K$

1: Initialize $z_1 = x_1$ and $y_1 = 0$.
2: **for** $k = 1, \ldots, K$ **do**
3: Query the SO and obtain gradient samples $\nabla f_{v_k}(y_k)$, $\nabla g_{w_k}(z_k)$:
4: Update the main iterate by

$$x_{k+1} = \text{prox}_{\alpha_k R(\cdot)}\left(x_k - \alpha_k \nabla g_{w_k}(z_k) \nabla f_{v_k}(y_k)\right):$$

5: Update auxillary iterates by an extrapolation-smoothing scheme:

$$\begin{aligned} z_{k+1} &= \left(1 - \frac{1}{\beta_k}\right) x_k + \frac{1}{\beta_k} x_{k+1}; \; y_{k+1} \\ &= (1 - \beta_k) y_k + \beta_k g_{w_{k+1}}(z_{k+1}); \end{aligned}$$

where the sample $g_{w_{k+1}}(z_{k+1})$ is obtained via querying the SO.
6: **end for**

In Algorithm 1, the extrapolation-smoothing scheme (i.e., the $(y; z)$-step) is critical to the acceleration of convergence. The acceleration is due to the fast running estimation of the unknown quantity $g(x_k) := E_w[g_w(x_k)]$. At iteration k, the running estimate y_k of $g(x_k)$ is obtained using a weighted smoothing scheme, corresponding to the y-step; while the new query point z_{k+1} is obtained through extrapolation, corresponding to the z-step. The updates are constructed in a way such that y_k is a nearly unbiased estimate of $g(x_k)$: To see how the extrapolation-smoothing scheme works, we let the weights be

$$\Gamma_t^{(k)} = \begin{cases} \beta_k \prod_{i=t+1}^{k-1}(1 - \beta_i), & \text{if } k > t \geq 0 \\ \beta_k, & \text{if } k = t \geq 0. \end{cases} \tag{3}$$

Then,we can verify the following important relations:

$$x_{k+1} = \sum_{t=0}^{k} {}^{(k)}_{t} z_{t+1}; \qquad y_{k+1} = \sum_{t=0}^{k} {}^{(k)}_{t} g_{w_{t+1}}(z_{t+1});$$

which essentially say that x_{k+1} is a damped weighted average of $\{z_{t+1}\}_0^{k+1}$ and y_{k+1} is a damped weighted average of $\{g_{w_{t+1}}(z_{t+1})\}_0^{k+1}$.

An Analytical Example of the Extrapolation-Smooth Scheme Now consider the special case where $g_w()$ is always a linear mapping $g_w(z) = A_w z + b_z$ and $_k = 1=(k+1)$. We can verify that ${}^{(k)}_{t} = 1=(k+1)$ for all t. Then we have

$$g(x_{k+1}) = \frac{1}{k+1} \sum_{t=0}^{k} E[A_w] z_{t+1} + E[b_w]; \qquad y_{k+1} = \frac{1}{k+1} \sum_{t=0}^{k} A_{w_{t+1}} z_{t+1} + \frac{1}{k+1} \sum_{t=0}^{k} b_{w_{t+1}}:$$

In this way, we can see that the scaled error

$$k(y_{k+1} \quad g(x_{k+1})) = \sum_{t=0}^{k} (A_{w_{t+1}} \quad E[A_w]) z_{t+1} + \sum_{t=0}^{k} (b_{w_{t+1}} \quad E[b_w])$$

is a zero-mean and zero-drift martingale. Under additional technical assumptions, we have

$$E[k y_{k+1} \quad g(x_{k+1})k^2] \quad O\left(\frac{1}{k}\right):$$

Note that the zero-drift property of the error martingale is the key to the fast convergence rate. The zero-drift property comes from the near-unbiasedness of y_k, which is due to the special construction of the extrapolation-smoothing scheme. In the more general case where g_w is not necessarily linear, we can use a similar argument to show that y_k is a nearly unbiased estimate of $g(x_k)$. As a result, the extrapolation-smoothing (y; z)-step ensures that y_k tracks the unknown quantity $g(x_k)$ efficiently.

3. Main Results

We present our main theoretical results in this section. Note that for ease of presentation, we defer all detailed proofs for the theorems, and technical lemmas to Appendix. Let us begin by stating our assumptions. Note that all assumptions involving random realizations of v; w hold with probability 1.

Assumption 1 The samples generated by the SO are unbiased in the following sense:

1. $E_{\{w_k; v_k\}}(r g^{>}_{w_k}(x) r f_{v_k}(y)) = r g^{>}(x) r f(y) \quad 8k = 1; 2; ; K; \qquad 8x; 8y.$
2. $E_{w_k}(g_{w_k}(x)) = g(x) \quad 8x.$

Note that w_k and v_k are not necessarily independent.

Assumption 2 The sample gradients and values generated by the SO satisfy

$$E_w(k g_w(x) \quad g(x)k^2) \quad {}^2 \qquad 8x:$$

Assumption 3 The sample gradients generated by the SO are uniformly bounded, and the penalty function R has bounded gradients.

$$\|\nabla f_v(x)\| \le \Gamma_1, \quad \|\nabla g_w(x)\| \le \Gamma_2, \quad \|\partial R(x)\| \le \Gamma_3 \quad \forall x, \forall w, \forall v$$

Assumption 4 There exists $L_F, L_f, L_g > 0$ such that

1. $F(z) \ge F(x) + \langle \nabla F(x), z - x \rangle + \frac{L_F}{2}\|z - x\|^2 \quad \forall x, \forall z.$
2. $\|\nabla f_v(y) - \nabla f_v(w)\| \le L_f \|y - w\| \quad \forall w, \forall y, \forall v.$
3. $\|g(x) - g(z) - \nabla g(z)^\top (x - z)\| \le \frac{L_g}{2}\|x - z\|^2 \quad \forall x, \forall z.$

Condition 1 of Assumption 4 requires that the function $F(\cdot)$ is L_f-smooth. Condition 2 requires that f_v has Lipschitz gradient for all v. Condition 3 requires that the function $g(\cdot)$ is L_g smooth. Note that the case where $L_f = 0$ or $L_g = 0$ coincides with the special case where either f_v or g is linear, respectively. The constants L_F, L_f, L_g jointly characterize the smoothness and complexity of stochastic composition optimization. They do not admit any straightforward dependence relation.

Our first main result concerns with general optimization problems which are not necessarily convex.

