Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions

Mengdi Wang1 · Ethan X. Fang2 · Han Liu1

Abstract Classical stochastic gradient methods are well suited for minimizing expected-value objective functions. However, they do not apply to the minimization of a nonlinear function involving expected values or a composition of two expected-value functions, i.e., the problem \( \min_x E_v \left[ f_v \left( E_w [g_w (x)] \right) \right] \). In order to solve this stochastic composition problem, we propose a class of stochastic compositional gradient descent (SCGD) algorithms that can be viewed as stochastic versions of quasi-gradient method. SCGD update the solutions based on noisy sample gradients of \( f_v \), \( g_w \) and use an auxiliary variable to track the unknown quantity \( E_w [g_w (x)] \). We prove that the SCGD converge almost surely to an optimal solution for convex optimization problems, as long as such a solution exists. The convergence involves the interplay of two iterations with different time scales. For nonsmooth convex problems, the SCGD achieves a convergence rate of \( O(k^{-1/4}) \) in the general case and \( O(k^{-2/3}) \) in the strongly convex case, after taking \( k \) samples. For smooth convex problems, the SCGD can be accelerated to converge at a rate of \( O(k^{-2/7}) \) in the general case and \( O(k^{-4/5}) \) in the strongly convex case. For nonconvex problems, we prove that any limit point generated by SCGD is a stationary point, for which we also provide the convergence rate analysis. Indeed, the stochastic setting where one wants to optimize compositions of expected-value functions...
value functions is very common in practice. The proposed SCGD methods find wide
applications in learning, estimation, dynamic programming, etc.

**Keywords** Stochastic gradient · Stochastic optimization · Convex optimization ·
Sample complexity · Simulation · Statistical learning

**Mathematics Subject Classification** 90C15 · 90C25 · 90C06 · 68W27

1 Introduction

Stochastic gradient descent (SGD) methods have been prominent in minimizing con-
 convex functions by using noisy gradients. They find wide applications in simulation,
distributed optimization, data-based optimization, statistical estimation, online learn-
ing, etc. SGD (also known as stochastic approximation, incremental gradient) methods
have been extensively studied and well recognized as fast first-order methods which
can be adapted to deal with problems involving large-scale or streaming data. For
certain special cases, it has been shown that SGD exhibit the optimal sample error
complexity in the statistical sense.

Classical SGD methods update iteratively by using “unbiased” samples of the iter-
ates’ gradients. In other words, the objective function is required to be linear in the
sampling probabilities. Indeed, this linearity is the key to analyze SGD and leads to
many nice properties of SGD. However, there has been little study on how to use SGD
and how it performs without linearity in the sampling probabilities.

In this paper, we aim to explore the regime where the linearity in sampling prob-
abilities is lost. We will develop a class of methods that we refer to as stochastic
compositional gradient methods (SCGD), analyze their convergence properties, and
demonstrate their potential applications to a broader range of stochastic problems.

Consider the optimization problem

$$\min_{x \in \mathcal{X}} \left\{ F(x) = (f \circ g)(x) \right\},$$

(1)

where \( f : \mathbb{R}^m \mapsto \mathbb{R} \) is a continuous function, \( g : \mathbb{R}^m \mapsto \mathbb{R}^m \) is a continuous mapping, \( F : \mathbb{R}^n \mapsto \mathbb{R} \) is the function composition defined as

\[
F(x) = (f \circ g)(x) = f(g(x)),
\]

for all \( x \in \mathbb{R}^n \), and \( \mathcal{X} \) is a convex and closed set in \( \mathbb{R}^n \). We assume throughout that there exists at least one optimal solution \( x^* \) to problem (1), and we denote by \( \mathcal{X}^* \) the
set of all optimal solutions. We mainly focus on the case where \( F = f \circ g \) is convex.
However, we do no require either the outer function \( f \) or the inner function \( g \) to be
convex or monotone. We require that \( f \) be continuously differentiable, but we allow
\( g \) and \( F \) to be nonsmooth.

We are interested in situations where \( f \) and \( g \) take the form of expected values

\[
f(y) = \mathbb{E}[f_v(y)], \quad g(x) = \mathbb{E}[g_w(x)],
\]

(2)
for all $y \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, where $f_\nu : \mathbb{R}^m \mapsto \mathbb{R}$, $g_\omega : \mathbb{R}^n \mapsto \mathbb{R}^m$ are functions parameterized by random variables $\nu$ and $\omega$, respectively, and $\mathbb{E} [\cdot]$ is an expectation taken over the probability space of $(\nu, \omega)$. We allow $\nu$, $\omega$ to be dependent to accommodate practical situations where $\omega$ can only be sampled conditioned on $\nu$. For readers who are not concerned with this, it is all right to view $\nu$, $\omega$ as independent random variables throughout this paper.

In this paper, we consider the first-order stochastic optimization setting where the functions $f$ or $g$ cannot be directly accessed. Suppose that we have access to a Sampling Oracle ($SO$) such that:

- Given some $x \in \mathcal{X}$, the oracle returns a random vector $g_\omega(x) \in \mathbb{R}^m$.
- Given some $\bar{x} \in \mathcal{X}$, the oracle returns an $n \times m$ matrix, $\tilde{\nabla} g_\omega(\bar{x}) \in \mathbb{R}^{n \times m}$, whose columns are noisy gradient/subgradient vectors.
- Given some $y \in \mathbb{R}^m$, the oracle returns a noisy gradient vector $\nabla f_\nu(y) \in \mathbb{R}^m$.

More detailed specifications on the $SO$ will be provided later. Our algorithmic objective is to solve problem (1)–(2) using fast updates while making queries to $SO$, without storing past query responses.

The stochastic problem of the form (1)–(2) is very common in operations research applications. Examples of applications include statistical learning, stochastic minimax problems, dynamic programming, estimation of large deviation rate, derivative-free optimization and risk-averse optimization. We will provide more details on these examples in Sects. 4 and 5. Indeed, the composition problem (1)–(2) is a ubiquitous model in practice, and is calling for efficient simulation-based algorithms.

Let us elaborate on the difficulty of problem (1)–(2) under the stochastic sampling framework. Before presenting our algorithmic solution, we consider two plausible alternative solutions: an approach using the Fenchel transform and an approach using the sample average approximation (SAA). In the special case where $f$ is convex and explicitly known, one may attempt to consider the Fenchel dual representation of problem (1), i.e.,

$$\min_{x \in \mathcal{X}} \sup_{\mu \in \mathbb{R}^m} \left\{ \mu' \mathbb{E}[g_\omega(x)] - f^*(\mu) \right\},$$

where $f^*$ is the convex conjugate of $f$. However, this saddle point problem is not necessarily convex-concave (unless $g$ is affine). This excludes the possibility of applying primal-dual stochastic approximation algorithms for saddle point problems (e.g., [23]) to our composition problem (1)–(2). In the more general case where $f$ may be nonconvex or not explicitly known, the Fenchel representation is not even available. Given
these restrictions, the Fenchel transform is not a valid approach for the stochastic composition problem (1)–(2).

To solve the stochastic problem (1), one may attempt to solve instead the SAA problem:

$$\min_{x \in \mathcal{X}} \frac{1}{N} \sum_{i=1}^{N} f_{v_i} \left( \frac{1}{N} \sum_{j=1}^{N} g_{w_{ij}}(x) \right),$$

in which the expectation is replaced by empirical means over some data set $$\{v_i, w_{ij}\}_{i=1}^{N} \{j=1}^{N}$$. This yields a deterministic problem involving large-scale data. It also bears similarity to SAA of three-stage stochastic programming. Two issues arise: (i) How many samples are needed for the SAA to be sufficiently accurate? (ii) How to solve the SAA problem efficiently? Regarding issue (i): As there are two nested levels of expectation in problem (1), we conjecture that the SAA might need $$O(N^2)$$ data points to reach a certain accuracy level, which can be reached by classical SAA using $$O(N)$$ samples if there is only one level of expectation. Analogous sample complexity results have been established for multi-stage stochastic programming; see [31] Section 5.8. We will show that, by using stochastic gradient-like updates, our SCGD algorithms achieve a substantially better error sample complexity. Regarding issue (ii): The SAA problem, although deterministic, can be very difficult to solve when the data size $$N^2$$ is large. For big data-based problem, classical optimization techniques no longer apply as the computer memory is not sufficient to store all the data samples. To address this issue, we may interpret the SAA problem as a stochastic problem of the form (1) defined over the empirical distribution. This allows the proposed SCGD algorithms be applied to the SAA problem by processing one data point (or a small subset of data) at every iteration and making fast updates. Therefore, our proposed algorithms are useful for not only the stochastic problem (1)–(2) but also its SAA.

Now let us demonstrate why classical SGD is not applicable to problem (1)–(2). We write the gradient of $$F = f \circ g$$ using the chain rule (assuming that $$f, g$$ are differentiable):

$$\nabla F(x) = \nabla g(x) \nabla f(g(x)).$$

To apply the SGD, we need the following unbiased sample of $$\nabla F$$ at a given $$x$$, e.g.,

$$\nabla g_{w}(x) \nabla f_{v}(g(x)).$$

However, this sample gradient is not available upon one single query to $$\mathcal{SO}$$. This is because the value $$g(x)$$ is unknown for any given $$x$$ but can only be estimated by noisy function evaluations.

In order to address these difficulties, we are motivated to adopt a quasi-gradient approach and approximate $$g(x)$$ using samples. We aim to construct the approximations in such a way that they can be calculated at a minimal computational cost with small memory overhead, or even on the run. For this purpose, we propose a class of methods.
that update based on queries to $SO$ and involve two iterations with different purposes and different stepsizes:

(i) One iteration maintains a sequence of estimates $\{x_k\}$ of $x^*$ by performing stochastic gradient-like updates. It uses a fast-diminishing stepsize $\{\alpha_k\}$ and converges at a relatively slow rate.

(ii) The other iteration maintains a sequence of running estimates $\{y_k\}$ of $g(x^*)$ by iterative averaging. It uses a slow-diminishing stepsize $\{\beta_k\}$ and converges at a relatively fast rate.

We refer to this class of methods as stochastic compositional gradient descent (SCGD). To guarantee the convergence of SCGD, the key observation is that we need two different stepsizes $\{\alpha_k\}$ and $\{\beta_k\}$, for maintaining estimates of $x^*$ and $g(x^*)$, respectively. The estimates of $g(x^*)$ and the estimates of $x^*$ can be viewed as two entangled stochastic processes. By controlling the stepsizes, we are able to show that $\{x_k\}$ converges to some optimal solution $x^*$ with probability 1 at favorable rates.

Related works The proposed SCGD is closely related to the classical SGD and stochastic approximation (SA) methods. In the case where $f$ is a linear function, our method reduces to the well-known SGD or SA which has been extensively studied. Similar to several sources on SA (see e.g., the textbooks by Kushner and Yin [19], by Benveniste et al. [3], by Borkar [9], by Bertsekas and Tsitsiklis [6]), we use a supermartingale convergence argument. In addition, we use the average of iterates to improve stability and convergence rate, where the averaging idea originates from Polyak and Juditsky [26].

