Stochastic Fixed Point Optimization Algorithm for Classifier Ensemble

Hideaki Iiduka

***Abstract*—This paper considers a classifier ensemble problem with sparsity and diversity learning, which arises in the field of machine learning, and shows that the classifier ensemble problem can be formulated as a convex stochastic optimization problem over the fixed point set of a quasi-nonexpansive mapping. Specifically, for such a problem, the present work proposes an algorithm referred to as the stochastic fixed point optimization algorithm and performs a convergence analysis for three types of step size: constant step size, decreasing step size, and a step size computed by line searches. In the case of a constant step size, the results indicate that a sufficiently small constant step size allows a solution to the problem to be approximated. In the case of a decreasing step size, conditions are shown under which the algorithm converges in probability to a solution. For the third case, a variation of the basic proposed algorithm also achieves convergence in probability to a solution. The high classification accuracies of the proposed algorithms are demonstrated through numerical comparisons with the conventional algorithm.**

***Index Terms*—convex stochastic optimization, classifier ensemble, fixed point, quasi-nonexpansive mapping, sparsity and diversity learning, stochastic fixed point optimization algorithm**

I. Introduction

THE *classifier ensemble problem* (see [1], [2], [3], [4], [5], [6], [7], [8] and references therein) is a significant, interesting problem that arises in the field of machine learning. One way of solving the classifier ensemble problem is to formulate it as a constrained convex optimization problem in which the objective function is the expectation of convex functions [9, Section 3]. Classically, the technique used to solve the problem is the *stochastic approximation* (SA) *method* [10, (5.4.1)], [11], which is applicable when unbiased estimates of (sub)gradients of an objective function are available. The usefulness of the method for stochastic optimization has already been proved [12], [13], [14], [15].

In this paper, we focus on the classifier ensemble problem with *sparsity and diversity learning*. This classifier ensemble problem can be restated as a convex stochastic optimization problem over the intersection of a half-space and the level sets of convex functions [4, (10)], [5, (15)]. Unfortunately, it would be difficult to apply the SA method and its variations to this classifier ensemble problem. This is because the constraint set of the problem is complicated in the sense that the projection onto the constraint set cannot be computed efficiently. The currently used method [5] for solving the classifier ensemble problem involves relaxing it as a convex quadratic programming problem and computing the closed-form solution to the relaxation problem.

H. Iiduka is with the Department of Computer Science, Meiji University, Kanagawa 214-8571, Japan (e-mail: iiduka@cs.meiji.ac.jp). This work was supported by JSPS KAKENHI Grant Number JP18K11184.

One way of addressing a convex optimization problem with a complicated constraint set—such as the intersection of level sets of convex functions [4], [5]—is to reformulate the constraint set as a *fixed point set* of a computable *nonexpansive* mapping. Then fixed point algorithms [16], [17], [18] for solving the problem can be developed based on the nonexpansive mapping. Sparsity and diversity learning methods [19] based on a previous fixed point algorithm [18] have been reported; however, unfortunately, these methods are suitable only for the deterministic case and need that an appropriate step size be chosen in order for the method to converge to a solution to the problem sufficiently quickly for practical use. Choosing an appropriate step size ahead of time is difficult because what step size is appropriate depends on, for example, the numbers of instances and attributes in the dataset. In particular, this difficulty means that these methods are unsuitable for many classifier ensemble problems, including multiclass classifier ensemble problems and classifier ensemble problems with multiple instances. Therefore, one goal of the present study is to fulfill the need for a stochastic optimization method that does not require setting the step sizes in advance in order to solve the classifier ensemble problem directly.

In particular, for the present study, an iterative algorithm was developed, which is referred to as the *stochastic fixed point optimization algorithm* (Algorithm 1), for solving the classifier ensemble problem that can be reformulated as a convex stochastic optimization problem over the fixed point set of a quasi-nonexpansive mapping (Problem II.1). The algorithm proposed herein combines the SA method [10, (5.4.1)], [11] and an existing fixed point algorithm [18]. We analyze the convergence of the proposed algorithm for three types of step size, as follows. For a constant step size, if the step size is sufficiently small, then the proposed algorithm approximates a solution to the problem (Theorem III.1). For a sequence of decreasing step sizes, the algorithm converges in probability to a solution to the problem (Theorem III.2). The main issue when using iterative algorithms is how to determine the appropriate step size in order to guarantee sufficiently fast convergence. Therefore, a variation of the basic proposed algorithm (Algorithm 2) is also presented that allows the step size to be computed by line searches. For this variation of the algorithm, we are able to show that the algorithm converges in probability to a solution to the problem (Corollary III.1). Since the algorithm is able to determine an appropriate step size separately for each iteration according to the current situation, it is able to obtain faster convergence.

Another contribution of the present study, in addition to the above-described convergence analysis, is that we show that sparsity and diversity learning methods based on the

proposed algorithm can be applied to classifier ensemble learning with sparsity and diversity. Considering the concrete classifier ensemble problems with LIBSVM datasets [20] and the UCI Machine Learning Repository datasets [21], we show that the proposed learning methods have higher classification accuracies than that of the conventional method [5], especially in the case of the variation using the Armijo line search algorithm (Section IV).

This paper is organized as follows. Section II gives the mathematical preliminaries and provides the main problem. Section III presents the stochastic fixed point optimization algorithm for solving the main problem and analyzes its convergence. Section IV numerically compares the behaviors of the proposed learning algorithms with those of the existing ones. Section V concludes the paper with a brief summary.

II. Mathematical Preliminaries

A. Definitions and Notation

We use the standard notation $\mathbb{N}$ for the natural numbers (positive integers) and $\mathbb{R}^N$ for the N-dimensional Euclidean space, for which we use $\langle\cdot,\cdot\rangle$ for the inner product and $\|\cdot\|$ for the associated norm. We define $\mathbb{R}^N_+ := \{(x_i)_{i=1}^N \in \mathbb{R}^N \colon x_i \geq 0 \ (i = 1, 2, \ldots, N)\}$. Further, for a matrix Z, we use notation $Z^\top$ to indicate its transpose. For a random variable Y, we use $\mathbb{E}[Y]$ to indicate its expectation and $\mathbb{P}[\hat{Y}]$ to indicate the probability of a realization $\hat{Y}$. Finally, the identity mapping on the Euclidean space is written as Id. A function $g\colon \mathbb{R}^N \to \mathbb{R}$ is *strictly convex* [22, Definition 8.6] if $\boldsymbol{x} \neq \boldsymbol{y}$ implies $g(\alpha\boldsymbol{x}+(1-\alpha)\boldsymbol{y}) < \alpha g(\boldsymbol{x})+(1-\alpha)g(\boldsymbol{y})$ for any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$ and $\alpha \in (0,1)$. For the same function, the *subdifferential* [22, Definition 16.1], [23, Section 23] at any $\boldsymbol{x} \in \mathbb{R}^N$ is defined as $\partial g(\boldsymbol{x}) := \{\boldsymbol{u} \in \mathbb{R}^N \colon g(\boldsymbol{y}) \geq g(\boldsymbol{x}) + \langle \boldsymbol{y}-\boldsymbol{x}, \boldsymbol{u}\rangle \ (\boldsymbol{y} \in \mathbb{R}^N)\}$. The elements $\boldsymbol{u}$ of $\partial g(\boldsymbol{x})$ for a given $\boldsymbol{x} \in \mathbb{R}^N$ comprise the *subgradients* of g at $\boldsymbol{x}$. When g is differentiable at $\boldsymbol{x} \in \mathbb{R}^N$, the subdifferential reduces to the gradient: $\{\nabla g(\boldsymbol{x})\} = \partial g(\boldsymbol{x})$.

A mapping $Q\colon \mathbb{R}^N \to \mathbb{R}^N$ is *quasi-nonexpansive* [22, Definition 4.1(iii)] if, for all $\boldsymbol{x} \in \mathbb{R}^N$ and all $\boldsymbol{y} \in \mathrm{Fix}(Q)$, $\|Q(\boldsymbol{x})-\boldsymbol{y}\| \leq \|\boldsymbol{x}-\boldsymbol{y}\|$, where $\mathrm{Fix}(Q)$ is defined by

$$\mathrm{Fix}(Q) := \left\{\boldsymbol{x} \in \mathbb{R}^N \colon Q(\boldsymbol{x}) = \boldsymbol{x}\right\},$$

and called the *fixed point set* of Q. The fixed point set of a quasi-nonexpansive mapping with at least one fixed point is closed and convex [24, Proposition 2.6]. If $\|Q(\boldsymbol{x})-\boldsymbol{y}\|^2 + \|(\mathrm{Id}-Q)(\boldsymbol{x})\|^2 \leq \|\boldsymbol{x}-\boldsymbol{y}\|^2$ for all $\boldsymbol{x} \in \mathbb{R}^N$ and all $\boldsymbol{y} \in \mathrm{Fix}(Q)$, then Q is said to be a *quasi-firmly nonexpansive* mapping [25, Section 3]. It has been shown that Q is quasi-firmly nonexpansive if and only if $R := 2Q - \mathrm{Id}$ is quasi-nonexpansive [22, Proposition 4.2], which is equivalent to saying that R being quasi-nonexpansive implies that $(1/2)(\mathrm{Id}+R)$ is quasi-firmly nonexpansive. If $f\colon \mathbb{R}^N \to \mathbb{R}$ is a convex function, then the *subgradient projection* [24, Proposition 2.3], [26, Subchapter 4.3],

$$Q_{\mathrm{sp},f}(\boldsymbol{x}) := \begin{cases} \boldsymbol{x} - \dfrac{f(\boldsymbol{x})}{\|\boldsymbol{u}\|^2}\boldsymbol{u} & \text{if } f(\boldsymbol{x}) > 0, \\ \boldsymbol{x} & \text{otherwise,} \end{cases} \tag{1}$$

where $\boldsymbol{u}$ is any vector in $\partial f(\boldsymbol{x})$, is quasi-firmly nonexpansive [25, Lemma 3.1]. Moreover, the fixed point set is simply

$$\mathrm{Fix}\left(Q_{\mathrm{sp},f}\right) = \mathrm{lev}_{\leq 0} f := \left\{\boldsymbol{x} \in \mathbb{R}^N \colon f(\boldsymbol{x}) \leq 0\right\}.$$

A particularly interesting example of f in (1) is $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_1 - \alpha$ $(\boldsymbol{x} \in \mathbb{R}^N)$, where $\alpha \in \mathbb{R}$ and $\|\cdot\|_1$ denotes the ℓ_1 norm, that is, $\|\boldsymbol{x}\|_1 := \sum_{i=1}^N |x_i|$ $(\boldsymbol{x} := (x_i)_{i=1}^N \in \mathbb{R}^N)$. In this case, $\boldsymbol{u} \in \partial(\|\cdot\|_1 - \alpha)(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathbb{R}^N)$ can be efficiently computed, so $Q_{\mathrm{sp},\|\cdot\|_1-\alpha}$ can also be computed. Then $\mathrm{Fix}(Q_{\mathrm{sp},\|\cdot\|_1-\alpha}) = \{\boldsymbol{x} \in \mathbb{R}^N \colon \|\boldsymbol{x}\|_1 \leq \alpha\}$.

