Fixed Point Theory and Algorithms for Sciences and Engineering

RESEARCH Open Access

Stochastic approximation method using diagonal positive-definite matrices for convex optimization with fixed point constraints

Hideaki Iiduka[1*]

*Correspondence:
iiduka@cs.meiji.ac.jp
[1]Department of Computer Science, Meiji University, Kanagawa, Japan

Abstract

This paper proposes a stochastic approximation method for solving a convex stochastic optimization problem over the fixed point set of a quasinonexpansive mapping. The proposed method is based on the existing adaptive learning rate optimization algorithms that use certain diagonal positive-definite matrices for training deep neural networks. This paper includes convergence analyses and convergence rate analyses for the proposed method under specific assumptions. Results show that any accumulation point of the sequence generated by the method with diminishing step-sizes almost surely belongs to the solution set of a stochastic optimization problem in deep learning. Additionally, we apply the learning methods based on the existing and proposed methods to classifier ensemble problems and conduct a numerical performance comparison showing that the proposed learning methods achieve high accuracies faster than the existing learning method.

MSC: 65K05; 65K15; 90C15

Keywords: Adaptive learning rate optimization algorithms; Convex stochastic optimization; Fixed point; Quasinonexpansive mapping; Stochastic fixed point optimization algorithm; Stochastic subgradient

1 Introduction

Convex stochastic optimization problems in which the objective function is the expectation of convex functions are considered important due to their occurrence in practical applications, such as machine learning and deep learning.

The classical method for solving these problems is the stochastic approximation (SA) method [1, (5.4.1)], [2, Algorithm 8.1], [3], which is applicable when unbiased estimates of (sub)gradients of an objective function are available. Modified versions of the SA method, such as the mirror descent SA method [4, Sects. 3 and 4], [5, Sect. 2.3] and the accelerated SA method [6, Sect. 3.1], have been reported as useful methods for solving these problems. Meanwhile, some stochastic optimization algorithms have been proposed with the rapid development of deep learning. For example, AdaGrad [7, Figs. 1 and 2] is an algorithm based on the mirror descent SA method, and Adam [8, Algorithm 1], [2, Algorithm 8.7] and AMSGrad [9, Algorithm 2] are well known as powerful tools for solving

convex stochastic optimization problems in deep neural networks. These algorithms use the inverses of diagonal positive-definite matrices at each iteration to adapt the learning rates of all model parameters. Hence, these algorithms are called *adaptive learning rate optimization algorithms*.

The above-mentioned methods commonly assume that metric projection onto a given constraint set is computationally possible. However, although the metric projection onto a simple convex set, such an affine subspace, half-space, or hyperslab, can be easily computed, the projection onto a complicated set, such as the intersections of simple convex sets, the set of minimizers of a convex function, or the solution set of a monotone variational inequality, cannot be easily computed. Accordingly, it is difficult to apply the above-mentioned methods to stochastic optimization problems with complicated constraints.

In order to solve a stochastic optimization problem over a complicated constraint set, we define a computable quasinonexpansive mapping whose fixed point set coincides with the constraint set, which is possible for the above-mentioned complicated convex sets (see Sect. 3.1 and Example 4.1 for examples of computable quasinonexpansive mappings). Accordingly, the present paper deals with a convex stochastic optimization problem over the *fixed point set* of a computable *quasinonexpansive* mapping.

Since useful fixed point algorithms have already been reported [10, Chap. 5], [11, Chaps. 2–9], [12–16], we can find fixed points of quasinonexpansive mappings, which are feasible points of the convex stochastic optimization problem. By combining the SA method with an existing fixed point algorithm, we could obtain algorithms [17, Algorithms 1 and 2] for solving convex stochastic optimization problems that can be applied to classifier ensemble problems [18, 19] (Example 4.1(ii)), which arise in the field of machine learning. However, the existing algorithms converge slowly [17] due to being stochastic first-order methods. In this paper, we propose an algorithm (Algorithm 1) for solving a convex stochastic optimization problem (Problem 3.1) that performs better than the algorithms in [17, Algorithms 1 and 2]. The algorithm proposed herein is based on useful adaptive learning rate optimization algorithms, such as Adam and AMSGrad, that use certain diagonal positive-definite matrices.[1] The first contribution of the present study is an analysis of the convergence of the proposed algorithm (Theorem 5.1). This analysis finds that, if sufficiently small constant step-sizes are used, then the proposed algorithm approximates a solution to the problem (Theorem 5.2). Moreover, for sequences of diminishing step-sizes, the convergence rates of the proposed algorithm can be specified (Theorem 5.3 and Corollary 5.1).

We compare the proposed algorithm with the existing adaptive learning rate optimization algorithms for a constrained convex stochastic optimization problem in deep learning (Example 4.1(i)). Although the existing adaptive learning rate optimization algorithms achieve low regret, they cannot solve the problem. The second contribution of the present study is to show that, unlike the existing adaptive learning rate optimization algorithms, the proposed algorithm can solve the problem (Corollaries 5.2 and 5.3) (see Sect. 5.2 for details). The third contribution is that we show that the proposed algorithm can solve classifier ensemble problems and that the learning methods based on the proposed algorithm perform better numerically than the existing learning method based on the existing algorithms in [17]. In particular, the numerical results indicate that the learning methods

[1] See (6) and (9) for the definitions of Adam and AMSGrad.

based on the proposed algorithm with constant step-sizes or step-sizes computed by the Armijo line search algorithm can solve classifier ensemble problems faster than the existing learning method based on the algorithms in [17]. As a result, the proposed learning methods achieve high accuracies faster than the existing learning method.

2 Mathematical preliminaries

2.1 Definitions and propositions

Let $\mathbb{N}$ be the set of all positive integers. Let $\mathbb{R}^N$ be an N-dimensional Euclidean space with the inner product $\langle \cdot, \cdot \rangle$ with the associated norm $\| \cdot \|$, and let $\mathbb{R}^N_+ := \{(x_i)_{i=1}^N \in \mathbb{R}^N : x_i \geq 0\ (i = 1, 2, \ldots, N)\}$. Let $X^\top$ denote the transpose of matrix X, let I denote the identity matrix, and let Id denote the identity mapping on $\mathbb{R}^N$. Let $\mathbb{S}^N$ be the set of $N \times N$ symmetric matrices, i.e., $\mathbb{S}^N = \{X \in \mathbb{R}^{N \times N} : X = X^\top\}$. Let $\mathbb{S}^N_{++}$ denote the set of symmetric positive-definite matrices, i.e., $\mathbb{S}^N_{++} = \{X \in \mathbb{S}^N : X \succ O\}$. Given $H \in \mathbb{S}^N_{++}$, the H-inner product of $\mathbb{R}^N$ and the H-norm can be defined for all $x, y \in \mathbb{R}^N$ by $\langle x, y \rangle_H := \langle x, Hy \rangle$ and $\|x\|_H^2 := \langle x, Hx \rangle$. Let $\mathsf{diag}(x_i)$ be an $N \times N$ diagonal matrix with diagonal components $x_i \in \mathbb{R}$ $(i = 1, 2, \ldots, N)$, and let $\mathbb{D}^N$ be the set of $N \times N$ diagonal matrices, i.e., $\mathbb{D}^N = \{X \in \mathbb{R}^{N \times N} : X = \mathsf{diag}(x_i), x_i \in \mathbb{R}\ (i = 1, 2, \ldots, N)\}$.

Let $\mathbb{E}[X]$ denote the expectation of random variable X. The history of the process $\xi_0, \xi_1, \ldots$ up to time n is denoted by $\xi_{[n]} = (\xi_0, \xi_1, \ldots, \xi_n)$. Let $\mathbb{E}[X|\xi_{[n]}]$ denote the conditional expectation of X given by $\xi_{[n]} = (\xi_0, \xi_1, \ldots, \xi_n)$. Unless stated otherwise, all relations between random variables are supported to hold almost surely.

The *subdifferential* [10, Definition 16.1], [20, Sect. 23] of a convex function $f : \mathbb{R}^N \to \mathbb{R}$ is defined for all $x \in \mathbb{R}^N$ by

$$\partial f(x) := \left\{ u \in \mathbb{R}^N : f(y) \geq f(x) + \langle y - x, u \rangle \ \left(y \in \mathbb{R}^N \right) \right\}.$$

A point $u \in \partial f(x)$ is called the *subgradient* of f at $x \in \mathbb{R}^N$.

Proposition 2.1 ([21, Theorem 4.1.3], [10, Propositions 16.14(ii), (iii)]) *Let $f : \mathbb{R}^N \to \mathbb{R}$ be convex. Then f is continuous and $\partial f(x) \neq \emptyset$ for every $x \in \mathbb{R}^N$. Moreover, for every $x \in \mathbb{R}^N$, there exists $\delta > 0$ such that $\partial f(B(x; \delta))$ is bounded, where $B(x; \delta)$ is the closed ball with center x and radius δ.*

When a mapping $Q : \mathbb{R}^N \to \mathbb{R}^N$ is considered under the H-norm $\| \cdot \|_H$, we denote it as $Q_H : \mathbb{R}^N \to \mathbb{R}^N$. We define $Q := Q_I$. A mapping $Q : \mathbb{R}^N \to \mathbb{R}^N$ is said to be *quasinonexpansive* [10, Definition 4.1(iii)] if

$$\left\| Q(x) - y \right\| \leq \|x - y\|$$

for all $x \in \mathbb{R}^N$ and all $y \in \mathrm{Fix}(Q)$, where $\mathrm{Fix}(Q)$ is the *fixed point set* of Q defined by $\mathrm{Fix}(Q) := \{x \in \mathbb{R}^N : x = Q(x)\}$. When a quasinonexpansive mapping has one fixed point, its fixed point set is closed and convex [22, Proposition 2.6]. Q is called a *firmly quasinonexpansive* mapping [23, Sect. 3] if $\|Q(x) - y\|^2 + \|(\mathrm{Id} - Q)(x)\|^2 \leq \|x - y\|^2$ for all $x \in \mathbb{R}^N$ and all $y \in \mathrm{Fix}(Q)$. Q is firmly quasinonexpansive if and only if $R := 2Q - \mathrm{Id}$ is quasinonexpansive [10, Proposition 4.2]. This means that $(1/2)(\mathrm{Id} + R)$ is firmly quasinonexpansive when R

is quasinonexpansive. Given $H \in \mathbb{S}^N_{++}$, we define the *subgradient projection*[2] relative to a convex function $f\colon \mathbb{R}^N \to \mathbb{R}$ by

$$Q_{f,H}(x) := \begin{cases} x - \frac{f(x)}{\|H^{-1}G(x)\|_H^2} H^{-1}G(x) & \text{if } f(x) > 0, \\ x & \text{otherwise,} \end{cases} \tag{1}$$

where $G(x)$ is any point in $\partial f(x)$ ($x \in \mathbb{R}^N$) and $\mathrm{lev}_{\leq 0} f := \{x \in \mathbb{R}^N : f(x) \leq 0\} \neq \emptyset$. The following proposition holds.

Proposition 2.2 *Let $H \in \mathbb{S}^N_{++}$ and let $f\colon \mathbb{R}^N \to \mathbb{R}$ be convex. Then $Q_{f,H}\colon \mathbb{R}^N \to \mathbb{R}^N$ defined by* (1) *satisfies the following*:

(i) $Q_f := Q_{f,I}$ *is firmly quasinonexpansive and* $\mathrm{Fix}(Q_f) = \mathrm{lev}_{\leq 0} f$;

(ii) $Q_{f,H}$ *is firmly quasinonexpansive under the H-norm with* $\mathrm{Fix}(Q_{f,H}) = \mathrm{Fix}(Q_f)$.

Proof (i) This follows from Proposition 2.3 in [22].

(ii) We first show that $\mathrm{lev}_{\leq 0} f = \mathrm{Fix}(Q_{f,H})$. From (1), we have that $\mathrm{lev}_{\leq 0} f \subset \mathrm{Fix}(Q_{f,H})$. Let $x \in \mathrm{Fix}(Q_{f,H})$ and assume that $x \notin \mathrm{lev}_{\leq 0} f$. Then the definition of the H-inner product and $G(x) \in \partial f(x)$ mean that, for all $y \in \mathrm{lev}_{\leq 0} f$,

$$\left\langle y - x, H^{-1}G(x) \right\rangle_H = \left\langle y - x, G(x) \right\rangle \leq f(y) - f(x) \leq -f(x) < 0, \tag{2}$$

which implies that $H^{-1}G(x) \neq 0$. From (1) and $x \in \mathrm{Fix}(Q_{f,H})$, we also have that

$$\frac{f(x)}{\|H^{-1}G(x)\|_H^2} H^{-1}G(x) = x - Q_{f,H}(x) = 0,$$

which, together with $f(x) > 0$, gives $H^{-1}G(x) = 0$, which is a contradiction. Hence, we have that $\mathrm{lev}_{\leq 0} f \supset \mathrm{Fix}(Q_{f,H})$, i.e., $\mathrm{lev}_{\leq 0} f = \mathrm{Fix}(Q_{f,H})$. Accordingly, (i) ensures that $\mathrm{Fix}(Q_{f,H}) = \mathrm{lev}_{\leq 0} f = \mathrm{Fix}(Q_f)$. For all $x \in \mathbb{R}^N \backslash \mathrm{lev}_{\leq 0} f$ and all $y \in \mathrm{lev}_{\leq 0} f$,

$$\begin{aligned} &\left\| Q_{f,H}(x) - y \right\|_H^2 \\ &\quad = \|x - y\|_H^2 + \frac{2f(x)}{\|H^{-1}G(x)\|_H^2} \left\langle y - x, H^{-1}G(x) \right\rangle_H + \frac{f(x)^2}{\|H^{-1}G(x)\|_H^2}, \end{aligned}$$

which, together with (2), implies that $Q_{f,H}$ is firmly quasinonexpansive under the H-norm. □

$Q\colon \mathbb{R}^N \to \mathbb{R}^N$ is said to be Lipschitz continuous (L-Lipschitz continuous) if there exists $L > 0$ such that $\|Q(x) - Q(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^N$. $Q\colon \mathbb{R}^N \to \mathbb{R}^N$ is said to be *nonexpansive* [10, Definition 4.1(ii)] if Q is 1-Lipschitz continuous, i.e., $\|Q(x) - Q(y)\| \leq \|x - y\|$ for all $x, y \in \mathbb{R}^N$. Any nonexpansive mapping satisfies the quasinonexpansivity condition. The *metric projection* [10, Subchapter 4.2, Chap. 28] onto a nonempty, closed convex set C ($\subset \mathbb{R}^N$), denoted by P_C, is defined for all $x \in \mathbb{R}^N$ by $P_C(x) \in C$ and $\|x - P_C(x)\| = \mathrm{d}(x, C) := \inf_{y \in C} \|x - y\|$. P_C is firmly nonexpansive, i.e., $\|P_C(x) - P_C(y)\|^2 + \|(\mathrm{Id} - P_C)(x) -$

[2] See [23, Lemma 3.1], [22, Proposition 2.3], [24, Subchapter 4.3] for the definition and properties of the subgradient projection when $H = I$.

$(\mathrm{Id} - P_C)(y)\|^2 \leq \|x - y\|^2$ for all $x, y \in \mathbb{R}^N$, with $\mathrm{Fix}(P_C) = C$ [10, Proposition 4.8, (4.8)]. The metric projection onto C under the H-norm is denoted by $P_{C,H}$. When C is an affine subspace, half-space, or hyperslab, the projection onto C can be computed within a finite number of arithmetic operations [10, Chap. 28].

3 Convex stochastic optimization problem over fixed point set

This paper considers the following problem.

Problem 3.1 Assume that

(A0) $(\mathsf{H}_n)_{n\in\mathbb{N}}$ is the sequence in $\mathbb{S}^N_{++} \cap \mathbb{D}^N$;

(A1) $Q_{\mathsf{H}_n} : \mathbb{R}^N \to \mathbb{R}^N$ is quasinonexpansive under the H_n-norm and $X := \bigcap_{n\in\mathbb{N}} \mathrm{Fix}(Q_{\mathsf{H}_n})$ $(\subset C)$ is nonempty, where $C \subset \mathbb{R}^N$ is a nonempty, closed convex set onto which the projection can be easily computed;

(A2) $f : \mathbb{R}^N \to \mathbb{R}$ defined for all $x \in \mathbb{R}^N$ by $f(x) := \mathbb{E}[F(x, \xi)]$ is well defined and convex, where ξ is a random vector whose probability distribution P is supported on a set $\Xi \subset \mathbb{R}^M$ and $F : \mathbb{R}^N \times \Xi \to \mathbb{R}$.

Then

$$\text{find } x^\star \in X^\star := \left\{x^\star \in X : f\left(x^\star\right) = f^\star := \inf_{x\in X} f(x)\right\},$$

where one assumes that $X^\star$ is nonempty.

Examples of Q_{H_n} satisfying (A0) and (A1) are described in Sect. 3.1 and Example 4.1.

The following are sufficient conditions [5, (A1), (A2), (2.5)] for being able to solve Problem 3.1.