The key feature of SCGD is the use of two entangled iterations, which use different stepsizes and exhibit convergent behaviors on different timescales—one is significantly faster than the other one. The idea of using multiple timescales and quasi-gradient has a long history in the literature of SA; see for example Kiefer and Wolfowitz [16], Korostelev [18], Borkar [8], Bhatnagara and Borkar [7], Konda and Tsitsiklis [17]. In the early Ukrainian literature on stochastic programming, Ermoliev [12] considered a simple two-timescale SA scheme for optimizing the composition of two expected-value functions, and showed its convergence under basic assumptions; see also [11] Section 6.7 for a brief discussion. However, there has been little further development of this idea since then, at least to the best knowledge of the authors.

The idea of stochastic gradient is also related to the class of incremental methods, which are developed for minimizing the sum of a large number of component functions. These methods update incrementally by making use of one component at a time, through a gradient-type or proximal-type iteration (see for example, [5,21,22]). These methods have been extended to deal with a large number of constraints $X = \cap X_i$ by random projection [20,34], and also to deal with stochastic variational inequalities [33]. However, existing incremental treatments do not apply to the optimization of nonlinear functions of the sum of many component functions, which is a special case of the problem we want to address here.

Within the statistical machine learning community, SGD for the unconstrained expectation minimization problem

$$\min_x \mathbb{E} [f_v(x)]$$

(3)
has drawn significant attention. Recent advances are mostly focused on the sample error complexity of SGD in dealing with independent or adversary samples. Without smoothness assumption, it has been shown that after $k$ samples/iterations, the average of the iterates has $O(1/k)$ optimization error for strongly convex objective, and $O(1/\sqrt{k})$ error for general convex objective (see Rakhlin et al. [27], Shamir and Zhang [30]). To complete the picture, for nonsmooth problems with noisy gradients, there are $\Theta(1/k)$ and $\Theta(1/\sqrt{k})$ minimax information-theoretic lower bounds for convex and strongly convex problems, respectively (see e.g., Agarwal et al. [1], and see the book by Nesterov and Yudin [24] for a comprehensive study on optimization complexities).

Regarding problem (3), there has been a line of works on optimal SA/SGD methods, e.g., the accelerated SA by Ghadimi and Lan [13,14], which achieves sample error complexity that is non-improvable not only in $k$ but also in problem parameters such as the modulus of strong convexity. In contrast, to the best knowledge of the authors, there is no prior work on the sample-error complexity for the stochastic composition problem (1)–(2) where the objective is a composition of multiple stochastic functions. In addition, there has been a recent work [10] that establishes central limit theorem for risk composition functions.

**Scope of work** We focus on the stochastic optimization problem (1)–(2) to minimize the composition of two expected-value functions. The difficulty lies in the fact that the objective function is nonlinear in the sampling/simulation probabilities. Algorithmically, we aim to develop fast solutions that update using noisy samples. Analytically, we aim to provide theoretical guarantee in terms of optimization error and sample complexity. This study provides new insights into the level of difficulty of stochastic optimization when there exists an interplay between nonlinearity and stochasticity.

The main contributions of this paper are four-folded:

- We formalize the stochastic optimization problem (1)–(2), which calls for efficient algorithms that make fast updates by querying a given sampling oracle. We propose two instances of such algorithms which we refer to as SCGD methods. Several practical instances of problem (1)–(2) have been identified, with applications in statistical learning, dynamic programming, risk management, etc.
- The first basic SCGD algorithm that comes to our mind is related to a long-dated two-timescale SA method for the root finding problem (see Ermoliev [12]). We provide a more general proof of convergence using a coupled supermartingale convergence argument. This proof of convergence suggests that the basic SCGD applies to a broader range of problems including constrained and nonsmooth optimization. We analyze its rate of convergence in terms of optimization error after making $k$ queries to the sampling oracle, for both convex and strongly convex problems. In addition, we prove that the limit point of SCGD must be stationary for nonconvex problems and analyze the corresponding convergence rate. To the best knowledge of the authors, this is the first result on convergence rate/sample complexity for such algorithms.
- Motivated by the basic SCGD, we propose a new accelerated algorithm which is tailored to the sampling process and has improved convergence rate for smooth optimization. The accelerated SCGD uses an additional extrapolation step before querying for a noisy function evaluation. The extrapolation step leverages the
Table 1  Summary of best known sample error complexities

|                      | \( \min_x (\mathbb{E}[f_v] \circ \mathbb{E}[g_w])(x) \) | \( \min_x \mathbb{E}[f_v(x)] \) |
|----------------------|-------------------------------------------------|---------------------------------|
| **Non-smooth**       | \( \mathcal{O}(k^{-1/4}) \)                   | \( \mathcal{O}(k^{-1/2}) \)    |
| **Smooth**           | \( \mathcal{O}(k^{-2/7}) \)                   | \( \mathcal{O}(k^{-4/5}) \)    |

gradient continuity of smooth problems and reduces the bias in the estimates. We prove that the accelerated SCGD converges with probability 1 to an optimal solution for convex problems. More importantly, we show that the accelerated SCGD exhibits improved rate of convergence/sample convexity with the presence of smoothness in the objective function. We analyze the error sample complexity for the algorithm, in the three cases of general convex problems, strongly convex problems, and nonconvex problems, respectively.

– Our sample error complexity analysis for SCGD provides the first few benchmarks for the stochastic composition problem (1)–(2). A summary of these results is given by Table 1, and is compared with the best known results for the expectation minimization problem (3). These new benchmarks suggest that, although there are two nested levels of expectations in problem (1)–(2), we need only sub-quadratic number of samples to obtain sufficiently small optimization error.

Let us make some comparisons between analogous problems and analogous algorithms according to Table 1. First we compare the SCGD and SGD, which apply to two different problems, (1)–(2) and (3), respectively. We note that the best known convergence rate for SGD is slightly better than that of SCGD. This coincides with our intuition that the nonlinear stochastic problem (1)–(2) is generally “harder” than problem (3). Second we compare the basic SCGD and the accelerated SCGD. We note that the presence of smoothness helps the convergence rate for the stochastic method. This is consistent with the belief that nonsmooth optimization is usually “harder” than smooth optimization. An interesting insight drawn from these comparisons is: the “level of difficulty” of stochastic optimization is related to both the nonlinearity of the objective function in sampling probabilities and the smoothness of the objective in the decision variables.

The rest of this paper is organized as follows. In Sect. 2, we analyze the basic SCGD and prove its almost sure convergence and rate of convergence for convex, strongly convex, and nonconvex problems, respectively. In Sect. 3, we propose the accelerated SCGD, and we prove that it exhibits improved convergence rates for smooth optimization problems in each of the cases of convex, strongly convex, and nonconvex objectives. In Sect. 4, we discuss several practical instances of the stochastic program (1)–(2) to which SCGD naturally applies, including statistical learning, multi-stage optimization, estimation of rare probabilities, etc. We also discuss the extension of SCGD to the derivative-free setting where only the zeroth-order information of \( f \) and \( g \) is available through sampling. In Sect. 5, we present two numerical experiments in which we apply the SCGD methods to a statistical learning problem and a dynamic programming problem, respectively.
Notations All vectors are considered as column vectors. For a vector $x \in \mathbb{R}^n$, we denote by $x'$ its transpose, and denote by $\|x\| = \sqrt{x'x}$ its Euclidean norm. For a matrix $A \in \mathbb{R}^{n \times n}$, we denote by $\|A\| = \max \{ \|Ax\| : \|x\| = 1 \}$ its Euclidean norm. For two sequences $\{a_k\}$, $\{b_k\}$, we denote by $a_k = O(b_k)$ if there exists $c > 0$ such that $\|a_k\| \leq c\|b_k\|$ for all $k$; we denote by $a_k = \Theta(b_k)$ if there exists $c_1 > c_2 > 0$ such that $c_2\|b_k\| \leq \|a_k\| \leq c_1\|b_k\|$ for all $k$. For a set $\mathcal{X} \subset \mathbb{R}^n$ and vector $y \in \mathbb{R}^n$, we denote by $\Pi_{\mathcal{X}}(y) = \text{argmin}_{x \in \mathcal{X}} \|y - x\|^2$ the Euclidean projection of $y$ on $\mathcal{X}$, where the minimization is always uniquely attained if $\mathcal{X}$ is nonempty, convex and closed. For a function $f(x)$, we denote by $\nabla f(x)$ its gradient at $x$ if $f$ is differentiable, denote by $\partial f(x)$ its subdifferential at $x$, and denote by $\widetilde{\nabla} f(x)$ some subgradient at $x$ (to be specified in the context). We denote by “$\rightarrow$” “converge to”, denote by “$\xrightarrow{a.s.}$” “converge almost surely to”, and denote by “$\xrightarrow{w.p.1}$” “with probability 1.”

2 Stochastic compositional gradient descent: a basic algorithm

We start with a basic instance of the stochastic compositional gradient descent algorithm, which we refer to as the basic SCGD; see Algorithm 1. This algorithm alternates between two steps: updating the solution $x_k$ by a stochastic quasi-gradient iteration, and estimating $g(x_k)$ by an iterative weighted average of past values $\{g_w(x_t)\}_{t=0}^k$.

Algorithm 1 Basic SCGD

Input: $x_0 \in \mathbb{R}^n$, $y_0 \in \mathbb{R}^m$, $\mathcal{SO}$, stepsizes $\{a_k\} \subset \mathbb{R}^+$, $\{\beta_k\} \subset (0, 1]$ satisfying $\frac{a_k-1}{\beta_k} \rightarrow 0$.

Output: The sequence $\{x_k\}$.

1: for $k = 0, 1, \ldots$ do
2: Query $\mathcal{SO}$ for the sample values of $g$ at $x_k$, obtaining $g_{w_k}(x_k)$ and $\widetilde{\nabla} g_{w_k}(x_k)$.
3: Update
   
   \[ y_{k+1} = (1 - \beta_k) y_k + \beta_k g_{w_k}(x_k). \]
4: Query $\mathcal{SO}$ for the sample gradient of $f$ at $y_{k+1}$, obtaining $\nabla f_{w_k}(y_{k+1})$.
5: Update
   
   \[ x_{k+1} = \Pi_{\mathcal{X}}\left\{ x_k - a_k \widetilde{\nabla} g_{w_k}(x_k) \nabla f_{w_k}(y_{k+1}) \right\}. \]

6: end for

Algorithm 1 is a natural iteration that comes to one’s mind for solving problem (1). A special case of the algorithm (when $\mathcal{X} = \mathbb{R}^n$) has been considered for the root finding problem in [12]. However, there is no known result on the convergence rate or sample complexity for this algorithm, which we establish next. Algorithm 1 can be viewed as a two-timescale stochastic approximation algorithm, which uses two stepsizes $a_k, \beta_k$. To ensure that the coupling process converges, it is necessary that the estimate of $g(x_k)$ is updated on a faster timescale comparing with the estimate of $x^*$. This requires that $a_k$ diminishes to zero at a faster rate than $\beta_k$. Throughout this paper, we assume implicitly that stepsizes $\{a_k\}$, $\{\beta_k\}$ are diminishing sequences and satisfy $\lim_{k \rightarrow \infty} a_{k-1}/\beta_k = 0$. 