A mapping $Q\colon \mathbb{R}^N \to \mathbb{R}^N$ is *nonexpansive* [22, Definition 4.1(ii)] when it satisfies $\|Q(\boldsymbol{x}) - Q(\boldsymbol{y})\| \leq \|\boldsymbol{x}-\boldsymbol{y}\|$ for arbitrary $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$. All nonexpansive mappings are also quasi-nonexpansive. If C $(\subset \mathbb{R}^N)$ is a nonempty, closed subset, then a *metric projection* onto C, denoted P_C, is defined for all $\boldsymbol{x} \in \mathbb{R}^N$ such that $P_C(\boldsymbol{x}) \in C$ and $\|\boldsymbol{x} - P_C(\boldsymbol{x})\| = \inf_{\boldsymbol{y}\in C}\|\boldsymbol{x}-\boldsymbol{y}\|$. Projection P_C is nonexpansive and is such that $\mathrm{Fix}(P_C) = C$ [22, Proposition 4.8, (4.8)].

A mapping $T\colon \mathbb{R}^N \to \mathbb{R}^N$ is said to be *fixed-point closed* if, for an arbitrary sequence $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ $(\subset \mathbb{R}^N)$, convergence of $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ to some $\boldsymbol{x} \in \mathbb{R}^N$ together with $\lim_{n\to+\infty}\|\boldsymbol{x}_n - T(\boldsymbol{x}_n)\| = 0$ imply $\boldsymbol{x} \in \mathrm{Fix}(T)$. The mapping $Q_{\mathrm{sp},g}$ defined by (1) is fixed-point closed [25, Lemma 3.1].

B. Classifier Ensemble Problem and Its Existing Method

The classifier ensemble problem with sparsity and diversity learning [4], [5] features prominently in machine learning. The problem of optimizing sparsity and diversity has been expressed as a complex quadratic problem. In this formulation, errors are minimized with a least-squares loss function and diversity is measured using Yule's Q statistic. We state the mathematical model [4], [5] of the classifier ensemble problem. In the case of ensemble learning used in classification problems, there is a one-to-one association between instances $\boldsymbol{i}$ with labels l. Suppose N classifiers $(h^n)_{n=1}^N$ are used to classify the instances $\boldsymbol{i}$ into K classes. Classifiers h^n $(n = 1, 2, \ldots, N)$ each output a discriminant measure z^n for a given process instance $\boldsymbol{i}$, written as vector $\boldsymbol{z} := (z^n)_{n=1}^N$. These outputs are fused to create the combined class similarity measure from which the classification for each $\boldsymbol{i}$ is determined. Specifically, a weighted measure [4, (1)], [5, (1)] is computed for each instance $\boldsymbol{i}$ by $H(\boldsymbol{i}) = \langle \boldsymbol{z}, \boldsymbol{x}\rangle$, where x^n $(n = 1, 2, \ldots, N)$ are the weights for the corresponding classifiers and $\boldsymbol{x} := (x^n)_{n=1}^N$.

For N classifiers and a sample set $\{(\boldsymbol{i}_m, l_m)\}_{m=1}^M$, which comprises M samples, we obtain a training set $S := \{(\boldsymbol{z}_m, l_m)\}_{m=1}^M$, where $\boldsymbol{z}_m := (z_m^n)_{n=1}^N$ $(m = 1, 2, \ldots, M)$ and z_m^n $(n = 1, 2, \ldots, N, m = 1, 2, \ldots, M)$ is the measure corresponding to the mth sample in the sample set and the nth classifier in an ensemble. The basic learning algorithm for a classifier ensemble seeks to minimize the empirical loss as a function of the classifier weights $\boldsymbol{x} = (x^n)_{n=1}^N$. Therefore, the general optimization problem [4, (2), (3)], [5, (2), (4)] seeks the classifier weights $\boldsymbol{x}$ that minimize the least-squares loss function

$$f(\boldsymbol{x}) = \mathbb{E}\Big[F(\boldsymbol{x}, \underbrace{(\boldsymbol{z}, l)}_{\boldsymbol{\xi}})\Big] := \mathbb{E}\left[\frac{1}{2}\left(\langle \boldsymbol{z}, \boldsymbol{x}\rangle - l\right)^2\right] \tag{2}$$

over

$$C_1 := \mathbb{R}^N_+, \tag{3}$$

where $(\boldsymbol{z}_m, l_m) \in S$ are independent and identically distributed.

The objective of *sparsity learning* [4, Subsection 2.2.2], [5, Subsection 3.2.2] in the context of combining multiple classifiers is minimizing f in (2) over the intersection of $C_1 := \mathbb{R}^N_+$ and

$$C_2 := \left\{\boldsymbol{x} \in \mathbb{R}^N \colon \|\boldsymbol{x}\|_1 \le t_1\right\}, \tag{4}$$

where t_1 is the sparsity control parameter. Thus, for sparsity learning, the learning of the classifier weights utilizes the ℓ_1-norm $\|\cdot\|_1$. In the case of *diversity learning* [5, Subsections 3.2.3 and 4.1], the objective is still minimizing f, but this time over the intersection of $C_1 := \mathbb{R}^N_+$ and

$$C_3 := \left\{\boldsymbol{x} \in \mathbb{R}^N \colon \underbrace{\sum_{m=1}^{M} \left\{\langle [\boldsymbol{z}_m], \boldsymbol{x}\rangle - \langle \boldsymbol{z}_m, \boldsymbol{x}\rangle^2\right\}}_{f_{\mathrm{div}}(\boldsymbol{x})} \ge t_2\right\}, \tag{5}$$

where $[\boldsymbol{z}_m] := ((z_m^1)^2, (z_m^2)^2, \ldots, (z_m^N)^2)^\top$ and t_2 is the diversity control parameter. For details on how to derive ensemble diversity measure f_{div}, see Subsection 4.1 in [5].

Using the notation above, the classifier ensemble problem with both sparsity and diversity learning [4, (10)], [5, (15)] can be represented in terms of f defined in (2) and C_i $(i = 1, 2, 3)$ defined in (3), (4), and (5), as follows:

$$\text{Minimize } f(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in \bigcap_{i=1,2,3} C_i. \tag{6}$$

The existing approach [5] for the classifier ensemble problem is to reformulate problem (6) as the following relaxation problem [5, (16)]:

$$\begin{aligned} &\text{minimize } \frac{1}{2}\sum_{m=1}^{M}(\langle \boldsymbol{z}_m, \boldsymbol{x}\rangle - l_m)^2 + \bar{\alpha}\|\boldsymbol{x}\|_1 - \bar{\beta} f_{\mathrm{div}}(\boldsymbol{x}) \\ &\text{subject to } \boldsymbol{x} \in C_1. \end{aligned} \tag{7}$$

Here, $\bar{\alpha}$ is a control parameter for sparsity regularization and $\bar{\beta}$ is that for the diversity calculation. To compute these, the grid search algorithm in [5, Figure 2] is applied. The closed-form solution $\boldsymbol{x}^*$ [5, (18)] to relaxation problem (7) is then obtained as

$${\boldsymbol{x}^*}^\top := \frac{1}{1 + 2\bar{\beta}}\left(\sum_{m=1}^{M}\left(l_m \boldsymbol{z}_m + \bar{\beta}[\boldsymbol{z}_m]\right) - \bar{\alpha} I\right)^\top Z, \tag{8}$$

where $I := (1, 1, \ldots, 1)^\top \in \mathbb{R}^N$ and $Z := (\sum_{m=1}^{M}(\boldsymbol{z}_m \boldsymbol{z}_m^\top))^{-1}$, unless $\sum_{m=1}^{M}(\boldsymbol{z}_m \boldsymbol{z}_m^\top)$ is singular, in which case the pseudo-inverse matrix of $\sum_{m=1}^{M}(\boldsymbol{z}_m \boldsymbol{z}_m^\top)$ is used in place of Z.

C. Fixed Point Problem Formulation of Classifier Ensemble

The mapping

$$Q_1 := P_{C_1}, \tag{9}$$

where P_{C_1} is the metric projection onto C_1 defined as in (3), satisfies the nonexpansivity condition with $\mathrm{Fix}(Q_1) = C_1$. From $C_1 := \mathbb{R}^N_+$, Q_1 can be easily computed within a finite number of arithmetic operations [22, Subchapter 28.3].

Let us define $f_0(\boldsymbol{x}) := \|\boldsymbol{x}\|_1 - t_1$ $(\boldsymbol{x} \in \mathbb{R}^N)$. Then f_0 is convex and $C_2 = \mathrm{lev}_{\le 0} f_0 \neq \emptyset$. The mapping Q_2 defined by

$$Q_2 = Q_{\mathrm{sp}, f_0}, \tag{10}$$

where Q_{sp, f_0} is the subgradient projection relative to f_0, satisfies $\mathrm{Fix}(Q_2) = C_2$ and is quasi-firmly nonexpansive [25, Lemma 3.1]. Since the subgradient of $f_0(\cdot) := \|\cdot\|_1 - t_1$ at any point in $\mathbb{R}^N$ can be efficiently calculated [22, Example 16.25], it is easy to compute Q_{sp, f_0}.

Let us define $g_0(\boldsymbol{x}) := t_2 - f_{\mathrm{div}}(\boldsymbol{x})$. Then g_0 is convex with $C_3 = \mathrm{lev}_{\le 0} g_0$. Hence,

$$Q_3 := Q_{\mathrm{sp}, g_0} \tag{11}$$

is computable and quasi-firmly nonexpansive with $\mathrm{Fix}(Q_3) = C_3$.

Define $Q \colon \mathbb{R}^N \to \mathbb{R}^N$ by

$$Q := \frac{1}{2}\left[\mathrm{Id} + \sum_{i=1}^{3}\omega_i Q_i\right], \tag{12}$$

where $(\omega_i)_{i=1,2,3} \subset (0, +\infty)$ satisfies $\sum_{i=1,2,3}\omega_i = 1$ and Q_i $(i = 1, 2, 3)$ are defined as in (9), (10), and (11). From [22, Proposition 4.34], we have

$$\mathrm{Fix}(Q) = \mathrm{Fix}\left(\sum_{i=1}^{3}\omega_i Q_i\right) = \bigcap_{i=1,2,3}\mathrm{Fix}(Q_i) = \bigcap_{i=1,2,3} C_i.$$

Therefore, it is shown that the classifier ensemble problem (6) can be expressed as the following problem:

Problem II.1 *Suppose that f is defined by* (2) *and Q is defined by* (12)*. Then*

$$\textit{find } \boldsymbol{x}^\star \in X^\star := \left\{\boldsymbol{x}^\star \in \mathrm{Fix}(Q) \colon f(\boldsymbol{x}^\star) = \underbrace{\inf_{\boldsymbol{x} \in \mathrm{Fix}(Q)} f(\boldsymbol{x})}_{f^\star}\right\}.$$

The following proposition lists the properties of Q and f. The proof is given in Appendix A.

Proposition II.1 *We have the following:*

(i) *Q defined by* (12) *is quasi-firmly nonexpansive and fixed-point closed;*
(ii) *There exists a bounded, closed convex set C such that $\mathrm{Fix}(Q) \subset C$ and P_C can be efficiently computed;*
(iii) *f defined by* (2) *is well defined and strictly convex. There exists a unique solution of Problem II.1.*

Problem II.1 is considered under the basic conditions (see, e.g., [14, (A1), (A2)]) for machine learning:

(B1) There is an independent and identically distributed sample $\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \ldots$ of realizations of the random vector $\boldsymbol{\xi}$;
(B2) There is an oracle which, for a given input point $(\boldsymbol{x}, \boldsymbol{\xi}) \in \mathbb{R}^N \times S$, returns a stochastic gradient $\mathsf{G}(\boldsymbol{x}, \boldsymbol{\xi}) := \nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{\xi})$.