(C1) There is an independent and identically distributed sample $\xi_0, \xi_1, \ldots$ of realizations of the random vector ξ;

(C2) There is an oracle which, for a given input point $(x, \xi) \in \mathbb{R}^N \times \Xi$, returns a stochastic subgradient $\mathsf{G}(x, \xi)$ such that $\mathsf{g}(x) := \mathbb{E}[\mathsf{G}(x, \xi)]$ is well defined and is a subgradient of f at x, i.e., $\mathsf{g}(x) \in \partial f(x)$;

(C3) There exists a positive number M such that, for all $x \in C$, $\mathbb{E}[\|\mathsf{G}(x, \xi)\|^2] \leq M^2$.

Suppose that $F(\cdot, \xi)$ $(\xi \in \Xi)$ is convex and consider the oracle which returns a stochastic subgradient $\mathsf{G}(x, \xi) \in \partial_x F(x, \xi)$ for given $(x, \xi) \in \mathbb{R}^N \times \Xi$. Then $f(\cdot) = \mathbb{E}[F(\cdot, \xi)]$ is well defined and convex, and $\partial f(x) = \mathbb{E}[\partial_x F(x, \xi)]$ [25, Theorem 7.51], [5, p.1575].

3.1 Related problems and their algorithms

Here, let us consider the following convex stochastic optimization problem [5, (1.1)]:

$$\text{minimize } f(x) = \mathbb{E}\big[F(x, \xi)\big] \text{ subject to } x \in C, \tag{3}$$

where $C \subset \mathbb{R}^N$ is nonempty, bounded, closed, and convex. The classical method for problem (3) under (C1)–(C3) is the *stochastic approximation* (SA) *method* [1, (5.4.1)], [2, Algorithm 8.1], [3] defined as follows: given $x_0 \in \mathbb{R}^N$ and $(\lambda_n)_{n\in\mathbb{N}} \subset (0, +\infty)$,

$$x_{n+1} = P_C\big(x_n - \lambda_n \mathsf{G}(x_n, \xi_n)\big) \quad (n \in \mathbb{N}). \tag{4}$$

The SA method requires the metric projection onto C, and hence can be applied only to cases where C is simple in the sense that P_C can be efficiently computed (e.g., C is a closed ball, half-space, or hyperslab [10, Chap. 28]). When C is not simple, the SA method requires solving the following subproblem at each iteration n:

$$\text{Find } x_{n+1} \in C \text{ such that } \{x_{n+1}\} = \operatorname*{argmin}_{y \in C} \left\| \left(x_n - \lambda_n \mathsf{G}(x_n, \xi_n)\right) - y \right\|.$$

The *mirror descent SA method* [4, Sects. 3 and 4], [5, Sect. 2.3] is useful for solving problem (3) and has been analyzed for the case of step-sizes that are constant or diminishing. For example, the mirror descent SA method [5, (2.32), (2.38), and (2.47)] with a constant step-size policy generates the following sequence $(\tilde{x}_1^n)_{n \in \mathbb{N}}$: given $x_0 \in X^o := \{x \in \mathbb{R}^N : \partial \omega(x) \neq \emptyset\}$,

$$x_{n+1} = \operatorname*{argmin}_{z \in C} \left\{ \left\langle \gamma_n \mathsf{G}(x_n, \xi_n), z - x_n \right\rangle + V(x_n, z) \right\}, \qquad \tilde{x}_1^{n+1} := \sum_{t=1}^{n+1} \frac{\gamma_t}{\sum_{i=1}^{n+1} \gamma_i} x_t, \tag{5}$$

where $\omega \colon C \to \mathbb{R}$ is differentiable and convex, $V \colon X^o \times C \to \mathbb{R}_+$ is defined for all $(x, z) \in X^o \times C$ by $V(x, z) := \omega(z) - [\omega(x) + \langle \nabla \omega(x), z - x \rangle]$, and γ_t ($t \in \mathbb{N}$) is a constant step-size. When $\omega(\cdot) = (1/2)\| \cdot \|^2$, x_{n+1} in (5) coincides with x_{n+1} in (4). Under certain assumptions, method (5) satisfies $\mathbb{E}[f(\tilde{x}_1^n) - f^\star] = \mathcal{O}(1/\sqrt{n})$ [5, (2.48)] (see [5, (2.57)] for the rate of convergence of the mirror descent SA method with a diminishing step-size policy).

As the field of deep learning has developed, it has produced some useful stochastic optimization algorithms, such as AdaGrad [7, Figs. 1 and 2], [2, Algorithm 8.4], RMSProp [2, Algorithm 8.5], and Adam [8, Algorithm 1], [2, Algorithm 8.7], for solving problem (3). The AdaGrad algorithm is based on the mirror decent SA method (5) (see also [7, (4)]), and the RMSProp algorithm is a variant of AdaGrad. The Adam algorithm is based on a combination of RMSProp and the momentum method [26, (9)], as follows: given $x_t, m_{t-1}, v_{t-1} \in \mathbb{R}^N$,

$$\begin{aligned}
&m_t := \beta_1 m_{t-1} + (1 - \beta_1) \nabla_x F(x_t, \xi_t), \\
&v_t := \beta_2 v_{t-1} + (1 - \beta_2) \nabla_x F(x_t, \xi_t) \odot \nabla_x F(x_t, \xi_t), \\
&\hat{m}_t := \frac{m_t}{1 - \beta_1^{t+1}}, \qquad \hat{v}_t := \frac{v_t}{1 - \beta_2^{t+1}}, \\
&\mathsf{d}_t := -\mathsf{diag}\left(\frac{1}{\sqrt{\hat{v}_{t,i}} + \epsilon} \right) \hat{m}_t = -\left(\frac{\hat{m}_{t,i}}{\sqrt{\hat{v}_{t,i}} + \epsilon} \right)_{i=1}^N, \\
&x_{t+1} := P_C[x_t + \lambda_t \mathsf{d}_t], \quad \text{i.e., } \{x_{t+1}\} = \operatorname*{argmin}_{y \in C} \left\| (x_t + \lambda_t \mathsf{d}_t) - y \right\|,
\end{aligned} \tag{6}$$

where $\beta_i > 0$ ($i = 1, 2$), $\epsilon > 0$, $(\lambda_n)_{n \in \mathbb{N}} \subset (0, 1)$ is diminishing step-size, and $A \odot B$ denotes the Hadamard product of matrices A and B. If we define matrix H_t as

$$\mathsf{H}_t := \mathsf{diag}(\sqrt{\hat{v}_{t,i}} + \epsilon), \tag{7}$$

then the Adam algorithm (6) can be expressed as

$$x_{n+1} = P_C\left[x_t - \lambda_t \mathsf{diag}\left(\frac{1}{\sqrt{\hat{v}_{t,i}} + \epsilon} \right) \hat{m}_t \right] = P_C\left[x_t - \lambda_t \mathsf{H}_t^{-1} \hat{m}_t \right]. \tag{8}$$

Unfortunately, there exists an explicit example of a simple convex optimization setting where Adam does not converge to the optimal solution [9, Theorem 2]. To guarantee convergence and preserve the practical benefits of Adam, AMSGrad [9, Algorithm 2] was proposed as follows: for $(\beta_{1,t})_{t\in\mathbb{N}} \subset (0, +\infty)$,

$$\begin{aligned}
&m_t := \beta_{1,t} m_{t-1} + (1-\beta_{1,t})\nabla_x F(x_t, \xi_t),\\
&v_t := \beta_2 v_{t-1} + (1-\beta_2)\nabla_x F(x_t, \xi_t) \odot \nabla_x F(x_t, \xi_t),\\
&\hat{v}_t := (\hat{v}_{t,i}) = \left(\max\{\hat{v}_{t-1,i}, v_{t,i}\}\right),\\
&\mathsf{H}_t := \mathsf{diag}(\sqrt{\hat{v}_{t,i}}),\\
&\mathsf{d}_t := -\mathsf{H}_t^{-1} m_t,\\
&x_{t+1} := P_{C,\mathsf{H}_t}[x_t + \lambda_t \mathsf{d}_t], \quad \text{i.e., } \{x_{t+1}\} = \operatorname*{argmin}_{y\in C} \left\|(x_t + \lambda_t \mathsf{d}_t) - y\right\|_{\mathsf{H}_t}.
\end{aligned} \tag{9}$$

The existing SA methods (4), (5), (6), and (9) (see also [6, 27], [2, Sect. 8.5], and [5, Sect. 2.3]) require minimizing a certain convex function over C at each iteration. Therefore, when C has a complicated form (e.g., C is expressed as the set of all minimizers of a convex function over a closed convex set, the solution set of a monotone variational inequality, or the intersection of closed convex sets), it is difficult to compute the point x_{n+1} generated by any of (4), (5), (6), and (9) at each iteration.

Meanwhile, the *fixed point theory* [10, 28–30] enables us to define a *computable* quasinonexpansive mapping of which the fixed point set is equal to the complicated set. For example, let $\mathrm{lev}_{\leq 0} f_i$ $(i = 1, 2, \ldots, I)$ be the level set of a convex function $f_i \colon \mathbb{R}^N \to \mathbb{R}$, and let X be the intersection of $\mathrm{lev}_{\leq 0} f_i$, i.e.,

$$X := \bigcap_{i=1}^{I} \mathrm{lev}_{\leq 0} f_i = \bigcap_{i=1}^{I} \left\{x \in \mathbb{R}^N : f_i(x) \leq 0\right\} \neq \emptyset. \tag{10}$$

Let $n \in \mathbb{N}$ be fixed arbitrarily, and let $\mathsf{H}_n \in \mathbb{S}^N_{++}$ (see (A0)). Let $Q_{f_i,\mathsf{H}_n} \colon \mathbb{R}^N \to \mathbb{R}^N$ $(i = 1, 2, \ldots, I)$ be the subgradient projection defined by (1) with $f := f_i$ and $H := \mathsf{H}_n$. Accordingly, Proposition 2.2 implies that Q_{f_i,H_n} is firmly quasinonexpansive under the H_n-norm and $\mathrm{Fix}(Q_{f_i,\mathsf{H}_n}) = \mathrm{lev}_{\leq 0} f_i$ $(i = 1, 2, \ldots, I)$. Under the condition that the subgradients of f_i can be efficiently computed (see, e.g., [10, Chap. 16] for examples of convex functions with computable subgradients), Q_{f_i,H_n} also can be computed. Here, let us define $Q_{\mathsf{H}_n} \colon \mathbb{R}^N \to \mathbb{R}^N$ as

$$Q_{\mathsf{H}_n} := \sum_{i=1}^{I} \omega_i Q_{f_i,\mathsf{H}_n}, \tag{11}$$

where $(\omega_i)_{i=1}^I \subset (0, +\infty)$ satisfies $\sum_{i=1}^I \omega_i = 1$. Then Q_{H_n} is quasinonexpansive under the H_n-norm [10, Exercise 4.11]. Moreover, we have that

$$X = \bigcap_{i=1}^{I} \mathrm{lev}_{\leq 0} f_i = \bigcap_{i=1}^{I} \mathrm{Fix}(Q_{f_i}) = \bigcap_{i=1}^{I} \bigcap_{n\in\mathbb{N}} \mathrm{Fix}(Q_{f_i,\mathsf{H}_n}) = \bigcap_{n\in\mathbb{N}} \mathrm{Fix}(Q_{\mathsf{H}_n}), \tag{12}$$

where the second equality comes from Proposition 2.2(i) (i.e., $\mathrm{Fix}(Q_{f_i}) = \mathrm{lev}_{\leq 0} f_i$ $(i = 1, 2, \ldots, I)$), the third equality comes from Proposition 2.2(ii) (i.e., $\mathrm{Fix}(Q_{f_i}) = \mathrm{Fix}(Q_{f_i,\mathsf{H}_n})$ for all $n \in \mathbb{N}$), and the fourth equality comes from [10, Proposition 4.34]. Therefore, (10), (11), and (12) imply that we can define a computable mapping Q_{H_n} satisfying (A1) of which the fixed point set is equal to the intersection of level sets. In the case where C is simple in the sense that $P_C = P_{C,I}$ can be easily computed, $I \succ O$ and $Q := P_C$ obviously satisfy (A0) and (A1) with $\mathrm{Fix}(P_C) = C =: X$. Accordingly, Problem 3.1 with $Q := P_C$ coincides with problem (3), which implies that Problem 3.1 is a generalization of problem (3).

Fixed point algorithms exist for searching for a fixed point of a nonexpansive mapping [10, Chap. 5], [11, Chaps. 2–9], [12–16]. The sequence $(x_n)_{n\in\mathbb{N}}$ is generated by the Halpern fixed point algorithm [11, Subchapter 6.5], [12, 16] as follows: for all $n \in \mathbb{N}$,

$$x_{n+1} := \alpha_n x_0 + (1 - \alpha_n) Q(x_n), \tag{13}$$

where $x_0 \in \mathbb{R}^N$, $(\alpha_n)_{n\in\mathbb{N}} \subset (0, 1)$ satisfies $\lim_{n\to+\infty} \alpha_n = 0$ and $\sum_{n=0}^{+\infty} \alpha_n = +\infty$, and $Q\colon \mathbb{R}^N \to \mathbb{R}^N$ is nonexpansive with $\mathrm{Fix}(Q) \neq \emptyset$. The sequence $(x_n)_{n\in\mathbb{N}}$ in (13) converges to the minimizer of the specific convex function $f_0(x) := (1/2)\|x - x_0\|^2$ $(x \in \mathbb{R}^N)$ over $\mathrm{Fix}(Q)$ (see, e.g., [11, Theorem 6.19]). From $\nabla f_0(x) = x - x_0$ $(x \in \mathbb{R}^N)$, the Halpern algorithm (13) can be expressed as follows (see [31, 32] for algorithms optimizing a general convex function):

$$x_{n+1} = Q(x_n) - \alpha_n\big(Q(x_n) - x_0\big) = Q(x_n) - \alpha_n \nabla f_0\big(Q(x_n)\big). \tag{14}$$

The following algorithm obtained by combining the SA method (4) with (14) for solving Problem 3.1 follows naturally from the above discussion: for all $n \in \mathbb{N}$,

$$x_{n+1} := P_C\big[Q_\alpha(x_n) - \lambda_n \mathsf{G}\big(Q_\alpha(x_n), \xi_n\big)\big], \tag{15}$$

where $Q_\alpha := \alpha \mathrm{Id} + (1 - \alpha)Q$ $(\alpha \in (0, 1))$. A convergence analysis of this algorithm for different step-size rules was performed in [17]. For example, algorithm (15) with a diminishing step-size was shown to converge in probability to a solution to Problem 3.1 with $X = \mathrm{Fix}(Q)$ [17, Theorem III.2]. The advantage of algorithm (15) is that it allows convex stochastic optimization problems with complicated constraints to be solved (see also (12)). From the fact stated in [17, Problem II.1] that the classifier ensemble problem [18, 19], which is a central issue in machine learning, can be formulated as a convex stochastic optimization problem with complicated constraints, the classifier ensemble problem can be regarded as an example of Problem 3.1. This result implies that algorithm (15) can solve the classifier ensemble problem. However, this algorithm suffers from slow convergence, as shown in [17]. Specifically, although the learning methods based on algorithm (15) have higher accuracies than the previously proposed learning methods, they have longer elapsed times. Accordingly, we should consider developing stochastic optimization techniques to accelerate algorithm (15). This paper proposes an algorithm (Algorithm 1) based on useful stochastic gradient descent algorithms, such as Adam [8, Algorithm 1] and AMSGrad [9, Algorithm 2], for solving Problem 3.1, as a replacement for the existing stochastic first-order method [17].

Algorithm 1 Stochastic approximation method for solving Problem 3.1

Require: $(\alpha_n)_{n\in\mathbb{N}}, (\beta_n)_{n\in\mathbb{N}}, (\lambda_n)_{n\in\mathbb{N}} \subset (0,1)$, C $(\supset X)$: nonempty, closed, convex

$n \leftarrow 0$, $x_0, m_{-1} \in \mathbb{R}^N$, $\mathsf{H}_0 \in \mathbb{S}^N_{++} \cap \mathbb{D}^N$

loop

$m_n := \beta_n m_{n-1} + (1-\beta_n)\mathsf{G}(x_n, \xi_n)$

$\mathsf{H}_n \in \mathbb{S}^N_{++} \cap \mathbb{D}^N$

Find $\mathsf{d}_n \in \mathbb{R}^N$ that solves $\mathsf{H}_n \mathsf{d} = -m_n$

$y_n := Q_{\mathsf{H}_n}(x_n + \lambda_n \mathsf{d}_n)$

$x_{n+1} := P_{C,\mathsf{H}_n}[\alpha_n x_n + (1-\alpha_n) y_n]$

$n \leftarrow n+1$

end loop

4 Proposed algorithm

Before giving some examples, we first prove the following lemma listing the basic properties of Algorithm 1.