Theorem 1 (Smooth (Nonconvex) Optimization) Let Assumptions 1, 2, 3, and 4 hold. Denote by $F(x) := (E_v(f_v) \circ E_w(g_w))(x)$ for short and suppose that $R(x) = 0$ in (1) and $E(F(x_k))$ is bounded from above. Choose $\alpha_k = k^{-a}$ and $\beta_k = 2k^{-b}$ where $a \in (0, 1)$ and $b \in (0, 1)$ in Algorithm 1. Then we have

$$\frac{\sum_{k=1}^{K} E(\|\nabla F(x_k)\|^2)}{K} \le O(K^{a-1} + L_f^2 L_g K^{4b-4a} I_{4a-4b=1} \log K + L_f^2 K^{-b} + K^{-a}). \tag{4}$$

If $L_g = 0$ and $L_f = 0$, choose $a = 5/9$ and $b = 4/9$, yielding

$$\frac{\sum_{k=1}^{K} E(\|\nabla F(x_k)\|^2)}{K} \le O(K^{-4/9}). \tag{5}$$

If $L_g = 0$ or $L_f = 0$, choose $a = b = 1/2$, yielding

$$\frac{\sum_{k=1}^{K} E(\|\nabla F(x_k)\|^2)}{K} \le O(K^{-1/2}). \tag{6}$$

The result of Theorem 1 strictly improves the best-known results given by Wang et al. (2017). First the result of (5) improves the finite-sample error bound from $O(k^{-2/7})$ to $O(k^{-4/9})$ for general convex and nonconvex optimization. This improves the best known convergence rate and provides a new benchmark for the stochastic composition problem. Note that it is possible to relax the condition "$E(F(x_k))$ is bounded from above" in Theorem 1. However, it would make the analysis more cumbersome and yield an additional term $\log K$ in the error bound.

Our second main result concerns strongly convex objective functions. We say that the objective function H is optimally strongly convex with parameter $\ell > 0$ if

$$H(x) \quad H(P_X(x)) \quad \ell kx \qquad P_X(x)k^2 \quad 8x: \tag{7}$$

(see Liu and Wright (2015)). Note that any strongly convex function is optimally strongly convex, but the reverse does not hold. For example, the objective function (2) in on-policy reinforcement learning is always optimally strongly convex (even if $E(A)$ is a rank deficient matrix), but not necessarily strongly convex.

We point out that there are other class of functions that are not strongly convex, for which first-order algorithms still achieve linear convergence. See Karimi et al. (2016) and Necoara et al. (2015) for examples. For ease of presentation, in this work, we only consider optimally strongly convex functions.

Theorem 2 (Strongly Convex Optimization) Suppose that the objective function $H(x)$ in (1) is optimally strongly convex with parameter $\ell > 0$ defined in (7). Set $_k = C_a k^{\ a}$ and $_k = C_b k^{\ b}$ where $C_a > 4$, $C_b > 2$, $a \ 2 \ (0; 1]$, and $b \ 2 \ (0; 1]$ in Algorithm 1. Under Assumptions 1, 2, 3, and 4, we have

$$E(kx_K \quad P_X(x_K)k^2) \quad O K^{\ a} + L^2 L_g K^{\ 4a+4b} + L^2 K^{\ b}: \tag{8}$$

If $L_g = 0$ and $L_f = 0$, choose $a = 1$ and $b = 4=5$, yielding

$$E(kx_K \quad P_X(x_K)k^2) \quad O(K^{\ 4=5}): \tag{9}$$

If $L_g = 0$ or $L_f = 0$, choose $a = 1$ and $b = 1$, yielding

$$E(kx_K \quad P_X(x_K)k^2) \quad O(K^{\ 1}): \tag{10}$$

Let us discuss the results of Theorem 2. In the general case where $L_f = 0$ and $L_g = 0$, the convergence rate in (9) is consistent with the result of Wang et al. (2017). Now consider the special case where $L_g = 0$, i.e., the inner mapping is linear. This result finds an immediate application to Bellman error minimization problem (2) which arises from reinforcement learning problem in (and with $'_1$ norm regularization). The proposed ASC-PG algorithm is able to achieve the optimal rate $O(1=K)$ without any extra assumption on the outer function f_v. To the best of our knowledge, this is the best (also optimal) sample-error complexity for on-policy reinforcement learning.

Theorem 3 (General Convex Optimization) Suppose that both $F(x)$ and $H(x)$ are convex, and the feasible set X is bounded. Set $_k = (k^{\ a})$ and $_k = (k^{\ b})$ with $a \ 2 \ (0; 1)$, and $b \ 2 \ (0; 1]$ in Algorithm 1. Under Assumptions 1, 2, 3, and 4, we have for any $c > 0$:

$$H(x_K) \quad H \quad O K^{a \ 1} + K^{a \ c} I^{\log K} + K^{\ a} \tag{11}$$
$$^{5a+4b+c \ \log K} \qquad ^{c=1} \qquad ^{2} \qquad ^{2}_{a \ b+c \ \log K}$$
$$^{6a \ 4b \ c=1} \qquad ^{2a+b \ c=1}$$
$$+ L_f L_g K \qquad I \qquad + L_f K \qquad I \qquad :$$

where $x_K := \frac{\sum_{k=2}^{K+1} k x_k}{\sum_{k=2}^{K+1} k}$. If $L_g = 0$ and $L_f = 0$, choose $a = 5/7$, $b = 4/7$, $c = 1$, yielding

$$H(x_K) - H^* = O(K^{-2/7}). \tag{12}$$

If $L_g = 0$ or $L_f = 0$, choose $a = 1/2$, $b = 1$, and $c = 1$, yielding

$$H(x_K) - H^* = O(K^{-1/2} \log K). \tag{13}$$

Theorem 3 gives stronger results than the best-known results of Wang et al. (2017) and requires milder assumptions. In the general case where $L_f = 0$ and $L_g = 0$, the convergence rate in (12) matches the result of Wang et al. (2017). When either $L_g = 0$ (i.e., the inner mapping is linear or $L_f = 0$ (i.e., the outer mapping is linear), the proposed ASC-PG algorithm is able to achieve the optimal rate $O(1/\sqrt{K})$ up to a logarithmic factor.

Remarks Theorems 1, 2, and 3 give important implications about the special cases where $L_f = 0$ or $L_g = 0$. In these cases, we argue that our convergence rate (10) is "optimal" with respect to the sample size K. To see this, it is worth pointing out the $O(1/K)$ rate of convergence is optimal for strongly convex expectation minimization problem. Because the expectation minimization problem is a special case of the problem (1), the $O(1/K)$ convergence rate must be optimal for the stochastic composition problem too. Similarly, the $O(1/\sqrt{K})$ rate of convergence is also optimal for general convex optimization due to the same argument.

- Consider the case where $L_f = 0$, which means that the outer function $f_v(\cdot)$ is linear with probability 1. Then the stochastic composition problem (1) reduces to an expectation minimization problem since $(E_v f_v \; E_w g_w)(x) = E_v(f_v(E_w g_w(x))) = E_v E_w(f_v \; g_w)(x)$. Therefore, it makes a perfect sense to obtain the optimal convergence rate.

- Consider the case where $L_g = 0$, which means that the inner function $g(\cdot)$ is a linear mapping. The result is quite surprising. Note that even $g(\cdot)$ is a linear mapping, it does not reduce problem (1) to an expectation minimization problem. However, the ASC-PG still achieves the optimal convergence rate. This suggests that, when inner linearity holds, the stochastic composition problem (1) is not fundamentally more difficult than the expectation minimization problem.

The convergence rate results unveiled in Theorems 1 and 2 are the best known results for the composition problem. We believe that they provide important new result which provides insights into the complexity of the stochastic composition problem.