III. Stochastic fixed point optimization algorithm

Algorithm 1 is the proposed algorithm for solving Problem II.1 under (B1) and (B2). Algorithm 1 is based on the stochastic approximation (SA) method [10, (5.4.1)], [11] defined as follows: given $\boldsymbol{x}_0 \in \mathbb{R}^N$ and $(\lambda_n)_{n\in\mathbb{N}} \subset (0, +\infty)$,

$$\boldsymbol{x}_{n+1} = P_{\mathrm{Fix}(Q)}\left(\boldsymbol{x}_n - \lambda_n \mathsf{G}(\boldsymbol{x}_n, \boldsymbol{\xi}_n)\right) \ (n \in \mathbb{N}). \tag{13}$$

The SA method needs to use the metric projection onto $\mathrm{Fix}(Q)$, which is the constraint set of Problem II.1, and hence, the method can be applied only to cases where $\mathrm{Fix}(Q)$ is simple in the sense that $P_{\mathrm{Fix}(Q)}$ can be efficiently computed (e.g., $\mathrm{Fix}(Q)$ is a closed ball, a half-space, or a hyperslab [22, Chapter 28]). Since $\mathrm{Fix}(Q)$ in Problem II.1 is equal to the complicated set $\bigcap_{i=1,2,3} C_i$, it would be difficult to apply the SA method to Problem II.1. Meanwhile, Algorithm 1 uses a computable quasi-firmly nonexpansive mapping Q (see Proposition II.1(i)), which implies Algorithm 1 can be applied to Problem II.1.

Algorithm 1 Stochastic fixed point optimization algorithm for Problem II.1

Require: $\alpha \in (0,1)$, $Q_\alpha := \alpha\mathrm{Id} + (1-\alpha)Q$, $(\lambda_n)_{n\in\mathbb{N}} \subset (0, +\infty)$

1: $n \leftarrow 0$, $\boldsymbol{x}_0 \in \mathbb{R}^N$
2: **loop**
3: $\quad \boldsymbol{x}_{n+1} := P_C[Q_\alpha(\boldsymbol{x}_n) - \lambda_n \mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n)]$
4: $\quad n \leftarrow n+1$
5: **end loop**

The stopping condition of Algorithm 1 can be, for example, any of the following: $n = 10^a$ $(a \in \mathbb{N})$, $\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\| < \epsilon$, and $\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\| < \epsilon$, where $\epsilon > 0$ is sufficiently small.

A. Constant step-size rule

Let us perform a convergence analysis of Algorithm 1 with a constant step size.

Theorem III.1 *Suppose that $(\lambda_n)_{n\in\mathbb{N}}$ in Algorithm 1 satisfies that, for all $n \in \mathbb{N}$, $\lambda_n := \lambda \in (0, +\infty)$. Then there exists a positive real number K_1 such that*

$$\liminf_{n\to+\infty} \mathbb{E}\left[\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\|^2\right] \le \frac{K_1\lambda}{\alpha(1-\alpha)}.$$

Moreover, if $\lim_{n\to+\infty} \mathbb{E}[\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\|^2]$ exists, then there exist positive real numbers B and K_2 such that

$$\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] \le B^2\lambda + K_2\sqrt{\frac{(1-\alpha)K_1\lambda}{\alpha}}.$$

Theorem III.1 indicates that Algorithm 1 with a small constant step size λ may find a solution of Problem II.1. The proof of Theorem III.1 is given in Appendix B.

B. Diminishing step-size rule

The following theorem establishes a convergence analysis of Algorithm 1 under a diminishing step size. The proof of Theorem III.2 is given in Appendix B.

Theorem III.2 *Suppose that $(\lambda_n)_{n\in\mathbb{N}}$ in Algorithm 1 satisfies that* (S1) $\lim_{n\to+\infty} \lambda_n = 0$ *and* (S2) $\sum_{n=0}^{+\infty} \lambda_n = +\infty$. *Then the sequence $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ generated by Algorithm 1 converges in probability to a unique solution to Problem II.1.*

C. Line search step-size rule

Although the simplest step size satisfying (S1) and (S2) is $\lambda_n = c/(n+1)$ $(n \in \mathbb{N})$, where $c > 0$ is a constant, it is too difficult to select a constant in advance that guarantees sufficiently quick convergence. This is because what constant is appropriate depends on various factors, such as the number of iterations, the number of dimensions, the shapes of objective functions and constraint sets, and the selection of subgradients. This subsection develops line search methods that can determine an appropriate step size at each iteration to make the convergence of Algorithm 1 faster.

The step size λ_n satisfying (S1) and (S2) must be decided before Algorithm 1 is executed. In contrast, in this subsection, we consider the selection of a *step range* $[\underline{\lambda}_n, \overline{\lambda}_n]$ satisfying the following:

(SR) The sequences $(\underline{\lambda}_n)_{n\in\mathbb{N}}$ and $(\overline{\lambda}_n)_{n\in\mathbb{N}}$ are such that, for all $n \in \mathbb{N}$, $\underline{\lambda}_n \le \overline{\lambda}_n$, $\lim_{n\to+\infty} \overline{\lambda}_n = 0$, and $\sum_{n=0}^{+\infty} \underline{\lambda}_n = +\infty$.

For an example of the selection of a step range, see Section IV. Algorithm 2 is obtained by replacing λ_n in Algorithm 1 with $\lambda_n \in [\underline{\lambda}_n, \overline{\lambda}_n]$. When $\underline{\lambda}_n = \overline{\lambda}_n$, Algorithm 2 coincides with Algorithm 1.

Algorithm 2 Modified stochastic fixed point optimization algorithm for Problem II.1

Require: $\alpha \in (0,1)$, $Q_\alpha := \alpha\mathrm{Id} + (1-\alpha)Q$, $(\underline{\lambda}_n)_{n\in\mathbb{N}}, (\overline{\lambda}_n)_{n\in\mathbb{N}} \subset (0, +\infty)$

1: $n \leftarrow 0$, $\boldsymbol{x}_0 \in \mathbb{R}^N$
2: **loop**
3: $\quad \lambda_n \in [\underline{\lambda}_n, \overline{\lambda}_n]$
4: $\quad \boldsymbol{x}_{n+1} := P_C[Q_\alpha(\boldsymbol{x}_n) - \lambda_n \mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n)]$
5: $\quad n \leftarrow n+1$
6: **end loop**

Since $\lambda_n \in [\underline{\lambda}_n, \overline{\lambda}_n]$ $(n \in \mathbb{N})$ satisfies $\lim_{n\to+\infty} \lambda_n = 0$ and $\sum_{n=0}^{+\infty} \lambda_n = +\infty$, Theorem III.2 implies the following corollary:

Corollary III.1 *Suppose that (SR) holds. Then the sequence $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ generated by Algorithm 2 converges in probability to a unique solution to Problem II.1.*

Step 3 in Algorithm 2 is implemented as line searches. A popular line search condition is the Armijo condition [27], [28, Subchapter 3.1, (3.4)] defined as follows: for some constant $c \in (0,1)$,

$$H(\boldsymbol{z}_n + \lambda\boldsymbol{d}_n) \le H(\boldsymbol{z}_n) + c\lambda\langle\nabla H(\boldsymbol{z}_n), \boldsymbol{d}_n\rangle, \tag{14}$$

where $H\colon \mathbb{R}^N \to \mathbb{R}$ is differentiable, $\boldsymbol{z}_{n+1} := \boldsymbol{z}_n + \lambda_n\boldsymbol{d}_n$ $(n \in \mathbb{N})$, $\boldsymbol{z}_0 \in \mathbb{R}^N$, $\boldsymbol{d}_n$ is the search direction, and $\lambda_n > 0$ $(n \in \mathbb{N})$. Algorithm 3 is a line search algorithm based on (14) with $H(\cdot) := F(\cdot, \boldsymbol{\xi}_n)$, $\boldsymbol{z}_n := Q_\alpha(\boldsymbol{x}_n)$, $\boldsymbol{d}_n := -\mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n)$, and $\nabla H(\boldsymbol{z}_n) := \mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n)$. If Algorithm 3 fails (step 8), then we set $\lambda_n := \underline{\lambda}_n$.

Algorithm 3 Armijo line search algorithm

Require: $a > 0$, $c \in (0, 1)$, $K \in \mathbb{N}\backslash\{0\}$
1: **for** $I = 1, 1/a, \ldots, 1/a^K$ **do**
2: $\quad \lambda_n \leftarrow I\overline{\lambda}_n + (1 - I)\underline{\lambda}_n$
3: $\quad \boldsymbol{y}_n := P_C[Q_\alpha(\boldsymbol{x}_n) - \lambda_n \mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n)]$
4: $\quad$ **if** $F(\boldsymbol{y}_n, \boldsymbol{\xi}_n) \leq F(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n) - c\lambda_n \langle \mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n)), \mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n) \rangle$ **then**
5: $\quad\quad$ **stop** (success)
6: $\quad$ **end if**
7: **end for**
8: **stop** (failed, $\lambda_n := \underline{\lambda}_n$)

IV. Numerical Comparisons

Let us compare the existing sparsity and diversity learning method [5] using $\boldsymbol{x}^*$ defined by (8) with the proposed learning methods using solutions to Problem II.1 with t_i $(i = 1, 2)$ given by the discussion in [4, Sections 2 and 3]. Problem II.1 can be solved by Algorithms 1 and 2 with a closed ball C $(\supset C_2)$, $\alpha := 1/2$, $\boldsymbol{x}_0 = \boldsymbol{0}$, and $\omega_i := 1/3$ $(i = 1, 2, 3)$ in (12). The algorithms used in the experiments are as follows:

- CF: The closed-from solution defined by (8) [5]
- C1: Algorithm 1 with $\lambda_n := 10^{-1}$
- C2: Algorithm 1 with $\lambda_n := 10^{-2}$
- C3: Algorithm 1 with $\lambda_n := 10^{-3}$
- D1: Algorithm 1 with $\lambda_n := 10^{-1}/(n + 1)$
- D2: Algorithm 1 with $\lambda_n := 10^{-2}/(n + 1)$
- D3: Algorithm 1 with $\lambda_n := 10^{-3}/(n + 1)$
- LS: Algorithm 2 with $\lambda_n \in [10^{-3}/(n + 1), 1/(n + 1)]$ computed by Algorithm 3 with $a = 2$, $K = 7$, and $c = 10^{-4}$ [27, Subsection 6.1], [28, Subchapter 3.1]

Although the sparsity and diversity learning methods were presented in [19], they are examples of Algorithm 1 with diminishing step sizes (i.e., D1, D2, and D3). Hence, the experiments compared the performances between the sparsity and diversity learning methods using the above algorithms. The previously reported results (see, e.g., [18], [19], [29], [30]) for fixed point algorithms used $\lambda_n := 10^{-3}/(n + 1)$ empirically. Accordingly, it would be natural to implement D3, which is Algorithm 1 with $\lambda_n := 10^{-3}/(n + 1)$. To check whether Algorithm 1 with $\lambda_n > 10^{-3}/(n+1)$ performs better than D3, we implemented D1 and D2. We also used constant step sizes $\lambda_n = 10^{-1}, 10^{-2}, 10^{-3}$ and a step range $[10^{-3}/(n + 1), 1/(n + 1)]$ satisfying (SR) to compare fairly the performances of D1, D2, and D3 with the ones of the proposed methods using constant step sizes and step range.