Lemma 4.1 *Suppose that* $\mathsf{H}_n \in \mathbb{S}^N_{++}$ $(n \in \mathbb{N})$, $(A1)$, $(A2)$, $(C1)$, *and* $(C2)$ *hold and consider the sequence* $(x_n)_{n\in\mathbb{N}}$ *defined for all* $n \in \mathbb{N}$ *by Algorithm 1. Then, for all* $x \in X$ *and all* $n \in \mathbb{N}$,

$$\begin{aligned}
&\mathbb{E}\big[\|x_{n+1} - x\|^2_{\mathsf{H}_n}\big] \\
&\quad \le \mathbb{E}\big[\|x_n - x\|^2_{\mathsf{H}_n}\big] + 2(1-\alpha_n)\lambda_n \big\{(1-\beta_n)\mathbb{E}\big[f(x) - f(x_n)\big] \\
&\qquad + \beta_n \mathbb{E}\big[\langle x - x_n, m_{n-1}\rangle\big]\big\} + (1-\alpha_n)\lambda_n^2 \mathbb{E}\big[\|\mathsf{d}_n\|^2_{\mathsf{H}_n}\big] \\
&\qquad - \alpha_n \mathbb{E}\big[\|x_{n+1} - x_n\|^2_{\mathsf{H}_n}\big] - (1-\alpha_n)\mathbb{E}\big[\|x_{n+1} - y_n\|^2_{\mathsf{H}_n}\big].
\end{aligned}$$

Moreover, under $(C3)$, $\mathbb{E}[\|m_n\|^2] \le \tilde{M}^2 := \max\{\|m_{-1}\|^2, M^2\}$ *holds for all* $n \in \mathbb{N}$. *If*

(A3) $h_\star := \sup\{\max_{i=1,2,\ldots,N} h_{n,i}^{-1/2} : n \in \mathbb{N}\}$ *is finite, where* $\mathsf{H}_n := \mathsf{diag}(h_{n,i})$,

then $\mathbb{E}[\|\mathsf{d}_n\|^2_{\mathsf{H}_n}] \le h_\star^2 \tilde{M}^2$ *holds for all* $n \in \mathbb{N}$.

Proof Let $x \in X \subset C$ and $n \in \mathbb{N}$ be fixed arbitrarily. The definition of x_{n+1} and the firm nonexpansivity of P_{C,H_n} guarantee that, almost surely,

$$\begin{aligned}
&\|x_{n+1} - x\|^2_{\mathsf{H}_n} \\
&\quad \le \big\|\big[\alpha_n x_n + (1-\alpha_n) y_n\big] - x\big\|^2_{\mathsf{H}_n} - \big\|x_{n+1} - \big[\alpha_n x_n + (1-\alpha_n) y_n\big]\big\|^2_{\mathsf{H}_n},
\end{aligned}$$

which, together with $\|\alpha x + (1-\alpha)y\|^2 = \alpha\|x\|^2 + (1-\alpha)\|y\|^2 - \alpha(1-\alpha)\|x-y\|^2$ $(x, y \in \mathbb{R}^N, \alpha \in \mathbb{R})$, implies that

$$\begin{aligned}
\|x_{n+1} - x\|^2_{\mathsf{H}_n} \le{}& \alpha_n \|x_n - x\|^2_{\mathsf{H}_n} + (1-\alpha_n)\|y_n - x\|^2_{\mathsf{H}_n} - \alpha_n \|x_{n+1} - x_n\|^2_{\mathsf{H}_n} \\
&- (1-\alpha_n)\|x_{n+1} - y_n\|^2_{\mathsf{H}_n}.
\end{aligned} \tag{16}$$

The definition of y_n and (A1) ensure that, almost surely,

$$\begin{aligned}
\|y_n - x\|^2_{\mathsf{H}_n} &\le \big\|(x_n - x) + \lambda_n \mathsf{d}_n\big\|^2_{\mathsf{H}_n} \\
&= \|x_n - x\|^2_{\mathsf{H}_n} + 2\lambda_n \langle x_n - x, \mathsf{d}_n\rangle_{\mathsf{H}_n} + \lambda_n^2 \|\mathsf{d}_n\|^2_{\mathsf{H}_n}.
\end{aligned}$$

The definitions of d_n and m_n in turn ensure that

$$\begin{aligned}\langle x_n - x, \mathsf{d}_n\rangle_{\mathsf{H}_n} &= \langle x - x_n, m_n\rangle \\ &= \beta_n\langle x - x_n, m_{n-1}\rangle + (1-\beta_n)\big\langle x - x_n, \mathsf{G}(x_n,\xi_n)\big\rangle.\end{aligned}$$

Hence, (16) implies that, almost surely,

$$\begin{aligned}\|x_{n+1} - x\|_{\mathsf{H}_n}^2 &\le \alpha_n\|x_n - x\|_{\mathsf{H}_n}^2 + (1-\alpha_n)\big\{\|x_n - x\|_{\mathsf{H}_n}^2 + 2\lambda_n\langle x_n - x, \mathsf{d}_n\rangle_{\mathsf{H}_n} \\ &\quad + \lambda_n^2\|\mathsf{d}_n\|_{\mathsf{H}_n}^2\big\} - \alpha_n\|x_{n+1} - x_n\|_{\mathsf{H}_n}^2 - (1-\alpha_n)\|x_{n+1} - y_n\|_{\mathsf{H}_n}^2 \\ &= \|x_n - x\|_{\mathsf{H}_n}^2 + 2(1-\alpha_n)\lambda_n\big\{\beta_n\langle x - x_n, m_{n-1}\rangle \\ &\quad + (1-\beta_n)\big\langle x - x_n, \mathsf{G}(x_n,\xi_n)\big\rangle\big\} + (1-\alpha_n)\lambda_n^2\|\mathsf{d}_n\|_{\mathsf{H}_n}^2 \\ &\quad - \alpha_n\|x_{n+1} - x_n\|_{\mathsf{H}_n}^2 - (1-\alpha_n)\|x_{n+1} - y_n\|_{\mathsf{H}_n}^2.\end{aligned} \tag{17}$$

Moreover, the condition $x_n = x_n(\xi_{[n-1]})$ ($n \in \mathbb{N}$) and (C1) guarantee that

$$\begin{aligned}\mathbb{E}\big[\big\langle x - x_n, \mathsf{G}(x_n,\xi_n)\big\rangle\big] &= \mathbb{E}\big[\mathbb{E}\big[\big\langle x - x_n, \mathsf{G}(x_n,\xi_n)\big\rangle|\xi_{[n-1]}\big]\big] \\ &= \mathbb{E}\big[\big\langle x - x_n, \mathbb{E}\big[\mathsf{G}(x_n,\xi_n)|\xi_{[n-1]}\big]\big\rangle\big] \\ &= \mathbb{E}\big[\big\langle x - x_n, \mathsf{g}(x_n)\big\rangle\big],\end{aligned}$$

which, together with (C2), implies that

$$\mathbb{E}\big[\big\langle x - x_n, \mathsf{G}(x_n,\xi_n)\big\rangle\big] \le \mathbb{E}\big[f(x) - f(x_n)\big].$$

Therefore, taking the expectation of (17) gives the first assertion of Lemma 4.1.

The definition of m_n and (C3), together with the convexity of $\|\cdot\|^2$, guarantee that, for all $n \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}\big[\|m_n\|^2\big] &\le \beta_n\mathbb{E}\big[\|m_{n-1}\|^2\big] + (1-\beta_n)\mathbb{E}\big[\big\|\mathsf{G}(x_n,\xi_n)\big\|^2\big] \\ &\le \beta_n\mathbb{E}\big[\|m_{n-1}\|^2\big] + (1-\beta_n)M^2.\end{aligned}$$

Induction thus ensures that, for all $n \in \mathbb{N}$,

$$\mathbb{E}\big[\|m_n\|^2\big] \le \tilde{M}^2 := \max\big\{\|m_{-1}\|^2, M^2\big\} < +\infty. \tag{18}$$

Given $n \in \mathbb{N}$, $\mathsf{H}_n \succ O$ ensures that there exists a unique matrix $\overline{\mathsf{H}}_n \succ O$ such that $\mathsf{H}_n = \overline{\mathsf{H}}_n^2$ [33, Theorem 7.2.6]. Since $\|x\|_{\mathsf{H}_n}^2 = \|\overline{\mathsf{H}}_n x\|^2$ holds for all $x \in \mathbb{R}^N$, the definition of d_n implies that, for all $n \in \mathbb{N}$,

$$\mathbb{E}\big[\|\mathsf{d}_n\|_{\mathsf{H}_n}^2\big] = \mathbb{E}\big[\big\|\overline{\mathsf{H}}_n^{-1}\mathsf{H}_n\mathsf{d}_n\big\|^2\big] \le \mathbb{E}\big[\big\|\overline{\mathsf{H}}_n^{-1}\big\|^2\|m_n\|^2\big],$$

where $\|\overline{\mathsf{H}}_n^{-1}\| = \|\mathsf{diag}(h_{n,i}^{-1/2})\| = \max_{i=1,2,\ldots,N} h_{n,i}^{-1/2}$ ($n \in \mathbb{N}$). From (18) and

$$h_\star := \sup\Big\{\max_{i=1,2,\ldots,N} h_{n,i}^{-1/2} : n \in \mathbb{N}\Big\} < +\infty$$

(by (A3)), we have that, for all $n \in \mathbb{N}$,

$$\mathbb{E}\left[\|\mathsf{d}_n\|_{\mathsf{H}_n}^2\right] \leq h_\star^2 \tilde{M}^2.$$

This completes the proof. □

The convergence analyses of Algorithm 1 in Sect. 5 depend on the following assumption:

(A4) [5, p.1574], [9, p.2] C ($\supset X$) is bounded.

Let us consider the case where H_n and v_n are defined for all $n \in \mathbb{N}$ by

$$\begin{aligned} v_n &:= \beta v_{n-1} + (1-\beta)\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n), \\ \hat{v}_n &:= (\hat{v}_{n,i}) = \left(\max\{\hat{v}_{n-1,i}, v_{n,i}\}\right), \\ \mathsf{H}_n &:= \mathsf{diag}(\sqrt{\hat{v}_{n,i}}), \end{aligned} \tag{19}$$

where $\beta \in (0,1)$ and $v_{-1} = \hat{v}_{-1} = 0 \in \mathbb{R}^N$ (see also (9)), and discuss the relationship between (A3) and (A4). Assumption (A4) implies that $(x_n)_{n\in\mathbb{N}} \subset C$ generated by Algorithm 1 is almost surely bounded. In the standard case of $\mathsf{G}(x_n, \xi_n) \in \partial_x F(x_n, \xi_n)$, Proposition 2.1 and (A4) imply that $(\mathsf{G}(x_n, \xi_n))_{n\in\mathbb{N}}$ is almost surely bounded, i.e., $M_1 := \sup_{n\in\mathbb{N}} \|\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n)\| < +\infty$. Since the triangle inequality and (19) guarantee that, almost surely, $\|v_n\| \leq \beta\|v_{n-1}\| + (1-\beta)\|\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n)\|$, induction shows that, for all $n \in \mathbb{N}$, almost surely, $\|v_n\| \leq M_1 < +\infty$. Accordingly, (19) leads to the almost sure boundedness of $(\hat{v}_n)_{n\in\mathbb{N}}$. Hence, $h_\star := \sup\{\max_{i=1,2,\ldots,N} \sqrt{\hat{v}_{n,i}} : n \in \mathbb{N}\} < +\infty$, which implies that (A3) holds. The above discussion shows that (A4) implies (A3) when H_n and v_n are as follows (see also (6) and (7)):

$$\begin{aligned} v_n &:= \beta v_{n-1} + (1-\beta)\mathsf{G}(x_n, \xi_n) \odot \mathsf{G}(x_n, \xi_n), \\ \hat{v}_n &:= (\hat{v}_{n,i}) = \left(\max\left\{\frac{v_{n,i}}{1-\beta^{n+1}}, \hat{v}_{n-1,i}\right\}\right), \\ \mathsf{H}_n &:= \mathsf{diag}(\sqrt{\hat{v}_{n,i}}). \end{aligned} \tag{20}$$

We provide some examples of Problem 3.1 with (A0)–(A4) that can be solved by Algorithm 1 under (C1)–(C3).

Example 4.1 (i) Deep learning problem [9, p.2]: At each time step t, stochastic optimization algorithms used in training deep networks pick a point $x_t \in X$ with the parameters of the model to be learned, where $X \subset \mathbb{R}^N$ is the simple, nonempty, bounded, closed convex feasible set of points, and then incur loss $f_t(x_t)$, where $f_t : \mathbb{R}^N \to \mathbb{R}$ is a convex loss function represented as the loss of the model with the chosen parameters in the next minibatch. Accordingly, the stochastic optimization problem in deep networks can be formulated as follows:

$$\text{minimize} \sum_{t=1}^{T} f_t(x) \text{ subject to } x \in X = \mathrm{Fix}(P_X) = \bigcap_{n\in\mathbb{N}} \mathrm{Fix}(P_{X,\mathsf{H}_n}), \tag{21}$$

where T is the total number of rounds in the learning process, and $(\mathsf{H}_n)_{n\in\mathbb{N}} \subset \mathbb{S}^N_{++} \cap \mathbb{D}^N$ defined by each of (19) and (20) satisfies (A0). $Q_{\mathsf{H}_n} := P_{X,\mathsf{H}_n}$ $(n \in \mathbb{N})$ satisfies (A1), and $f(\cdot) = \mathbb{E}[f_\xi(\cdot)] := (1/T)\sum_{t=1}^T f_t(\cdot)$ satisfies (A2). Setting $C := X$ ensures (A4), which implies (A3). Algorithm 1 for solving problem (21) is as follows:

$$x_{n+1} := \alpha_n x_n + (1-\alpha_n) P_{X,\mathsf{H}_n}\left(x_n - \lambda_n \mathsf{H}_n^{-1} m_n\right). \tag{22}$$

(ii) Classifier ensemble problem [18, Sect. 2.2.2], [19, Sect. 3.2.2] (see also [17, Problem II.1]): For a training set $S = \{(z_m, l_m)\}_{m=1}^M \subset \mathbb{R}^N \times \mathbb{R}$, where $z_m := (z_m^n)_{n=1}^N$ and z_m^n is the measure corresponding to the mth sample in the sample set and the nth classifier in an ensemble. The classifier ensemble problem with sparsity learning is the following:

$$\begin{aligned} &\text{minimize } f(x) = \mathbb{E}\left[\frac{1}{2}\left(\langle z, x\rangle - l\right)^2\right] \\ &\text{subject to } x \in X := \mathbb{R}^N_+ \cap \left\{x \in \mathbb{R}^N : \|x\|_1 \le t_1\right\}, \end{aligned} \tag{23}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm and t_1 is the sparsity control parameter. Suppose that H_n is as each of (19) and (20), which satisfies (A0), and define a mapping $Q_{\mathsf{H}_n} : \mathbb{R}^N \to \mathbb{R}^N$ by

$$Q_{\mathsf{H}_n} := P_{\mathbb{R}^N_+,\mathsf{H}_n} P_{\{x\in\mathbb{R}^N :\, \|x\|_1 \le t_1\},\mathsf{H}_n}. \tag{24}$$

Since the projections $P_{\mathbb{R}^N_+,\mathsf{H}_n}$ and $P_{\{x\in\mathbb{R}^N :\, \|x\|_1 \le t_1\},\mathsf{H}_n}$ can be easily computed [34, Lemma 1.1], Q_{H_n} defined by (24) can be also computed. Moreover, Q_{H_n} defined by (24) is nonexpansive with $X = \bigcap_{n\in\mathbb{N}} \mathrm{Fix}(Q_{\mathsf{H}_n})$, i.e., (A1) holds. Since $\{x \in \mathbb{R}^N : \|x\|_1 \le t_1\}$ is bounded, we can set a simple, bounded set C such that $X \subset C$, i.e., (A4) holds. Moreover, f in problem (23) satisfies (A2).

The classifier ensemble problem with both sparsity and diversity learning is as follows:

$$\begin{aligned} &\text{minimize } f(x) = \mathbb{E}\left[\frac{1}{2}\left(\langle z, x\rangle - l\right)^2\right] \\ &\text{subject to } x \in X := \left\{x \in \mathbb{R}^N_+ : \|x\|_1 \le t_1\right\} \cap \left\{x \in \mathbb{R}^N : f_{\mathrm{div}}(x) \ge t_2\right\}, \end{aligned} \tag{25}$$

where t_2 is the diversity control parameter, $f_{\mathrm{div}}(x) := \sum_{m=1}^M \{\langle [z_m], x\rangle - \langle z_m, x\rangle^2\}$ $(x \in \mathbb{R}^N)$, and $[z_m] := ((z_m^i)^2)_{i=1}^N \in \mathbb{R}^N$. From the discussion regarding (10), (11), and (12), a mapping

$$Q_{\mathsf{H}_n} := \omega_1 P_{\mathbb{R}^N_+,\mathsf{H}_n} + \omega_2 Q_{\|\cdot\|_1 - t_1,\mathsf{H}_n} + \omega_3 Q_{-f_{\mathrm{div}}(\cdot)+t_2,\mathsf{H}_n}, \tag{26}$$

with $(\mathsf{H}_n)_{n\in\mathbb{N}} \subset \mathbb{S}^N_{++} \cap \mathbb{D}^N$ defined by each of (19) and (20), is quasinonexpansive under the H_n-norm satisfying $X = \bigcap_{n\in\mathbb{N}} \mathrm{Fix}(Q_{\mathsf{H}_n})$, i.e., (A1) holds. The discussion in the previous paragraph implies that (A0), (A2), and (A4) again hold.