4. Application to Reinforcement Learning

In this section, we apply the proposed ASC-PG algorithm to conduct policy value evaluation in reinforcement learning through attacking Bellman equations. Suppose that there are in total S states. Let the policy of interest be π. Denote the value function of states by $V \in \mathbb{R}^S$, where $V(s)$ denotes the value of being at state s under policy π. The Bellman equation of the problem is

$$V(s_1) = E[r_{s_1,s_2} + \gamma V(s_2) | s_1] \text{ for all } s_1, s_2 \in \{1, \ldots, S\},$$

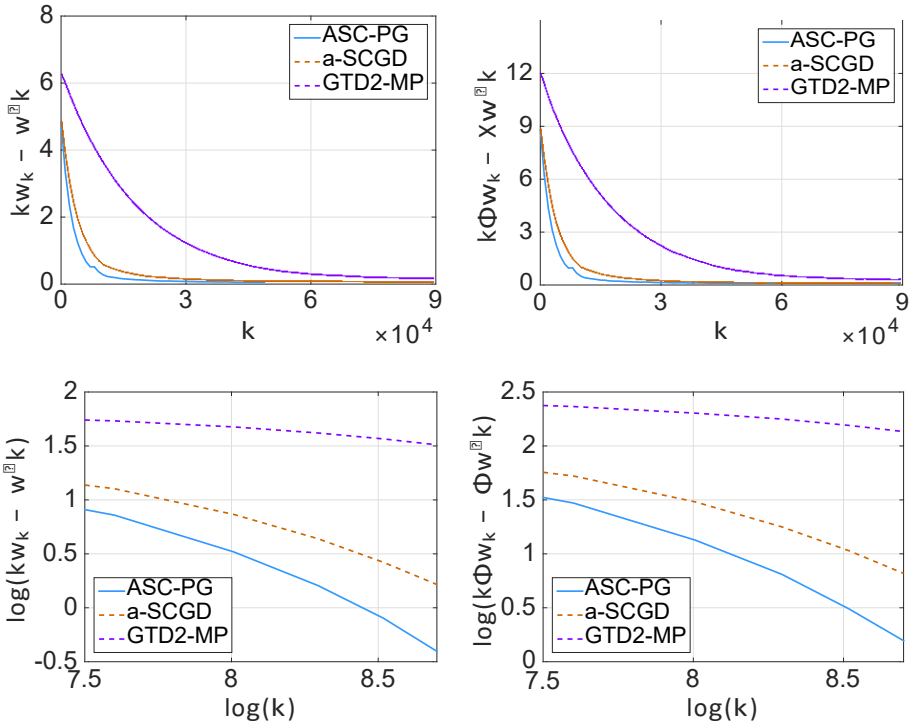

Figure 1: Empirical convergence rate of the ASC-PG algorithm and the GTD2-MP algorithm under Experiment 1 averaged over 100 runs, where w_k denotes the solution at the k-th iteration.

where $r_{s_1;s_2}$ denotes the reward of moving from state s_1 to s_2, and the expectation is taken over all possible future state s_2 conditioned on current state s_1 and the policy . We have that the solution $V \in \mathbb{R}^S$ to the above equation satisfies that $V = \Gamma V$. Here a moderately large S will make solving the Bellman equation directly impractical. To resolve the curse of dimensionality, in many practical applications, we approximate the value of each state by some linear map of its feature $\phi_s \in \mathbb{R}^d$, where $d < S$ to reduce the dimension. In particular, we assume that $V(s) \approx \phi_s^T w$ for some $w \in \mathbb{R}^m$.

To compute w, we formulate the problem as a Bellman residual minimization problem that

$$\min_w \sum_{s=1}^{S} (\phi_s^T w - \Gamma q_{;s}(w))^2;$$

where $q_{;s}(w) = \mathbb{E}[r_{s;s'} + \gamma \phi_{s'}^T w] = \sum_{s'} P_{ss'}(r_{s;s'} + \gamma \phi_{s'}^T w)$; $\gamma < 1$ is a discount factor, and $r_{s;s'}$ is the random reward of transition from state s to state s'. It is clearly seen that the proposed ASC-PG algorithm could be directly applied to solve this problem where we take

$$g(w) = (\phi_1^T w; q_{;1}(w); \ldots; \phi_S^T w; q_{;S}(w)) \in \mathbb{R}^{2S};$$

$$f(\phi_1^T w; q_{;1}(w); \ldots; \phi_S^T w; q_{;S}(w)) = \sum_{s=1}^{S} (\phi_s^T w - q_{;s}(w))^2 \quad \forall \in \mathbb{R}:$$

We point out that the g() function here is a linear map. By our theoretical analysis, we expect to achieve a faster O(1=k) rate of convergence, which is justified empirically in our later simulation study.

We consider three experiments, where in the first two experiments, we compare our proposed accelerated ASC-PG algorithm with accelerated SCGD (a-SCGD) algorithm (Wang et al., 2017) and the recently proposed GTD2-MP algorithm (Liu et al., 2015). Also, in the first two experiments, we do not add any regularization term, i.e. R() = 0. We point out here that in experiments 1 and 2, since there is no penalization term, i.e., R() = 0, the ASC-PG algorithm essentially matches the a-SCGD algorithm in Wang et al. (2017). However, note that the choice of stepsizes $f_k g$ and $f_k g$ of ASC-PG are different from the stepsizes of a-SCGD, where for ASC-PG $_k$ = k ^{1} and $_k$ = k ^{1}, and for a-SCGD $_k$ = k ^{1} and $_k$ = k $^{4=5}$. In the third experiment, we add an $'_1$-penalization term kwk_1. In all cases, we choose the step sizes via comparison studies as in Dann et al. (2014):

Experiment 1: We use the Baird's example (Baird, 1995), which is a well-known example to test the off-policy convergent algorithms. This example contains S = 6 states, and two actions at each state. We refer the readers to Baird (1995) for more detailed information of the example.

Experiment 2: We generate a Markov decision problem (MDP) using similar setup as in White and White (2016). In each instance, we randomly generate an MDP which contains S = 100 states, and three actions at each state. The dimension of the Given one state and one action, the agent can move to one of four next possible states. In our simulation, we generate the transition probabilities for each MDP instance uniformly from [0; 1] and normalize the sum of transitions to one, and we generate the reward for each transition also uniformly in [0; 1].

Experiment 3: We generate the data same as Experiment 2 except that we have a larger d = 100 dimensional feature space, where only the first 4 components of w are non-zeros. We add an $'_1$-regularization term, kwk_1, to the objective function.

Denote by w_k the solution at the k-th iteration. For the first two experiments, we report the empirical convergence performance kw_k wk and kw_k wk, where = $(_1; :::; _S)^T$ 2 $<^{S d}$ and w = V , and all w_k's are averaged over 100 runs, in the first two subfigures of Figures 1 and 2. It is seen that the ASC-PG algorithm achieves the fastest convergence rate empirically in both experiments. To further evaluate our theoretical results, we plot log(t) vs. log(kw_k wk) (or log(kw_k wk) averaged over 100 runs for the first two experiments in the second two subfigures of Figures 1 and 2. The empirical results further support our theoretical analysis that kw_k wk^2 = O(1=k) for the ASC-PG algorithm when g() is a linear mapping.

For Experiment 3, as the optimal solution is unknown, we run the ASC-PG algorithm for one million iterations and take the corresponding solution as the optimal solution w^ and we report kw_k wk and kw_k wk_k averaged over 100 runs in Figure 3. It is seen the the ASC-PG algorithm achieves fast empirical convergence rate.