The experiments used Mac Pro (Late 2013) with a 3 GHz 8-core Intel Xeon E5 CPU, 32 GB 1800 MHz DDR3 memory, and macOS Mojave version 10.14.3 operating system. The algorithms used in the experiments were written in Python 3.6.8 with the NumPy 1.15.4 package. The Moore-Penrose pseudo-inverse provided as `linalg.pinv` in the NumPy package was used to compute the (pseudo-)inverse matrix in CF. The experiments used the datasets from the LIBSVM [20] and the UCI Machine Learning Repository [21] for which information is shown in Table I. In this experiments, stratified 10-fold cross-validation for the datasets was performed. For this validation, the `StratifiedKFold` class in the scikit-learn 0.20.1 package was used.

Ensembles of support vector classifiers were constructed by the `BaggingClassifier` class in the scikit-learn 0.20.1 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 (OvR) multiclass classification strategy implemented as the `OneVsRestClassifier` class in the scikit-learn 0.20.1 package was used. The stopping condition for Algorithms 1 and 2 was any of the following: $n = \hat{M}$ and $n = 2\hat{M}$, where $\hat{M}$ is the number of training data needed for each classifier to learn the weights.

TABLE I: Datasets used for classification

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

The performances of the methods in the experiments were verified from the classification accuracy and elapsed time. In particular, for each dataset, the experiments compared the accuracy of the existing sparsity and diversity learning method using (8) (CF) with the accuracies of the proposed learning methods using Algorithm 1 (C1, C2, C3, D1, D2, and D3) and Algorithm 2 (LS) by using the T.TEST function in Microsoft Excel. The function value is the probability associated with a t-test, and the significance level is set at 5%; i.e., if the value of the function is less than 0.05, then there is a significant difference between the existing learning method and the proposed learning methods, and hence, the performance of the existing learning method is significantly different from the performances of the proposed learning methods.

Tables II and III show the classification accuracies and the elapsed times for the methods when the stopping conditions of the proposed algorithms were $n = \hat{M}$ and $n = 2\hat{M}$. Let us consider the results of the binary classification. The performances of the methods using C1, C2, C3, D1, D2, and D3 were sometimes good and sometimes not. For example, for the "leukemia" dataset, the performance of the existing learning method using CF was significantly different from that of the proposed learning methods using C3 and D3 (the values of the T.TEST function for C3 and D3 when

the stopping condition was $n = \hat{M}$ were 0.004 and 0.001, respectively). For the "breast-cancer" dataset, the performance of the existing learning method using CF was not significantly different from that of the proposed learning methods using C3 and D3 (the values of the T.TEST function for C3 and D3 when the stopping condition was $n = \hat{M}$ were 0.447 and 0.785, respectively). This is because the selection of a suitable step size depends on the instances and the attributes of the datasets (see also Subsection III-C). Moreover, these tables show that, for the "madelon" dataset, the values of the T.TEST function for the learning methods using C1, D1, and D2 when $n = \hat{M}$ were less than 0.05, while those when $n = 2\hat{M}$ were greater than 0.05. Meanwhile, it can be seen that the learning method using LS was robust for the cases of both $n = \hat{M}$ and $n = 2\hat{M}$. This is because Algorithm 2 with a step size computed by Algorithm 3 can determine an appropriate step size at each iteration. When the datasets except for the "leukemia" dataset were used, the elapsed time for the existing learning method using CF was shorter than the elapsed time for the proposed learning methods. Meanwhile, since the "leukemia" dataset has larger attributes than other datasets, the existing learning method using CF was time-consuming. This is because the method using CF needs to compute the inverses of large matrices.

Next, let us consider the results of the multiclass classification. For all of the datasets, the elapsed time for the proposed learning methods was longer than the elapsed time for the existing learning method. For the datasets except for the "iris", "wine", and "digits" datasets, the performance of the existing learning method using CF was significantly different from that of the proposed learning methods. For the "iris" and "wine" datasets, all of the learning methods used in the experiments had high classification accuracies. The step size $\lambda_n = 10^{-3}/(n+1)$ was useful for solving fixed point problems faster (see, e.g., [18], [19], [29], [30]). In fact, when the "iris" dataset was used and $n = 2\hat{M}$, only the value of the T.TEST function for the proposed learning method for D3 using $\lambda_n = 10^{-3}/(n+1)$ was less than 0.05 (the value of the T.TEST function for D3 when $n = 2\hat{M}$ was 0.002). However, when the "digits" dataset was used, the values of the T.TEST function for the proposed learning methods except for D3 using $\lambda_n = 10^{-3}/(n+1)$ were less than 0.05 (the values of the T.TEST function for D3 when $n = \hat{M}$ and $n = 2\hat{M}$ were 0.085 and 0.113, respectively). This result implies that a use of $\lambda_n = 10^{-3}/(n+1)$ was not always desirable, in contrast to [18], [19], [29], [30] and that it is too difficult to select suitable step sizes of learning methods for each dataset before implementing them. Meanwhile, the learning method using LS was robust for the multiclass classification. In particular, the learning method using LS had high accuracies, and for almost the datasets, the performance of the existing learning method using CF was significantly different from that of the learning method using LS. This is because Algorithm 2 with a step size computed by Algorithm 3 can determine an appropriate step size at each iteration and because the learning method using LS (Algorithm 2) can find solutions to the classifier ensemble problems at an early stage.

The above discussion shows that the performances of the learning methods using constant and diminishing step sizes were sometimes good and sometimes not, while the performance of the learning method using a step size computed by line searches is reliably good. Therefore, the sparsity and diversity learning method using LS is superior for solving Problem II.1.

V. Conclusion

This paper presented a stochastic fixed point algorithm for solving the classifier ensemble problem to minimize the expectation of convex functions over the fixed point set of a quasi-nonexpansive mapping and showed its convergence. This proposed algorithm and a variation with a line search step size were compared numerically with the existing learning method with respect to the classifier ensemble problem with sparsity and diversity learning for LIBSVM and UCI Machine Learning Repository datasets. The results demonstrated the optimality and efficiency of the proposed algorithms. In particular, the learning method using the algorithm with a line search step size is well suited for this classifier ensemble problem.

Acknowledgments

I am sincerely grateful to Editor-in-Chief Jun Wang and the three anonymous referees for helping me improve the original manuscript. I am grateful to Yoichi Hayashi for introducing me to convex ensemble learning, which is a significant application of convex optimization. I also thank Kazuhiro Hishinuma for his input on the numerical evaluation.

Appendix A
Proofs of Proposition II.1 and Lemmas

For the random process $\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \ldots$, let $\mathbb{E}[X|\boldsymbol{\xi}_{[n]}]$ denote the conditional expectation of X given $\boldsymbol{\xi}_{[n]} = (\boldsymbol{\xi}_0, \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_n)$. Unless stated otherwise, all relations between random variables are supported to hold almost surely.

We first prove Proposition II.1.

Proof of Proposition II.1: (i) The mapping Q_i $(i = 1, 2, 3)$ defined by one of (9), (10), and (11) is computable and quasi-firmly nonexpansive (see Subsection II-C). The quasi-nonexpansivity of Q_i $(i = 1, 2, 3)$ implies that $\sum_{i=1}^{3} \omega_i Q_i$ is quasi-nonexpansive [22, Exercise 4.11]. Accordingly, Q defined by (12) is computable and quasi-firmly nonexpansive. The continuity of $Q_1 := P_{C_1}$ ensures that Q_1 is fixed-point closed. Since Q_i $(i = 2, 3)$ is fixed-point closed (see Subsection II-A), Q is also fixed-point closed.

(ii) The boundedness condition of C_2 defined by (4) guarantees that $\mathrm{Fix}(Q)$ $(\subset C_2)$ is bounded. From the closed form of C_2, we can choose a simple, closed convex set C such that $C \supset C_2 \supset \mathrm{Fix}(Q)$ (e.g., C is a closed ball with a large enough radius).

(iii) Theorems 7.47 and 7.51 in [31] and the continuity and convexity of $F(\cdot, \boldsymbol{\xi})$ $(\boldsymbol{\xi} \in S)$ guarantee that the function f defined by (2) is well defined and convex. Moreover, it can be shown that, for a given $\boldsymbol{\xi} \in \mathbb{R}^N \backslash \{\mathbf{0}\} \times \mathbb{R}$, $F(\cdot, \boldsymbol{\xi})$ is strictly convex. Since at least one of x_m^n $(n = 1, 2, \ldots, N, m = 1, 2, \ldots, M)$ is not equal to 0, we have that f defined by (2) is strictly convex. The continuity of f and the boundedness of

TABLE II: Classification accuracies (%), elapsed times (s), and the value of the T.TEST function in Microsoft Excel for CF and each of the proposed algorithms when the stopping condition of both the proposed algorithms was $n = \hat{M}$

	CF		**C1**			**C2**			**C3**			**D1**			**D2**			**D3**			**LS**		
	acc.	time	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test
australian	55.93	0.016	69.86	1.777	0.002	83.03	1.747	0.000	85.07	1.723	0.000	84.48	1.737	0.000	83.47	1.723	0.000	83.05	1.725	0.000	85.51	1.793	0.000
breast-cancer	92.84	0.012	71.47	1.718	0.000	94.44	1.709	0.364	94.29	1.690	0.447	92.24	1.697	0.781	93.43	1.635	0.787	93.43	1.675	0.785	93.99	1.639	0.582
diabetes	33.98	0.009	65.10	2.134	0.000	64.45	2.110	0.000	63.02	2.130	0.000	56.64	2.031	0.000	53.64	2.055	0.000	29.03	2.143	0.005	60.81	2.051	0.000
ionosphere	73.29	0.015	73.25	0.644	0.987	71.86	0.634	0.561	71.31	0.614	0.418	72.43	0.600	0.725	71.03	0.627	0.351	71.32	0.648	0.433	71.84	0.605	0.547
leukemia	35.66	5794	54.66	26.44	0.058	34.83	26.09	0.925	70.33	25.48	0.004	53.00	24.82	0.123	54.83	23.36	0.067	71.83	18.57	0.001	77.66	22.07	0.000
madelon	46.20	2.196	50.25	59.15	0.033	50.05	52.91	0.026	50.00	52.76	0.022	49.95	52.62	0.024	50.05	52.66	0.021	49.95	52.76	0.024	50.05	51.84	0.021
splice	69.59	0.032	50.37	3.524	0.000	48.48	3.497	0.000	42.09	3.463	0.000	45.98	3.410	0.000	42.58	3.441	0.000	44.00	3.538	0.000	44.08	3.417	0.000
iris	84.00	0.031	84.00	0.795	1.000	80.00	0.644	0.383	81.33	0.657	0.566	81.33	0.636	0.566	74.66	0.649	0.039	68.66	0.670	0.002	80.00	0.669	0.383
svmguide2	26.78	0.029	56.54	2.274	0.002	56.54	2.381	0.002	56.54	2.347	0.002	56.54	2.404	0.002	56.54	2.358	0.002	56.54	2.355	0.002	56.54	2.460	0.002
wine	93.41	0.021	96.17	0.815	0.430	93.31	0.790	0.978	92.24	0.798	0.750	92.75	0.800	0.860	89.38	0.800	0.310	91.68	0.832	0.650	93.31	0.764	0.978
vehicle	14.28	0.052	55.80	9.820	0.000	43.02	9.876	0.000	37.37	9.868	0.000	42.26	9.631	0.000	39.74	9.753	0.000	37.60	10.03	0.000	43.99	9.958	0.000
glass	26.04	0.058	45.32	2.083	0.003	44.78	2.020	0.001	44.78	2.074	0.001	46.73	1.972	0.003	45.31	2.054	0.001	47.34	2.040	0.001	44.05	2.187	0.004
segment	12.72	0.130	75.93	98.19	0.000	74.93	97.17	0.000	71.29	97.68	0.000	71.03	98.11	0.000	72.68	95.39	0.000	74.32	97.04	0.000	71.55	98.43	0.000
digits	29.17	0.427	78.30	98.11	0.000	82.72	97.07	0.000	79.35	97.14	0.000	79.78	99.86	0.000	52.69	97.42	0.005	41.74	97.28	0.085	80.45	100.7	0.000
usps	14.36	12.03	16.70	6141	0.000	72.01	6097	0.000	70.63	6060	0.000	48.11	6044	0.000	68.08	6034	0.000	67.67	6054	0.000	64.94	5994	0.000