Algorithm 1 for solving each of problems (23) and (25) is represented as follows:

$$x_{n+1} := P_{C,\mathsf{H}_n}\left[\alpha_n x_n + (1-\alpha_n) Q_{\mathsf{H}_n}\left(x_n - \lambda_n \mathsf{H}_n^{-1} m_n\right)\right]. \tag{27}$$

In contrast to Adam (6) and AMSGrad (9) that can solve a convex stochastic optimization problem with a simple constraint (3) (see also problem (21)), algorithm (27) can be

applied to a convex stochastic optimization problem with complicated constraints, such as problems (23) and (25).

(iii) Network utility maximization problem [35, (6), (7)] (see also [36, Problem II.1]): The network resource allocation problem is to determine the source rates that maximize the utility aggregated over all sources over the link capacity constraints and source constraints. This problem can be formulated as the following network utility maximization problem:

$$\text{maximize} \sum_{s\in\mathcal{S}} u_s(x_s) \text{ subject to } x = (x_s)_{s\in\mathcal{S}} \in X := \bigcap_{l\in\mathcal{L}} C_l \cap \bigcap_{s\in\mathcal{S}} C_s, \tag{28}$$

where x_s denotes the transmission rate of source $s \in \mathcal{S}$, u_s is a concave utility function of source s, $\mathcal{S}(l)$ denotes the set of sources that use link $l \in \mathcal{L}$, C_l is the capacity constraint set of link l having capacity $c_l \in \mathbb{R}_+$ defined by $C_l := \{x = (x_s)_{s\in\mathcal{S}} : \sum_{s\in\mathcal{S}(l)} x_s \le c_l\}$, and C_s is the constraint set of source s having the maximum allowed rate M_s defined by $C_s := \{x = (x_s)_{s\in\mathcal{S}} : x_s \in [0, M_s]\}$. Since C_l and C_s are half-spaces, the projections P_{C_l,H_n} and P_{C_s,H_n} are easily computed,[3] where $(\mathsf{H}_n)_{n\in\mathbb{N}} \subset \mathbb{S}^N_{++} \cap \mathbb{D}^N$ is defined by each of (19) and (20). For example, we can define a nonexpansive mapping $Q_{\mathsf{H}_n} := \prod_{l\in\mathcal{L}} P_{C_l,\mathsf{H}_n} \prod_{s\in\mathcal{S}} P_{C_s,\mathsf{H}_n}$ satisfying $X = \bigcap_{n\in\mathbb{N}} \mathrm{Fix}(Q_{\mathsf{H}_n})$. The boundedness of $\bigcap_{s\in\mathcal{S}} C_s$ allows us to set a simple, bounded set C satisfying $C \supset \bigcap_{s\in\mathcal{S}} C_s \supset X$. Algorithm (27) with $\mathsf{G}(x_n, \xi_n) \in \partial(-u_{\xi_n})(x_n)$ can be applied to problem (28).

5 Convergence analyses and comparisons

5.1 Convergence analyses of Algorithm 1

For convergence analyses of Algorithm 1, we prove the following theorem.

Theorem 5.1 *Suppose that* (A0)–(A4) *and* (C1)–(C3) *hold and that* $(\alpha_n)_{n\in\mathbb{N}}$, $(\beta_n)_{n\in\mathbb{N}}$, $(\lambda_n)_{n\in\mathbb{N}}$, *and* $(\gamma_n)_{n\in\mathbb{N}}$ *defined by* $\gamma_n := (1-\alpha_n)(1-\beta_n)\lambda_n$ $(n \in \mathbb{N})$ *satisfy*

$$0 < \liminf_{n\to+\infty} \alpha_n \le \limsup_{n\to+\infty} \alpha_n < 1, \qquad \limsup_{n\to+\infty} \beta_n < 1, \quad \text{and} \quad \gamma_{n+1} \le \gamma_n \quad (n \in \mathbb{N}) \tag{29}$$

and that $\mathsf{H}_n = \mathsf{diag}(h_{n,i})$ *satisfies* [4]

$$h_{n+1,i} \ge h_{n,i} \quad (n \in \mathbb{N}, i = 1, 2, \ldots, N). \tag{30}$$

Then Algorithm 1 is such that the following are satisfied for all $n \ge 1$:

$$\mathbb{E}\big[f(\tilde{x}_n) - f^\star\big] \le \frac{D}{2\tilde{a}\tilde{b}n\lambda_n}\mathbb{E}\left[\sum_{i=1}^N h_{n,i}\right] + \frac{\tilde{M}\sqrt{DN}}{\tilde{b}n}\sum_{k=1}^n \beta_k + \frac{h_\star^2\tilde{M}^2}{2\tilde{b}n}\sum_{k=1}^n \lambda_k,$$

where $\tilde{x}_n := (1/n)\sum_{k=1}^n x_k$, $\tilde{M}$ *and* $h_\star$ *are defined as in Lemma 4.1,*

$$D := \max_{i=1,2,\ldots,N} \sup\big\{(x_{k+1,i} - x_i)^2 : k \in \mathbb{N}\big\} < +\infty,$$

[3] The projection P_{C,H_n} onto a half-space $C := \{x \in \mathbb{R}^N : \langle a, x\rangle \le b\} = \mathrm{Fix}(P_C) = \mathrm{Fix}(P_{C,\mathsf{H}_n})$ under the H_n-norm, where $a \ne 0$ and $b \in \mathbb{R}$, can be defined for all $x \in \mathbb{R}^N$ by $P_{C,\mathsf{H}_n}(x) := x + [(b - \langle a, x\rangle_{\mathsf{H}_n})/\|a\|^2_{\mathsf{H}_n}]a$ $(x \notin C)$ or $P_{C,\mathsf{H}_n}(x) := x$ $(x \in C)$.

[4] Condition (30) is satisfied when H_n is defined by either (19) or (20).

$(\alpha_n)_{n\in\mathbb{N}} \subset [c,a] \subset (0,1)$, $(\beta_n)_{n\in\mathbb{N}} \subset (0,b] \subset (0,1)$, $\tilde{a} := 1-a$, $\tilde{b} := 1-b$, $\tilde{c} := 1-c$, *and* $\hat{M} := \sup\{\mathbb{E}[f(x)-f(x_n)] : n \in \mathbb{N}\} < +\infty$ *for some* $x \in X$. *If*

(A1)' $Q_{\mathsf{H}_n} : \mathbb{R}^N \to \mathbb{R}^N$ *is nonexpansive under the* H_n*-norm,*

then, for all $n \geq 1$,

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\left\|x_k - Q_{\mathsf{H}_k}(x_k)\right\|_{\mathsf{H}_k}^2\right] \\
&\quad\leq 4\left(\frac{1}{\tilde{a}}+\frac{1}{c}\right)\left\{\frac{D}{n}\mathbb{E}\left[\sum_{i=1}^{N}h_{n,i}\right]+\frac{2\tilde{c}\hat{M}}{n}\sum_{k=1}^{n}(1-\beta_k)\lambda_k+\frac{2\tilde{c}\hat{M}\sqrt{DN}}{n}\sum_{k=1}^{n}\beta_k\lambda_k\right\} \\
&\qquad+\left\{4\left(\frac{1}{\tilde{a}}+\frac{1}{c}\right)\tilde{c}+2\right\}\frac{h_\star^2\tilde{M}^2}{n}\sum_{k=1}^{n}\lambda_k^2.
\end{aligned}$$

Proof Let $x \in X$ be fixed arbitrarily. Lemma 4.1 guarantees that, for all $k \in \mathbb{N}$,

$$\begin{aligned}
\mathbb{E}\left[f(x_k)-f(x)\right] &\leq \frac{1}{2\gamma_k}\left\{\mathbb{E}\left[\|x_k-x\|_{\mathsf{H}_k}^2\right]-\mathbb{E}\left[\|x_{k+1}-x\|_{\mathsf{H}_k}^2\right]\right\} \\
&\quad+\frac{\beta_k}{1-\beta_k}\mathbb{E}\left[\langle x-x_k, m_{k-1}\rangle\right]+\frac{\lambda_k}{2(1-\beta_k)}\mathbb{E}\left[\|\mathsf{d}_k\|_{\mathsf{H}_k}^2\right].
\end{aligned}$$

Summing the above inequality ensures that, for all $n \geq 1$,

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[f(x_k)-f(x)\right] \\
&\quad\leq \underbrace{\frac{1}{2n}\sum_{k=1}^{n}\frac{1}{\gamma_k}\left\{\mathbb{E}\left[\|x_k-x\|_{\mathsf{H}_k}^2\right]-\mathbb{E}\left[\|x_{k+1}-x\|_{\mathsf{H}_k}^2\right]\right\}}_{\Gamma_n} \\
&\qquad+\underbrace{\frac{1}{n}\sum_{k=1}^{n}\frac{\beta_k}{1-\beta_k}\mathbb{E}\left[\langle x-x_k, m_{k-1}\rangle\right]}_{B_n}+\frac{1}{2\tilde{b}n}\underbrace{\sum_{k=1}^{n}\lambda_k\mathbb{E}\left[\|\mathsf{d}_k\|_{\mathsf{H}_k}^2\right]}_{\Lambda_n},
\end{aligned} \tag{31}$$

where (29) implies that $b > 0$ exists such that, for all $n \in \mathbb{N}$, $\beta_n \leq b < 1$ and $\tilde{b} := 1-b$. The definition of Γ_n and $\mathbb{E}[\|x_{n+1}-x\|_{\mathsf{H}_n}^2]/\gamma_n \geq 0$ imply that

$$\Gamma_n \leq \frac{\mathbb{E}[\|x_1-x\|_{\mathsf{H}_1}^2]}{\gamma_1}+\underbrace{\sum_{k=2}^{n}\left\{\frac{\mathbb{E}[\|x_k-x\|_{\mathsf{H}_k}^2]}{\gamma_k}-\frac{\mathbb{E}[\|x_k-x\|_{\mathsf{H}_{k-1}}^2]}{\gamma_{k-1}}\right\}}_{\tilde{\Gamma}_n}. \tag{32}$$

Given $k \in \mathbb{N}$, $\mathsf{H}_k \succ O$ ensures that there exists a unique matrix $\overline{\mathsf{H}}_k \succ O$ such that $\mathsf{H}_k = \overline{\mathsf{H}}_k^2$ [33, Theorem 7.2.6]. Since $\|x\|_{\mathsf{H}_k}^2 = \|\overline{\mathsf{H}}_k x\|^2$ holds for all $x \in \mathbb{R}^N$, we have that, for all $k \in \mathbb{N}$,

$$\tilde{\Gamma}_n = \mathbb{E}\left[\sum_{k=2}^{n}\left\{\frac{\|\overline{\mathsf{H}}_k(x_k-x)\|^2}{\gamma_k}-\frac{\|\overline{\mathsf{H}}_{k-1}(x_k-x)\|^2}{\gamma_{k-1}}\right\}\right]. \tag{33}$$

Since H_k ($k \in \mathbb{N}$) is diagonal, we can express H_k as $\mathsf{H}_k = \mathsf{diag}(h_{k,i})$, where $h_{k,i} > 0$ ($k \in \mathbb{N}$, $i = 1, 2, \ldots, N$). Accordingly, for all $k \in \mathbb{N}$ and all $x := (x_i)_{i=1}^N \in \mathbb{R}^N$,

$$\overline{\mathsf{H}}_k = \mathsf{diag}\big(h_{k,i}^{\frac{1}{2}}\big) \quad \text{and} \quad \|\overline{\mathsf{H}}_k x\|^2 = \sum_{i=1}^N h_{k,i} x_i^2. \tag{34}$$

Hence, (33) ensures that, for all $n \in \mathbb{N}$,

$$\tilde{\Gamma}_n = \mathbb{E}\left[\sum_{k=2}^n \sum_{i=1}^N \left(\frac{h_{k,i}}{\gamma_k} - \frac{h_{k-1,i}}{\gamma_{k-1}}\right)(x_{k,i} - x_i)^2\right].$$

From $\gamma_k \leq \gamma_{k-1}$ ($k \geq 1$) (see (29)) and (30), we have that $h_{k,i}/\gamma_k - h_{k-1,i}/\gamma_{k-1} \geq 0$ ($k \geq 1$, $i = 1, 2, \ldots, N$). Moreover, (A4) implies that $D := \max_{i=1,2,\ldots,N} \sup\{(x_{n,i} - x_i)^2 : n \in \mathbb{N}\} < +\infty$. Accordingly, for all $n \in \mathbb{N}$,

$$\tilde{\Gamma}_n \leq D\mathbb{E}\left[\sum_{k=2}^n \sum_{i=1}^N \left(\frac{h_{k,i}}{\gamma_k} - \frac{h_{k-1,i}}{\gamma_{k-1}}\right)\right] = D\mathbb{E}\left[\sum_{i=1}^N \left(\frac{h_{n,i}}{\gamma_n} - \frac{h_{1,i}}{\gamma_1}\right)\right].$$

Hence, (32), together with $\mathbb{E}[\|x_1 - x\|_{\mathsf{H}_1}^2]/\gamma_1 \leq D\mathbb{E}[\sum_{i=1}^N h_{1,i}/\gamma_1]$, implies that, for all $n \in \mathbb{N}$,

$$\Gamma_n \leq D\mathbb{E}\left[\sum_{i=1}^N \frac{h_{1,i}}{\gamma_1}\right] + D\mathbb{E}\left[\sum_{i=1}^N \left(\frac{h_{n,i}}{\gamma_n} - \frac{h_{1,i}}{\gamma_1}\right)\right] = \frac{D}{\gamma_n}\mathbb{E}\left[\sum_{i=1}^N h_{n,i}\right],$$

which, together with the existence of $a > 0$ such that, for all $n \in \mathbb{N}$, $\alpha_n \leq a < 1$ (by (29)) and $\tilde{a} := 1 - a$, implies that

$$\Gamma_n \leq \frac{D}{\tilde{a}\tilde{b}\lambda_n}\mathbb{E}\left[\sum_{i=1}^N h_{n,i}\right]. \tag{35}$$

The Cauchy–Schwarz inequality, together with $D := \max_{i=1,2,\ldots,N} \sup\{(x_{n,i} - x_i)^2 : n \in \mathbb{N}\} < +\infty$ and $\mathbb{E}[\|m_n\|] \leq \tilde{M} := \sqrt{\max\{\|m_{-1}\|^2, M^2\}}$ ($n \in \mathbb{N}$) (by Lemma 4.1), guarantees that, for all $n \in \mathbb{N}$,

$$\begin{aligned} B_n &\leq \sum_{k=1}^n \frac{\beta_k}{1-\beta_k}\mathbb{E}\big[\|x - x_k\|\|m_{k-1}\|\big] \leq \frac{\sqrt{DN}}{\tilde{b}}\sum_{k=1}^n \beta_k \mathbb{E}\big[\|m_{k-1}\|\big] \\ &\leq \frac{\tilde{M}\sqrt{DN}}{\tilde{b}}\sum_{k=1}^n \beta_k. \end{aligned} \tag{36}$$

Since $\mathbb{E}[\|\mathsf{d}_n\|_{\mathsf{H}_n}^2] \leq h_\star^2 \tilde{M}^2$ ($n \in \mathbb{N}$) holds (by Lemma 4.1), we have that, for all $n \in \mathbb{N}$,

$$\Lambda_n := \sum_{k=1}^n \lambda_k \mathbb{E}\big[\|\mathsf{d}_k\|_{\mathsf{H}_k}^2\big] \leq h_\star^2 \tilde{M}^2 \sum_{k=1}^n \lambda_k. \tag{37}$$

Therefore, (31), (35), (36), and (37), together with the convexity of f, imply that, for all $n \in \mathbb{N}$,

$$\mathbb{E}\big[f(\tilde{x}_n) - f(x)\big] \leq \frac{D}{2\tilde{a}\tilde{b}n\lambda_n}\mathbb{E}\left[\sum_{i=1}^{N} h_{n,i}\right] + \frac{\tilde{M}\sqrt{DN}}{\tilde{b}n}\sum_{k=1}^{n}\beta_k + \frac{h_\star^2\tilde{M}^2}{2\tilde{b}n}\sum_{k=1}^{n}\lambda_k.$$

Lemma 4.1 ensures that, for all $n \in \mathbb{N}$,

$$\begin{aligned}
&\tilde{a}\sum_{k=1}^{n}\mathbb{E}\big[\|x_{k+1} - y_k\|_{\mathsf{H}_k}^2\big] \\
&\quad\leq \underbrace{\sum_{k=1}^{n}\big\{\mathbb{E}\big[\|x_k - x\|_{\mathsf{H}_k}^2\big] - \mathbb{E}\big[\|x_{k+1} - x\|_{\mathsf{H}_k}^2\big]\big\}}_{X_n} + \sum_{k=1}^{n}(1-\alpha_k)\lambda_k^2\mathbb{E}\big[\|\mathsf{d}_k\|_{\mathsf{H}_k}^2\big] \\
&\qquad + 2\sum_{k=1}^{n}(1-\alpha_k)\lambda_k\big\{(1-\beta_k)\mathbb{E}\big[f(x) - f(x_k)\big] + \beta_k\mathbb{E}\big[\langle x - x_k, m_{k-1}\rangle\big]\big\}.
\end{aligned}$$