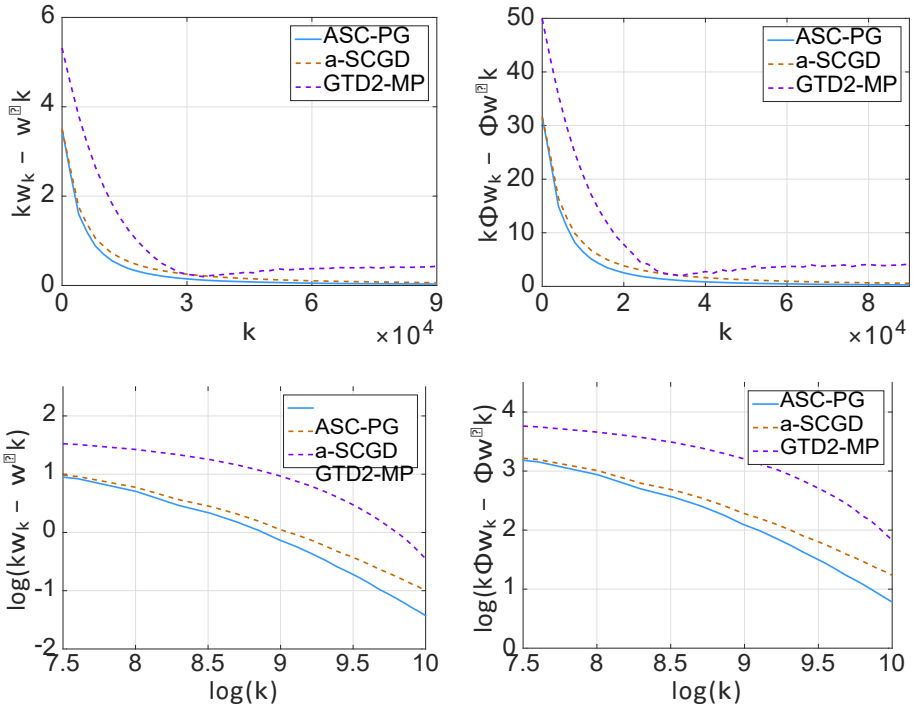

Figure 2: Empirical convergence rate of the ASC-PG algorithm and the GTD2-MP algorithm under Experiment 2 averaged over 100 runs, where w_k denotes the solution at the k-th iteration.

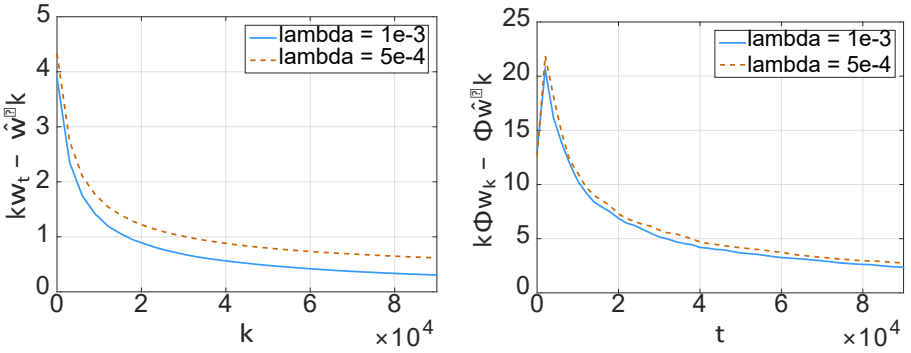

Figure 3: Empirical convergence rate of the ASC-PG algorithm with the ℓ_1-regularization term $\|w\|_1$ under Experiment 3 averaged over 100 runs, where w_k denotes the solution at the t-th iteration.

5. Conclusion

We develop the first proximal gradient method for stochastic composition optimization with nonsmooth penalties. The algorithm updates by interacting with a stochastic first-order

oracle. Finite-sample convergence rates are established under a variety of assumptions, which provide new rate benchmarks that improve the best-known results prior to this paper. Application of the ASC-PG to reinforcement learning leads to a new on-policy learning algorithm, which achieves faster convergence than the best known algorithms. For future research, it remains open whether or under what circumstances the current $O(K^{-4/9})$ can be further improved. Another direction is to customize and adapt the algorithm and analysis to more specific problems arising from reinforcement learning and risk-averse optimization, in order to fully exploit the potential of the proposed method.

Acknowledgments

We thank the Editor, the Action Editor and two anonymous reviewers for the careful consideration and invaluable suggestions, which help us further improve the paper. This project is in part supported by NSF grants CCF1718513, CNS-1548078, DMS-10009141, CMMI-1653435, and by the National Institute on Drug Abuse grant P50 DA039838. The content of this manuscript is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute on Drug Abuse or the National Institutes of Health.

Appendix

This appendix includes all detailed proofs for the theorems in the main text. The key idea of deriving the rate convergence of our algorithm is to find a sequence of random vectors associated with the output solutions at each iteration, which converge at fast rates. To better analyze the convergence process, we have constructed two auxiliary sequences of random variables $\{m_k\}$ and $\{n_k\}$, which are defined in the beginng of Lemma 7, that converge jointly to 0 at a fast rate. The convergence of $\{m_k\}$ and $\{n_k\}$ further imply that $\{x_k\}$ and $\{y_k\}$ converge rapidly to their limits respectively. Then we are able to improve the convergence rate of y_k. Intuitively, we show that the improved iterate y_k tracks the unknown quantity $E[g(x_k)]$ more closely than in the previous work. This explains why our new algorithm achieves an improved rate of convergence and sample complexity.

A. Proof to Theorem 1

In this section, we provide the detailed proof for Theorem 1. We start by some technical lemmas, The first one provides a convergence result of the generated solutions, where we bound the term $\|x_k - x_{k+1}\|^2$ as k increases.

Lemma 4 Under Assumption 3, two subsequent iterates in Algorithm 1 satisfy

$$\|x_k - x_{k+1}\|^2 \quad (^2): \quad _k$$

Proof From the definition of the proximal operation, we have

$$x_{k+1} = \text{prox}_{_k R()}(x_k \quad _k r g_{w_k}^{>}(x_k) r f_{v^k}(y_k))$$

$$= \arg\min_x \frac{1}{2}\|x - x_k + \eta_k \nabla g_{w_k}^{\top}(x_k)\nabla f_{v_k}(y_k)\|^2 + \eta_k R(x).$$

The optimality condition suggests the following equality:

$$x_{k+1} - x_k = -\eta_k(\nabla g_{w_k}^{\top}(x_k)\nabla f_{v_k}(y_k) + s_{k+1}) \tag{14}$$

where $s_{k+1} \in \partial R(x_{k+1})$ is some vector in the sub-differential set of $R(\cdot)$ at x_{k+1}. Then apply the boundedness condition in Assumption 3 to yield

$$\begin{aligned}\|x_{k+1} - x_k\| &= \eta_k\|(\nabla g_{w_k}^{\top}(x_k)\nabla f_{v_k}(y_k) + s_{k+1})\| \\ &\overset{\text{(Assumption 3)}}{\leq} \eta_k(\|\nabla g_{w_k}(x_k)\nabla f_{v_k}(y_k)\| + \|s_{k+1}\|) \\ &\leq \eta_k(\|\nabla g_{w_k}(x_k)\|\|\nabla f_{v_k}(y_k)\| + \|s_{k+1}\|) \\ &\leq (1)\eta_k,\end{aligned}$$

which implies the claim. ■

Next, we provide a simple bound which quantifies the difference between the random variable $\|\nabla g_w^{\top}(x)\nabla_v f(g(x)) - \nabla g_w^{\top}(x)\nabla_v f(y)\|$ and its population version $\|y - g(x)\|$.