Springer
To analyze the sample complexity of Algorithm 1, we consider its averaged iterates given by
\[ \hat{x}_k = \frac{1}{N_k} \sum_{t=k-N_k}^k x_t, \]
where \( N_k \in (0, k] \) is a positive integer selected according to \( k \). As long as \( N_k = \Theta(k) \), the asymptotic behavior of \( \hat{x}_k \) is close to that of \( x_k \) and is more stable. This is known to be true for classical stochastic approximation and stochastic gradient descent method (e.g., see [26]). Throughout the rest of the paper, we take for simplicity that \( N_k = \lceil k/2 \rceil \) for all \( k \). For other reasonable choices of \( N_k \) [e.g., \( N_k = \gamma k \) for some \( \gamma \in (0, 1) \)], our analysis can be easily adapted to yield convergence and convergence rate results that are analogous to the current ones. A more detailed account on choices of \( N_k \) and variants of averaging schemes is beyond the scope of this paper. For readers interested in this subject, we refer to the works [27,30] and references therein.

2.1 Preliminaries

In this paper, we focus on convex problems and also provide results for nonconvex but differentiable objectives. For simplicity of notation, we denote by \( \partial F(x) \) the subdifferential of \( F \) when it is convex and the singleton set \( \{\nabla F(x)\} \) when \( F \) is differentiable but nonconvex. Throughout our discussions, we make the following assumptions.

**Assumption 1** Let \( C_g, C_f, V_g, L_f \) be positive scalars.

(i) The outer function \( f \) is continuously differentiable, the inner function \( g \) is continuous, the feasible set \( \mathcal{X} \) is closed and convex, and there exists at least one optimal solution \( x^* \) to problem (1).

(ii) The random variables, \((w_0, v_0), (w_1, v_1), (w_2, v_2), \ldots\), are independent and identically distributed with a complete probability measure, and
\[
E[g_{w_0}(x)] = g(x), \quad E[\tilde{\nabla} g_{w_0}(x) \nabla f_{v_0}(g(x))] \in \partial F(x), \quad \forall x \in \mathcal{X}.
\]

(iii) The function \( g(\cdot) \) is Lipschitz continuous with parameter \( C_g \), and the samples \( g_{w_k}(\cdot) \) and \( \tilde{\nabla} g_{w_k}(\cdot) \) have bounded second moments such that with probability 1
\[
E[\|\tilde{\nabla} g_{w_0}(x)\|^2 | v_0] \leq C_g, \quad E[\|g_{w_0}(x) - g(x)\|^2] \leq V_g,
\]
for all \( x \in \mathcal{X} \).

(iv) The functions \( f \) and \( f_v \) have Lipschitz continuous gradients, such that with probability 1
\[
E[\|\nabla f_{v_0}(y)\|^2] \leq C_f, \quad \|\nabla f_{v_0}(y) - \nabla f_{v_0}(\overline{y})\| \leq L_f \|y - \overline{y}\|,
\]
for all \( y, \overline{y} \in \mathbb{R}^m \).
Assumption 1 is quite general and can be verified easily in practice. It essentially requires that: (i) the outer and inner functions $f, g$ are sufficiently smooth; (ii) the sample first-order information are unbiased; (iii) random variables generated by the Sampling Oracle ($SO$) have bounded second moments. The smoothness of $f, g$ needs to be verified case-by-case when specific problem instances are given. The unbiasedness and second-moment boundedness are typical assumptions in stochastic sampling. They are very mild and are satisfied by the majority of estimation and machine learning applications, as long as the underlying probabilistic distributions have reasonable tail distributions.

In Algorithm 1 and Assumption 1, we intentionally do not specify what $\tilde{\nabla} g_w(x)$ is. Let us view $\tilde{\nabla} g_w(x)$ as a noisy sample of some generalized gradient. If $g$ is differentiable, $\tilde{\nabla} g_w(x)$ can be a noisy gradient. If $g$ is convex but nonsmooth, $\tilde{\nabla} g_w(x)$ can be a noisy subgradient. If $g$ is nonconvex and nonsmooth, $\tilde{\nabla} g_w(x)$ can be some directional derivative under additional conditions. As long as the chain rule holds in expectation [Assumption 1 (ii)], i.e.,

$$ E[\tilde{\nabla} g_{w_0}(x)\nabla f_{v_0}(g(x))] \in \partial F(x), $$

the SCGD algorithm is guaranteed to work.

We denote by $F_k$ the collection of random variables

$$ \{x_0, \ldots, x_k, y_0, \ldots, y_k, w_0, \ldots, w_{k-1}, v_0, \ldots, v_{k-1}\}. $$

Let us first derive some basic inequalities based on Assumption 1. By using Assumption 1 (iii), (iv), we can show that $\|x_{k+1} - x_k\|^2$ is of order $\alpha_k^2$, i.e.,

$$ E[\|x_{k+1} - x_k\|^2 | F_k] \leq \alpha_k^2 E[\|\tilde{\nabla} g_{w_k}(x_k)\|^2 \|\nabla f_{v_k}(y_{k+1})\|^2 | F_k] $$

$$ = \alpha_k^2 E[E[\|\tilde{\nabla} g_{w_k}(x_k)\|^2 | v_k]\|\nabla f_{v_k}(y_{k+1})\|^2 | F_k] $$

$$ \leq \alpha_k^2 C_g C_f, \quad (4) $$

for all $k$, with probability 1.

While analyzing Algorithm 1, we observe that the two error sequences $\{x_k - x^*\}$ and $\{g(x_k) - y_{k+1}\}$, are coupled together in their asymptotic behaviors. Then we use the following coupled supermartingale convergence lemma by Wang and Bertsekas [34] to prove their almost sure convergence. This lemma generalizes the earlier supermartingale convergence lemma by Robbins and Siegmund [29]. It characterizes the inherent convergence of two random processes that are coupled together.

**Lemma 1** (Coupled supermartingale convergence Lemma) Let $\{\xi_k\}, \{\zeta_k\}, \{u_k\}, \{\overline{u}_k\}, \{\eta_k\}, \{\theta_k\}, \{e_k\}, \{\mu_k\},$ and $\{v_k\}$ be sequences of nonnegative random variables such that

$$ E[\xi_{k+1} | G_k] \leq (1 + \eta_k)\xi_k - u_k + c\theta_k \xi_k + \mu_k, $$

$$ E[\zeta_{k+1} | G_k] \leq (1 - \theta_k)\zeta_k - \overline{u}_k + e_k \xi_k + v_k, $$
where $G_k$ is the collection $\{\xi_0, \ldots, \xi_k, \zeta_0, \ldots, \zeta_k, u_0, \ldots, u_k, \overline{u}_0, \ldots, \overline{u}_k, \eta_0, \ldots, \eta_k, \theta_0, \ldots, \theta_k, \epsilon_0, \ldots, \epsilon_k, \mu_0, \ldots, \mu_k, v_0, \ldots, v_k\}$, and $c$ is a positive scalar. Also, assume that

$$
\sum_{k=0}^{\infty} \eta_k < \infty, \quad \sum_{k=0}^{\infty} \epsilon_k < \infty, \quad \sum_{k=0}^{\infty} \mu_k < \infty, \quad \sum_{k=0}^{\infty} v_k < \infty, \quad w.p.1.
$$

Then $\xi_k$ and $\zeta_k$ converge almost surely to two nonnegative random variables, respectively, and we have

$$
\sum_{k=0}^{\infty} u_k < \infty, \quad \sum_{k=0}^{\infty} \overline{u}_k < \infty, \quad \sum_{k=0}^{\infty} \theta_k \xi_k < \infty, \quad w.p.1.
$$

Now we analyze the $y$-step in the basic SCGD Algorithm 1 and provide a recursive error bound for $\{y_{k+1} - g(x_k)\}$. The motivation of introducing the auxiliary variable $y_{k+1}$ is to track the unknown quantity $g(x_k)$ by averaging past samples $\{g_u, (x_t)\}_{t=0}^{k}$. We show in the next lemma that the distance between $y_{k+1}$ and $g(x_k)$ decreases “in expectation” according to a supermartingale-type inequality. Intuitively, this suggests that the tracking variable $y_{k+1}$ and the unknown $g(x_k)$ “converge” to each other. It is worth pointing out that this result does not require the convexity of the objective function.

**Lemma 2** Let Assumption 1 hold, and let $\{(x_k, y_k)\}$ be generated by Algorithm 1. Then:

(a) With probability 1, we have

$$
\mathbb{E} \left[ \| y_{k+1} - g(x_k) \|^2 \mid F_k \right] 
\leq (1 - \beta_k) \| y_k - g(x_{k-1}) \|^2 + \beta_k^{-1} C_g \| x_k - x_{k-1} \|^2 + 2 V_g \beta_k^2. \quad (5)
$$

(b) If $\sum_{k=1}^{\infty} \alpha_k^2 \beta_k^{-1} < \infty$, we have $\sum_{k=0}^{\infty} \beta_k^{-1} C_g \| x_k - x_{k-1} \|^2 < \infty$ with probability 1.

(c) There exists $D_y \geq 0$ such that $\mathbb{E} \left[ \| y_{k+1} - g(x_k) \|^2 \right] \leq D_y$ for all $k$.

**Proof** See Supplementary Materials Section G.1 for the detailed proof.

Next we analyze the improvement of the optimality error $\| x_k - x^* \|$ as the algorithm proceeds. As we show in the following lemma, the optimality improvement is also coupled with the estimation error $\| g(x_k) - y_{k+1} \|$.

**Lemma 3** Let Assumption 1 hold, and let $F = f \circ g$ be convex. Then Algorithm 1 generates a sequence $\{(x_k, y_k)\}$ such that, with probability 1

$$
\mathbb{E} \left[ \| x_{k+1} - x^* \|^2 \mid F_k \right] 
\leq \left( 1 + L_f^2 C_g \beta_k^{-1} \alpha_k^2 \right) \| x_k - x^* \|^2 - 2 \alpha_k (F(x_k) - F^*) + C_f C_g \alpha_k^2 + \beta_k \mathbb{E} \left[ \| g(x_k) - y_{k+1} \|^2 \mid F_k \right]. \quad (6)
$$
Proof. See Supplementary Materials Section G.2 for the detailed proof.

Remarks on the $O(\cdot)$ Notation In the rest of this paper, the convergence and convergence rate analysis is quite tedious. It involves asymptotic analysis of a number of sequences. For simplicity, we will use the $O(\cdot)$ notation frequently. This is for two reasons:

(i) We use $O(\cdot)$ to avoid defining too many constants $c_1, c_2, c_3, c_4, c_5, \ldots$ that are irrelevant with parameters of problem (1) and $SO$. In the analysis, we try to keep problem-dependent parameters, such as $C_f, C_g, V_g, L_F$, inside the $O(\cdot)$ in order to provide insights. We remark that the constants in the $O(\cdot)$ are not material to our analysis. Our key analysis revolves around the iteration/sample complexity, i.e., the polynomial order of $k$ inside the $O(\cdot)$.