A discussion similar to the one for obtaining (35) implies that

$$X_n \leq D\mathbb{E}\left[\sum_{i=1}^{N} h_{1,i}\right] + D\mathbb{E}\left[\sum_{i=1}^{N}(h_{n,i} - h_{1,i})\right] = D\mathbb{E}\left[\sum_{i=1}^{N} h_{n,i}\right].$$

The continuity of f (see (A2)) and (A4) mean that $\hat{M} := \sup\{\mathbb{E}[f(x) - f(x_n)] \colon n \in \mathbb{N}\} < +\infty$. Accordingly, an argument similar to the one for obtaining (36) and (37) guarantees that, for all $n \in \mathbb{N}$,

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\big[\|x_{k+1} - y_k\|_{\mathsf{H}_k}^2\big] \\
&\quad\leq \frac{D}{\tilde{a}n}\mathbb{E}\left[\sum_{i=1}^{N} h_{n,i}\right] + \frac{2\hat{M}}{\tilde{a}n}\sum_{k=1}^{n}(1-\alpha_k)(1-\beta_k)\lambda_k + \frac{2\tilde{M}\sqrt{DN}}{\tilde{a}n}\sum_{k=1}^{n}(1-\alpha_k)\beta_k\lambda_k \\
&\qquad + \frac{h_\star^2\tilde{M}^2}{\tilde{a}n}\sum_{k=1}^{n}(1-\alpha_k)\lambda_k^2.
\end{aligned}$$

From (29), there exists $c > 0$ such that, for all $n \in \mathbb{N}$, $c \leq \alpha_n$. Setting $\tilde{c} := 1 - c$, it follows that, for all $n \in \mathbb{N}$,

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\big[\|x_{k+1} - y_k\|_{\mathsf{H}_k}^2\big] \\
&\quad\leq \frac{D}{\tilde{a}n}\mathbb{E}\left[\sum_{i=1}^{N} h_{n,i}\right] + \frac{2\tilde{c}\hat{M}}{\tilde{a}n}\sum_{k=1}^{n}(1-\beta_k)\lambda_k + \frac{2\tilde{c}\tilde{M}\sqrt{DN}}{\tilde{a}n}\sum_{k=1}^{n}\beta_k\lambda_k \\
&\qquad + \frac{\tilde{c}h_\star^2\tilde{M}^2}{\tilde{a}n}\sum_{k=1}^{n}\lambda_k^2.
\end{aligned} \tag{38}$$

A discussion similar to the one for obtaining (38) ensures that, for all $n \in \mathbb{N}$,

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\|x_{k+1}-x_k\|_{\mathsf{H}_k}^2\right] \\
&\quad\le \frac{D}{cn}\mathbb{E}\left[\sum_{i=1}^{N}h_{n,i}\right]+\frac{2\tilde{c}\hat{M}}{cn}\sum_{k=1}^{n}(1-\beta_k)\lambda_k+\frac{2\tilde{c}\tilde{M}\sqrt{DN}}{cn}\sum_{k=1}^{n}\beta_k\lambda_k \\
&\qquad+\frac{\tilde{c}h_\star^2\tilde{M}^2}{cn}\sum_{k=1}^{n}\lambda_k^2.
\end{aligned} \tag{39}$$

Suppose that (A1)' holds. Then we have that, for all $k \in \mathbb{N}$, almost surely $\|y_k - Q_{\mathsf{H}_k}(x_k)\|_{\mathsf{H}_k} = \|Q_{\mathsf{H}_k}(x_k+\lambda_k \mathsf{d}_k) - Q_{\mathsf{H}_k}(x_k)\|_{\mathsf{H}_k} \le \lambda_k\|\mathsf{d}_k\|_{\mathsf{H}_k}$, which, together with $\|x-y\|^2 \le 2\|x\|^2+2\|y\|^2$ $(x,y\in\mathbb{R}^N)$, implies that

$$\begin{aligned}
\mathbb{E}\left[\left\|x_k-Q_{\mathsf{H}_k}(x_k)\right\|_{\mathsf{H}_k}^2\right] &\le 2\mathbb{E}\left[\|x_k-y_k\|_{\mathsf{H}_k}^2\right]+2\mathbb{E}\left[\left\|y_k-Q_{\mathsf{H}_k}(x_k)\right\|_{\mathsf{H}_k}^2\right] \\
&\le 2\mathbb{E}\left[\|x_k-y_k\|_{\mathsf{H}_k}^2\right]+2\lambda_k^2\mathbb{E}\left[\|\mathsf{d}_k\|_{\mathsf{H}_k}^2\right].
\end{aligned}$$

Accordingly, (38) and (39) guarantee that, for all $n \in \mathbb{N}$,

$$\begin{aligned}
&\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\left\|x_k-Q_{\mathsf{H}_k}(x_k)\right\|_{\mathsf{H}_k}^2\right] \\
&\quad\le \frac{4}{n}\sum_{k=1}^{n}\mathbb{E}\left[\|x_k-x_{k+1}\|_{\mathsf{H}_k}^2\right]+\frac{4}{n}\sum_{k=1}^{n}\mathbb{E}\left[\|x_{k+1}-y_k\|_{\mathsf{H}_k}^2\right]+\frac{2}{n}\sum_{k=1}^{n}\lambda_k^2\mathbb{E}\left[\|\mathsf{d}_k\|_{\mathsf{H}_k}^2\right] \\
&\quad\le 4\left(\frac{1}{\tilde{a}}+\frac{1}{c}\right)\left\{\frac{D}{n}\mathbb{E}\left[\sum_{i=1}^{N}h_{n,i}\right]+\frac{2\tilde{c}\hat{M}}{n}\sum_{k=1}^{n}(1-\beta_k)\lambda_k+\frac{2\tilde{c}\tilde{M}\sqrt{DN}}{n}\sum_{k=1}^{n}\beta_k\lambda_k\right\} \\
&\qquad+\left\{4\left(\frac{1}{\tilde{a}}+\frac{1}{c}\right)\tilde{c}+2\right\}\frac{h_\star^2\tilde{M}^2}{n}\sum_{k=1}^{n}\lambda_k^2,
\end{aligned}$$

which completes the proof. □

5.1.1 Constant step-size rule

The following theorem indicates that sufficiently small constant step-sizes $\beta_n := \beta$ and $\lambda_n := \lambda$ allow a solution to the problem to be approximated.

Theorem 5.2 *Suppose that the assumptions in Theorem 5.1 hold and also assume that, for all $i = 1,2,\ldots,N$, there exists a positive number B_i such that*[5]

$$\sup\left\{\mathbb{E}[h_{n,i}]\colon n\in\mathbb{N}\right\} \le B_i. \tag{40}$$

Then Algorithm 1 with $\alpha_n := \alpha$, $\beta_n := \beta$, and $\lambda_n := \lambda$ $(n \in \mathbb{N})$ satisfies that

$$\liminf_{n\to+\infty}\mathbb{E}\left[\|x_n-x_{n+1}\|_{\mathsf{H}_n}^2\right] \le \frac{2\tilde{\alpha}}{\alpha}\left\{\hat{M}(1-\beta)+\tilde{M}\sqrt{DN}\beta+\frac{h_\star^2\tilde{M}^2}{2}\lambda\right\}\lambda, \tag{41}$$

[5]Condition (40) is satisfied when H_n is defined by either (19) or (20).

$$\liminf_{n\to+\infty}\mathbb{E}\left[\|x_{n+1}-y_n\|_{\mathsf{H}_n}^2\right]\leq 2\left\{\hat{M}(1-\beta)+\tilde{M}\sqrt{DN}\beta+\frac{h_\star^2\tilde{M}^2}{2}\lambda\right\}\lambda, \tag{42}$$

$$\liminf_{n\to+\infty}\mathbb{E}\left[f(x_n)-f^\star\right]\leq\frac{\tilde{M}\sqrt{DN}}{1-\beta}\beta+\frac{h_\star^2\tilde{M}^2}{2(1-\beta)}\lambda, \tag{43}$$

$$\mathbb{E}\left[f(\tilde{x}_n)-f^\star\right]\leq\mathcal{O}\left(\frac{1}{n}\right)+\frac{\tilde{M}\sqrt{DN}}{1-\beta}\beta+\frac{h_\star^2\tilde{M}^2}{2(1-\beta)}\lambda, \tag{44}$$

where $\tilde{x}_n:=(1/n)\sum_{k=1}^n x_k$ *and* $\tilde{\alpha}:=1-\alpha$. *Under* (A1)′, *we have*

$$\begin{aligned}&\frac{1}{n}\sum_{k=1}^n\mathbb{E}\left[\left\|x_k-Q_{\mathsf{H}_k}(x_k)\right\|_{\mathsf{H}_k}^2\right]\\&\quad\leq\mathcal{O}\left(\frac{1}{n}\right)+\frac{4}{\alpha}\left\{2\hat{M}(1-\beta)+2\tilde{M}\sqrt{DN}\beta+2h_\star^2\tilde{M}^2\lambda\right\}\lambda+2h_\star^2\tilde{M}^2\lambda^2.\end{aligned} \tag{45}$$

Proof We first show that, for all $\epsilon>0$,

$$\begin{aligned}\liminf_{n\to+\infty}\mathbb{E}\left[\|x_n-x_{n+1}\|_{\mathsf{H}_n}^2\right]\leq\frac{2\tilde{\alpha}}{\alpha}&\left\{\hat{M}(1-\beta)+\tilde{M}\sqrt{DN}\beta+\frac{h_\star^2\tilde{M}^2}{2}\lambda\right\}\lambda\\&+D\epsilon+\epsilon.\end{aligned} \tag{46}$$

If (46) does not hold, then there exists $\epsilon_0>0$ such that

$$\begin{aligned}\liminf_{n\to+\infty}\mathbb{E}\left[\|x_n-x_{n+1}\|_{\mathsf{H}_n}^2\right]>\frac{2\tilde{\alpha}}{\alpha}&\left\{\hat{M}(1-\beta)+\tilde{M}\sqrt{DN}\beta+\frac{h_\star^2\tilde{M}^2}{2}\lambda\right\}\lambda\\&+D\epsilon_0+\epsilon_0.\end{aligned} \tag{47}$$

Let $x\in X$ and $\chi_n:=\mathbb{E}[\|x_n-x\|_{\mathsf{H}_n}^2]$ for all $n\in\mathbb{N}$. Lemma 4.1, together with the proofs of (36) and (37), implies that, for all $n\in\mathbb{N}$,

$$\begin{aligned}\chi_{n+1}\leq\chi_n+\underbrace{\chi_{n+1}-\mathbb{E}\left[\|x_{n+1}-x\|_{\mathsf{H}_n}^2\right]}_{\tilde{X}_n}&-\alpha\mathbb{E}\left[\|x_{n+1}-x_n\|_{\mathsf{H}_n}^2\right]\\&+2\tilde{\alpha}\lambda\left\{\hat{M}(1-\beta)+\tilde{M}\sqrt{DN}\beta+\frac{h_\star^2\tilde{M}^2}{2}\lambda\right\}.\end{aligned} \tag{48}$$

From (34) and (A4), for all $n\in\mathbb{N}$,

$$\tilde{X}_n=\mathbb{E}\left[\sum_{i=1}^N(h_{n+1,i}-h_{n,i})(x_{n+1,i}-x_i)^2\right]\leq D\mathbb{E}\left[\sum_{i=1}^N(h_{n+1,i}-h_{n,i})\right].$$

Accordingly, (30) and (40) ensure that there exists $n_0\in\mathbb{N}$ such that, for all $n\geq n_0$,

$$\tilde{X}_n\leq D\alpha\epsilon_0. \tag{49}$$

Hence, (48) implies that, for all $n \geq n_0$,

$$\chi_{n+1} \leq \chi_n + D\alpha\epsilon_0 - \alpha\mathbb{E}\left[\|x_{n+1} - x_n\|_{\mathsf{H}_n}^2\right] + 2\tilde{\alpha}\lambda\left\{\hat{M}(1-\beta) + \tilde{M}\sqrt{DN}\beta + \frac{h_\star^2\tilde{M}^2}{2}\lambda\right\}.$$

From (47), there exists $n_1 \in \mathbb{N}$ such that, for all $n \geq n_1$,

$$\mathbb{E}\left[\|x_n - x_{n+1}\|_{\mathsf{H}_n}^2\right] > \frac{2\tilde{\alpha}}{\alpha}\left\{\hat{M}(1-\beta) + \tilde{M}\sqrt{DN}\beta + \frac{h_\star^2\tilde{M}^2}{2}\lambda\right\}\lambda + D\epsilon_0 + \frac{\epsilon_0}{2}.$$

Therefore, for all $n \geq n_2 := \max\{n_0, n_1\}$,

$$\begin{aligned}\chi_{n+1} &\leq \chi_n + D\alpha\epsilon_0 - 2\tilde{\alpha}\lambda\left\{\hat{M}(1-\beta) + \tilde{M}\sqrt{DN}\beta + \frac{h_\star^2\tilde{M}^2}{2}\lambda\right\} - D\alpha\epsilon_0 - \frac{\alpha\epsilon_0}{2} \\ &\quad + 2\tilde{\alpha}\lambda\left\{\hat{M}(1-\beta) + \tilde{M}\sqrt{DN}\beta + \frac{h_\star^2\tilde{M}^2}{2}\lambda\right\} \\ &= \chi_n - \frac{\alpha\epsilon_0}{2} \\ &\leq \chi_{n_2} - \frac{\alpha\epsilon_0}{2}(n+1-n_2),\end{aligned}$$

which is a contradiction since the right-hand side of the above inequality approaches minus infinity as n increases. Hence, (46) holds for all ϵ, which implies that (41) holds. A discussion similar to the one for showing (46) leads to (42). We next show that, for all $\epsilon > 0$,

$$\liminf_{n\to+\infty}\mathbb{E}\left[f(x_n) - f^\star\right] \leq \frac{\tilde{M}\sqrt{DN}}{1-\beta}\beta + \frac{h_\star^2\tilde{M}^2}{2(1-\beta)}\lambda + \frac{D\alpha\epsilon}{2\tilde{\alpha}(1-\beta)\lambda} + \epsilon. \tag{50}$$

If (50) does not hold for all $\epsilon > 0$, then there exist $\epsilon_0 > 0$ and $n_3 \in \mathbb{N}$ such that, for all $n \geq n_3$,

$$\mathbb{E}\left[f(x_n) - f^\star\right] > \frac{\tilde{M}\sqrt{DN}}{1-\beta}\beta + \frac{h_\star^2\tilde{M}^2}{2(1-\beta)}\lambda + \frac{D\alpha\epsilon_0}{2\tilde{\alpha}(1-\beta)\lambda} + \frac{\epsilon_0}{2}.$$

Lemma 4.1, together with (48) and (49), ensures that, for all $n \geq n_0$,

$$\chi_{n+1} \leq \chi_n + D\alpha\epsilon_0 - 2\tilde{\alpha}(1-\beta)\lambda\mathbb{E}\left[f(x_n) - f^\star\right] + \left\{2\tilde{M}\sqrt{DN}\beta + h_\star^2\tilde{M}^2\lambda\right\}\tilde{\alpha}\lambda.$$

Accordingly, for all $n \geq n_4 := \max\{n_0, n_3\}$,

$$\begin{aligned}&\chi_{n+1} \\ &\quad\leq \chi_n + D\alpha\epsilon_0 - 2\tilde{\alpha}(1-\beta)\lambda\left\{\frac{\tilde{M}\sqrt{DN}}{1-\beta}\beta + \frac{h_\star^2\tilde{M}^2}{2(1-\beta)}\lambda + \frac{D\alpha\epsilon_0}{2\tilde{\alpha}(1-\beta)\lambda} + \frac{\epsilon_0}{2}\right\} \\ &\qquad + \left\{2\tilde{M}\sqrt{DN}\beta + h_\star^2\tilde{M}^2\lambda\right\}\tilde{\alpha}\lambda \\ &\quad= \chi_n - \tilde{\alpha}(1-\beta)\lambda\epsilon_0 \\ &\quad\leq \chi_{n_4} - \tilde{\alpha}(1-\beta)\lambda\epsilon_0(n+1-n_4),\end{aligned}$$

which is a contradiction. Since (50) holds for all $\epsilon > 0$, we have (43). Conditions (44) and (45) follow from Theorem 5.1, which completes the proof. □

5.1.2 Diminishing step-size rule

Lemma 4.1 and Theorem 5.1 give us the following theorem as a convergence analysis of Algorithm 1 with a diminishing step-size.