Lemma 5 Under Assumptions 3 and 4, we have

$$\|\nabla g_w^{\top}(x)\nabla_v f(g(x)) - \nabla g_w^{\top}(x)\nabla_v f(y)\| \leq (L_f\|y - g(x)\|).$$

Proof We have

$$\begin{aligned}\|\nabla g_w^{\top}(x)\nabla f_v(g(x)) - \nabla g_w^{\top}(x)\nabla f_v(y)\| &\leq \|\nabla g_w^{\top}(x)\|\|\nabla f_v(g(x)) - \nabla f_v(y)\| \\ &\overset{\text{(Assumption 3)}}{\leq} (1)\|\nabla f_v(g(x)) - \nabla f_v(y)\| \\ &\overset{\text{(Assumption 4)}}{\leq} (L_f)\|y - g(x)\|.\end{aligned}$$

It completes the proof. ■

Then, we prove two lemmas which provides the convergence rate of $E(\|y_k - g(x_k)\|^2)$.

Lemma 6 Given two sequences of positive scalars $\{s_k\}_{k=1}^{1}$ and $\{\eta_k\}_{k=1}^{1}$ satisfying

$$s_{k+1} \leq (1 - \eta_k + C_1\eta^2)s_k + C_2\eta^{a} \tag{15}$$

where $C_1 > 0$, $C_2 > 0$, and $a > 0$. Letting $\eta_k \in \langle$ as $\eta_k = C_3 k^{-b}$ where $b \in (0, 1]$ and $C_3 > 2$, the sequence can be bounded by

$$v_k \leq Ck^{-c}$$

where C and c are defined as

$$C := \max_{k \leq (C_1 C_3)^{1/b}+1} s_k k^c + \frac{C_2}{C_3 - 2} \quad \text{and} \quad c := a - b.$$

In other words, we have

$$s_k = (k^{-(a-b)}).$$

Proof We prove it by induction. First it is easy to verify that the claim holds for $k \geq (C_1C^2)^{1/(b-a)}$ from the definition for C. Next we prove from "k" to "k + 1", that is, given $s_k \leq Ck^{-c}$ for $k > (C_1C^2)^{1/(b-a)}$, we need to prove $s_{k+1} \leq C(k+1)^{-c}$.

$$\begin{aligned} s_{k+1} &\leq (1 - \alpha_k + C_1\alpha_k^2)s_k + C_2k^{-a} \\ &\leq (1 - C_3k^{-b} + C_1C_3^2k^{-2b})Ck^{-c} + C_2k^{-a} \\ &= Ck^{-c} - CC_3k^{-b-c} + CC_1C_3^2k^{-2b-c} + C_2k^{-a}. \end{aligned} \tag{16}$$

To prove that (16) is bounded by $C(k+1)^{-c}$, it suffices to show that

$$\Delta := (k+1)^{-c} - k^{-c} + C_3k^{-b-c} - C_1C_3^2k^{-2b-c} \geq 0 \quad \text{and} \quad C \geq \frac{C_2k^{-a}}{\Delta}.$$

From the convexity of function $h(t) = t^{-c}$, we have the inequality $(k+1)^{-c} \geq k^{-c} - ck^{-c-1}$. Therefore we obtain

$$\begin{aligned} \Delta &\geq ck^{-c-1} + C_3k^{-b-c} - C_1C_3^2k^{-2b-c} \\ &\overset{(b\leq 1;\ k>(C_1C_3^2)^{1/(b-a)})}{\geq} (C_3 - 2)(k^{-b-c}) \\ &\overset{(C_3>2)}{>} 0. \end{aligned}$$

To verify the second one, we have

$$\frac{C_2k^{-a}}{\Delta} \leq \frac{C_2}{C_3 - 2}k^{a+b+c} \overset{(c=a+b)}{=} \frac{C_2}{C_3 - 2} \leq C$$

where the last inequality is due to the definition of C. It completes the proof. ■

Lemma 7 Choose α_k to be $\alpha_k = C_bk^{-b}$ where $C_b > 2$, $b \in (0, 1]$, and $\beta_k = C_ak^{-a}$. Under Assumptions 1 and 2, we have

$$\mathbb{E}(\|y_k - g(x_k)\|^2) \leq L_g(k^{-4a+4b}) + (k^{-b}). \tag{17}$$

Proof Denote by m_{k+1}

$$m_{k+1} := \sum_{t=0}^{k} \prod_{(t)}^{(k)} \|x_{k+1} - z_{t+1}\|^2$$

and n_{k+1}

$$n_{k+1} := \sum_{t=0}^{k} \prod_{t}^{(k)} (g_{w_{t+1}}(z_{t+1}) - g(z_{t+1}))$$

for short.

From Lemma 10 in (Wang et al., 2017), we have

$$\|y_k - g(x_k)\|^2 \leq \frac{L_g}{2}m_k + n_k^2 \leq L_g m_k + 2n_k^2. \tag{18}$$

From Lemma 11 in (Wang et al., 2017), m_{k+1} can be bounded by

$$m_{k+1} \quad (1 \quad {}_k)m_k + {}_k q_k + \frac{k\overline{x}_k^2}{k} \quad x_{k+1}k^2 \tag{19}$$

where q_k is bounded by

$$\begin{aligned} q_{k+1} & \quad (1 \quad {}_k)q_k + \frac{k\overline{x}_k x_{k+1}}{k} \quad x_k k^2 \\ & \quad (1 \quad {}_k)q_k + \frac{(1)^2 \quad k}{k} \\ & \quad (1 \quad {}_k)q_k + (k^{\ 2a+b}): \end{aligned} \qquad \text{(Lemma 4)}$$

Taking $s_k = q_k$, and ${}_k = {}_k$, applying Lemma 6 implies the following decay rate

$$q_k \ (k^{\ 2a+b+b}) = (k^{\ 2a+2b}):$$

Together with (19), we have

$$\begin{aligned} m_{k+1} & \quad (1 \quad {}_k)m_k + {}_k q_k + \frac{k\overline{x}_k^2}{k} \quad x_{k+1}k^2 \\ & \quad (1 \quad {}_k)m_k + (k^{\ 2a+b}) + (k^{\ 2a+b}) \quad (1 \\ & \quad {}_k)m_k + (k^{\ 2a+b}); \end{aligned}$$

which leads to

$$m_k \ (k^{\ 2a+2b}) \qquad \text{and} \quad m_k^2 \ (k^{\ 4a+4b}): \tag{20}$$

by using Lemma 6 again. Then we estimate the upper bound for $E(n_k^2)$. From Lemma 11 in (Wang et al., 2017), we know $E(n_k^2)$ is bounded by

$$E(n_{k+1}^2) \ (1 \quad {}_k)^2 E(kn_k k^2) + {}^{22}\overline{k}\ \overline{g}\ (1 \quad 2_k + {}^2)E_k(kn_k k^2) + {}^{22}: \ {}_k \ {}_g$$

By using Lemma 6 again, we have

$$E(n_k^2) \ (k^{\ b}): \tag{21}$$

Now we are ready to estimate the upper bound of $ky_{k+1} \quad g(x_{k+1})k^2$ by following (18)

$$\begin{aligned} E(ky_k \quad g(x_k)k^2) & \quad L_g E(m_k^2) + 2E(n_k^2) \\ & \quad L_g(k^{\ 4a+4b}) + (k^{\ b}): \end{aligned} \qquad \text{(20)+(21)}$$

It completes the proof. ■

Now we are ready to prove Theorem 1.