(ii) We use $O(\cdot)$ to avoid situations such that “some recursive inequality holds for $k$ sufficiently large.” Let $\{a_k\}, \{b_k\}$ be two nonnegative sequences. Whenever we have $a_k \leq O(b_k)$ for $k$ sufficiently large, we have $a_k \leq O(b_k)$ for all $k$ with a larger constant hidden in the $O(\cdot)$ notation.

Due to the complicated nature of the convergence rate analysis, we find $O(\cdot)$ to be a useful tool to keep the analysis compact and readable. Note that the use of $O(\cdot)$ only hides constants that are problem-independent. It does not affect the convergence and rate of convergence results.

2.2 Almost sure convergence

In what follows, we present a more general proof of the almost sure convergence, based on a coupled martingale convergence argument. Specifically, we show that the basic SCGD is guaranteed to find an optimal solution for convex optimization problems if such a solution exists. We also show that any limit point generated by the basic SCGD must be stationary for problems that are not necessarily convex. This proof of convergence sheds light on the convergence rate and error-sample complexity analysis which we develop later.

**Theorem 1** (Almost sure convergence of basic SCGD) Let Assumption 1 hold, and let the stepsizes $\{\alpha_k\}$ and $\{\beta_k\}$ be such that

$$\sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} \beta_k = \infty, \quad \sum_{k=0}^{\infty} \left( \alpha_k^2 + \beta_k^2 + \frac{\alpha_k^2}{\beta_k} \right) < \infty.$$ 

Let $\{(x_k, y_k)\}$ be the sequence generated by the basic SCGD Algorithm 1 starting with an arbitrary initial point $(x_0, y_0)$. Then:

(a) If $F$ is convex, $x_k$ converges almost surely to a random point in the set of optimal solutions of problem (1).

(b) If $F$ has Lipschitz continuous gradient, $X = \mathbb{R}^n$, and all samples generated by the $SO$ are uniformly bounded, any limit point of the sequence $\{x_k\}$ is a stationary point with probability 1.
Proof (a) By Lemma 2, we have $\sum_{k=0}^{\infty} \frac{C_g}{\beta_k} \|x_k - x_{k-1}\|^2 < \infty$ with probability 1. This, together with the assumption $\sum_{k=0}^{\infty} (\alpha_k^2 + \beta_k^2 + \alpha_k^2/\beta_k) < \infty$ and the fact $F(x_k) - F^* \geq 0$, suggests that the supermartingale convergence Lemma 1 applies to Eqs. (5) and (6).

Take arbitrary $x^* \in X^*$. Let us apply Lemma 1 to Eq. (5) and Eq. (6) with

$$\xi_k = \|x_k - x^*\|^2, \quad \zeta_k = \|g(x_{k-1}) - y_k\|^2, \quad u_k = 2\alpha_k (F(x_k) - F^*), \quad \overline{\alpha}_k = 0,$$

$$\mu_k = C_f C_g \alpha_k^2, \quad v_k = \beta_k^{-1} C_g \|x_k - x_{k-1}\|^2 + 2V_g \beta_k^2.$$

Then we obtain that $\{\|x_k - x^*\|\}$ and $\{\|y_{k+1} - g(x_k)\|\}$ converge almost surely to two random variables, and

$$\liminf_{k \to \infty} \alpha_k (F(x_k) - F^*) < \infty, \quad \liminf_{k \to \infty} \beta_k \|y_{k+1} - g(x_k)\|^2 < \infty, \quad w.p.1,$$

which further implies that

$$\liminf_{k \to \infty} F(x_k) = F^*, \quad \liminf_{k \to \infty} \|y_{k+1} - g(x_k)\|^2 = 0, \quad w.p.1.$$

Let $\Omega_{x^*}$ be the collection of sample paths that $\Omega_{x^*} = \{\omega : \lim_{k \to \infty} \|x_k(\omega) - x^*\| \text{ exists}\}$. It has been proved that $P(\Omega_{x^*}) = 1$ for any $x^* \in X^*$. We claim that $\cap_{x^* \in X^*} \Omega_{x^*}$ is measurable and $P(\cap_{x^* \in X^*} \Omega_{x^*}) = 1$. To see this, we consider a countable dense subset $X^*_{\omega}$ of $X^*$. By the fact that $F$ is convex, we have that the set $X^*_{\omega} \subset \mathbb{R}^n$ is separable, and such $X^*_{\omega}$ exists. So the probability of non-convergence for some $x^* \in X^*_{\omega}$ is the probability of a union of countably many sets, each having probability 0. As a result, we have

$$P \left( \cap_{X^*_{\omega}} \Omega_{x^*} \right) = 1 - P \left( \cup_{X^*_{\omega}} \Omega_{x^*}^c \right) \geq 1 - \sum_{x^* \in X^*_{\omega}} P \left( \Omega_{x^*}^c \right) = 1.$$

Now we consider an arbitrary $\tilde{x} \in X^*$, which is the limit of a sequence of optimal solutions $\{\tilde{x}_j\}_{j=1}^{\infty} \subset X^*_{\omega}$. By using a limit point argument, it is not hard to see that $\|x_k(\omega) - \tilde{x}\|$ is convergent as long as $\|x_k(\omega) - \tilde{x}_j\|$ is convergent for all $j$. Consider arbitrary $\omega \in \cap_{X^*_{\omega}} \Omega_{x^*}$. We have the triangle inequalities:

$$\|x_k(\omega) - \tilde{x}_j\| - \|\tilde{x}_j - \tilde{x}\| \leq \|x_k(\omega) - \tilde{x}\| \leq \|x_k(\omega) - \tilde{x}_j\| + \|\tilde{x}_j - \tilde{x}\|.$$

By taking $k \to \infty$ and using the fact that $\lim_{k \to \infty} \|x_k(\omega) - \tilde{x}_j\|$ exists, we obtain

$$\lim_{k \to \infty} \|x_k(\omega) - \tilde{x}_j\| - \|\tilde{x}_j - \tilde{x}\| \leq \liminf_{k \to \infty} \|x_k(\omega) - \tilde{x}\| \leq \limsup_{k \to \infty} \|x_k(\omega) - \tilde{x}_j\| \leq \lim_{k \to \infty} \|x_k(\omega) - \tilde{x}_j\| + \|\tilde{x}_j - \tilde{x}\|.$$
So we have
\[
\limsup_{k \to \infty} \|x_k(\omega) - \bar{x}\| - \liminf_{k \to \infty} \|x_k(\omega) - \bar{x}\| \leq 2\|\bar{x}_j - \bar{x}\|.
\]
By taking \(j \to \infty\), we have \(\|\bar{x}_j - \bar{x}\| \to 0\) and
\[
\limsup_{k \to \infty} \|x_k(\omega) - \bar{x}\| = \liminf_{k \to \infty} \|x_k(\omega) - \bar{x}\|.
\]
It follows that the limit of \(\|x_k(\omega) - \bar{x}\|\) exists, therefore \(\omega \in \Omega_{\bar{x}}\). Now we have proved that \(\cap \chi^*_Q \Omega_{x^*} \subset \Omega_{\bar{x}}\) for all \(\bar{x} \in \mathcal{X}^*\). As a result, we have \(\cap \chi^*_Q \Omega_{x^*} \subset \cap \chi^* \Omega_{x^*}\). Note that \(P(\cap \chi^*_Q \Omega_{x^*}) = 1\) and that the considered probability measure is complete. We have that \((\cap \chi^* \Omega_{x^*})^c\) is the subset of a null set \((\cap \chi^*_Q \Omega_{x^*})^c\), therefore it is measurable and satisfies \(P((\cap \chi^*_Q \Omega_{x^*})^c) = 0\). It follows that \(\cap \chi^* \Omega_{x^*}\) is measurable and satisfies \(P(\cap \chi^* \Omega_{x^*}) = 1\). In other words, \(\|x_k - \bar{x}\|\) is convergent for all \(\bar{x} \in \mathcal{X}^*\), with probability 1.

Consider an arbitrary sample trajectory \(\{x_k(\omega)\}\) such that \(\omega \in \cap \chi^* \Omega_{x^*}\) and \(\liminf_{k \to \infty} F(x_k(\omega)) = F^*\). Take arbitrary \(x^* \in \mathcal{X}^*\). Since \(\|x_k(\omega) - x^*\|\) converges, the sequence is bounded. By the continuity of \(F\), the sequence \(\{x_k(\omega)\}\) must have a limit point \(\bar{x}\) being an optimal solution, i.e., \(F(\bar{x}) = F^*\) and \(\bar{x} \in \mathcal{X}^*\). Since \(\omega \in \cap \chi^* \Omega_{x^*} \subset \Omega_{\bar{x}}\), we obtain that \(\|x_k(\omega) - \bar{x}\|\) is also a convergent sequence. Since it is convergent while having a limit point 0, we have \(\|x_k(\omega) - \bar{x}\| \to 0\). Then \(x_k(\omega) \to \bar{x}\) on this sample trajectory. Note that \(\bar{x}\) depends on the sample path \(\omega\), so it is a random variable. Also note that the set of all such sample paths has a probability measure equaling to 1. Therefore \(x_k\) converges almost surely to a random point in the set of optimal solutions of problem (1).

(b) See Supplementary Materials Section A for the detailed proof.

Theorem 1(a) establishes that “for convex optimization, \(x_k\) converges almost surely to an optimal solution.” We remark that the limit optimal solution is a random variable that depends on the realization of specific sample paths. For each realization of the sample path \(\{x_k(\omega)\}\), the iterate is a convergent sequence and the corresponding limit is an optimal solution.

Theorem 1(b) establishes that “for nonconvex optimization, any limit point generated by Algorithm 1 is a stationary point with probability 1.” We remark that this does not exclude the possibility that \(\{x_k\}\) does not have any limit point. If \(F(x)\) is a “bad” function, e.g., \(F(x) = 1/|x|\), it is possible for \(x_k\) to diverge without any limit point. This is a general limitation of the class of gradient descent methods for nonconvex optimization. To guarantee the existence of a limit point, we need some additional assumptions or mechanisms to ensure the boundedness of iterates.
2.3 Rate of convergence

Now we establish the convergence rate and sample complexity of the basic SCGD algorithm. To do this, we consider the averaged iterates of Algorithm 1, given by

$$\hat{x}_k = \frac{1}{N_k} \sum_{t=k-N_k}^k x_t,$$

where we take $N_k = \lceil k/2 \rceil$ for simplicity. Note that the convergence rate is related to the stepsizes of the algorithm. By convention, we choose stepsizes $\{\alpha_k\}$ and $\{\beta_k\}$ as powers of $k$, i.e.,

$$\alpha_k = k^{-a}, \quad \beta_k = k^{-b},$$

and we aim to minimize the error bound over the parameters $a, b$.

In what follows, we analyze three cases separately: (i) the case where $F = f \circ g$ is a convex function; (ii) the case where $F = f \circ g$ is a strongly convex function; (iii) the case where $F = f \circ g$ is not necessarily convex. For the convex and strongly convex cases (i) and (ii), we consider the rate of convergence of Algorithm 1 in terms of the optimality error $F(x_k) - F^*$ and the distance to optimal solution $\|x_k - x^*\|$, respectively. For the nonconvex case (iii), we consider the convergence rate in terms of a metric of nonstationarity.