Theorem 5.3 *Suppose that the assumptions in Theorem 5.1 and (40) hold. Let $(\beta_n)_{n\in\mathbb{N}}$ and $(\lambda_n)_{n\in\mathbb{N}}$ satisfy the following*:

$$\lim_{n\to+\infty}\beta_n = 0, \qquad \sum_{n=0}^{+\infty}\lambda_n = +\infty, \qquad \sum_{n=0}^{+\infty}\lambda_n^2 < +\infty, \quad \text{and} \quad \sum_{n=0}^{+\infty}\beta_n\lambda_n < +\infty. \tag{51}$$

Then Algorithm 1 satisfies that

$$\liminf_{n\to+\infty}\mathbb{E}\left[\|x_n - x_{n+1}\|_{\mathsf{H}_n}\right] = 0, \qquad \liminf_{n\to+\infty}\mathbb{E}\left[\|x_{n+1} - y_n\|_{\mathsf{H}_n}\right] = 0, \tag{52}$$

$$\liminf_{n\to+\infty}\mathbb{E}\left[f(x_n) - f^\star\right] \le 0. \tag{53}$$

Moreover, if (A1)' holds, then we have

$$\liminf_{n\to+\infty}\mathbb{E}\left[\left\|x_n - Q_{\mathsf{H}_n}(x_n)\right\|_{\mathsf{H}_n}\right] = 0.$$

Let $(\beta_n)_{n\in\mathbb{N}}$ and $(\lambda_n)_{n\in\mathbb{N}}$ satisfy the following:

$$\lim_{n\to+\infty}\frac{1}{n\lambda_n} = 0, \qquad \lim_{n\to+\infty}\frac{1}{n}\sum_{k=1}^{n}\lambda_k = 0, \quad \text{and} \quad \lim_{n\to+\infty}\frac{1}{n}\sum_{k=1}^{n}\beta_k = 0. \tag{54}$$

Then the sequence $(\tilde{x}_n)_{n\in\mathbb{N}}$ defined by $\tilde{x}_n := (1/n)\sum_{k=1}^{n} x_k$ satisfies

$$\limsup_{n\to+\infty}\mathbb{E}\left[f(\tilde{x}_n) - f^\star\right] \le 0$$

with

$$\mathbb{E}\left[f(\tilde{x}_n) - f^\star\right] \le \frac{D\sum_{i=1}^{N}B_i}{2\tilde{a}\tilde{b}n\lambda_n} + \frac{\tilde{M}\sqrt{DN}}{\tilde{b}n}\sum_{k=1}^{n}\beta_k + \frac{h_\star^2\tilde{M}^2}{2\tilde{b}n}\sum_{k=1}^{n}\lambda_k.$$

Moreover, if (A1)' holds, then we have

$$\lim_{n\to+\infty}\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\left\|x_k - Q_{\mathsf{H}_k}(x_k)\right\|_{\mathsf{H}_k}^2\right] = 0$$

with

$$\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\left[\left\|x_k - Q_{\mathsf{H}_k}(x_k)\right\|_{\mathsf{H}_k}^2\right]$$

$$\leq 4\left(\frac{1}{\tilde{a}}+\frac{1}{c}\right)\left\{\frac{D\sum_{i=1}^{N}B_i}{n}+\frac{2\tilde{c}\hat{M}}{n}\sum_{k=1}^{n}(1-\beta_k)\lambda_k+\frac{2\tilde{c}\tilde{M}\sqrt{DN}}{n}\sum_{k=1}^{n}\beta_k\lambda_k\right\}$$
$$+\left\{4\left(\frac{1}{\tilde{a}}+\frac{1}{c}\right)\tilde{c}+2\right\}\frac{h_\star^2\tilde{M}^2}{n}\sum_{k=1}^{n}\lambda_k^2.$$

Proof We first show (52). Lemma 4.1, together with (36), (37), and (48), implies that, for all $n\in\mathbb{N}$,

$$\left.\begin{array}{l}\alpha_n\mathbb{E}[\|x_{n+1}-x_n\|_{\mathsf{H}_n}^2]\\(1-\alpha_n)\mathbb{E}[\|x_{n+1}-y_n\|_{\mathsf{H}_n}^2]\end{array}\right\}\leq \chi_n(x)-\chi_{n+1}(x)+D\mathbb{E}\left[\sum_{i=1}^{N}(h_{n+1,i}-h_{n,i})\right]$$
$$+2\hat{M}\lambda_n+2\tilde{M}\sqrt{DN}\beta_n\lambda_n+h_\star^2\tilde{M}^2\lambda_n^2, \tag{55}$$

where $\chi_n(x):=\mathbb{E}[\|x_n-x\|_{\mathsf{H}_n}^2]$ for all $x\in X$ and all $n\in\mathbb{N}$. Consider (Case 1): For all $x\in X$, there exists $m_0\in\mathbb{N}$ such that, for all $n\in\mathbb{N}$, $n\geq m_0$ implies $\chi_{n+1}(x)\leq\chi_n(x)$. This case guarantees the existence of $\lim_{n\to+\infty}\chi_n(x)$ for all $x\in X$. From (30) and (40), we have that $\lim_{n\to+\infty}\mathbb{E}[\sum_{i=1}^{N}(h_{n+1,i}-h_{n,i})]=0$. Moreover, (51) ensures that $\lim_{n\to+\infty}\beta_n=\lim_{n\to+\infty}\lambda_n=0$. Accordingly, (55) and $0<\liminf_{n\to+\infty}\alpha_n\leq\limsup_{n\to+\infty}\alpha_n<1$ (by (29)) imply that

$$\lim_{n\to+\infty}\mathbb{E}\left[\|x_{n+1}-x_n\|_{\mathsf{H}_n}\right]=0\quad\text{and}\quad\lim_{n\to+\infty}\mathbb{E}\left[\|x_{n+1}-y_n\|_{\mathsf{H}_n}\right]=0. \tag{56}$$

Consider (Case 2): There exists $x_0\in X$, for all $m\in\mathbb{N}$, there exists $n\in\mathbb{N}$ such that $n\geq m$ and $\chi_{n+1}(x_0)>\chi_n(x_0)$. In this case, there exists $(x_{n_i})_{i\in\mathbb{N}}\subset(x_n)_{n\in\mathbb{N}}$ such that, for all $i\in\mathbb{N}$, $\chi_{n_i+1}(x_0)>\chi_{n_i}(x_0)$. From (55), we have that, for all $i\in\mathbb{N}$,

$$\left.\begin{array}{l}\alpha_{n_i}\mathbb{E}[\|x_{n_i+1}-x_{n_i}\|_{\mathsf{H}_{n_i}}^2]\\(1-\alpha_n)\mathbb{E}[\|x_{n_i+1}-y_{n_i}\|_{\mathsf{H}_{n_i}}^2]\end{array}\right\}<D\mathbb{E}\left[\sum_{j=1}^{N}(h_{n_i+1,j}-h_{n_i,j})\right]$$
$$+2\hat{M}\lambda_{n_i}+2\tilde{M}\sqrt{DN}\beta_{n_i}\lambda_{n_i}+h_\star^2\tilde{M}^2\lambda_{n_i}^2.$$

A discussion similar to the one for showing (56) guarantees that

$$\lim_{i\to+\infty}\mathbb{E}\left[\|x_{n_i+1}-x_{n_i}\|_{\mathsf{H}_{n_i}}\right]=0\quad\text{and}\quad\lim_{i\to+\infty}\mathbb{E}\left[\|x_{n_i+1}-y_{n_i}\|_{\mathsf{H}_{n_i}}\right]=0. \tag{57}$$

Therefore, we have (52). If (A1)' holds, then Lemma 4.1 implies that, for all $n\in\mathbb{N}$,

$$\mathbb{E}\left[\left\|y_n-Q_{\mathsf{H}_n}(x_n)\right\|_{\mathsf{H}_n}\right]\leq h_\star\tilde{M}\lambda_n,$$

which implies that $\lim_{n\to+\infty}\mathbb{E}[\|y_n-Q_{\mathsf{H}_n}(x_n)\|_{\mathsf{H}_n}]=0$. In (Case 1), (56) and the triangle inequality mean that $\lim_{n\to+\infty}\mathbb{E}[\|x_n-y_n\|_{\mathsf{H}_n}]=0$. Accordingly, the triangle inequality and $\lim_{n\to+\infty}\mathbb{E}[\|y_n-Q_{\mathsf{H}_n}(x_n)\|_{\mathsf{H}_n}]=0$ imply that $\lim_{n\to+\infty}\mathbb{E}[\|x_n-Q_{\mathsf{H}_n}(x_n)\|_{\mathsf{H}_n}]=0$. In (Case 2), (57) and the triangle inequality mean that $\lim_{i\to+\infty}\mathbb{E}[\|x_{n_i}-y_{n_i}\|_{\mathsf{H}_{n_i}}]=0$. Accordingly, the triangle inequality and $\lim_{i\to+\infty}\mathbb{E}[\|y_{n_i}-Q_{\mathsf{H}_{n_i}}(x_{n_i})\|_{\mathsf{H}_{n_i}}]=0$ imply that $\lim_{i\to+\infty}\mathbb{E}[\|x_{n_i}-Q_{\mathsf{H}_{n_i}}(x_{n_i})\|_{\mathsf{H}_{n_i}}]=0$. Thus, we have that

$$\liminf_{n\to+\infty}\mathbb{E}\left[\left\|x_n-Q_{\mathsf{H}_n}(x_n)\right\|_{\mathsf{H}_n}\right]=0.$$

Next, we show (53). Lemma 4.1, together with (36) and (37), ensures that, for all $x^\star \in X^\star$ and all $k \in \mathbb{N}$,

$$2(1-\alpha_k)(1-\beta_k)\lambda_k \mathbb{E}\big[f(x_k) - f^\star\big] \leq \chi_k^\star - \chi_{k+1}^\star + D\mathbb{E}\left[\sum_{i=1}^{N}(h_{k+1,i} - h_{k,i})\right] + 2\tilde{M}\sqrt{DN}\beta_k\lambda_k + h_\star^2\tilde{M}^2\lambda_k^2,$$

where $\chi_n^\star := \chi_n(x^\star)$ for all $x^\star \in X^\star$ and all $n \in \mathbb{N}$. Summing the above inequality from $k = 0$ to $k = n$ gives that, for all $n \in \mathbb{N}$,

$$2\sum_{k=0}^{n}(1-\alpha_k)(1-\beta_k)\lambda_k \mathbb{E}\big[f(x_k) - f^\star\big] \leq \chi_0^\star + D\mathbb{E}\left[\sum_{i=1}^{N} h_{n+1,i}\right] + 2\tilde{M}\sqrt{DN}\sum_{k=0}^{n}\beta_k\lambda_k + h_\star^2\tilde{M}^2\sum_{k=0}^{n}\lambda_k^2,$$

which, together with (40) and (51), implies that

$$\sum_{k=0}^{+\infty}(1-\alpha_k)(1-\beta_k)\lambda_k \mathbb{E}\big[f(x_k) - f^\star\big] < +\infty.$$

If (53) does not hold, then there exist $\zeta > 0$ and $m_1 \in \mathbb{N}$ such that, for all $k \geq m_1$, $\mathbb{E}[f(x_k) - f^\star] \geq \zeta$. Hence, we have that

$$+\infty = \zeta\sum_{k=0}^{+\infty}(1-\alpha_k)(1-\beta_k)\lambda_k \leq \sum_{k=0}^{+\infty}(1-\alpha_k)(1-\beta_k)\lambda_k \mathbb{E}\big[f(x_k) - f^\star\big] < +\infty,$$

where the first equation comes from $\limsup_{n\to+\infty}\alpha_n < 1$, $\sum_{n=0}^{+\infty}\lambda_n = +\infty$, and $\sum_{n=0}^{+\infty}\beta_n\lambda_n < +\infty$ (by (29) and (51)). Since we have a contradiction, (53) holds. Theorem 5.1, together with (40) and (54), ensures that

$$\limsup_{n\to+\infty}\mathbb{E}\big[f(\tilde{x}_n) - f^\star\big] \leq 0 \quad \text{and} \quad \lim_{n\to+\infty}\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\big[\|x_k - Q_{\mathsf{H}_k}(x_k)\|_{\mathsf{H}_k}^2\big] = 0$$

with the convergence rate in Theorem 5.3. □

Theorem 5.3 leads to the following corollary.

Corollary 5.1 *Suppose that the assumptions in Theorem 5.3 and (A1)' hold, and consider Algorithm 1 with* $\lambda_n := 1/n^\eta$ $(\eta \in [1/2, 1])$ *and* $(\beta_n)_{n\in\mathbb{N}}$ *such that* $\sum_{n=1}^{+\infty}\beta_n < +\infty$. *Under* $\eta \in (1/2, 1]$, *we have that*

$$\liminf_{n\to+\infty}\mathbb{E}\big[f(x_n) - f^\star\big] \leq 0, \qquad \liminf_{n\to+\infty}\mathbb{E}\big[\|x_n - Q_{\mathsf{H}_n}(x_n)\|_{\mathsf{H}_n}\big] = 0.$$

Under $\eta \in [1/2, 1)$, *we have that*

$$\limsup_{n\to+\infty}\mathbb{E}\big[f(\tilde{x}_n) - f^\star\big] \leq 0, \qquad \lim_{n\to+\infty}\frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\big[\|x_k - Q_{\mathsf{H}_k}(x_k)\|_{\mathsf{H}_k}^2\big] = 0$$

with the rate of convergence

$$\mathbb{E}\big[f(\tilde{x}_n)-f^\star\big]\leq\mathcal{O}\left(\frac{1}{n^{1-\eta}}\right),\qquad \frac{1}{n}\sum_{k=1}^{n}\mathbb{E}\big[\big\|x_k-Q_{\mathsf{H}_k}(x_k)\big\|_{\mathsf{H}_k}^2\big]=\mathcal{O}\left(\frac{1}{n^{\eta}}\right).$$

Proof The step-size $\lambda_n := 1/n^\eta$ ($\eta \in (1/2, 1]$) and $(\beta_n)_{n\in\mathbb{N}}$ such that $\sum_{n=1}^{+\infty}\beta_n < +\infty$ satisfy (51). Accordingly, Theorem 5.3 with (A1)' implies that $\liminf_{n\to+\infty}\mathbb{E}[f(x_n)-f^\star]\leq 0$, and $\liminf_{n\to+\infty}\mathbb{E}[\|x_n-Q_{\mathsf{H}_n}(x_n)\|_{\mathsf{H}_n}]=0$. The step-size $\lambda_n := 1/n^\eta$ ($\eta \in [1/2, 1)$) satisfies

$$\lim_{n\to+\infty}\frac{1}{n\lambda_n}=\lim_{n\to+\infty}\frac{1}{n^{1-\eta}}=0.$$

Moreover, we have that

$$\frac{1}{n}\sum_{k=1}^{n}\lambda_k^2\leq\frac{1}{n}\sum_{k=1}^{n}\lambda_k\leq\frac{1}{n}\left\{1+\int_1^n\frac{\mathrm{d}t}{t^\eta}\right\}=\frac{1}{n}\left\{\frac{n^{1-\eta}}{1-\eta}-\frac{\eta}{1-\eta}\right\}\leq\frac{1}{1-\eta}\frac{1}{n^\eta}. \tag{58}$$

Hence, $\lim_{n\to+\infty}(1/n)\sum_{k=1}^{n}\lambda_k=\lim_{n\to+\infty}(1/n)\sum_{k=1}^{n}\lambda_k^2=0$. The condition $\sum_{n=1}^{+\infty}\beta_n<+\infty$ implies that $\lim_{n\to+\infty}(1/n)\sum_{k=1}^{n}\beta_k=0$ and $\lim_{n\to+\infty}(1/n)\sum_{k=1}^{n}\beta_k\lambda_k=0$. Hence, (54) is satisfied. Accordingly, from Theorem 5.3 with (A1)' and (58), we have the convergence rate of Algorithm 1 in Corollary 5.1. □

5.2 Comparisons of Algorithm 1 with the existing adaptive learning rate optimization algorithms

The main objective of the existing adaptive learning rate optimization algorithms is to minimize $\sum_{t=1}^{T}f_t(x)$ subject to $x\in X$, where T is the total number of rounds in the learning process, $f_t:\mathbb{R}^N\to\mathbb{R}$ ($t=1,2,\ldots,T$) is a differentiable, convex loss function, and $X\subset\mathbb{R}^N$ is bounded, closed, and convex (see also problem (21) in Example 4.1(i)). We would like to achieve low regret on the sequence $(f_t(x_t))_{t=1}^T$, measured as

$$R(T):=\sum_{t=1}^{T}f_t(x_t)-\min_{x\in X}\sum_{t=1}^{T}f_t(x)=\sum_{t=1}^{T}f_t(x_t)-\sum_{t=1}^{T}f_t\big(x^*\big),$$

where $x^*\in X$ is a minimizer of $\sum_{t=1}^{T}f_t(x)$ over X, and $(x_t)_{t=1}^T\subset X$ is the sequence generated by a learning algorithm. Although Theorem 4.1 in [8] indicates that Adam [8, Algorithm 1], [2, Algorithm 8.7] (algorithm (6)) is such that there exists a positive real number D such that $R(T)/T\leq D/\sqrt{T}$, the proof of Theorem 4.1 in [8] is incomplete [9, Theorem 1]. AMSGrad [9, Algorithm 2] (algorithm (9)) is such that the following result holds [9, Theorem 4, Corollary 1]: Suppose that $\beta_{1,t}:=\beta_1\lambda^{t-1}$ ($\beta_1,\lambda\in(0,1)$), $\gamma:=\beta_1/\sqrt{\beta_2}<1$, and $\lambda_t:=\alpha/\sqrt{t}$ ($\alpha>0$). Then there exist positive real numbers $\hat{D}_i$ ($i=1,2,3$) such that

$$\begin{aligned}\frac{R(T)}{T}&=\frac{1}{T}\sum_{t=1}^{T}f_t(x_t)-\frac{1}{T}\sum_{t=1}^{T}f_t\big(x^*\big)\\&\leq\frac{\hat{D}_1N}{\alpha\tilde{\beta}_1\sqrt{T}}+\frac{\beta_1\hat{D}_2}{2\tilde{\beta}_1(1-\lambda)^2T}+\frac{\alpha\sqrt{1+\ln T}}{\tilde{\beta}_1{}^2(1-\gamma)\sqrt{1-\beta_2}T}\sum_{i=1}^{N}\|g_{1:T,i}\|,\end{aligned}$$

where $\tilde{\beta}_1 := 1-\beta_1$, $g_t := \nabla_x F(x_t, \xi_t)$, [6] and $\|g_{1:T,i}\| := \sqrt{\sum_{t=1}^{T} g_{t,i}^2} \leq \hat{D}_3\sqrt{T}$. Hence, with AMSGrad, there exists a positive real number $\hat{D}$ such that

$$\frac{R(T)}{T} = \frac{1}{T}\sum_{t=1}^{T} f_t(x_t) - \frac{1}{T}\sum_{t=1}^{T} f_t(x^*) \leq \hat{D}\sqrt{\frac{1+\ln T}{T}}. \tag{59}$$

We apply Algorithm 1 with $\lambda_n := 1/n^\eta$ ($\eta \in [1/2, 1)$) (see also algorithm (22)) to Problem 3.1 for the special case where $f(\cdot) = \mathbb{E}[f_\xi(\cdot)] := (1/T)\sum_{t=1}^{T} f_t(\cdot)$, $Q_{\mathsf{H}_n} := P_{X,\mathsf{H}_n}$ ($n \in \mathbb{N}$), H_n is defined by either (19) or (20), and $C = X$ (see also problem (21)). Then Theorem 5.2 has the following corollary.