Proof to Theorem 1. From the Lipschitzian condition in Assumption 4, we have

$$F(x_{k+1}) \quad F(x_k)$$

$$\langle \nabla F(x_k),\, x_{k+1} - x_k \rangle + \frac{L_F}{2}\|x_{k+1} - x_k\|^2$$

(Lemma 6)

$$= \alpha_k \langle \nabla F(x_k),\, \nabla g_{w_k}(x_k) f_{v_k}(y_k) \rangle + (\alpha_k)^2$$

$$= \alpha_k \|\nabla F(x_k)\|^2 + \alpha_k \underbrace{\langle \nabla F(x_k),\, \nabla F(x_k) - \nabla g_{w_k}(x_k) f_{v_k}(y_k) \rangle}_{=:T} + (\alpha_k)^2 \tag{22}$$

Next we estimate the upper bound for $E(T)$:

$$\begin{aligned}
E(T) &= E\langle \nabla F(x_k),\, \nabla F(x_k) - \nabla g_{w_k}(x_k) f_{v_k}(g(x_k)) \rangle \\
&\quad + E\langle \nabla F(x_k),\, \nabla g_{w_k}(x_k) f_{v_k}(g(x_k)) - \nabla g_{w_k}(x_k) f_{v_k}(y_k) \rangle \\
&\overset{\text{(Assumption 1)}}{=} E\langle \nabla F(x_k),\, \nabla g_{w_k}(x_k) f_{v_k}(g(x_k)) - \nabla g_{w_k}(x_k) f_{v_k}(y_k) \rangle \\
&\quad \frac{1}{2} E\|\nabla F(x_k)\|^2 + \frac{1}{2} E\|\nabla g_{w_k}(x_k) f_{v_k}(g(x_k)) - \nabla g_{w_k}(x_k) f_{v_k}(y_k)\|^2 \\
&\overset{\text{(Lemma 5)}}{\leq} \frac{1}{2} E\|\nabla F(x_k)\|^2 + (L_f^2) E\|y_k - g(x_k)\|^2.
\end{aligned}$$

Take expectation on both sides of (22) and substitute $E(T)$ by its upper bound:

$$\begin{aligned}
&\frac{\alpha_k}{2}\|\nabla F(x_k)\|^2 \\
&E(F(x_k)) - E(F(x_{k+1})) + (L_f^2 \alpha_k) E\|y_k - g(x_k)\|^2 + (\alpha_k)^2 \\
&\overset{\text{(Lemma 7)}}{\leq} E(F(x_k)) - E(F(x_{k+1})) + L_g (L_f^2 \beta_k)(\beta_k^{4a+4b}) + (L_f^2 \beta_k^{b}) \beta_k + (\alpha_k)^2 \\
&E(F(x_k)) - E(F(x_{k+1})) + L_f^2 L_g (\beta_k^{5a+4b}) + L_f^2 (\beta_k^{a+b}) + (\beta_k^{2a})
\end{aligned}$$

which suggests that

$$\begin{aligned}
&E(\|\nabla F(x_k)\|^2) \\
&2\alpha_k^{-1} E(F(x_k)) - 2\alpha_k^{-1} E(F(x_{k+1})) + L_f^2 L_g (\beta_k^{4a+4b}) + L_f^2 (\beta_k^{b}) + (\beta_k^{a}) \\
&2k^{a} E(F(x_k)) - 2k^{a} E(F(x_{k+1})) + L_f^2 L_g (k^{-4a+4b}) + L_f^2 (k^{-b}) + (k^{-a}).
\end{aligned} \tag{23}$$

Sum Eq. (23) from $k = 1$ to K and obtain

$$\begin{aligned}
\frac{\sum_{k=1}^{K} E(\|\nabla F(x_k)\|^2)}{K} &\leq 2K^{-1}\alpha_1^{-1} F(x_1) + K^{-1} \sum_{k=2}^{K} ((k+1)^a - k^a) E(F(x_k)) \\
&\quad + K^{-1} \sum_{k=1}^{K} L_f^2 L_g (k^{-4a+4b}) + K^{-1} L_f^2 \sum_{k=1}^{K} (k^{-b}) + K^{-1} \sum_{k=1}^{K} (k^{-a}) \\
&\leq 2K^{-1}\alpha_0^{-1} F(x_0) + K^{-1} \sum_{k=2}^{K} a k^{a-1} E(F(x_k)) \\
&\quad + K^{-1} \sum_{k=1}^{K} L_f^2 L_g (k^{-4a+4b}) + K^{-1} L_f^2 \sum_{k=1}^{K} (k^{-b}) + K^{-1} \sum_{k=1}^{K} (k^{-a})
\end{aligned}$$

$$O(K^{a-1} + L^2 L_{fg} K^{4b-4a} I_{4a}^{\log K}_{4b=1} + L^2 K^{-b} + K^{-a});$$

where the second inequality uses the fact that $h(t) = t^a$ is a concave function suggesting $(k+1)^a - k^a \le ak^{a-1}$, and the last inequality uses the condition $E(F(x_k)) - (1)$.

Letting $a = 5/9$ and $b = 4/9$, we obtain the desired convergence rate $O(K^{-4/9})$. ■

B. Proof to Theorem 2

In this section, we provide the detailed proof for Theorem 2. The proof is based on the following key lemma, which provides a "contraction" property of $(E(H(x_{k+1})) - H)$ and $E(\|x_{k+1} - P_X(x_{k+1})\|^2)$.