**Theorem 2** (Convergence rate of basic SCGD for convex problems) Suppose that Assumption 1 holds and $F$ is convex. Let $D_x > 0$ be such that $\sup_{x \in \mathcal{X}} \|x - x^*\|^2 \leq D_x$, and let $D_y > 0$ be the scalar defined in Lemma 2. Let the stepsizes be $\alpha_k = k^{-a}, \beta_k = k^{-b}$, where $a, b$ are scalars in $(0, 1)$. Then:

(a) The averaged iterates generated by Algorithm 1 is such that

$$E[F(\hat{x}_k) - F^*] = O\left((1 + L_f^2 C_g^2)(D_x + D_y)(k^{a-1} + k^{b-a} (\log k)^{1-a+b+1}) + C_f C_g k^{-a} (\log k)^{1-a+1} + V_g k^{a-2b} (\log k)^{1-a+2b+1} + C_g^2 C_f k^{b-a} (\log k)^{1-a+b+1}\right).$$

(b) For $k$ sufficiently large, the preceding bound is minimized at $a = 3/4, b = 1/2$, yielding

$$E[F(\hat{x}_k) - F^*] = O\left(\frac{(1 + L_f^2 C_g^2)(D_x + D_y) + V_g + C_g^2 C_f}{k^{1/4}} + \frac{C_f C_g}{k^{3/4}}\right) = O\left(\frac{1}{k^{1/4}}\right).$$

**Proof** Define the random variable

$$J_k = \|x_k - x^*\|^2 + \|y_k - g(x_{k-1})\|^2.$$
so we have $E[J_k] \leq D_x + D_y \equiv D$ for all $k$. We multiply Eq. (5) by $(1 + \beta_k)$ and take its sum with Eq. (6), and we obtain

$$E[J_{k+1} | \mathcal{F}_k] \leq \left(1 + L_f^2 C_g^2 \frac{\alpha_k^2}{\beta_k^2}\right) J_k - 2\alpha_k (F(x_k) - F^*) + \left(C_f C_g \alpha_k^2 + 2V_g \beta_k^2 (1 + \beta_k) + \frac{C_g(1 + \beta_k)}{\beta_k} \|x_k - x_{k-1}\|^2.\right)

Taking expectation of both sides and using the fact $1 + \beta_k \leq 2$, we obtain

$$E[J_{k+1}] \leq \left(1 + L_f^2 C_g^2 \frac{\alpha_k^2}{\beta_k^2}\right) E[J_k] - 2\alpha_k E[F(x_k) - F^*] + C_f C_g \alpha_k^2 + 4V_g \beta_k^2 + 2C_g^2 C_f \frac{\alpha_k^2}{\beta_k^2}.\right.

Let $N > 0$. By reordering the terms in the preceding relation and taking its sum over $k - N, \ldots, k$, we have

$$2 \sum_{t=k-N}^{k} E[F(x_t) - F^*] \leq \left(1 + L_f^2 C_g^2 \frac{\alpha_t^2}{\beta_t^2}\right) E[J_t] - 1 \alpha_t E[J_{t+1}] + \frac{1}{\alpha_{t-1}} E[J_{t-1}] - \frac{1}{\alpha_k} E[J_{k+1}] + \frac{1}{\alpha_{k-N-1}} E[J_{k-N}]

\begin{align*}
&+ L_f^2 C_g^2 \sum_{t=k-N}^{k} \frac{\alpha_t}{\beta_t} E[J_t] + C_f C_g \sum_{t=k-N}^{k} \alpha_t + 4V_g \sum_{t=k-N}^{k} \frac{\beta_t^2}{\alpha_t} + 2C_g^2 C_f \sum_{t=k-N}^{k} \frac{\alpha_t}{\beta_t} \\
&\leq \sum_{t=k-N}^{k} \left(1 - \frac{1}{\alpha_{t-1}}\right) D + \frac{1}{\alpha_{k-N-1}} D + L_f^2 C_g^2 \sum_{t=k-N}^{k} \frac{\alpha_t}{\beta_t} D \\
&+ C_f C_g \sum_{t=k-N}^{k} \alpha_t + 4V_g \sum_{t=k-N}^{k} \frac{\beta_t^2}{\alpha_t} + 2C_g^2 C_f \sum_{t=k-N}^{k} \frac{\alpha_t}{\beta_t} \\
&\leq \frac{1}{\alpha_k} D + L_f^2 C_g^2 \left(\sum_{t=k-N}^{k} \frac{\alpha_t}{\beta_t}\right) D + C_f C_g \sum_{t=k-N}^{k} \alpha_t \\
&+ 4V_g \sum_{t=k-N}^{k} \frac{\beta_t^2}{\alpha_t} + 2C_g^2 C_f \sum_{t=k-N}^{k} \frac{\alpha_t}{\beta_t}.\right.\]
Let $\alpha_k = k^{-a}$, and $\beta_k = k^{-b}$, where $a$, $b$ are scalars in $(0, 1)$. We have

$$2 \sum_{t=k-N}^{k} \mathbb{E} \left[ F(x_t) - F^* \right]$$

$$\leq \mathcal{O} \left( k^a D + L_f^2 C_g^2 \left( k^{1+b-a} - (k-N)^{1+b-a} \right) D \left( \log k \right)^{1_{a=b+1}} \right.$$  

$$\left. + C_f C_g (k^{1-a} - (k-N)^{1-a}) (\log k)^{1_{a=1}} \right. 

$$\left. + V_g (k^{1+a-2b} - (k-N)^{1+a-2b}) (\log k)^{1_{1+a=2b}} \right. 

$$\left. + C_g^2 C_f (k^{1+b-a} - (k-N)^{1+b-a}) (\log k)^{1_{a=b+1}} \right),$$

where $1_A = 1$ if $A$ is true and $1_A = 0$ if $A$ is false. Note that the log $k$ term only occurs in rare situations when we take the sum $\sum_{t=1}^{k} \frac{1}{t}$, which is not of substantial importance to our analysis. Using the convexity of $F$ and taking $N = N_k = k/2$, we obtain

$$\mathbb{E} \left[ F(\hat{x}_k) - F^* \right] \leq \frac{1}{N_k} \sum_{t=k-N_k}^{k} \mathbb{E} \left[ F(x_t) - F^* \right]$$

$$\leq \mathcal{O} \left( (k^{a-1} + L_f^2 C_g^2 k^{b-a} (\log k)^{1_{a=b+1}}) D + C_f C_g k^{-a} (\log k)^{1_{a=1}} \right.$$  

$$\left. + V_g k^{a-2b} (\log k)^{1_{1+a=2b}} + C_g^2 C_f k^{b-a} (\log k)^{1_{a=b+1}} \right).$$

The order of the bound is minimized when $a = 3/4$ and $b = 1/2$, which completes the proof.

Next we consider a special case where $F = f \circ g$ is strongly convex in the following sense: there exists a scalar $\sigma > 0$ such that

$$F(x) - F^* \geq \sigma \|x - x^*\|^2, \quad \forall \ x \in \mathcal{X}. \quad (7)$$

In the next theorem we show that a faster convergence rate can be obtained assuming strong convexity. This is consistent with the well known complexity results for convex optimization. The proof is given at Supplementary Materials Section B.

**Theorem 3** (Convergence rate of basic SCGD for strongly convex problems) *Suppose that Assumption 1 holds and $F$ is strongly convex satisfying (7). Let the stepsizes be*

$$\alpha_k = \frac{1}{k \sigma}, \quad \beta_k = \frac{1}{k^{2/3}}.$$

*Then*

$$\mathbb{E} \left[ \|x_k - x^*\|^2 \right] = \mathcal{O} \left( \frac{C_f C_g \log k}{\sigma^2} + \frac{L_f^2 C_g (C_f C_g^2 / \sigma^2 + V_g) k^{1/3}}{k^{2/3}} \right) = \mathcal{O} \left( \frac{1}{k^{2/3}} \right).$$
and
\[
\mathbb{E} \left[ \| \hat{x}_k - x^* \|^2 \right] = \mathcal{O}\left( \frac{1}{k^{2/3}} \right).
\]

Let us compare the rates of convergence for convex and strongly convex problems after \( k \) queries to the \( \mathcal{SO} \). For convex problems, the error is of the order of \( k^{-1/4} \); while for strongly convex problems, the error is of the order of \( k^{-2/3} \). It is as expected that strongly convex problems are “easier” to solve than those problems lacking strong convexity.

Finally we analyze the behavior of the basic SCGD for nonconvex optimization problems. Without convexity, the algorithm is no longer guaranteed to find a global optimum. However, we have shown that any limit point of iterates produced by the algorithm must be a stationary point for the nonconvex problem. In the next theorem, we provide an estimate that quantifies how fast the non-stationary metric \( \| \nabla F(x_k) \| \) decrease to zero.

**Theorem 4** (Convergence rate of basic SCGD for nonconvex problems) *Suppose Assumption 1 holds, \( F \) is Lipschitz differentiable with parameter \( L_F \), and \( X = \mathbb{R}^n \). Let the stepsizes be*

\[
\alpha_k = k^{-a}, \quad \beta_k = k^{-b},
\]

*where \( a, b \) are scalars in \((0, 1)\) satisfying \( b < a < 2b \). Let*

\[
T_\epsilon = \min \left\{ k : \inf_{0 \leq t \leq k} \mathbb{E} \left[ \| \nabla F(x_t) \|^2 \right] \leq \epsilon \right\}
\]

*then*

\[
T_\epsilon \leq \mathcal{O}(\epsilon^{-1/p}),
\]

*where \( p = \min\{1 - a, a - b, 2b - a, a\} \). By minimizing the complexity bound over \( a, b \), we obtain \( T_\epsilon \leq \mathcal{O}(\epsilon^{-4}) \) with \( a = 3/4, b = 1/2 \).*

*Proof* See Supplementary Materials Section C for the detailed proof. □

### 3 Acceleration for smooth convex optimization

In this section, we propose an accelerated version of SCGD that achieves faster rate of convergence when the objective function is differentiable, which we refer to as *accelerated SCGD*.

Recall our optimization problem

\[
\min_{x \in X} (f \circ g)(x), \text{ where } g(x) = \mathbb{E}[g_w(x)], \quad f(y) = \mathbb{E}[f_v(y)], \quad \forall \ x \in \mathbb{R}^n, \ y \in \mathbb{R}^m
\]

\( \odot \) Springer
In Sect. 2, we have obtained the convergence rate for basic SCGD in the case where \( g_w \) may be nondifferentiable. In this section, we will show this performance can be improved if we restrict to smooth problems where both \( f_v \) and \( g_w \) are differentiable. We state the \textit{accelerated stochastic compositional gradient descent} (accelerated SCGD) method as follows.

