Optimality and Convergence for Convex Ensemble Learning with Sparsity and Diversity based on Fixed Point Optimization[☆]

Yoichi Hayashi[a,*], Hideaki Iiduka[b]

[a]*Department of Computer Science, Meiji University, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan.*
[b]*Department of Computer Science, Meiji University, Tama-ku, Kawasaki, Kanagawa 214-8571, Japan.*

Abstract

This paper discusses the classifier ensemble problem with sparsity and diversity learning, which is a central issue in machine learning. The current approach for reducing the size and increasing the accuracy of a classifier ensemble is to formulate it as a convex quadratic programming problem, which is a relaxation problem, and then solve it by using the existing methods for convex quadratic programming or by computing closed-form solutions. This paper presents a novel computational approach for solving the classifier ensemble problem with sparsity and diversity learning without any recourse to relaxation problems and their associated methods. We first show that the classifier ensemble problem can be expressed as a minimization problem for the sum of certain convex functions over the intersection of fixed point sets of quasi-nonexpansive mappings. Next, we propose fixed point optimization algorithms for solving the minimization problem and show that the algorithms converge to the solution of the minimization problem. It is shown that the proposed algorithms can directly solve the classifier ensemble problem with sparsity and diversity learning. Finally, we compare the

[☆]This work was supported in part by the International Collaborative Research Project supported by Meiji University, and in part by the Japan Society for the Promotion of Science through a Grant-in-Aid for Scientific Research (C) (15K04763).

[*]Corresponding author

Email addresses: hayashiy@cs.meiji.ac.jp (Yoichi Hayashi), iiduka@cs.meiji.ac.jp (Hideaki Iiduka)

performance of the proposed sparsity and diversity learning methods against an existing method in classification experiments using data sets from the UCI machine learning repository and the LIBSVM. The experimental results show that the proposed methods have higher classification accuracies than the existing method.

Keywords: convex ensemble learning, incremental subgradient method, fixed point, quasi-nonexpansive mapping
2000 MSC: 65K05, 68Q32, 90C25

---

1. Introduction

Methods for selecting classifiers to be combined in an ensemble have been gaining attention in the literature. When classifiers are aggregated, the resulting ensemble will generally perform better than any individual component [27]. Techniques such as bagging [5] and boosting [21] have been used extensively; however, although these methods are effective, they can be computationally expensive, particularly when using unlabeled test data.

A crucial factor when constructing an ensemble is the diversity and accuracy of the individual classifiers. If individual classifiers in a set are highly correlated, then there will be little improvement in the accuracy by creating an ensemble from them [26]. It is thus desirable to select diverse classifiers to improve the accuracy. Methods for pruning ensembles have been developed to reduce their size while increasing their accuracy [18]. The resulting ensembles are called pruned or sparse ensembles.

In the unordered bagging algorithm, the generalization error typically decreases with the inclusion of additional classifiers, which favors larger ensemble sizes. In [18], it was shown that by modifying the ordering of aggregation in the bagging algorithm, the generalization error can be minimized using an intermediate number of variables. This is in agreement with similar studies on neural networks which found it may be better to create an ensemble from a subset, instead of using all of the available networks [28]. Linear programming methods have also been proposed to sparsely combine multiple classifiers using a sparse weight vector [26]. Ideally, a sparse model will be more computationally efficient as well as having a better generalization performance.

The above ensemble classifier algorithms have been developed to optimize diversity and sparsity independently. However, algorithms which combine

these objectives are clearly preferable. A number of methods have been proposed to meet both of these objectives simultaneously. Examples include pruning the error correcting output code by utilizing diversity and accuracy information simultaneously [19], or using ordered aggregation to measure each base classifier's accuracy for a given problem [29]. For some ensembles, it was shown that using a small subset of the original members could improve the generalization performance [29]. Studies have also investigated how to simultaneously optimize both diversity and sparsity for ensembles of neural networks using a multiobjective approach, e.g., [7, 8].