Lemma 8 Assume that both $F(x)$ and $R(x)$ are convex. Under Assumptions 1, 2, 3, and 4, the iterates generated by Algorithm 1 satisfies for any sequence of positive scalars $\{\gamma_k\}$:

$$\begin{aligned} &2\gamma_k(E(H(x_{k+1})) - H) + E(\|x_{k+1} - P_X(x_{k+1})\|^2) \qquad (24)\\ &\le (1+\gamma_k)E(\|x_k - P_X(x_k)\|^2) + \Delta(\gamma_k) + (L_f^3\gamma_k^2/\gamma_k)E(\|y_k - g(x_k)\|^2) + \Delta(\gamma_k). \end{aligned}$$

Proof Following the line of the proof to Lemma 4, we have

$$x_{k+1} - x_k = \gamma_k(\nabla g_{w_k}(x_k)\nabla f_{v_k}(y_k) + s_{k+1}) \qquad (25)$$

where $s_{k+1} \in \partial R(x_{k+1})$ is some vector in the sub-differential set of $R(\cdot)$ at x_{k+1}. Then we consider $\|x_{k+1} - P_X(x_{k+1})\|^2$:

$$\begin{aligned} &\|x_{k+1} - P_X(x_{k+1})\|^2\\ &\|x_{k+1} - x_k + x_k - P_X(x_k)\|^2\\ =\ &\|x_k - P_X(x_k)\|^2 - \|x_{k+1} - x_k\|^2 + 2\langle x_{k+1} - x_k; x_{k+1} - P_X(x_k)\rangle\\ \overset{(25)}{=}\ &\|x_k - P_X(x_k)\|^2 - \|x_{k+1} - x_k\|^2 - 2\gamma_k\langle\nabla g_{w_k}(x_k)\nabla f_{v_k}(y_k) + s_{k+1}; x_{k+1} - P_X(x_k)\rangle\\ =\ &\|x_k - P_X(x_k)\|^2 - \|x_{k+1} - x_k\|^2 + 2\gamma_k\langle\nabla g_{w_k}(x_k)\nabla f_{v_k}(y_k); P_X(x_k) - x_{k+1}\rangle\\ &+2\gamma_k\langle s_{k+1}; P_X(x_k) - x_{k+1}\rangle\\ &\|x_k - P_X(x_k)\|^2 - \|x_{k+1} - x_k\|^2 + 2\gamma_k\langle\nabla g_{w_k}(x_k)\nabla f_{v_k}(y_k); P_X(x_k) - x_{k+1}\rangle\\ &+2\gamma_k(R(P_X(x_k)) - R(x_{k+1})) \quad \text{(due to the convexity of } R(\cdot))\\ &\|x_k - P_X(x_k)\|^2 - \|x_{k+1} - x_k\|^2 + 2\gamma_k\underbrace{\langle\nabla F(x_k); P_X(x_k) - x_{k+1}\rangle}_{T_1}\\ &+2\gamma_k\underbrace{\langle\nabla g_{w_k}(x_k)\nabla f_{v_k}(y_k) - \nabla F(x_k); P_X(x_k) - x_{k+1}\rangle}_{T_2}\\ &+2\gamma_k(R(P_X(x_k)) - R(x_{k+1})) \qquad (26) \end{aligned}$$

where the second equality follows from $\|a+b\|^2 = \|b\|^2 - \|a\|^2 + 2\langle a; a+b\rangle$ with $a = x_{k+1} - x_k$ and $b = x_k - P_X(x_k)$. We next estimate the upper bound for T_1 and T_2 respectively:

$$T_1 = \langle\nabla F(x_k); x_k - x_{k+1}\rangle + \langle\nabla F(x_k); x_k - P_X(x_k)\rangle$$

$$F(x_k) \quad F(x_{k+1}) + \underbrace{\frac{L_F}{2}\|x_{k+1} \quad x_k\|^2}_{\text{due to Assumption 4}} + \underbrace{F(P_X(x_k)) \quad F(x_k)}_{\text{due to the convexity of } F()}$$

$$= \quad F(P_X(x_k)) \quad F(x_{k+1}) + \frac{L_F}{2}\|x_{k+1} \quad x_k\|^2$$

$$F(P_X(x_k)) \quad F(x_{k+1}) + \langle \ _k; \ \rangle^2$$

where the last inequality uses Lemma 4.

$$\begin{aligned} T_2 \ = \ & \langle \nabla F(x_k) \quad \nabla g_{w_k}^{>}(x_k) \nabla f_{v^k}(y_k); \ x_k \quad P_X(x_k) \rangle \\ & + \langle \nabla F(x_k) \quad \nabla g_{w_k}^{>}(x_k) \nabla f_{v^k}(y_k); \ x_{k+1} \quad x_k \rangle \\ & \underbrace{\langle \nabla F(x_k) \quad \nabla g_{w_k}^{>}(x_k) \nabla f_{v_k}(g(x_k)); \ x_k \quad P_X(x_k) \rangle}_{T_{2;1}} \\ & + \underbrace{\langle \nabla g_{w_k}^{>}(x_k) \nabla f_{v^k}(g(x_k)) \quad \nabla g_{w_k}^{>}(x_k) \nabla f_{v^k}(y_k); \ x_k \quad P_X(x_k) \rangle}_{T_{2;2}} \\ & + \frac{_k}{2}\underbrace{\|\nabla F(x_k) \quad \nabla g_{w_k}^{>}(x_k) \nabla f_{v^k}(y_k)\|^2}_{T_{2;3}} + \frac{1}{2_k}\|x_k \quad x_{k+1}\|^2 \end{aligned}$$

where the last line is due to the inequality $\langle a; b \rangle \ \frac{1}{2_k}\|a\|^2 + \frac{_k}{2}\|b\|^2$. For $T_{2;1}$, we have $E(T_{2;1}) = 0$ due to Assumption 1. For $T_{2;2}$, we have

$$\begin{aligned} T_{2;2} & \ \frac{_k}{2}\|\nabla g_{w_k}^{>}(x_k) \nabla f_{v^k}(g(x_k)) \quad \nabla g_{w_k}^{>}(x_k) \nabla f_{v^k}(y_k)\|^2 + \frac{_k}{2}\|x_k \quad P_X(x_k)\|^2 \\ & \text{(Lemma 5)} \quad L^2 \ _k^{\ 2} \ \|y_k \quad g(x_k)\|^2 + \frac{_k}{2}\|x_k \quad x_{k+1}\|^2: \end{aligned}$$

$T_{2;3}$ can be bounded by a constant

$$T_{2;3} \ 2\|\nabla F(x_k)\|^2 + 2\|\nabla g_{w_k}^{>} \nabla f_{v^k}(y_k)\|^2 \ \text{(Assumption 3)} \ (1):$$

Take expectation on T_2 and put all pieces into it:

$$E(T_2) \quad L^2 \ _k^{\ 2} \ \|y_k \quad g(x_k)\|^2 + \frac{1}{2_k}(_k\|x_k \quad P_X(x_k)\|^2 + \|x_k \quad x_{k+1}\|^2) + \ _k:$$

Taking expectation on both sides of (26) and plugging the upper bounds of T_1 and T_2 into it, we obtain

$$\begin{aligned} & 2_k(E(H(x_{k+1})) \quad H) + E(\|x_{k+1} \quad P_X(x_{k+1})\|^2) \\ & \quad (1 + \ _k)E(\|x_k \quad P_X(x_k)\|^2) + \ _k + 3(L_f \ _k = _k)E^2(\|y_k \quad g(x_k)\|^2) + \langle \ _k; \ \rangle^2 \end{aligned}$$

which completes the proof. ■

Then, we prove the Theorem 2 based on the previous lemma.