**Algorithm 2** Accelerated SCGD (a-SCGD)

**Input:** \( x_0 \in \mathbb{R}^n, y_0 \in \mathbb{R}^m, \mathcal{SO}, \) stepsizes \( \{\alpha_k\} \subset \mathbb{R}^+, \{\beta_k\} \subset (0, 1] \) satisfying \( \alpha_k / \beta_k \to 0. \)

**Output:** The sequence \( \{x_k\}. \)

1: for \( k = 0, 1, 2, \ldots \) do
2: Query \( \mathcal{SO} \) for gradients of \( g, f \) at \( x_k, y_k \) respectively, obtaining \( \nabla g_w(x_k) \) and \( \nabla f_{v_k}(y_k). \)
3: Update
   \[ x_{k+1} = \Pi_X \left[ x_k - \alpha_k \nabla g_w(x_k) \nabla f_{v_k}(y_k) \right], \]
   \[ z_{k+1} = - \left( \frac{1}{\beta_k} - 1 \right) x_k + \frac{1}{\beta_k} x_{k+1}. \]
4: Query \( \mathcal{SO} \) for the sample value of \( g \) at \( z_{k+1} \), obtaining \( g_{w_{k+1}}(z_{k+1}). \)
5: Update
   \[ y_{k+1} = (1 - \beta_k) y_k + \beta_k g_{w_{k+1}}(z_{k+1}). \]
6: end for

In Algorithm 2, \( x_k, z_k \in \mathbb{R}^n, y_k \in \mathbb{R}^m, \nabla g_w \) is the \( n \times m \) matrix with each column being the gradient of the corresponding entry of \( g_w \), and \( \{\alpha_k\}, \{\beta_k\} \) are pre-specified sequences of positive scalars. Similar to the basic-SCGD, we need the stepsize \( \alpha_k \) to converge to 0 faster than \( \beta_k \) converges. Again, it is assumed throughout that \( \{\alpha_k\}, \{\beta_k\} \) are diminishing sequences and \( \alpha_k / \beta_k \to 0. \)

Note that the iteration for \( x_k \) in Algorithm 2 takes the same form as in Algorithm 1. What makes Algorithm 2 different is the auxiliary variable \( z_{k+1} \), which is a new query point obtained by extrapolation. Equivalently, the variable \( x_{k+1} \) can be viewed as an interpolated point given by

\[ x_{k+1} = (1 - \beta_k) x_k + \beta_k z_{k+1}. \]

Compared with the basic SCGD, the new estimate \( y_k \) is a weighted average of \( \{g_{w_t}(z_t)\}_{t=0}^k \), which are samples of function \( g(\cdot) \) at the \textit{extrapolated points}. Our purpose is to find an “unbiased” approximation of \( g(x_k) \) in the following sense:

\[ y_k = \text{weighted average of } \{g_{w_t}(z_t)\}_{t=0}^k \approx g(x_k) = g(\text{weighted average of } \{z_t\}_{t=0}^k). \]

In Algorithm 2, the new estimate \( y_k \) “tracks” the unknown quantity \( g(x_k) \) with a smaller bias and at a faster rate. As a result, the accelerated algorithm achieves an improved error complexity of \( O(k^{-2/7}) \) for general convex problems and \( O(k^{-4/5}) \) for strongly convex problems.
The acceleration of our algorithm comes from the extrapolation step that calculates $z_k$. Note that this idea is fundamentally different from the accelerated SA [13, 14]. The accelerated SA uses the idea of Nesterov smoothing [25] from deterministic convex optimization to improve the error dependency of SA on model parameters. In contrast, our accelerated SCGD solves the two-level stochastic problem (1), and it uses an extrapolation technique to balance noises associated with random samples, which reduces bias in the estimates and improves the error dependency on the number of samples $k$.

### 3.1 Almost sure convergence

In addition to Assumption 1, we make the following assumption.

**Assumption 2** The function $g$ has Lipschitz continuous gradients, i.e., there exists a scalar $L_g > 0$ such that

$$
\| \nabla g(x) - \nabla g(y) \| \leq L_g \| x - y \|, \quad \forall x, y \in \mathcal{X}.
$$

The sample gradients of $g$ and $f$ satisfy with probability 1 that

$$
E[\| \nabla g_{w_0}(x) \|^4 | v_0] \leq C_g^2, \quad E[\| \nabla f_{v_0}(y) \|^4] \leq C_f^2, \quad \forall x \in \mathcal{X}, \ y \in \mathbb{R}^m.
$$

Now we state the almost sure convergence result of the accelerated SCGD. We will establish it through a series of lemmas.

**Theorem 5** (Almost sure convergence of the accelerated SCGD) Suppose Assumptions 1 and 2 hold. Let the stepsizes $\{\alpha_k\}$ and $\{\beta_k\}$ satisfy

$$
\sum_{k=0}^{\infty} \alpha_k = \infty, \quad \sum_{k=0}^{\infty} \beta_k = \infty, \quad \sum_{k=0}^{\infty} \left( \alpha_k^2 + \beta_k^2 + \frac{\alpha_k^2}{\beta_k} + \frac{\alpha_k^4}{\beta_k^3} \right) < \infty.
$$

Let $\{(x_k, y_k)\}$ be generated by the accelerated SCGD Algorithm 2 starting from an arbitrary initial solution $(x_0, y_0)$. Then,

(a) If $F$ is convex, $\{x_k\}$ converges almost surely to a random point in the set of optimal solutions.

(b) If $F$ has Lipschitz continuous gradient, $\mathcal{X} = \mathbb{R}^n$, and all random variables generated by the SO are uniformly bounded, any limit point of $\{x_k\}$ is a stationary point with probability 1.

**Proof** See Supplementary Materials Section D for the detailed proof.

### 3.2 Accelerated rate of convergence

Next we analyze the rate of convergence for the accelerated SCGD Algorithm 2. We consider two cases: convex problems and strongly convex problems. Similar to Sect. 2, we also consider the averaged iterates given by

$$
\mathbb{E}\left[ \| \nabla \frac{1}{n} \sum_{i=1}^{n} g_{w_0}(x_i) \| \right] \leq C_g^2, \quad \mathbb{E}[\| \nabla \frac{1}{n} \sum_{i=1}^{n} f_{v_0}(y_i) \|] \leq C_f^2, \quad \forall x \in \mathcal{X}, \ y \in \mathbb{R}^m.
$$

Let $\{(x_k, y_k)\}$ be generated by the accelerated SCGD Algorithm 2 starting from an arbitrary initial solution $(x_0, y_0)$. Then,

(a) If $F$ is convex, $\{x_k\}$ converges almost surely to a random point in the set of optimal solutions.

(b) If $F$ has Lipschitz continuous gradient, $\mathcal{X} = \mathbb{R}^n$, and all random variables generated by the SO are uniformly bounded, any limit point of $\{x_k\}$ is a stationary point with probability 1.

**Proof** See Supplementary Materials Section D for the detailed proof.
\[ \hat{x}_k = \frac{1}{N_k} \sum_{t=k-N_k}^{k} x_t, \]

where \( N_k = \lceil k/2 \rceil \). The key component of our analysis is the recursive improvement in \( \| y_k - g(x_k) \| \), which has been established in Lemma D.3. As Lemma D.3 gives a sharper bound compared with Lemma 2, the accelerated SCGD generates more accurate estimates of \( g(x_k) \), resulting in faster rate of convergence.

**Theorem 6** (Accelerated convergence rate for general convex problems) Suppose that Assumptions 1 and 2 hold and \( F \) is a convex function. Let \( D > 0 \) be a constant such that \( \sup_{x \in \mathcal{X}} \| x - x^\ast \|^2 \leq D \), and let the stepsizes be

\[ \alpha_k = k^{-a}, \quad \beta_k = k^{-b}, \]

where \( a, b \) are scalars in \( (0, 1) \). Then:

(a) The averaged iterates generated by Algorithm 2 is such that

\[
E \left[ F(\hat{x}_k) - F^\ast \right] = \mathcal{O} \left( D k^{a-1} + C_f C_g k^{-a} + C_1 k^{-b/2} (\log k)^{1_{b=1/2}} + C_2 k^{-2a + 2b} (\log k)^{1_{a=3b=1}} \right),
\]

where \( C_1 = \sqrt{DC_g V_g L_f} \), and \( C_2 = \sqrt{DC_g L_f L_g C_g C_f} \).

(b) For \( k \) sufficiently large, the preceding bound is minimized at \( a = \frac{5}{7}, b = \frac{4}{7} \), yielding

\[
E \left[ F(\hat{x}_k) - F^\ast \right] = \mathcal{O} \left( k^{-2/7} \right).
\]

**Proof** See Supplementary Materials Section E for the detailed proof. \( \square \)

Next we consider the case when the problem is strongly convex, i.e., there exists \( \sigma > 0 \) and a unique optimal solution \( x^\ast \) such that \( F(x) - F^\ast \geq \sigma \| x - x^\ast \|^2 \) for all \( x \in \mathcal{X} \). The proof is given at Supplementary Materials Section F.

**Theorem 7** (Accelerated convergence rate for strongly convex problems) Let Assumptions 1 and 2 hold, and let \( F \) be strongly convex with parameter \( \sigma \). Let the stepsizes be

\[ \alpha_k = \frac{1}{\sigma k}, \quad \beta_k = \frac{1}{k^{4/5}}. \]

Then the iterates generated by Algorithm 2 is such that

\[
E \left[ \| x_k - x^\ast \|^2 \right] = \mathcal{O} \left( \frac{L_g^2 C_f^2 C_g^2 / \sigma^3}{\sigma^2 k^{4/5}} + \frac{C_f C_g \log k}{\sigma^2 k} \right) = \mathcal{O} \left( k^{-4/5} \right).
\]
Lastly, we consider the case where the objective function is not necessarily convex. We obtain the following convergence rate in terms of the nonstationarity metric $E\left[\|\nabla F(x_k)\|^2\right]$.

**Theorem 8** (Convergence rate of accelerated SCGD for nonconvex problems) Suppose that Assumptions 1 and 2 hold, $F$ has Lipschitz continuous gradient, and $X = \mathbb{R}^n$. Let the stepsizes be

$$\alpha_k = k^{-5/7}, \quad \beta_k = k^{-4/7}.$$ 

Let $T_\epsilon = \min \{ k : \inf_{0 \leq t \leq k} E[\|\nabla F(x_t)\|^2] \leq \epsilon \}$, then $T_\epsilon = O\left(\epsilon^{-7/2}\right)$.

The proof is analogous to Theorem 4, which we omit to avoid repetition.

**Remarks on sample error complexity**

An open issue remaining is whether the sample error complexity obtained by SCGD algorithms is improvable for problem (1)–(2). The best convergence rates we have obtained are $O \left( k^{-2/7} \right)$ and $O \left( k^{-4/5} \right)$ for convex and strongly convex problems, respectively. They are still inferior to the best known rates for the classical problem (3), i.e., $O \left( k^{-1/2} \right)$ and $O \left( 1/k \right)$, which are known to be optimal in terms of $k$ (see [1]). However, the stochastic problem (1)–(2) considered here is more generic than problem (3), and in fact, contains the latter as a special case. We conjecture that, the stochastic problem (1)–(2) where one minimizes compositions of two expected-value functions is intrinsically harder than the expectation minimization problem (3). For future work, it will be very interesting to analyze the sample complexity lower bounds for the stochastic composition problem (1), either to show that the current rates are non-improvable, or to motivate faster algorithms.