Corollary 5.2 *Consider problem* (21) *and suppose that the assumptions in Theorem* 5.1 *hold. Then algorithm* (22) *satisfies that*

$$\liminf_{n\to+\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f_t(x_n) - \frac{1}{T}\sum_{t=1}^{T} f_t(x^*)\right] \leq \frac{\tilde{M}\sqrt{DN}}{1-\beta}\beta + \frac{h_\star^2\tilde{M}^2}{2(1-\beta)}\lambda,$$
$$\limsup_{n\to+\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} f_t(\tilde{x}_n) - \frac{1}{T}\sum_{t=1}^{T} f_t(x^*)\right] \leq \frac{\tilde{M}\sqrt{DN}}{1-\beta}\beta + \frac{h_\star^2\tilde{M}^2}{2(1-\beta)}\lambda,$$

where $\tilde{x}_n := (1/n)\sum_{k=1}^{n} x_k$ *and* $(x_n)_{n\in\mathbb{N}} \subset X$ *is the sequence in algorithm* (22).

In contrast to Adam and AMSGrad with diminishing step-sizes, Corollary 5.2 indicates that algorithm (22) with constant step-sizes may approximate a solution of problem (21).

Corollary 5.1 implies the following corollary.

Corollary 5.3 *Suppose that the assumptions in Corollary* 5.1 *hold and* $\lambda_n := 1/n^\eta$ ($\eta \in [1/2, 1]$), *and* $(\beta_n)_{n\in\mathbb{N}}$ *is such that* $\sum_{n=1}^{+\infty} \beta_n < +\infty$. *Under* $\eta \in (1/2, 1]$, *algorithm* (22) *satisfies that*

$$\liminf_{n\to+\infty} \mathbb{E}\left[\sum_{t=1}^{T} f_t(x_n) - \sum_{t=1}^{T} f_t(x^*)\right] = 0.$$

Moreover, under $\eta \in [1/2, 1)$, *any accumulation point of* $(\tilde{x}_n := (1/n)\sum_{k=1}^{n} x_k)_{n\in\mathbb{N}}$ *almost surely belongs to the solution set of problem* (21), *and algorithm* (22) *achieves the following rate of convergence*:

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(\tilde{x}_n) - \sum_{t=1}^{T} f_t(x^*)\right] = \mathcal{O}\left(\frac{1}{n^{1-\eta}}\right).$$

Proof For problem (21), Corollary 5.3 implies that $0 \leq \liminf_{n\to+\infty} \mathbb{E}[f(x_n) - f^\star] \leq 0$ and $0 \leq \limsup_{n\to+\infty} \mathbb{E}[f(\tilde{x}_n) - f^\star] \leq 0$, where $f := (1/T)\sum_{t=1}^{T} f_t$. The second inequality guarantees that $\lim_{n\to+\infty} \mathbb{E}[f(\tilde{x}_n) - f^\star] = 0$. Let $\hat{x} \in X$ be an arbitrary accumulation point of $(\tilde{x}_n)_{n\in\mathbb{N}} \subset X$. Since there exists $(\tilde{x}_{n_i})_{i\in\mathbb{N}} \subset (\tilde{x}_n)_{n\in\mathbb{N}}$ such that $(\tilde{x}_{n_i})_{i\in\mathbb{N}}$ converges almost surely

[6] Since AMSGrad is applied to constrained convex optimization, in general, $\lim_{T\to+\infty} \|g_{1:T,i}\| \neq 0$ and $\|g_{1:T,i}\| \leq \hat{D}_3\sqrt{T}$ hold [8, Corollary 4.2].

to $\hat{x} \in X$, the continuity of f ensures that $0 = \lim_{i\to+\infty} \mathbb{E}[f(\tilde{x}_{n_i}) - f^\star] = \mathbb{E}[f(\hat{x}) - f^\star]$, i.e., $\hat{x} \in X^\star$. The rate of convergence of $(\tilde{x}_n)_{n\in\mathbb{N}}$ is obtained from Corollary 5.1. □

It is not guaranteed that x_T defined by AMSGrad with $\lambda_t := \alpha/\sqrt{t}$ optimizes $\sum_{t=1}^T f_t$ over X since (59) depends on a given parameter T, i.e.,

$$\frac{R(T)}{T} \leq \mathcal{O}\left(\sqrt{\frac{1+\ln T}{T}}\right).$$

Meanwhile, Corollary 5.3 implies that any accumulation point of $(\tilde{x}_n)_{n\in\mathbb{N}}$ defined by algorithm (22) with $\lambda_n := 1/\sqrt{n}$ almost surely belongs to the set of minimizers of $\sum_{t=1}^T f_t$ over X and $(\tilde{x}_n)_{n\in\mathbb{N}}$ achieves an $\mathcal{O}(1/\sqrt{n})$ convergence rate, i.e.,

$$\mathbb{E}\left[\sum_{t=1}^T f_t(\tilde{x}_n) - \sum_{t=1}^T f_t(x^*)\right] = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right).$$

5.3 Numerical comparisons

In this section, we consider the classifier ensemble problem [18, Sect. 2.2.2], [19, Sect. 3.2.2], [17, Problem II.1] (see problems (23) and (25) in Example 4.1 (ii)) and compare the performances of the learning methods based on the following algorithms which used commonly $\beta = 0.99$ [9, Sect. 5] and $\alpha_n = 1/2$ ($n \in \mathbb{N}$).

- SG: Stochastic gradient algorithm (15) with $\lambda_n \in [10^{-3}/(n+1), 1/(n+1)]$ computed by the Armijo line search algorithm [17, Algorithms 2 and 3, LS].
- C1: Algorithm 1 with (19) and $\beta_n = \lambda_n = 10^{-1}$.
- C2: Algorithm 1 with (19) and $\beta_n = \lambda_n = 10^{-3}$.
- C3: Algorithm 1 with (20) and $\beta_n = \lambda_n = 10^{-1}$.
- C4: Algorithm 1 with (20) and $\beta_n = \lambda_n = 10^{-3}$.
- D1: Algorithm 1 with (19), $\beta_n = 0.9/2^n$, and $\lambda_n = 10^{-1}/\sqrt{n+1}$.
- D2: Algorithm 1 with (19), $\beta_n = 0.9/2^n$, and $\lambda_n = 10^{-3}/\sqrt{n+1}$.
- D3: Algorithm 1 with (19), $\beta_n = 0.9/2^n$, and $\lambda_n \in [10^{-3}/\sqrt{n+1}, 1/\sqrt{n+1}]$ computed by the Armijo line search algorithm.
- D4: Algorithm 1 with (20), $\beta_n = 0.9/2^n$, and $\lambda_n = 10^{-1}/\sqrt{n+1}$.
- D5: Algorithm 1 with (20), $\beta_n = 0.9/2^n$, and $\lambda_n = 10^{-3}/\sqrt{n+1}$.
- D6: Algorithm 1 with (20), $\beta_n = 0.9/2^n$, and $\lambda_n \in [10^{-3}/\sqrt{n+1}, 1/\sqrt{n+1}]$ computed by the Armijo line search algorithm.

The step-size $\beta_n := 0.9/2^n$ used in D1–D6 was based on [9, Sect. 5]. The numerical results in [17] showed that the learning method based on SG performed better than the existing methods in [19, (18)]. Therefore, we compare the performance of the learning method based on SG with the one of the learning methods based on C1–D6. See Corollary 1 in [17], Theorems 5.2 and 5.3, and Corollary 5.1 for convergence analyses of the above algorithms for solving problems (23) and (25).

The experiments used Mac Pro (Late 2013) with a 3.5 GHz 6-core Intel Xeon E5 CPU, 32 GB 1866 MHz DDR3 memory, and macOS Catalina version 10.15.1 operating system. The algorithms used in the experiments were written in Python 3.7.5 with the NumPy 1.17.4 package. The experiments used the datasets from LIBSVM [37] and the UCI Machine Learning Repository [38] for which information is shown in Table 1. In these experiments, stratified 10-fold cross-validation for the datasets was performed. For this validation, the

Table 1 Datasets used for classification

Dataset		Classes	Instances	Attributes
1.	australian	2	690	14
2.	breast-cancer	2	683	10
3.	diabetes	2	768	8
4.	ionosphere	2	351	34
5.	leukemia	2	72	7129
6.	madelon	2	2600	500
7.	splice	2	3175	60
8.	iris	3	150	4
9.	svmguide2	3	391	20
10.	wine	3	178	13
11.	vehicle	4	846	18
12.	glass	6	214	9
13.	segment	7	2310	19
14.	digits	10	1797	64
15.	usps	10	9298	256

`StratifiedKFold` class in the scikit-learn 0.21.3 package was used. Ensembles of support vector classifiers were constructed by the `BaggingClassifier` class in the scikit-learn 0.21.3 package. The number of base estimators was set as the default value of the scikit-learn package. For learning multiclass classification tasks with the classifiers used in the experiments, the one-vs-the-rest multiclass classification strategy implemented as the `OneVsRestClassifier` class in the scikit-learn 0.21.3 package was used. The stopping condition for the algorithms used in the experiments was $n = 100$.

Let us consider problem (23) and compare the performances of the sparsity learning methods based on the algorithms with Q_{H_n} defined by (24). Although we can consider problem (25) and compare the performances of the sparsity and diversity learning methods based on the algorithms with Q_{H_n} defined by (26), we omit the details due to lack of space.[7]

Tables 2 and 3 show that the accuracy of the learning method based on SG was almost the same as that of the learning methods based on C1, C2, C3, C4, D3, D4, and D6. These tables also show that the elapsed times for the proposed learning methods were shorter than the elapsed times for the learning method based on SG.

The average accuracies and elapsed times of the existing learning method (SG) were compared to the average accuracies and elapsed times of the proposed learning methods (C1–D6) by using an analysis of variance (ANOVA) test and Tukey–Kramer's honestly significant difference (HSD) test. The `scipy.stats.f_oneway` method in the SciPy library was used as the implementation of the ANOVA test, and the `statsmodels.stats.multicomp.pairwise_tukeyhsd` method in the StatsModels package was used as the implementation of Tukey–Kramer's HSD test. Recall that the ANOVA test examines whether the hypothesis that the given groups have the same population mean is rejected, whereas Tukey–Kramer's HSD test can be used to find specifically which pair has a significant difference in groups. The significance level was set at 5% (0.05) for the ANOVA and Tukey–Kramer's HSD tests. The p-value computed by the ANOVA test for the accuracies was about 4.09×10^{-19} (< 0.05). Table 4 indicates that the adjusted p-value between each of the learning methods based on C1, C2, C3, C4, D3, D4, and D6 and the

[7] We checked that the sparsity and diversity learning methods based on C1, C2, C3, C4, D3, D4, and D6 with Q_{H_n} defined by (26) perform better than the learning method based on SG, as seen in the results (Tables 2, 3, 4, and 5) for ensemble learning with sparsity.

Table 2 Classification accuracies (%) and elapsed times (s) for the sparsity learning methods based on SG, C1, C2, C3, and C4 applied to the datasets in Table 1

#	*SG*		*C1*		*C2*		*C3*		*C4*	
	acc.	time	acc.	time	acc.	time	acc.	time	acc.	time
1	80.59	0.531	81.74	0.206	83.34	0.197	84.20	0.208	83.92	0.213
2	95.52	0.499	94.45	0.203	94.01	0.206	94.44	0.205	93.43	0.209
3	65.10	0.510	64.06	0.205	63.15	0.205	63.41	0.210	63.67	0.211
4	71.29	0.433	74.78	0.206	71.03	0.209	72.14	0.210	71.03	0.212
5	75.16	39.848	48.16	9.405	68.66	9.402	57.83	9.364	75.16	9.246
6	50.00	4.107	48.65	0.801	50.05	0.815	50.30	0.819	49.95	0.805
7	45.70	0.697	46.68	0.220	43.89	0.219	43.89	0.224	42.19	0.230
8	87.33	0.916	83.33	0.601	81.33	0.599	82.66	0.605	82.00	0.615
9	56.54	1.080	56.54	0.605	56.54	0.616	40.63	0.632	13.53	0.626
10	96.72	1.015	96.72	1.015	89.91	0.613	92.13	0.621	91.09	0.625
11	45.89	2.236	48.05	0.806	44.44	0.829	42.68	0.850	43.97	0.856
12	42.17	2.111	46.77	1.211	46.22	1.201	46.67	1.238	45.43	1.226
13	68.05	7.326	75.06	1.517	72.94	1.500	72.98	1.521	71.60	1.532
14	70.24	10.197	66.78	2.298	65.58	2.278	75.62	2.358	40.73	2.303
15	60.91	95.861	64.99	11.571	71.20	11.594	58.69	11.604	69.95	11.611
Ave.	67.41	11.158	66.04	2.030	66.82	2.032	65.22	2.045	62.51	2.035

Table 3 Classification accuracies (%) and elapsed times (s) for the sparsity learning methods based on D1, D2, D3, D4, D5, and D6 applied to the datasets in Table 1

#	*D1*		*D2*		*D3*		*D4*		*D5*		*D6*	
	acc.	time	acc.	time	acc.	time	acc.	time	acc.	time	acc.	time
1	77.84	0.210	82.75	0.207	83.92	0.298	82.47	0.210	83.33	0.213	83.78	0.229
2	95.52	0.180	89.76	0.206	94.44	0.287	93.57	0.206	91.81	0.208	94.15	0.254
3	27.86	0.202	51.17	0.206	64.32	0.280	56.76	0.212	59.11	0.209	64.06	0.237
4	76.45	0.187	71.03	0.200	71.58	0.312	71.32	0.213	71.01	0.212	71.86	0.267
5	39.00	9.383	54.00	9.365	46.16	9.697	51.5	9.525	66.16	9.584	68.66	10.190
6	49.90	0.795	51.35	0.822	50.20	1.068	50.8	0.805	49.65	0.849	50.00	0.974
7	43.49	0.222	43.08	0.225	43.60	0.352	44.39	0.223	42.49	0.229	43.48	0.298
8	63.33	0.607	74.66	0.600	84.66	0.780	77.33	0.621	78.66	0.613	81.33	0.690
9	25.01	0.615	39.24	0.612	56.54	0.722	16.79	0.625	23.28	0.629	56.54	0.694
10	62.47	0.592	69.50	0.603	91.55	0.823	88.71	0.630	94.53	0.616	91.65	0.717
11	29.28	0.841	32.14	0.829	40.94	1.150	40.08	0.835	37.49	0.843	43.86	1.006
12	22.38	1.221	25.62	1.205	45.80	1.617	31.02	1.234	33.95	1.246	49.02	1.469
13	50.95	1.497	41.47	1.507	72.25	2.182	67.44	1.527	53.03	1.527	76.66	1.937
14	64.78	2.304	34.18	2.322	66.33	3.319	74.17	2.356	37.78	2.358	66.40	3.079
15	32.06	11.604	46.01	11.585	67.63	13.472	62.63	11.620	55.46	11.671	66.20	13.259
Ave.	50.69	2.031	53.73	2.033	65.33	2.424	60.60	2.056	58.52	2.067	67.18	2.353

existing learning method based on SG was greater than 0.05. This implies that the existing and proposed methods based on C1, C2, C3, C4, D3, D4, and D6 had almost the same performances in the sense of accuracy. The p-value computed by the ANOVA test for the elapsed time was about 2.67×10^{-29} (< 0.05). Table 5 indicates that there is a significant difference in the sense of the elapsed time between each of the proposed methods and the existing method based on SG. Therefore, the proposed methods ran significantly faster than the existing method based on SG.