Proof to Theorem 2 Apply the optimally strong convexity in (7) to Lemma 8, yielding

$$(1+2_k)E(kx_{k+1} \quad P_X(x_{k+1})k^2)$$
$$(1+_k)E(kx_k \quad P_X(x_k)k^2) + (_k) + {}^3(L_f{}_k=_k)E^2(k^2y_k \quad g(x_k)k^2) + (_k): {}^2$$

It follows by dividing $1+2_k$ on both sides

$$E(kx_{k+1} \quad P_X(x_{k+1})k^2)$$
$$\frac{1+}{1+2_k}E(k^kx_k \quad P_X(x_k)k^2) + (^3) + {}_k(L^{22}=_k)E(k_ky_k \quad g(x_k)k^2) + (^2): {}_k$$

Choosing $_k = {}_k \quad 2^{22} \quad 0{:}5_k$ yields

$$E(kx_{k+1} \quad P_X(x_{k+1})k^2)$$
$$(1 \quad _k)E(kx_k \quad P_X(x_k)k^2) + (^2) + {}_k\frac{(L^2{}_k)}{E(k^fg(x_k)} \quad y_kk^2)$$
$$(1 \quad _k)E(kx_k \quad P_X(x_k)k^2) + (k^{\ 2a}) + (L_gL^2k^{\ 5a+4b}_f + L^2k^{\ a}_f{}^{\ b}):$$

Apply Lemma 6 and substitute the subscript k by K to obtain the first claim in (8)

$$E(kx_K \quad P_X(x_K)k^2) \ OK^{\ a} + L^2L_gK^{\ 4a+4b}_f + L^2K^{\ b}_f:$$

The followed specification of a and b can easily verified. ■

C. Proof to Theorem 3

In what follows, we provide the detailed proof to Theorem 3.

Proof to Theorem 3 Let $P_k = 2_k(E(H(x_{k+1})) \quad H) + E(kx_{k+1} \quad P_X(x_{k+1})k^2)$, which is the left hand side of inequality (24). Summing P_k up from $k = 1$ to $k = K$ and taking $_k = (k^{\ c})$ yields

$$2\sum_{k=1}^{K}{}_k(E(H(x_{k+1})) \quad H) + E(kx_{K+1} \quad P_X(x_{K+1})k^2)$$
$$(1+_1)E(kx_1 \quad P_X(x_1)k^2) + \sum_{k=2}^{K}{}_kE(kx_k \quad P_X(x_k)k^2)$$
$$+ \sum^{K}(L^{22}=_{fk})E(k_ky_k \quad g(x_k)k^2) + (^2)\,k\bar{k}2$$
$$(1) + \sum_{k=2}^{K}{}_k \ ! \ + (K^{1\ 2a}) + \sum_{k=2}^{K}(L^{22}=_k)E(k_ky_k \quad g(x_k)k^2)$$

$$(1) + K^{1\ c} I^{\log K}_{c=1} + O(K^{1\ 2a}) + \sum_{k=2}^{K} (L^{2\ 2} =_k) E(k y_k \quad g(x_k)k^2): \quad (27)$$

Note that the second inequality holds by the condition that the feasible set X is bounded, and thus $kx_k \quad P_X(x_k)k^2$ is bounded.

We continue to bound the last term on the right hand side of (27):

$$\begin{aligned} & \sum_{k=2}^{K} (L^{2\ 2} =_k) E(k y_k \quad g(x_k)k^2) \\ (\text{Lemma 7}) \quad & \sum_{k=2}^{K} L^2 L_g (k^{\ 6a+4b\ c}) + L^2 (k^{\ 2a\ b\ c}) \\ & L_f^2 L_g (K^{1\ 6a+4b\ c} I_{6a\ 4b\ c=1}) + L^2 (K^{1\ 2a\ b\ c} I_{2a+b\ c=1}): \end{aligned} \quad (28)$$

Plugging (28) into (27) and dividing $2^{\sum_{k=1}^{K} k}$ on both sides:

$$\begin{aligned} & \frac{2^{\sum_{k=1}^{K} k} H(x_{k+1}) \quad H}{2^{\sum_{k=1}^{K} k}} \\ & \frac{(1) + K^{1\ c} I^{\log K}_{c=1} + (K^{1\ 2a})}{2^{\sum_{k=1}^{K}}} \quad \frac{k}{\log K} \\ & + \frac{L_f^2 L_g (K^{1\ 6a+4b+c} I_{6a\ 4b\ c=1}) + L_f (K^{1\ 2a\ b+c} I_{2a\ b+c=1})}{2^{\sum_{k=1}^{K} k}} \\ & O(K^{a\ 1}) + O(K^{a\ c} I^{\log K}_{c=1}) + O(K^{\ a}) \\ & \quad + O(L_f^2 L_g K^{\ 5a+4b+c} I^{\log K}_{6a\ 4b\ c=1}) + O(L_f^2 K^{\ a\ b+c} I^{\log K}_{2a+b\ c=1}) \\ = & \ O\left(K^{a\ 1} + K^{a\ c} I^{\log K}_{c=1} + K^{\ a} + L_f^2 L_g K^{\ 5a+4b+c} I^{\log K}_{6a\ 4b\ c=1} + L_f^2 K^{\ a\ b+c} I^{\log K}_{2a+b\ c=1}\right): \end{aligned}$$

where the last inequality uses the fact $\sum_{k=1}^{K} k \quad (K^{1\ a})$. To finish the proof of (11), we use Jensen's inequality

$$\frac{2^{\sum_{k=1}^{K} k H(x_{k+1})}}{2^{\sum_{k=1}^{K} k}} H(x_K): \quad k=1$$

■

References

L Baird. Residual algorithms: Reinforcement learning with function approximation. In International Conference on Machine Learning (ICML), pages 30–37, 1995.

B. Dai, N. He, Y. Pan, B. Boots, and L. Song. Learning from conditional distributions via dual embeddings. In Artificial Intelligence and Statistics (AISTATS), pages 1458–1467, 2017.

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

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

Y. M. Ermoliev. Methods of Stochastic Programming. Monographs in Optimization and OR, Nauka, Moscow, 1976.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (KDD), pages 795–811. Springer, 2016.

B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan, and M. Petrik. Finite-sample analysis of proximal gradient td algorithms. In Conference on Uncertainty in Artificial Intelligence (UAI), 2015.

J. Liu and S. J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. SIAM Journal on Optimization, 25(1):351–376, 2015.

I Necoara, Yu Nesterov, and F Glineur. Linear convergence of first order methods for non-strongly convex optimization. arXiv preprint arXiv:1504.06298, 2015.

A. Nedić and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. SIAM Journal on Optimization, 12:109–138, 2001.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. SIAM Journal on Optimization, 19:1574–1609, 2009.

R. S. Sutton and A. G. Barto. Reinforcement Learning: An Introduction. MIT press, 1998.

M. Wang and D. P. Bertsekas. Stochastic first-order methods with random constraint projection. SIAM Journal on Optimization, 26(1):681–717, 2016.

M. Wang and J. Liu. A stochastic compositional subgradient method using Markov samples. Winter Simulation Conference, 2016.

M. Wang, Y. Chen, J. Liu, and Y. Gu. Random multi-constraint projection: Stochastic gradient methods for convex optimization with many constraints. arXiv preprint arXiv:1511.03760, 2015.

M. Wang, J. Liu, and E. X. Fang. Accelerating stochastic composition optimization. In Advances in Neural Information Processing Systems (NIPS), pages 1714–1722, 2016.

M. Wang, E. X. Fang, and H. Liu. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. Mathematical Programming, 161 (1-2):419–449, 2017.

A. White and M. White. Investigating practical linear temporal difference learning. In International Conference on Autonomous Agents & Multiagent Systems (AAMAS), pages 494–502, 2016.