### 4 Applications and extensions

Optimization involving compositions of expected-value functions is a generic model of real-world decision making problems. In this section, we collect several instances of problem (1)–(2), and demonstrate the potential applications of the SCGD methods. We provide three examples in this section, and provide more applications in Supplementary Materials Section H. We believe that the stochastic optimization model considered in this paper will motivate new applications as well as new approaches to tackle important practical problems.

#### 4.1 Statistical learning

We first demonstrate the application of SCGD to an important statistical learning problem on estimating sparse additive models (SpAM) [28]. SpAM is a class of models for high-dimensional sparse nonparametric regression. More specifically, suppose we are given a number of $d$-dimensional input vectors $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \in \mathbb{R}^d$. 

\[ \text{$\copyright$ Springer} \]
and responses \( y_i \), where \( i = 1, \ldots, n \). SpAM assumes that each paired sample \((x_i, y_i)\) satisfies \( y_i = \alpha + \sum_{j=1}^{d} h_j(x_{ij}) + \epsilon_i \), where each \( h_j : \mathbb{R} \to \mathbb{R} \) is a feature component function, and each \( \epsilon_i \) is some zero-mean noise. Without loss of generality, we always assume \( \alpha = 0 \) (This can be achieved by centering the data).

To make the model interpretable, SpAM assumes that most of the feature functions \( h_j(\cdot) \) are zero functions. To induce sparsity, penalized regression methods have been well studied in the past decade (see [2,32,35] for examples), in which the component function \( h_j \)'s are assumed to be linear most of the time. When \( h_j \)'s are linear, the model simplifies to \( y_i = x_i^T \beta + \epsilon_i \) for some \( \beta \in \mathbb{R}^d \). In this case, \( \ell_1 \)-penalized regression [32] is popularly known as the LASSO estimator, written as

\[
\hat{\beta} = \arg \min_{\beta \in \mathbb{R}^d} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \| \beta \|_1.
\]

The LASSO estimator has strong statistical guarantee and can be computed efficiently.

In nonparametric models, the feature functions \( h_j \) are usually nonlinear and the \( \ell_1 \) penalty no longer induces sparsity. SpAM estimates the feature functions \( h_j(\cdot) \) by solving the following stochastic minimization problem:

\[
\min_{h_j \in \mathcal{H}_j, j=1,\ldots,d} \mathbb{E} \left[ Y - \sum_{j=1}^{d} h_j(X_j) \right]^2 + \lambda \sum_{j=1}^{d} \sqrt{\mathbb{E}[h_j^2(X_j)]}, \tag{8}
\]

where \( \mathcal{H}_j \) is some pre-specified functional space which ensures that the model is identifiable, and the expectation is taken over the sample pair \((X, Y)\). The SpAM estimator is known to have strong statistical guarantee [15,28]. However, there has been no known stochastic algorithm to solve the SpAM problem efficiently. In particular, even though the problem in (8) is convex, a provably convergent stochastic algorithm for solving (8) is lacking. Since the SpAM estimator (8) takes the form of (1), we can apply the SCGD methods to solve the estimation problem (8). The stochastic nature of SCGD allows it to sequentially update the solution when a new pair of data points \((X, Y)\) arrives. In many statistical applications, the feasible sets \( \mathcal{H}_j \) are usually compact and the feature functions \( h_j \)'s are continuously differentiable. Moreover, it is common to assume that the random variables \( X_j \)'s and \( Y \) have light-tail distributions (such as sub-Gaussian distributions). As a result, it is straightforward to check that the sample gradients have bounded second or fourth moments. Therefore Assumptions 1 and 2 are satisfied in a majority of these problems. We will provide a numerical example for solving SpAM using SCGD in Sect. 5.1.

4.2 Minimax problems

The SCGD methods are suitable for online solution of the stochastic minimax problems

\[
\min_{x \in \mathcal{X}} \max_{\omega \in \mathcal{O}} \left\{ \mathbb{E}[g^{(1)}(x)], \ldots, \mathbb{E}[g^{(I)}(x)] \right\},
\]

\( \mathcal{O} \) Springer
where $I$ is a positive integer; $E[g_w^{(i)}(x)]$ are convex functions. More generally, we may formulate the minimax problem to involve two stages of stochasticity, i.e.,

$$\min_{x \in X} \mathbb{E}_v \left[ \max \{ \mathbb{E}[g_w^{(1)}(x) \mid v], \ldots, \mathbb{E}[g_w^{(I)}(x) \mid v] \} \right].$$

These problems can be viewed as stochastic games with random payoffs against an adversarial opponent. They can be seen to take the form of (1)–(2), if we consider the max operator as the outer function $f$. Note that the max operator is nondifferentiable. Thus we may use a smooth approximation to replace the max operator, then apply the SCGD methods to solve the approximate minimax problem.

### 4.3 Dynamic programming

Dynamic programming is a rich area that involves sequential decision making, Monte Carlo estimation, and stochastic optimization. We consider the Markov decision problem (MDP) with states $i = 1, \ldots, n$. Finding an optimal policy for the MDP can be equivalently casted into solving the fixed-point Bellman equation, i.e., finding $J \in \mathbb{R}^n$ such that

$$J = \max_{a \in A} \{ g_a + \gamma P_a J \},$$

where $J \in \mathbb{R}^n$ is the cost-to-go vector, $a \in A$ is an action, $g_a \in \mathbb{R}^n$ is the transition cost vector given action $a$, $P_a \in \mathbb{R}^{n \times n}$ is the matrix of transition probabilities given action $a$, $\gamma \in (0, 1]$ is a discount factor, and the maximization is taken elementwise. The optimal policy is the state to action mapping that achieves the maximization.

Meanwhile, the Bellman residual minimization approach is to solve the problem

$$\min_{J \in \mathbb{R}^n} \| J - \max_{a} \{ g_a + \gamma P_a J \} \|^2,$$

whose set of optimal solutions coincides with that of the Bellman equation. In approximate dynamic programming, we may solve the high-dimensional Bellman equation approximately by restricting $J$ to belong to some parametric family, which translates to a constraint in the residual minimization problem.

We consider the simulation setting where $P_a$ and $g_a$ are not explicitly given. Instead, we are given a simulator of the MDP in which, at the current state $x_k$, we can choose an arbitrary action $a_k$ to obtain a random transition cost $g_k$ and to arrive at a future state $i_{k+1}$. This generates a sample trajectory of state, action, and cost triplets according to the unknown transition probabilities:

$$\{(i_0, a_0, g_0), \ldots, (i_k, a_k, g_k), (i_{k+1}, a_{k+1}, g_{k+1}), \ldots\}.$$

We rewrite the Bellman residual minimization problem as

$$\min_{J \in \mathbb{R}^n} \sum_{i=1}^{n} \left( J(i) - \max_{a \in A} \{ \mathbb{E}[g_k + \gamma J(i_{k+1}) \mid i_{k} = i, a_k = a] \} \right)^2,$$
which takes the form of problem (1)–(2). Therefore we are able to apply SCGD to the Bellman minimization problem. The SCGD can update online based on the simulation trajectories \( \{(i_k, a_k, g_k)\} \), without knowing the underlying transition probabilities. This suggests a new way to solve dynamic programming online.

5 Numerical results

In this section, we conduct numerical experiment of the SCGD methods for two problems.

5.1 Sparse additive model

We apply the SCGD algorithms to solve the SpAM optimization problem in (8). More specifically, let \( \phi_1, \phi_2, \ldots, \) be a set of countable basis functions of the functional space \( \mathcal{H}_j \)'s, i.e., any function \( h \in \mathcal{H}_j \) can be represented by \( h(x) = \sum_{i=1}^{\infty} a_i \phi_i(x) \), where \( a_i \in \mathbb{R} \). In applications, we generally choose a number \( p \) and approximate the function class \( \mathcal{H}_j \) by \( \tilde{\mathcal{H}}^p_j := \{ h : \text{there exist } a_1, \ldots, a_p \in \mathbb{R}, \text{ such that } h(x) = \sum_{i=1}^{p} a_i \phi_i(x) \} \).

Using the above approximation, it is readily seen that problem (8) can be written into the form of problem (1). In particular, let

\[
g(\eta) = E \left[ \left( Y - \sum_{j=1}^{d} \sum_{k=1}^{p} \eta_{jk} \phi_k(X_k) \right)^2, \left( \sum_{k=1}^{p} \eta_{1k} \phi_1(X_1) \right)^2, \ldots, \left( \sum_{k=1}^{p} \eta_{dk} \phi_d(X_d) \right)^2 \right]^T,
\]

where \( \eta \in \mathbb{R}^{d \times p} \), and

\[
f(z) = z_1 + \sum_{j=1}^{d} \sqrt{z_{j+1}}.
\]

We see that the estimation problem (8) becomes \( \min_{\eta} f(g(\eta)) \). Estimating the feature functions \( h_1, \ldots, h_d \) reduces to finding the weights \( \eta \). The SCGD method is a natural fit for this stochastic optimization problem.

We simulate data from the model \( y_i = \sum_{j=1}^{d} h_j(x_{ij}) + \epsilon_i \), where we choose the underlying feature functions as \( h_1(t) = t^3, h_2(t) = t, h_3(t) = \cdots = h_d(t) = 0 \) and \( \epsilon_i \sim \mathcal{N}(0, 0.1) \). Let \( d = 30 \) and \( n = 200 \). We use the span of cubic B-spline with five evenly distributed knots as the functional space \( \tilde{\mathcal{H}}_j = \text{span}\{\phi_1, \ldots, \phi_p\} \) (this choice of \( \tilde{\mathcal{H}}_j \) is consistent with [15]), and we choose the tuning parameter \( \lambda \) by threefold cross-validation. Let all \( x_i \)'s be independently drawn from the uniform distribution \( U[0, 1]^d \). We estimate the feature functions \( h_j \)'s by adopting basic SCGD Algorithm 1, where at the \( k \)-th iteration, we choose the stepsizes \( \alpha_k = k^{-3/4} \) and \( \beta_k = k^{-1/2} \).
We run the algorithm for 20,000 iterations, and we plot the resulting solutions for $h_1$, $h_2$, $h_3$ and $h_4$ in Fig. 1. We also investigate the empirical rate of convergence of the basic SCGD and the accelerated SCGD. Since the underlying true $\eta^*$ is unknown, we first let Algorithm 1 run $K = 1,000,000$ iterations and take the resulting solution as optimal $\eta^*$. Then, we run basic SCGD in Algorithm 1 with stepsizes $\alpha_k = k^{-3/4}$ and $\beta_k = k^{-1/2}$ and accelerated SCGD in Algorithm 2 with stepsizes $\alpha_k = k^{-5/7}$ and $\beta_k = k^{-4/7}$. At the $k$-th iteration, we compute the distance between the intermediate solution $\eta^k$ and $\eta^*$ by computing $\|\eta^k - \eta^*\|^2$, where $\eta^k$ is generated either by basic or accelerated SCGD. Then, we plot the $\log(k)$ versus $\log(\|\eta^k - \eta^*\|)$ for $k = 10,000–40,000$ in Fig. 2. Similarly, we also plot $\log(t)$ versus $\log(\|\eta^t - \eta^*\|)$ where $t$ is the running time in seconds. The figure supports our theoretical analysis: a faster convergence of the accelerated SCGD is clearly seen.