TABLE III: Classification accuracies (%), elapsed times (s), and the value of the T.TEST function in Microsoft Excel for CF and each of the proposed algorithms when the stopping condition of both the proposed algorithms was $n = 2\hat{M}$

	CF		**C1**			**C2**			**C3**			**D1**			**D2**			**D3**			**LS**		
	acc.	time	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test	acc.	time	t.test
australian	55.93	0.016	66.50	3.500	0.029	83.33	3.409	0.000	85.65	3.380	0.000	85.37	3.379	0.000	83.76	3.433	0.000	82.60	3.394	0.000	85.51	3.434	0.000
breast-cancer	92.84	0.012	67.93	3.446	0.000	93.85	3.280	0.367	93.41	3.297	0.580	93.12	3.302	0.683	92.99	3.256	0.730	93.58	3.309	0.528	92.63	3.400	0.849
diabetes	33.98	0.009	65.10	4.147	0.000	63.67	4.178	0.000	61.98	4.146	0.000	52.99	4.026	0.000	52.34	4.077	0.000	26.42	4.153	0.000	57.68	4.079	0.000
ionosphere	73.29	0.015	74.42	1.312	0.762	72.11	1.256	0.526	71.86	1.264	0.471	71.86	1.249	0.471	71.58	1.277	0.406	71.03	1.270	0.286	71.55	1.295	0.399
leukemia	35.66	5794	45.66	48.98	0.412	60.16	53.25	0.023	73.66	46.16	0.003	41.83	49.85	0.689	58.66	34.69	0.099	75.16	23.42	0.001	77.66	36.41	0.000
madelon	46.20	2.196	48.85	118.4	0.160	51.25	105.1	0.018	50.15	105.2	0.044	49.95	105.1	0.053	49.95	105.2	0.053	50.05	105.0	0.048	50.10	104.3	0.047
splice	69.59	0.032	47.68	7.168	0.000	46.98	6.842	0.000	43.49	6.802	0.000	43.99	6.914	0.000	43.89	6.916	0.000	40.39	6.913	0.000	44.08	6.860	0.000
iris	84.00	0.031	82.00	1.430	0.575	78.00	1.255	0.127	81.33	1.276	0.473	81.33	1.215	0.473	80.00	1.288	0.309	69.33	1.326	0.002	78.00	1.338	0.127
svmguide2	26.78	0.029	56.54	4.559	0.001	56.54	4.657	0.001	56.54	4.705	0.001	56.54	4.641	0.001	56.54	4.741	0.001	56.54	4.712	0.001	56.54	4.678	0.001
wine	93.41	0.021	95.58	1.710	0.486	92.79	1.633	0.979	92.69	1.590	0.956	92.23	1.503	0.857	92.06	1.599	0.826	89.52	1.647	0.409	93.81	1.601	0.794
vehicle	14.28	0.052	58.25	19.55	0.000	42.78	19.66	0.000	37.97	19.34	0.000	42.44	19.12	0.000	42.46	19.34	0.000	39.49	20.12	0.000	42.82	19.94	0.000
glass	26.04	0.058	48.54	4.006	0.000	41.08	3.971	0.006	47.65	3.953	0.000	48.04	3.895	0.000	45.28	4.077	0.001	45.24	4.034	0.005	47.15	3.984	0.000
segment	12.72	0.130	75.41	196.3	0.000	75.45	195.1	0.000	70.69	195.0	0.000	72.42	192.8	0.000	70.95	195.5	0.000	75.10	199.2	0.000	71.42	198.5	0.000
digits	29.17	0.427	77.37	195.8	0.000	84.90	197.7	0.000	79.38	194.9	0.000	78.24	195.8	0.000	56.46	194.4	0.002	45.91	195.6	0.113	79.73	200.5	0.000
usps	14.36	12.03	16.70	12258	0.000	73.10	12179	0.000	70.87	12133	0.000	48.31	12082	0.000	66.64	12090	0.000	67.27	12088	0.000	64.17	12013	0.000

$\mathrm{Fix}(Q)$ ensure the nonempty condition of the solution set of Problem II.1. Moreover, the strict convexity of f guarantees the uniqueness of the solution to Problem II.1. □

Proposition II.1 leads to the following lemma.

Lemma A.1 *Under (B1) and (B2), we have the following:*

(i) *For a given $(\boldsymbol{x}, \boldsymbol{\xi}) \in C \times S$, $\mathsf{g}(\boldsymbol{x}) := \mathbb{E}[\mathsf{G}(\boldsymbol{x}, \boldsymbol{\xi})]$ is well defined and $\mathsf{g}(\boldsymbol{x}) = \nabla f(\boldsymbol{x})$;*

(ii) *There exists $B \in \mathbb{R}$ such that, for all $\boldsymbol{x} \in C$ and all $n \in \mathbb{N}$, $\mathbb{E}[\|\mathsf{G}(\boldsymbol{x}, \boldsymbol{\xi}_n)\|^2] \leq B^2$.*

Proof: (i) Theorem 7.49 in [31], the boundedness of C_2 defined by (4), and the definition of $F(\cdot, \boldsymbol{\xi})$ ($\boldsymbol{\xi} \in S$) ensure that there exists a bounded set $C \supset C_2 \supset \mathrm{Fix}(Q)$ such that $f(\cdot) = \mathbb{E}[F(\cdot, \boldsymbol{\xi})]$ is differentiable on C. Moreover, for all $\boldsymbol{x} \in C$, $\mathsf{g}(\boldsymbol{x}) = \mathbb{E}[\nabla_{\boldsymbol{x}} F(\boldsymbol{x}, \boldsymbol{\xi})] = \nabla f(\boldsymbol{x})$ [31, Theorem 7.49(c)].

(ii) Given $\boldsymbol{\xi} \in S$, the subdifferential of $F(\cdot, \boldsymbol{\xi})$ is bounded on a bounded set C [32, Theorem 4.1.3], [22, Propositions 16.14(ii), (iii)]. Hence, Lemma A.1(ii) holds. □

The following lemma lists the basic properties of Algorithm 1.

Lemma A.2 *Suppose that $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ is the sequence generated by Algorithm 1 and $\boldsymbol{x} \in \mathrm{Fix}(Q)$ and define $X_n := \|\boldsymbol{x}_n - \boldsymbol{x}\|^2$, $\mathsf{G}_n := \mathsf{G}(Q_\alpha(\boldsymbol{x}_n), \boldsymbol{\xi}_n)$, and $\tilde{\mathsf{g}}_n := \mathsf{g}(Q_\alpha(\boldsymbol{x}_n))$ for all $n \in \mathbb{N}$. Then, for all $n \in \mathbb{N}$,*

$$\begin{aligned}
\text{(i) } \mathbb{E}[X_{n+1}] &\leq \mathbb{E}[X_n] - 2\alpha(1-\alpha)\,\mathbb{E}\left[\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\|^2\right] \\
&\quad + 2\lambda_n \left\{\lambda_n \mathbb{E}\left[\|\mathsf{G}_n\|^2\right] + \mathbb{E}\left[\langle \boldsymbol{x} - \boldsymbol{x}_n, \tilde{\mathsf{g}}_n\rangle\right]\right\}, \\
\text{(ii) } \mathbb{E}[X_{n+1}] &\leq \mathbb{E}[X_n] + 2\lambda_n \big\{\lambda_n \mathbb{E}\left[\|\mathsf{G}_n\|^2\right] \\
&\quad + \mathbb{E}\left[f(\boldsymbol{x}) - f(\boldsymbol{x}_n)\right] \\
&\quad + \mathbb{E}\left[\langle \boldsymbol{x}_n - Q_\alpha(\boldsymbol{x}_n), \nabla f(\boldsymbol{x}_n) - \tilde{\mathsf{g}}_n\rangle\right]\big\}.
\end{aligned}$$

Proof: (i) Let $n \in \mathbb{N}$ be fixed arbitrarily. Set $\boldsymbol{y}_n := Q_\alpha(\boldsymbol{x}_n) - \lambda_n \mathsf{G}_n$. The equation $-2\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \|\boldsymbol{x}-\boldsymbol{y}\|^2 - \|\boldsymbol{x}\|^2 - \|\boldsymbol{y}\|^2$ ($\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$) implies that

$$\begin{aligned}
&2\left\langle \boldsymbol{y}_n - \boldsymbol{x}_n + \lambda_n \mathsf{G}_n, \boldsymbol{x}_n - \boldsymbol{x}\right\rangle \\
&= -2\left\langle \boldsymbol{x}_n - \boldsymbol{y}_n, \boldsymbol{x}_n - \boldsymbol{x}\right\rangle + 2\lambda_n \left\langle \mathsf{G}_n, \boldsymbol{x}_n - \boldsymbol{x}\right\rangle \\
&= \|\boldsymbol{y}_n - \boldsymbol{x}\|^2 - \|\boldsymbol{x}_n - \boldsymbol{y}_n\|^2 - X_n + 2\lambda_n \left\langle \mathsf{G}_n, \boldsymbol{x}_n - \boldsymbol{x}\right\rangle.
\end{aligned}$$

From [18, Proposition 2.1], we have that, for all $\boldsymbol{x} \in \mathbb{R}^N$ and all $\boldsymbol{y} \in \mathrm{Fix}(Q)$, $\langle \boldsymbol{x} - Q_\alpha(\boldsymbol{x}), \boldsymbol{x} - \boldsymbol{y}\rangle \geq (1-\alpha)\|\boldsymbol{x} - Q(\boldsymbol{x})\|^2$. Accordingly,

$$\begin{aligned}
-2(1-\alpha)\left\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|^2 &\geq 2\left\langle Q_\alpha(\boldsymbol{x}_n) - \boldsymbol{x}_n, \boldsymbol{x}_n - \boldsymbol{x}\right\rangle \\
&= 2\left\langle \boldsymbol{y}_n - \boldsymbol{x}_n + \lambda_n \mathsf{G}_n, \boldsymbol{x}_n - \boldsymbol{x}\right\rangle.
\end{aligned}$$