6 Conclusion

In this paper, we proposed a stochastic approximation method based on adaptive learning rate optimization algorithms for solving a convex stochastic optimization problem over the fixed point set of a quasinonexpansive mapping. It also presented convergence analyses

Table 4 Multiple comparison for accuracies for the sparsity learning methods applied to the datasets in Table 1 using Tukey–Kramer's HSD test at the 5% significance level ("meandiffs" indicates the pairwise mean differences between Groups 1 and 2, "*p*-adj" indicates the adjusted *p*-value, and "Lower" (resp. "Upper") indicates the lower (resp. upper) value of the confidence interval for the pairwise mean differences)

Group 1	Group 2	meandiffs	*p*-adj	Lower	Upper	Reject
C1	C2	0.7823	0.9	−6.969	8.5335	FALSE
C1	C3	−0.8189	0.9	−8.5702	6.9323	FALSE
C1	C4	−3.5273	0.9	−11.2785	4.2239	FALSE
C1	D1	−15.4512	0.001	−23.2024	−7.6999	TRUE
C1	D2	−12.3071	0.001	−20.0583	−4.5559	TRUE
C1	D3	−0.7095	0.9	−8.4607	7.0417	FALSE
C1	D4	−5.4384	0.4642	−13.1897	2.3128	FALSE
C1	D5	−7.5201	0.0668	−15.2713	0.2311	FALSE
C1	D6	1.1391	0.9	−6.6122	8.8903	FALSE
C1	SG	1.3916	0.9	−6.3596	9.1428	FALSE
C2	C3	−1.6012	0.9	−9.3524	6.15	FALSE
C2	C4	−4.3096	0.7575	−12.0608	3.4416	FALSE
C2	D1	−16.2334	0.001	−23.9847	−8.4822	TRUE
C2	D2	−13.0894	0.001	−20.8406	−5.3382	TRUE
C2	D3	−1.4918	0.9	−9.243	6.2594	FALSE
C2	D4	−6.2207	0.2564	−13.9719	1.5305	FALSE
C2	D5	−8.3023	0.0241	−16.0536	−0.5511	TRUE
C2	D6	0.3568	0.9	−7.3944	8.108	FALSE
C2	SG	0.6093	0.9	−7.1419	8.3605	FALSE
C3	C4	−2.7084	0.9	−10.4596	5.0428	FALSE
C3	D1	−14.6322	0.001	−22.3834	−6.881	TRUE
C3	D2	−11.4882	0.001	−19.2394	−3.737	TRUE
C3	D3	0.1094	0.9	−7.6418	7.8606	FALSE
C3	D4	−4.6195	0.6775	−12.3707	3.1317	FALSE
C3	D5	−6.7011	0.1642	−14.4524	1.0501	FALSE
C3	D6	1.958	0.9	−5.7932	9.7092	FALSE
C3	SG	2.2105	0.9	−5.5407	9.9617	FALSE
C4	D1	−11.9238	0.001	−19.6751	−4.1726	TRUE
C4	D2	−8.7798	0.0121	−16.531	−1.0286	TRUE
C4	D3	2.8178	0.9	−4.9334	10.569	FALSE
C4	D4	−1.9111	0.9	−9.6623	5.8401	FALSE
C4	D5	−3.9928	0.8393	−11.744	3.7585	FALSE
C4	D6	4.6664	0.6654	−3.0848	12.4176	FALSE
C4	SG	4.9189	0.6002	−2.8323	12.6701	FALSE
D1	D2	3.144	0.9	−4.6072	10.8953	FALSE
D1	D3	14.7416	0.001	6.9904	22.4929	TRUE
D1	D4	10.0127	0.0016	2.2615	17.7639	TRUE
D1	D5	7.9311	0.0398	0.1799	15.6823	TRUE
D1	D6	16.5902	0.001	8.839	24.3414	TRUE
D1	SG	16.8427	0.001	9.0915	24.594	TRUE
D2	D3	11.5976	0.001	3.8464	19.3488	TRUE
D2	D4	6.8687	0.1379	−0.8825	14.6199	FALSE
D2	D5	4.787	0.6343	−2.9642	12.5383	FALSE
D2	D6	13.4462	0.001	5.6949	21.1974	TRUE
D2	SG	13.6987	0.001	5.9475	21.4499	TRUE
D3	D4	−4.7289	0.6493	−12.4801	3.0223	FALSE
D3	D5	−6.8106	0.1467	−14.5618	0.9407	FALSE
D3	D6	1.8486	0.9	−5.9027	9.5998	FALSE
D3	SG	2.1011	0.9	−5.6501	9.8523	FALSE
D4	D5	−2.0816	0.9	−9.8329	5.6696	FALSE
D4	D6	6.5775	0.1849	−1.1737	14.3287	FALSE
D4	SG	6.83	0.1437	−0.9212	14.5812	FALSE
D5	D6	8.6591	0.0145	0.9079	16.4104	TRUE
D5	SG	8.9117	0.0099	1.1604	16.6629	TRUE
D6	SG	0.2525	0.9	−7.4987	8.0037	FALSE

Table 5 Multiple comparison for elapsed time for the sparsity learning methods applied to the datasets in Table 1 using Tukey–Kramer's HSD test at the 5% significance level ("meandiffs" indicates the pairwise mean differences between Groups 1 and 2, "*p*-adj" indicates the adjusted *p*-value, and "Lower" (resp. "Upper") indicates the lower (resp. upper) value of the confidence interval for the pairwise mean differences)

Group 1	Group 2	meandiffs	*p*-adj	Lower	Upper	Reject
C1	C2	0.0019	0.9	–3.0351	3.0389	FALSE
C1	C3	0.0142	0.9	–3.0227	3.0512	FALSE
C1	C4	0.0043	0.9	–3.0327	3.0413	FALSE
C1	D1	0.0003	0.9	–3.0367	3.0372	FALSE
C1	D2	0.0026	0.9	–3.0344	3.0395	FALSE
C1	D3	0.3937	0.9	–2.6433	3.4307	FALSE
C1	D4	0.0258	0.9	–3.0111	3.0628	FALSE
C1	D5	0.0366	0.9	–3.0003	3.0736	FALSE
C1	D6	0.323	0.9	–2.714	3.3599	FALSE
C1	SG	9.1275	0.001	6.0905	12.1645	*TRUE*
C2	C3	0.0123	0.9	–3.0246	3.0493	FALSE
C2	C4	0.0024	0.9	–3.0346	3.0394	FALSE
C2	D1	–0.0016	0.9	–3.0386	3.0353	FALSE
C2	D2	0.0007	0.9	–3.0363	3.0376	FALSE
C2	D3	0.3918	0.9	–2.6452	3.4288	FALSE
C2	D4	0.0239	0.9	–3.013	3.0609	FALSE
C2	D5	0.0347	0.9	–3.0022	3.0717	FALSE
C2	D6	0.3211	0.9	–2.7159	3.358	FALSE
C2	SG	9.1256	0.001	6.0886	12.1626	*TRUE*
C3	C4	–0.0099	0.9	–3.0469	3.027	FALSE
C3	D1	–0.014	0.9	–3.051	3.023	FALSE
C3	D2	–0.0117	0.9	–3.0486	3.0253	FALSE
C3	D3	0.3795	0.9	–2.6575	3.4164	FALSE
C3	D4	0.0116	0.9	–3.0254	3.0485	FALSE
C3	D5	0.0224	0.9	–3.0146	3.0593	FALSE
C3	D6	0.3087	0.9	–2.7282	3.3457	FALSE
C3	SG	9.1132	0.001	6.0763	12.1502	*TRUE*
C4	D1	–0.004	0.9	–3.041	3.0329	FALSE
C4	D2	–0.0017	0.9	–3.0387	3.0352	FALSE
C4	D3	0.3894	0.9	–2.6476	3.4264	FALSE
C4	D4	0.0215	0.9	–3.0155	3.0585	FALSE
C4	D5	0.0323	0.9	–3.0046	3.0693	FALSE
C4	D6	0.3187	0.9	–2.7183	3.3556	FALSE
C4	SG	9.1232	0.001	6.0862	12.1602	*TRUE*
D1	D2	0.0023	0.9	–3.0347	3.0393	FALSE
D1	D3	0.3935	0.9	–2.6435	3.4304	FALSE
D1	D4	0.0256	0.9	–3.0114	3.0625	FALSE
D1	D5	0.0364	0.9	–3.0006	3.0733	FALSE
D1	D6	0.3227	0.9	–2.7143	3.3597	FALSE
D1	SG	9.1272	0.001	6.0903	12.1642	*TRUE*
D2	D3	0.3911	0.9	–2.6458	3.4281	FALSE
D2	D4	0.0232	0.9	–3.0137	3.0602	FALSE
D2	D5	0.0341	0.9	–3.0029	3.071	FALSE
D2	D6	0.3204	0.9	–2.7166	3.3574	FALSE
D2	SG	9.1249	0.001	6.088	12.1619	*TRUE*
D3	D4	–0.3679	0.9	–3.4049	2.6691	FALSE
D3	D5	–0.3571	0.9	–3.3941	2.6799	FALSE
D3	D6	–0.0707	0.9	–3.1077	2.9662	FALSE
D3	SG	8.7338	0.001	5.6968	11.7707	*TRUE*
D4	D5	0.0108	0.9	–3.0262	3.0478	FALSE
D4	D6	0.2972	0.9	–2.7398	3.3341	FALSE
D4	SG	9.1017	0.001	6.0647	12.1386	*TRUE*
D5	D6	0.2863	0.9	–2.7506	3.3233	FALSE
D5	SG	9.0909	0.001	6.0539	12.1278	*TRUE*
D6	SG	8.8045	0.001	5.7676	11.8415	*TRUE*

of the proposed method with constant and diminishing step-sizes. The analyses confirm that any accumulation point of the sequence generated by the proposed method almost surely belongs to the solution set of the stochastic optimization problem in deep learning. We also compared the proposed algorithm with the existing adaptive learning rate optimization algorithms and showed that the proposed algorithm achieved an $\mathcal{O}(1/\sqrt{n})$ convergence rate which was not achieved for the existing adaptive learning rate optimization algorithms. Numerical results for the classifier ensemble problems demonstrated that the proposed learning methods achieve high accuracies faster than the existing learning method based on the first-order algorithm. In particular, the proposed methods with constant step-sizes or Armijo line search step-sizes solve the classifier ensemble problems faster than the existing method based on the first-order algorithm.

Acknowledgements
The author would like to thank Professor Heinz Bauschke, Professor Yunier Bello-Cruz, Professor Radu Ioan Bot, Professor Robert Csetnek, and Professor Alexander Zaslavski for giving him a chance to submit his paper to this special issue. The author is sincerely grateful to Editor-in-Chief Yunier Bello-Cruz and the two anonymous reviewers for helping him improve the original manuscript. The author thanks Hiroyuki Sakai for his input on the numerical examples.

Funding
This work was supported by the Japan Society for the Promotion of Science (JSPS KAKENHI Grant Number JP18K11184).

Availability of data and materials
Not applicable.

Competing interests
The author declares that they have no competing interests.

Authors' contributions
HI developed the mathematical methods. HI discussed the results and contributed to the final manuscript. All authors read and approved the final manuscript.

Publisher's Note

Received: 10 December 2020 Accepted: 26 March 2021 Published online: 12 April 2021

References

1. Borkar, V.S.: Stochastic Approximation: A Dynamical Systems Viewpoint. Cambridge University Press, Cambridge (2008)
2. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
3. Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**, 400–407 (1951)
4. Nedić, A., Lee, S.: On stochastic subgradient mirror-descent algorithm with weighted averaging. SIAM J. Optim. **24**, 84–107 (2014)
5. Nemirovski, A., Juditsky, A., Lan, G., Shapiro, A.: Robust stochastic approximation approach to stochastic programming. SIAM J. Optim. **19**, 1574–1609 (2009)
6. Ghadimi, S., Lan, G.: Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: a generic algorithmic framework. SIAM J. Optim. **22**, 1469–1492 (2012)
7. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
8. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations, pp. 1–15 (2015)
9. Reddi, S.J., Kale, S., Kumar, S.: On the convergence of Adam and beyond. In: Proceedings of the International Conference on Learning Representations, pp. 1–23 (2018)
10. Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer, New York (2011)
11. Berinde, V.: Iterative Approximation of Fixed Points. Springer, Berlin (2007)
12. Halpern, B.: Fixed points of nonexpanding maps. Bull. Am. Math. Soc. **73**, 957–961 (1967)
13. Krasnosel'skiĭ, M.A.: Two remarks on the method of successive approximations. Usp. Mat. Nauk **10**, 123–127 (1955)
14. Mann, W.R.: Mean value methods in iteration. Proc. Am. Math. Soc. **4**, 506–510 (1953)
15. Nakajo, K., Takahashi, W.: Strong convergence theorems for nonexpansive mappings and nonexpansive semigroups. J. Math. Anal. Appl. **279**, 372–379 (2003)
16. Wittmann, R.: Approximation of fixed points of nonexpansive mappings. Arch. Math. **58**, 486–491 (1992)
17. Iiduka, H.: Stochastic fixed point optimization algorithm for classifier ensemble. IEEE Trans. Cybern. **50**, 4370–4380 (2020)

18. Yin, X.C., Huang, K., Hao, H.W., Iqbal, K., Wang, Z.B.: A novel classifier ensemble method with sparsity and diversity. Neurocomputing **134**, 214–221 (2014)
19. Yin, X.C., Huang, K., Yang, C., Hao, H.W.: Convex ensemble learning with sparsity and diversity. Inf. Fusion **20**, 49–58 (2014)
20. Rockafellar, R.T.: Convex Analysis. Princeton University Press, Princeton (1970)
21. Borwein, J.M., Lewis, A.S.: Convex Analysis and Nonlinear Optimization: Theory and Examples. Springer, New York (2000)
22. Bauschke, H.H., Combettes, P.L.: A weak-to-strong convergence principle for Fejér-monotone methods in Hilbert space. Math. Oper. Res. **26**, 248–264 (2001)
23. Bauschke, H.H., Chen, J.: A projection method for approximating fixed points of quasi nonexpansive mappings without the usual demiclosedness condition. J. Nonlinear Convex Anal. **15**, 129–135 (2014)
24. Vasin, V.V., Ageev, A.L.: Ill-Posed Problems with a Priori Information. V.S.P. Intl. Science, Utrecht (1995)
25. Shapiro, A., Dentcheva, D., Ruszczyński, A.: Lectures on Stochastic Programming: Modeling and Theory, 2nd edn. MOS-SIAM Series on Optimization. SIAM, Philadelphia (2014)
26. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature **323**, 533–536 (1986)
27. Ghadimi, S., Lan, G.: Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization II: shrinking procedures and optimal algorithms. SIAM J. Optim. **23**, 2061–2089 (2013)
28. Goebel, K., Kirk, W.A.: Topics in Metric Fixed Point Theory. Cambridge Studies in Advanced Mathematics. Cambridge University Press, New York (1990)
29. Goebel, K., Reich, S.: Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings. Dekker, New York (1984)
30. Takahashi, W.: Nonlinear Functional Analysis. Yokohama Publishers, Yokohama (2000)
31. Yamada, I.: The hybrid steepest descent method for the variational inequality problem over the intersection of fixed point sets of nonexpansive mappings. In: Butnariu, D., Censor, Y., Reich, S. (eds.) Inherently Parallel Algorithms for Feasibility and Optimization and Their Applications, pp. 473–504. Elsevier, New York (2001)
32. Yamada, I., Ogura, N.: Hybrid steepest descent method for variational inequality problem over the fixed point set of certain quasi-nonexpansive mappings. Numer. Funct. Anal. Optim. **25**, 619–655 (2004)
33. Horn, R.A., Johnson, C.R.: Matrix Analysis. Cambridge University Press, Cambridge (1985)
34. Wanka, G., Wilfer, O.: Formulae of epigraphical projection for solving minimax location problems. Pac. J. Optim. **16**, 289–313 (2020)
35. Nedić, A., Ozdaglar, A.: Approximate primal solutions and rate analysis for dual subgradient methods. SIAM J. Optim. **19**, 1757–1780 (2009)
36. Iiduka, H.: Distributed optimization for network resource allocation with nonsmooth utility functions. IEEE Trans. Control Netw. Syst. **6**, 1354–1365 (2019)
37. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**, 27 (2011)
38. Dua, D., Graff, C.: UCI Machine learning repository. School Inf. Comput. Sci., Univ. California at Irvine, Irvine, CA, USA (2019)