5.2 Stochastic shortest path problem

In this section, we consider a stochastic shortest path problem, as illustrated in Fig. 3. Our objective is to minimize the expected length of path from node A to node B. At
Fig. 2 The empirical rate of convergences of basic and accelerated SCGD. Left $\eta^k$ is the solution at the $k$-th iteration, and $\eta^*$ is the optimal solution. Right $\eta^t$ is the solution at time $t$ in seconds. The accelerated SCGD converges at a faster rate compared with the basic SCGD.

Fig. 3 The stochastic shortest path problem. If we choose action $a_1$, we will randomly move to one of the adjacent nodes as in figure (i) with equal probability. If we choose action $a_2$, we will randomly move to one of the adjacent nodes as in figure (ii) with equal probability.

every node $i = 1, \ldots, n$, we have two possible actions: choose action $a_1$ and randomly move to an arrow-pointed adjacent node $j$ in Fig. 3(i) with equal probability at a cost $c(i, j)$; choose action $a_2$ and randomly move to an arrow-pointed adjacent node $j$ in Fig. 3(ii) with equal probability at a cost $\hat{c}(i, j)$. The Bellman equation for this problem is

$$J(i) = \min \left\{ \mathbb{E}[c(i, j) + J(j) \mid a_1], \mathbb{E}[\hat{c}(i, j) + J(j) \mid a_2] \right\}, \quad J(B) = 0,$$

for $i, j = 1, \ldots, n$, where the expectation is taken over possible future state $j$ conditioned on current state $i$ and the action $a_1$ or $a_2$. We denote by $J^* \in \mathbb{R}^n$ the optimal solution to the Bellman equation, in which $J^*(i)$ is the optimal expected path length from node $i$ to node $B$. Solving the Bellman equation finds the optimal decision rule automatically.
Consider the Bellman residual minimization problem

$$\min_{J \in \mathbb{R}^n} \sum_{i=1}^{n} (J(i) - \min \{ q_{i,1}(J), q_{i,2}(J) \})^2,$$

where we define $q_{i,1}, q_{i,2}: \mathbb{R}^n \mapsto \mathbb{R}$ to be functions given by

$$q_{i,1}(J) = E[c(i, j) + J(j) | a_1], \quad q_{i,2}(J) = E[\hat{c}(i, j) + J(j) | a_2].$$

We note that this problem is nonconvex. However, it is not hard to argue that the optimal $J^*$ is the only stationary solution. Therefore by applying the SCGD algorithm, we are guaranteed to find the optimal solution $J^*$. Note the “min” operation is nondifferentiable. In order to apply SCGD, we approximate $\max\{y_1, y_2\}$ by a differentiable function $h_\epsilon(y_1, y_2)$, as defined by

$$h_\epsilon(y_1, y_2) = \begin{cases} 
  y_2 & \text{if } y_1 > y_2 + \epsilon, \\
  y_1 & \text{if } y_2 > y_1 + \epsilon, \\
  \frac{(y_1 - y_2)^2}{4\epsilon} + \frac{y_1 + y_2}{2} + \frac{\epsilon}{4} & \text{otherwise}.
\end{cases}$$

Let us consider the smoothed Bellman residual minimization, i.e.,

$$\min_{J \in \mathbb{R}^n} \sum_{i=1}^{n} \left( J(i) - h_\epsilon(q_{i,1}(J), q_{i,2}(J)) \right)^2,$$

which takes the form of (1) if we take

$$f([J, Q_{1,1}, Q_{1,2}, \ldots, Q_{n,1}, Q_{n,2}]) = \sum_{i=1}^{n} (J(i) - h_\epsilon(Q_{i,1}, Q_{i,2}))^2,$$

$$g(J) = [J, q_{1,1}(J), q_{1,2}(J), \ldots, q_{n,1}(J), q_{n,2}(J)],$$

where $g(J)$ is a mapping that involves expected values. Here $f$ is a known deterministic function (without random variable $v$) and its gradient can be easily calculated. Given a simulator of the random walk, we can calculate sample gradients/values of $g$ by using simulated state transitions.

In the numerical experiment, we generate the weighted graph arbitrarily and build a simulator of the random walk. We apply the basic and accelerated SCGD algorithms using stepsizes suggested by preceding theorems. The SCGD algorithms update the estimate of $J^*$ every time after observing a new state transition. As a byproduct, the SCGD provides estimates of the $Q$ values as the algorithm proceeds. Given the optimal cost vector $J^*$ and associated $Q$ values, we easily obtain the optimal strategy by comparing whether $Q_{i,1} > Q_{i,2}$ for every node $i$. This yields an online decision strategy for the stochastic shortest path problem.

The decision rule found by SCGD is demonstrated in Fig. 4, which coincides with the optimal solution computed by the value iteration (see the textbook [4]).
Fig. 4 The optimal decision rule found by SCGD for the stochastic shortest path problem. If a node is colored white, the optimal action is $a_1$; if a node is colored grey, the optimal action is $a_2$. The transition probability matrix corresponding to the optimal decision rule is demonstrated by the arcs. The possible movements of action $a_1$ are blue solid arcs, and the possible movements of action $a_2$ are purple dashed arcs (color figure online).

Fig. 5 The empirical convergence rate of the estimate $J^k$ to the optimal $J^*$. Left $J^k$ is computed by basic or accelerated SCGD at the $k$-th iteration. Right $J^t$ is computed by basic or accelerated SCGD at the time $t$ in seconds. The accelerated SCGD converges at a faster rate compared with the basic SCGD.

6 Conclusions

In this paper, we have considered the minimization of the composition of two expected-value functions. This problem involves two levels of stochasticity, so that the classical stochastic gradient no longer applies. We have proposed a class of stochastic compositional gradient methods that update based on random gradient evaluations of the inner and outer functions. We present a comprehensive convergence and rate of convergence analysis for these algorithms. A summary of the convergence rate and sample error complexity results is given in Table 1. For future research, one theoretical question is whether the current sample complexity is improvable or not. Another interesting direction is to extend the current work to optimization problems that involve compositions of more than two stochastic functions.
References

1. Agarwal, A., Bartlett, P., Ravikumar, P., Wainwright, M.: Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. IEEE Trans. Inf. Theory 58(5), 3235–3249 (2012)

2. Amini, A.A., Wainwright, M.J.: High-dimensional analysis of semidefinite relaxations for sparse principal components. Ann. Stat. 37(5B), 2877–2921 (2009)

3. Benveniste, A., Mézard, M., Priouret, P.: Adaptive Algorithms and Stochastic Approximations. Springer, Berlin (2012)

4. Bertsekas, D.P.: Dynamic Programming and Optimal Control, Volume I–II, 4th edn. Athena Scientific, Belmont (2007)

5. Bertsekas, D.P.: Incremental proximal methods for large scale convex optimization. Math. Program. Ser. B 129, 163–195 (2011)

6. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and Distributed Computation: Numerical Methods. Athena Scientific, Belmont (1989)

7. Bhatnagar, S., Borkar, V.S.: A two timescale stochastic approximation scheme for simulation-based parametric optimization. Prob. Eng. Inf. Sci. 12, 519–531 (1998)

8. Borkar, V.S.: Stochastic approximation with two time scales. Syst. Control Lett. 29, 291–294 (1997)

9. Borkar, V.S.: Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge University Press, Cambridge (2008)

10. Dentcheva, D., Penev, S., Ruszczynski, A.: Statistical estimation of composite risk functionals and risk optimization problems. arXiv preprint. arXiv:1504.02658 (2015)

11. Ermoliev, Y., Wets, R.: Numerical Techniques for Stochastic Optimization. Springer, New York (2011)

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

13. Ghadimi, S., Lan, G.: Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: a generic algorithmic framework. SIAM J. Optim. 22(4), 1469–1492 (2012)

14. Ghadimi, S., Lan, G.: Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. SIAM J. Optim. 23(4), 2061–2089 (2013)

15. Huang, J., Horowitz, J.L., Wei, F.: Variable selection in nonparametric additive models. Ann. Stat. 38(4), 2282 (2010)

16. Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. Ann. Math. Stat. 23(3), 462–466 (1952)

17. Konda, R., Tsitsiklis, J.N.: Convergence rate of linear two-time-scale stochastic approximation. Ann. Appl. Prob. 14, 796–819 (2004)

18. Korostelev, A.P.: Stochastic Recurrent Procedures. Nauka, Moscow (1984)

19. Kushner, H.J., Yin, G.: Stochastic Approximation and Recursive Algorithms and Applications. Springer, New York (2003)

20. Nedić, A.: Random algorithms for convex minimization problems. Math. Prog. Ser. B 129, 225–253 (2011)

21. Nedić, A., Bertsekas, D.P: Incremental subgradient methods for nondifferentiable optimization. SIAM J. Optim. 12, 109–138 (2001)

22. Nedić, A., Bertsekas, D.P., Borkar, V.S.: Distributed asynchronous incremental subgradient methods. Stud. Comput. Math. 8, 381–407 (2001)

23. Nemirovski, A., Rubinstein, R.Y.: An efficient stochastic approximation algorithm for stochastic saddle point problems. In: Dror, M., L’Ecuyer, P., Szidarovszky, F. (eds.) Modeling Uncertainty, pp. 156–184. Springer (2005)

24. Nemirovsky, A., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. Wiley, New York (1983)

25. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Sov. Math. Dokl. 27(2), 372–376 (1983)

26. Polyak, B.T., Juditsky, A.: Acceleration of stochastic approximation by averaging. SIAM J. Control Optim. 30(4), 838–855 (1992)

27. Rakhlin, A., Shamir, O., Sridharan, K.: Making gradient descent optimal for strongly convex stochastic optimization. In: Proceedings of the 29th International Conference on Machine Learning, pp. 449–456 (2012)
28. Ravikumar, P., Lafferty, J., Liu, H., Wasserman, L.: Sparse additive models. J. R. Stat. Soc. Ser. B 71(5), 1009–1030 (2009)
29. Robbins, H., Siegmund, D.: A convergence theorem for nonnegative almost supermartingales and some applications. J.S. Rustagi, Optimizing Methods in Statistics, pp. 233–257. Academic Press, NY (1985)
30. Shamir, O., Zhang, T.: Stochastic gradient descent for non-smooth optimization: convergence results and optimal averaging schemes. In: Proceedings of the 30th International Conference on Machine Learning, pp. 71–79 (2013)
31. Shapiro, A., Dentcheva, D., Ruszczynski, A.: Lectures on stochastic programming: modeling and theory. MOS-SIAM series on optimization. SIAM, Philadelphia (2009)
32. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B 58, 267–288 (1996)
33. Wang, M., Bertsekas, D.P.: Incremental constraint projection methods for variational inequalities. Math. Program. Ser. A 150(2), 321–363 (2015)
34. Wang, M., Bertsekas, D.P.: Incremental constraint projection-proximal methods for nonsmooth convex optimization. SIAM J. Optim. (2014) (in press)
35. Yuan, M., Lin, Y.: Model selection and estimation in the Gaussian graphical model. Biometrika 94(1), 19–35 (2007)