The nonexpansivity of P_C with $\boldsymbol{x} = P_C(\boldsymbol{x})$ implies that $\|\boldsymbol{x}_{n+1} - \boldsymbol{x}\| = \|P_C(\boldsymbol{y}_n) - P_C(\boldsymbol{x})\| \leq \|\boldsymbol{y}_n - \boldsymbol{x}\|$. Hence,

$$\begin{aligned}
X_{n+1} \leq X_n + \left\|\boldsymbol{x}_n - \boldsymbol{y}_n\right\|^2 - 2(1-\alpha)\left\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|^2 \\
- 2\lambda_n \left\langle \mathsf{G}_n, \boldsymbol{x}_n - \boldsymbol{x}\right\rangle. \qquad (15)
\end{aligned}$$

From $\|\boldsymbol{x} - \boldsymbol{y}\|^2 \leq 2\|\boldsymbol{x}\|^2 + 2\|\boldsymbol{y}\|^2$ ($\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^N$) and the definition of $\boldsymbol{y}_n$,

$$\left\|\boldsymbol{y}_n - \boldsymbol{x}_n\right\|^2 \leq 2\left\|Q_\alpha(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|^2 + 2\lambda_n^2 \left\|\mathsf{G}_n\right\|^2. \qquad (16)$$

Accordingly, (15), (16), and the definition of Q_α guarantee that

$$\begin{aligned}
X_{n+1} \leq X_n - 2\alpha(1-\alpha)\left\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|^2 + 2\lambda_n^2 \left\|\mathsf{G}_n\right\|^2 \\
+ 2\lambda_n \left\langle \boldsymbol{x} - \boldsymbol{x}_n, \mathsf{G}_n\right\rangle. \qquad (17)
\end{aligned}$$

The definition of the expectation and the condition $\boldsymbol{x}_n = \boldsymbol{x}_n(\boldsymbol{\xi}_{[n-1]})$ ($n \in \mathbb{N}$) ensure that, for all $n \in \mathbb{N}$,

$$\begin{aligned}
\mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_n, \mathsf{G}_n\right\rangle\right] &= \mathbb{E}\left[\mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_n, \mathsf{G}_n\right\rangle | \boldsymbol{\xi}_{[n-1]}\right]\right] \\
&= \mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_n, \mathbb{E}\left[\mathsf{G}_n | \boldsymbol{\xi}_{[n-1]}\right]\right\rangle\right] \qquad (18) \\
&= \mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_n, \mathsf{g}(Q_\alpha(\boldsymbol{x}_n))\right\rangle\right].
\end{aligned}$$

Therefore, taking the expectation of (17), together with (18), we have the result that

$$\begin{aligned}
\mathbb{E}\left[X_{n+1}\right] \leq \mathbb{E}\left[X_n\right] - 2\alpha(1-\alpha)\,\mathbb{E}\left[\left\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|^2\right] \\
+ 2\lambda_n B_n, \qquad (19)
\end{aligned}$$

where

$$B_n := \lambda_n \mathbb{E}\left[\left\|\mathsf{G}_n\right\|^2\right] + \mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_n, \mathsf{g}(Q_\alpha(\boldsymbol{x}_n))\right\rangle\right]. \qquad (20)$$

(ii) Let $n \in \mathbb{N}$ be fixed arbitrarily and define $\tilde{\mathsf{g}}_n := \mathsf{g}(Q_\alpha(\boldsymbol{x}_n))$. Then, we have that

$$\begin{aligned}
&\mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_n, \tilde{\mathsf{g}}_n\right\rangle\right] \\
&= \mathbb{E}\left[\left\langle \boldsymbol{x} - Q_\alpha(\boldsymbol{x}_n), \tilde{\mathsf{g}}_n\right\rangle\right] + \mathbb{E}\left[\left\langle Q_\alpha(\boldsymbol{x}_n) - \boldsymbol{x}_n, \tilde{\mathsf{g}}_n\right\rangle\right] \\
&\leq \mathbb{E}\left[f(\boldsymbol{x}) - f(Q_\alpha(\boldsymbol{x}_n))\right] + \mathbb{E}\left[\left\langle Q_\alpha(\boldsymbol{x}_n) - \boldsymbol{x}_n, \tilde{\mathsf{g}}_n\right\rangle\right],
\end{aligned}$$

where the second inequality comes from $\{\nabla f(\boldsymbol{x})\} = \partial f(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^N$) and the definition of ∂f. Moreover, we have that $f(\boldsymbol{x}_n) - f(Q_\alpha(\boldsymbol{x}_n)) \leq \langle \boldsymbol{x}_n - Q_\alpha(\boldsymbol{x}_n), \nabla f(\boldsymbol{x}_n)\rangle$. Accordingly,

$$\mathbb{E}\left[\left\langle \boldsymbol{x} - \boldsymbol{x}_n, \tilde{\mathsf{g}}_n\right\rangle\right] \leq \mathbb{E}\left[f(\boldsymbol{x}) - f(\boldsymbol{x}_n)\right] + \bar{B}_n, \qquad (21)$$

where

$$\bar{B}_n := \mathbb{E}\left[\left\langle \boldsymbol{x}_n - Q_\alpha(\boldsymbol{x}_n), \nabla f(\boldsymbol{x}_n) - \tilde{\mathsf{g}}_n\right\rangle\right]. \qquad (22)$$

Hence, (19), (20), (21), and (22) lead to Lemma A.2(ii). This completes the proof. □

Appendix B
Proofs of Theorems III.1 and III.2

Proof of Theorem III.1: Fix $\boldsymbol{x} \in \mathrm{Fix}(Q)$ arbitrarily. Since Proposition II.1(i) and (ii) imply the almost sure boundedness of $(Q_\alpha(\boldsymbol{x}_n))_{n\in\mathbb{N}}$, $(\mathsf{g}(Q_\alpha(\boldsymbol{x}_n)))_{n\in\mathbb{N}}$ is almost surely bounded, where $\mathsf{g}(Q_\alpha(\boldsymbol{x}_n)) = \nabla f(Q_\alpha(\boldsymbol{x}_n))$ ($n \in \mathbb{N}$). The Cauchy-Schwarz inequality ensures that there exists $\bar{B} \in \mathbb{R}$ such that, for all $n \in \mathbb{N}$, $\mathbb{E}[\langle \boldsymbol{x} - \boldsymbol{x}_n, \mathsf{g}(Q_\alpha(\boldsymbol{x}_n))\rangle] \leq \bar{B}$. Accordingly, Lemma A.1(ii) implies that, for all $n \in \mathbb{N}$,

$$B_n \leq B^2 \lambda_n + \bar{B}, \qquad (23)$$

where B_n ($n \in \mathbb{N}$) is defined as in (20). Hence, Lemma A.2(i) (see also (19)) and (23) imply that, for all $n \in \mathbb{N}$,

$$\begin{aligned}
\mathbb{E}\left[X_{n+1}\right] \leq \mathbb{E}\left[X_n\right] - 2\alpha(1-\alpha)\,\mathbb{E}\left[\left\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|^2\right] \\
+ 2K_1\lambda, \qquad (24)
\end{aligned}$$

where $K_1 := B^2\lambda + \bar{B} < +\infty$.

Define $q_n := \mathbb{E}[\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\|^2]$ for all $n \in \mathbb{N}$. Let us assume that $\alpha(1-\alpha)\liminf_{n\to+\infty} q_n \leq K_1\lambda$ does not hold,

i.e., $\alpha(1-\alpha)\liminf_{n\to+\infty} q_n > K_1\lambda$. Then there exists $\delta > 0$ such that

$$\alpha(1-\alpha)\liminf_{n\to+\infty} q_n > K_1\lambda + 2\delta.$$

The definition of the limit inferior of $(q_n)_{n\in\mathbb{N}}$ thus guarantees that there exists $n_0 \in \mathbb{N}$ such that, for all $n \geq n_0$,

$$\alpha(1-\alpha)\liminf_{n\to+\infty} q_n - \delta \leq \alpha(1-\alpha)q_n.$$

Accordingly, for all $n \geq n_0$,

$$\alpha(1-\alpha)q_n > K_1\lambda + \delta. \tag{25}$$

Hence, from (24) and (25), for all $n \geq n_0$,

$$\mathbb{E}\left[X_{n+1}\right] < \mathbb{E}\left[X_n\right] - 2\delta < \mathbb{E}\left[X_{n_0}\right] - 2\delta\left(n+1-n_0\right),$$

which leads to a contradiction since the right-hand side of the above inequality approaches minus infinity when n diverges. Therefore,

$$\liminf_{n\to+\infty} \mathbb{E}\left[\left\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\right\|^2\right] \leq \frac{K_1\lambda}{\alpha(1-\alpha)}. \tag{26}$$

If the limit of $(q_n)_{n\in\mathbb{N}}$ exists, then (26) and Jensen's inequality guarantee that, for all $\epsilon > 0$, there exists $n_1 \in \mathbb{N}$ such that, for all $n \geq n_1$,

$$\mathbb{E}\left[\left\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\right\|\right] \leq \sqrt{\frac{K_1\lambda}{\alpha(1-\alpha)} + \epsilon}. \tag{27}$$

From the Cauchy-Schwarz inequality and the boundedness conditions of $(\mathrm{g}(Q_\alpha(\boldsymbol{x}_n)))_{n\in\mathbb{N}}$ and $(\nabla f(\boldsymbol{x}_n))_{n\in\mathbb{N}}$, there exists $K_2 \in \mathbb{R}$ such that, for all $n \in \mathbb{N}$,

$$\begin{aligned}\bar{B}_n &\leq K_2\mathbb{E}\left[\left\|\boldsymbol{x}_n - Q_\alpha(\boldsymbol{x}_n)\right\|\right] \\ &= (1-\alpha)K_2\mathbb{E}\left[\left\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\right\|\right],\end{aligned} \tag{28}$$

where $\bar{B}_n$ $(n \in \mathbb{N})$ is defined as in (22). Accordingly, Lemma A.2(ii) (see also (19), (20), and (21)) and (28) imply that, for all $n \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}\left[X_{n+1}\right] \leq \mathbb{E}\left[X_n\right] &+ 2B^2\lambda^2 + 2\lambda\mathbb{E}\left[f(\boldsymbol{x}) - f(\boldsymbol{x}_n)\right] \\ &+ 2(1-\alpha)K_2\lambda\mathbb{E}\left[\left\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\right\|\right].\end{aligned} \tag{29}$$

Let us show that, for all $\epsilon > 0$,

$$\begin{aligned}\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] \leq (1-\alpha)K_2\sqrt{\frac{K_1\lambda}{\alpha(1-\alpha)} + \epsilon} + \epsilon \\ + B^2\lambda.\end{aligned} \tag{30}$$

Suppose that (30) does not hold for all $\epsilon > 0$; that is, there exists $\epsilon_0 > 0$ such that

$$\begin{aligned}\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] > (1-\alpha)K_2\sqrt{\frac{K_1\lambda}{\alpha(1-\alpha)} + \epsilon_0} + \epsilon_0 \\ + B^2\lambda.\end{aligned} \tag{31}$$

Since the definition of the limit inferior of $(\mathbb{E}[f(\boldsymbol{x}_n) - f^\star])_{n\in\mathbb{N}}$ ensures the existence of $n_2 \in \mathbb{N}$ such that, for all $n \geq n_2$,

$$\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] - \frac{1}{2}\epsilon_0 \leq \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right],$$

(31) implies that, for all $n \geq n_2$,

$$\mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] > B^2\lambda + (1-\alpha)K_2\sqrt{\frac{K_1\lambda}{\alpha(1-\alpha)} + \epsilon_0} + \frac{1}{2}\epsilon_0. \tag{32}$$

Therefore, from (29) with $\boldsymbol{x} := \boldsymbol{x}^\star \in X^\star$, (27), and (32), for all $n \geq n_3 := \max\{n_1, n_2\}$,

$$\begin{aligned}&\mathbb{E}\left[X_{n+1}\right] \\ <&\mathbb{E}\left[X_n\right] + 2(1-\alpha)K_2\lambda\sqrt{\frac{K_1\lambda}{\alpha(1-\alpha)} + \epsilon_0} + 2B^2\lambda^2 \\ &- 2\lambda\left\{B^2\lambda + (1-\alpha)K_2\sqrt{\frac{K_1\lambda}{\alpha(1-\alpha)} + \epsilon_0} + \frac{1}{2}\epsilon_0\right\} \\ =&\mathbb{E}\left[X_n\right] - \lambda\epsilon_0,\end{aligned}$$

which implies that

$$\mathbb{E}\left[X_{n+1}\right] < \mathbb{E}\left[X_{n_3}\right] - \lambda\epsilon_0\left(n+1-n_3\right).$$

Since the right-hand side of the above inequality approaches minus infinity when n diverges, we have a contradiction. Hence, (30) holds for all $\epsilon > 0$. The lack of restriction on ϵ leads to the assertions in Theorem III.1. This completes the proof. □

Proof of Theorem III.2: Suppose that there exists $m_0 \in \mathbb{N}$ such that, for all $n \geq m_0$, $\mathbb{E}[X_{n+1}^\star] \leq \mathbb{E}[X_n^\star]$, where $\{\boldsymbol{x}^\star\} = X^\star$ and $X_n^\star := \|\boldsymbol{x}_n - \boldsymbol{x}^\star\|$ $(n \in \mathbb{N})$. In this case, there exists $\lim_{n\to+\infty}\mathbb{E}[X_n^\star]$. Lemma A.2(i) (see (19) and (20)) implies that, for all $n \geq m_0$,

$$\begin{aligned}&2\alpha\left(1-\alpha\right)\mathbb{E}\left[\left\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|^2\right] \\ &\leq \mathbb{E}\left[X_n^\star\right] - \mathbb{E}\left[X_{n+1}^\star\right] + 2\lambda_n(B^2\lambda_n + \bar{B}),\end{aligned}$$

where $\bar{B}$ is defined as in (23). Accordingly, from Jensen's inequality and $\lim_{n\to+\infty}\lambda_n = 0$,

$$\lim_{n\to+\infty} \mathbb{E}\left[\left\|\boldsymbol{x}_n - Q(\boldsymbol{x}_n)\right\|\right] = 0. \tag{33}$$

Furthermore, Lemma A.2(ii), (22), and (28) guarantee that, for all $n \geq m_0$ and all $\boldsymbol{x} \in \mathrm{Fix}(Q)$,

$$2\lambda_n D_n(\boldsymbol{x}) \leq \mathbb{E}\left[X_n(\boldsymbol{x})\right] - \mathbb{E}\left[X_{n+1}(\boldsymbol{x})\right], \tag{34}$$

where $X_n(\boldsymbol{x})$ and $D_n(\boldsymbol{x})$ are defined for all $n \in \mathbb{N}$ and all $\boldsymbol{x} \in \mathrm{Fix}(Q)$ by $X_n(\boldsymbol{x}) := \|\boldsymbol{x}_n - \boldsymbol{x}\|^2$ and

$$\begin{aligned}D_n(\boldsymbol{x}) := -B^2\lambda_n + \mathbb{E}\left[f(\boldsymbol{x}_n) - f(\boldsymbol{x})\right] \\ - (1-\alpha)K_2\mathbb{E}\left[\left\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\right\|\right].\end{aligned} \tag{35}$$

Summing (34) from $n = 0$ to $n = k$ $(k \in \mathbb{N})$ yields that, for all $\boldsymbol{x} \in \mathrm{Fix}(Q)$,

$$\begin{aligned}2\sum_{n=0}^{k}\lambda_n D_n(\boldsymbol{x}) &\leq \mathbb{E}\left[X_0(\boldsymbol{x})\right] - \mathbb{E}\left[X_{k+1}(\boldsymbol{x})\right] \\ &\leq \mathbb{E}\left[X_0(\boldsymbol{x})\right] < +\infty.\end{aligned} \tag{36}$$

Let us prove that, for all $\boldsymbol{x} \in \mathrm{Fix}(Q)$,

$$\liminf_{n\to+\infty} D_n(\boldsymbol{x}) \leq 0. \tag{37}$$

If there exists $\bar{\boldsymbol{x}} \in \mathrm{Fix}(Q)$ such that $\liminf_{n\to+\infty} D_n(\bar{\boldsymbol{x}}) > 0$, then there exist $m_0 \in \mathbb{N}$ and $\gamma > 0$ such that, for all $n \geq m_0$, $D_n(\bar{\boldsymbol{x}}) \geq \gamma$. Hence, from (36) and $\sum_{n=0}^{+\infty} \lambda_n = +\infty$,

$$+\infty = 2\gamma \sum_{n=m_0}^{+\infty} \lambda_n \leq 2 \sum_{n=m_0}^{+\infty} \lambda_n D_n(\bar{\boldsymbol{x}}) < +\infty,$$

which is a contradiction. Accordingly, for all $\boldsymbol{x} \in \mathrm{Fix}(Q)$, (37) holds, which implies that, for all $\boldsymbol{x} \in \mathrm{Fix}(Q)$,

$$\begin{aligned}&\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f(\boldsymbol{x})\right]\\ &\leq \limsup_{n\to+\infty} \left\{B^2\lambda_n + (1-\alpha)K_2\mathbb{E}\left[\|Q(\boldsymbol{x}_n) - \boldsymbol{x}_n\|\right]\right\}.\end{aligned}$$

Hence, from (33) and $\lim_{n\to+\infty} \lambda_n = 0$,

$$\liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f(\boldsymbol{x})\right] \leq 0 \text{ for all } \boldsymbol{x} \in \mathrm{Fix}(Q). \quad (38)$$

Moreover, (38) guarantees the existence of a subsequence $(\boldsymbol{x}_{n_i})_{i\in\mathbb{N}}$ of $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ such that

$$\lim_{i\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_{n_i}) - f^\star\right] = \liminf_{n\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_n) - f^\star\right] \leq 0. \quad (39)$$

Since $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ is almost surely bounded, there exists $\bar{S} \subset S$ such that $\mathbb{P}[\bar{S}] = 1$ and for all $\boldsymbol{\xi} \in \bar{S}$, $(\boldsymbol{x}_{n_i}(\boldsymbol{\xi}))_{i\in\mathbb{N}}$ is bounded. Let $\boldsymbol{\xi} \in \bar{S}$ be chosen arbitrarily. Then there exists $(\boldsymbol{x}_{n_{i_j}}(\boldsymbol{\xi}))_{j\in\mathbb{N}} \subset (\boldsymbol{x}_{n_i}(\boldsymbol{\xi}))_{i\in\mathbb{N}}$ such that $(\boldsymbol{x}_{n_{i_j}}(\boldsymbol{\xi}))_{j\in\mathbb{N}}$ converges to $\boldsymbol{x}^*(\boldsymbol{\xi})$. Proposition II.1(i) and (33) guarantee that $\mathbb{E}[\|\boldsymbol{x}^* - Q(\boldsymbol{x}^*)\|] = 0$, i.e., $\boldsymbol{x}^* \in \mathrm{Fix}(Q)$. Proposition II.1(iii) guarantees the continuity of f. Accordingly, (39) and the definition of $f^\star$ allow us to deduce that

$$\begin{aligned}0 \leq \mathbb{E}\left[f(\boldsymbol{x}^*) - f^\star\right] &= \lim_{j\to+\infty} \mathbb{E}\left[f\left(\boldsymbol{x}_{n_{i_j}}\right) - f^\star\right]\\ &= \lim_{i\to+\infty} \mathbb{E}\left[f(\boldsymbol{x}_{n_i}) - f^\star\right] \leq 0,\end{aligned}$$

which implies that $f(\boldsymbol{x}^*) = f^\star$. The uniqueness condition of the solution to Problem II.1 thus ensures that $\boldsymbol{x}^* = \boldsymbol{x}^\star$. Let $(\boldsymbol{x}_{n_{i_k}}(\boldsymbol{\xi}))_{k\in\mathbb{N}}$ be another subsequence of $(\boldsymbol{x}_{n_i}(\boldsymbol{\xi}))_{i\in\mathbb{N}}$ which converges to $\boldsymbol{x}_*$. A discussion similar to the one for obtaining $\boldsymbol{x}^* = \boldsymbol{x}^\star$ leads to $\boldsymbol{x}_* = \boldsymbol{x}^\star$. This implies that any subsequence of $(\boldsymbol{x}_{n_i}(\boldsymbol{\xi}))_{i\in\mathbb{N}}$ converges to $\boldsymbol{x}^\star$. Furthermore, the uniqueness condition of the solution to Problem II.1 guarantees that $(\boldsymbol{x}_n(\boldsymbol{\xi}))_{n\in\mathbb{N}}$ admits a unique cluster point $\boldsymbol{x}^\star$. Therefore, $(\boldsymbol{x}_n(\boldsymbol{\xi}))_{n\in\mathbb{N}}$ converges to $\boldsymbol{x}^\star$; i.e., $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ converges almost surely to $\boldsymbol{x}^\star$.

Suppose that, for all $n \in \mathbb{N}$, there exists $m(n) \geq n$, $\mathbb{E}[X^\star_{m+1}] > \mathbb{E}[X^\star_m]$. This implies that there exists $(\boldsymbol{x}_{n_l})_{l\in\mathbb{N}} \subset (\boldsymbol{x}_n)_{n\in\mathbb{N}}$ such that, for all $l \in \mathbb{N}$, $\mathbb{E}[X^\star_{n_l+1}] > \mathbb{E}[X^\star_{n_l}]$. Lemma 2.1 in [33] thus guarantees that there exists $m_1 \in \mathbb{N}$ such that, for all $n \geq m_1$, $\mathbb{E}[X^\star_{\tau(n)}] \leq \mathbb{E}[X^\star_{\tau(n)+1}]$, where $\tau(n) := \max\{k \leq n \colon \mathbb{E}[X^\star_k] < \mathbb{E}[X^\star_{k+1}]\}$ $(n \in \mathbb{N})$ satisfies $\lim_{n\to+\infty} \tau(n) = +\infty$. Since Lemma A.2 implies that, for all $n \geq m_1$,

$$\alpha\left(1-\alpha\right)\mathbb{E}\left[\left\|Q\left(\boldsymbol{x}_{\tau(n)}\right) - \boldsymbol{x}_{\tau(n)}\right\|^2\right] \leq \lambda_{\tau(n)}(B^2\lambda_{\tau(n)} + \bar{B}),$$

we find that

$$\lim_{n\to+\infty} \mathbb{E}\left[\left\|\boldsymbol{x}_{\tau(n)} - Q\left(\boldsymbol{x}_{\tau(n)}\right)\right\|\right] = 0. \quad (40)$$

Inequality (34) with $\boldsymbol{x} = \boldsymbol{x}^\star$ ensures that, for all $n \geq m_1$, $2\lambda_{\tau(n)}D_{\tau(n)}(\boldsymbol{x}^\star) \leq \mathbb{E}[X^\star_{\tau(n)}] - \mathbb{E}[X^\star_{\tau(n)+1}] \leq 0$. Hence, for all $n \geq m_1$,

$$D_{\tau(n)}(\boldsymbol{x}^\star) \leq 0, \quad (41)$$

which implies that

$$\begin{aligned}&\limsup_{n\to+\infty} \mathbb{E}\left[f\left(\boldsymbol{x}_{\tau(n)}\right) - f^\star\right]\\ &\leq \limsup_{n\to+\infty} \left\{B^2\lambda_{\tau(n)} + (1-\alpha)K_2\mathbb{E}\left[\left\|\boldsymbol{x}_{\tau(n)} - Q\left(\boldsymbol{x}_{\tau(n)}\right)\right\|\right]\right\}.\end{aligned}$$

Accordingly, (40) and $\lim_{n\to+\infty} \lambda_{\tau(n)} = 0$ lead to

$$\limsup_{n\to+\infty} \mathbb{E}\left[f\left(\boldsymbol{x}_{\tau(n)}\right) - f^\star\right] \leq 0. \quad (42)$$

Let $(\boldsymbol{x}_{\tau(n_j)})_{j\in\mathbb{N}}$ be an arbitrary subsequence of $(\boldsymbol{x}_{\tau(n)})_{n\geq m_1}$. Then (42) implies that

$$\begin{aligned}\limsup_{j\to+\infty} \mathbb{E}\left[f\left(\boldsymbol{x}_{\tau(n_j)}\right) - f^\star\right] &\leq \limsup_{n\to+\infty} \mathbb{E}\left[f\left(\boldsymbol{x}_{\tau(n)}\right) - f^\star\right]\\ &\leq 0. \quad (43)\end{aligned}$$

Proposition II.1(ii) guarantees that there exists $\hat{S} \subset S$ such that $\mathbb{P}[\hat{S}] = 1$ and for all $\boldsymbol{\xi} \in \hat{S}$, $(\boldsymbol{x}_{\tau(n_j)}(\boldsymbol{\xi}))_{j\in\mathbb{N}}$ is bounded. Let $\boldsymbol{\xi} \in \hat{S}$. Then there exists $(\boldsymbol{x}_{\tau(n_{j_k})}(\boldsymbol{\xi}))_{k\in\mathbb{N}} \subset (\boldsymbol{x}_{\tau(n_j)}(\boldsymbol{\xi}))_{j\in\mathbb{N}}$ such that $(\boldsymbol{x}_{\tau(n_{j_k})}(\boldsymbol{\xi}))_{k\in\mathbb{N}}$ converges to $\boldsymbol{x}_\star(\boldsymbol{\xi})$. The discussion for the proof of $\boldsymbol{x}^* \in \mathrm{Fix}(Q)$, together with (40), implies $\boldsymbol{x}_\star \in \mathrm{Fix}(Q)$. The continuity of f and (43) yield that

$$\begin{aligned}0 \leq \mathbb{E}\left[f(\boldsymbol{x}_\star) - f^\star\right] &= \limsup_{k\to+\infty} \mathbb{E}\left[f\left(\boldsymbol{x}_{\tau(n_{j_k})}\right) - f^\star\right]\\ &\leq \limsup_{j\to+\infty} \mathbb{E}\left[f\left(\boldsymbol{x}_{\tau(n_j)}\right) - f^\star\right] \leq 0,\end{aligned}$$

which, together with the uniqueness condition of the solution to Problem II.1, implies that $\boldsymbol{x}_\star = \boldsymbol{x}^\star$. Choose $(\boldsymbol{x}_{\tau(n_{j_l})}(\boldsymbol{\xi}))_{l\in\mathbb{N}} \subset (\boldsymbol{x}_{\tau(n_j)}(\boldsymbol{\xi}))_{j\in\mathbb{N}}$. A discussion similar to the one for showing the convergence of $(\boldsymbol{x}_{\tau(n_{j_k})}(\boldsymbol{\xi}))_{k\in\mathbb{N}}$ to $\boldsymbol{x}^\star$ ensures that $(\boldsymbol{x}_{\tau(n_{j_l})}(\boldsymbol{\xi}))_{l\in\mathbb{N}}$ converges to the same $\boldsymbol{x}^\star$. This implies that $(\boldsymbol{x}_{\tau(n_j)}(\boldsymbol{\xi}))_{j\in\mathbb{N}}$ converges to $\boldsymbol{x}^\star$. Since any subsequence of $(\boldsymbol{x}_{\tau(n)})_{n\geq m_1}$ converges almost surely to $\boldsymbol{x}^\star$, it is guaranteed that $(\boldsymbol{x}_{\tau(n)})_{n\geq m_1}$ converges almost surely to $\boldsymbol{x}^\star$, which implies that $\lim_{n\to+\infty} \mathbb{E}[X^\star_{\tau(n)}] = 0$. Since Lemma 2.1 in [33] guarantees that, for all $n \geq m_1$, $\mathbb{E}[X^\star_n] \leq \mathbb{E}[X^\star_{\tau(n)+1}]$, we find that $\limsup_{n\to+\infty} \mathbb{E}[\|\boldsymbol{x}_n - \boldsymbol{x}^\star\|^2] \leq \limsup_{n\to+\infty} \mathbb{E}[\|\boldsymbol{x}_{\tau(n)+1} - \boldsymbol{x}^\star\|^2] = 0$, i.e.,

$$\lim_{n\to+\infty} \mathbb{E}\left[\|\boldsymbol{x}_n - \boldsymbol{x}^\star\|^2\right] = 0.$$

Therefore, $(\boldsymbol{x}_n)_{n\in\mathbb{N}}$ converges in probability to $\boldsymbol{x}^\star$. This completes the proof. □

REFERENCES

[1] P. Yang, P. D. Yoo, J. Fernando, B. B. Zhou, Z. Zhang, and A. Y. Zomaya, "Sample subset optimization techniques for imbalanced and ensemble learning problems in bioinformatics applications," *IEEE Transactions on Cybernetics*, vol. 44, no. 3, pp. 445–455, 2014.

[2] C. Yu, M. Zhang, and F. Ren, "Collective learning for the emergence of social norms in networked multiagent systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2342–2355, 2014.

[3] S. Özöğür-Akyü, T. Windeatt, and R. Smith, "Pruning of error correcting output codes by optimization of accuracy-diversity trade off," *Machine Learning*, vol. 22, pp. 1751–1783, 2015.

[4] X. C. Yin, K. Huang, H. W. Hao, K. Iqbal, and Z. B. Wang, "A novel classifier ensemble method with sparsity and diversity," *Neurocomputing*, vol. 134, pp. 214–221, 2014.

[5] X. C. Yin, K. Huang, C. Yang, and H. W. Hao, "Convex ensemble learning with sparsity and diversity," *Information Fusion*, vol. 20, pp. 49–58, 2014.

[6] L. Zhang and W. Zhou, "Sparse ensemble using weighted combination methods based on linear programming," *Pattern Recognition*, vol. 44, pp. 97–106, 2011.

[7] Y. Zhang, S. Burer, and W. Street, "Ensemble pruning via semi-definite programming," *Journal of Machine Learning Research*, vol. 7, pp. 1315–1338, 2006.

[8] M. Perez-Ortiz, P. A. Gutierrez, and C. Hervas-Martinez, "Projection-based ensemble learning for ordinal regression," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 681–694, 2014.

[9] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.

[10] V. S. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint*. New York: Cambridge University Press, Cambridge, 2008.

[11] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.

[12] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic algorithmic framework," *SIAM Journal on Optimization*, vol. 22, pp. 1469–1492, 2012.

[13] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization II: Shrinking procedures and optimal algorithms," *SIAM Journal on Optimization*, vol. 23, pp. 2061–2089, 2013.

[14] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM Journal on Optimization*, vol. 19, pp. 1574–1609, 2009.

[15] S. Sundhar Ram, A. Nedić, and V. V. Veeravalli, "Incremental stochastic subgradient algorithms for convex optimization," *SIAM Journal on Optimization*, vol. 20, pp. 691–717, 2009.

[16] I. Yamada, "The hybrid steepest descent method for the variational inequality problem over the intersection of fixed point sets of nonexpansive mappings," in *Inherently Parallel Algorithms for Feasibility and Optimization and Their Applications* (D. Butnariu, Y. Censor, and S. Reich, eds.), pp. 473–504, New York: Elsevier, 2001.

[17] P. L. Combettes, "A block-iterative surrogate constraint splitting method for quadratic signal recovery," *IEEE Transactions on Signal Processing*, vol. 51, pp. 1771–1782, 2003.

[18] H. Iiduka, "Convergence analysis of iterative methods for nonsmooth convex optimization over fixed point sets of quasi-nonexpansive mappings," *Mathematical Programming*, vol. 159, pp. 509–538, 2016.

[19] Y. Hayashi and H. Iiduka, "Optimality and convergence for convex ensemble learning with sparsity and diversity based on fixed point optimization," *Neurocomputing*, vol. 273, pp. 367–372, 2018.

[20] C. C. Chang and C. J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27 https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/, 2011.

[21] D. Dheeru and E. Karra Taniskidou, "UCI machine learning repository," 2017.

[22] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York: Springer, 2011.

[23] R. T. Rockafellar, *Convex Analysis*. New Jersey: Princeton University Press, 1970.

[24] H. H. Bauschke and P. L. Combettes, "A weak-to-strong convergence principle for Fejér-monotone methods in Hilbert space," *Mathematics of Operations Research*, vol. 26, pp. 248–264, 2001.

[25] H. H. Bauschke and J. Chen, "A projection method for approximating fixed points of quasi nonexpansive mappings without the usual demiclosedness condition," *Journal of Nonlinear and Convex Analysis*, vol. 15, pp. 129–135, 2014.

[26] V. V. Vasin and A. L. Ageev, *Ill-posed problems with a priori information*. Utrecht: V.S.P. Intl Science, 1995.

[27] A. S. Lewis and M. L. Overton, "Nonsmooth optimization via quasi-Newton methods," *Mathematical Programming*, vol. 141, pp. 135–163, 2013.

[28] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, New York: Springer, 2nd ed., 2006.

[29] H. Iiduka, "Fixed point optimization algorithms for distributed optimization in networked systems," *SIAM Journal on Optimization*, vol. 23, pp. 1–26, 2013.

[30] H. Iiduka, "Proximal point algorithms for nonsmooth convex optimization with fixed point constraints," *European Journal of Operational Research*, vol. 253, pp. 503–513, 2016.

[31] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*. MOS-SIAM Series on Optimization, SIAM, Philadelphia, 2nd ed., 2014.

[32] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. New York: Springer, 2000.

[33] P. E. Maingé, "The viscosity approximation process for quasi-nonexpansive mappings in Hilbert spaces," *Computers and Mathematics with Applications*, vol. 59, pp. 74–79, 2010.

Hideaki Iiduka received the Ph.D. degree in mathematical and computing science from Tokyo Institute of Technology, Tokyo, Japan, in 2005. From 2005 to 2007, he was a Research Assistant in the Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, Japan. From 2007 to 2008, he was a Research Fellow (PD) of the Japan Society for the Promotion of Science. From October 2008 to March 2013, he was an Associate Professor in the Network Design Research Center, Kyushu Institute of Technology, Tokyo, Japan. From April 2013 to March 2019, he was an Associate Professor in the Department of Computer Science, School of Science and Technology, Meiji University, Kanagawa, Japan. Since April 2019, he has been a Professor in the same department. His research field is optimization theory and its applications to mathematical information science. He is a member of SIAM and MOS.