Alternatively, the ensemble subset selection problem can be treated as a quadratic integer programming problem, where the solution is obtained using a semi-definite programming technique. This approach has been shown to outperform the heuristic methods in the literature in computational experiments [27]. An example of such an approach was presented in Kim et al. [16]. A meta-evolutionary approach, called evolutionary pruning, was developed to optimize ensembles, rather than create an ensemble of individually optimized members. This method used a Bayesian approach, where the ensemble size is specified in advance, and then ensembles competed based on their predictive performance [16]. Using this approach, both diversity and accuracy could be optimized.

A related approach for optimizing both sparsity and diversity was developed in [23]. The optimization of the sparsity and diversity was formulated as a complex quadratic problem, where errors were minimized with a least squares function. This was a relaxation problem for the selection of ensemble members, where diversity was measured using Yule's Q statistic. This method was found to perform favorably in a number of different aspects compared to other algorithms, such as the bagging algorithm [23]. This work was extended in [24], by introducing a penalty for sparsity and diversity.

This study proposes a new approach for the classifier ensemble problem, with sparsity and diversity learning, without recourse to relaxation problems and their related methods.

The approach presented here uses fixed point theory for nonexpansive mappings [1], [10, Chapter 3], [11, Chapter 1]. Here, classifier ensemble optimization is expressed as a minimization problem for the sum of certain convex functions over the intersection of fixed point sets of certain quasi-nonexpansive mappings. Fixed point optimization algorithms [14] (see also [12, 13, 15]) are used to solve this minimization problem and the iterations of the algorithm are shown to converge. This convergence guarantees that

the algorithms can optimize the classifier ensemble in terms of sparsity and diversity learning.

This paper is organized as follows. Section 2 introduces the relevant theory and definitions and expresses the classifier ensemble problem as a convex optimization problem over fixed point constraints. Section 3 proposes methods for solving the convex optimization problem and shows that the algorithm converges with diminishing step size. Section 4 presents numerical experiments using the UCI classification and LIBSVM data sets and compares the performances of the proposed methods with the method proposed in [24]. Section 5 summarizes the key findings and conclusions of this paper.

2. Mathematical Preliminaries

Let X^{T} denote the transpose of a matrix X. Let $\mathbb{R}^N$ be an N-dimensional Euclidean space with inner product $\langle \cdot, \cdot \rangle$ and induced norm $\| \cdot \|$, and let $\mathbb{R}^N_+ := \{(x_i)_{i=1}^N \colon x_i \geq 0 \ (i = 1, 2, \ldots, N)\}$. Let $\mathbb{N}$ denote the set of all positive integers including zero. The identity mapping on $\mathbb{R}^N$ is denoted by Id.

2.1. Convexity and subdifferentiability

A function $f \colon \mathbb{R}^N \to \mathbb{R}$ is said to be *strictly convex* [4, Definition 8.6] if, for all $\alpha \in (0, 1)$ and for all $x, y \in \mathbb{R}^N$, $x \neq y$ implies $f(\alpha x + (1 - \alpha)y) < \alpha f(x) + (1 - \alpha) f(y)$. The *subdifferential* [4, Definition 16.1], [20, Section 23] of a convex function $f \colon \mathbb{R}^N \to \mathbb{R}$ at $x \in \mathbb{R}^N$ is

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

If $f \colon \mathbb{R}^N \to \mathbb{R}$ is convex and differentiable, then $\partial f(x) = \{\nabla f(x)\}$ $(x \in \mathbb{R}^N)$ [4, Proposition 17.26]. Suppose that f is differentiable; then, f is strictly convex if and only if ∇f is strictly monotone; i.e., $\langle x - y, \nabla f(x) - \nabla f(y) \rangle > 0$ $(x, y \in \mathbb{R}^N$ with $x \neq y)$ [25, Proposition 25.10].

2.2. Quasi nonexpansivity and fixed-point closedness

The fixed point set of $Q \colon \mathbb{R}^N \to \mathbb{R}^N$ is denoted by

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

Q is said to be *quasi-nonexpansive* [4, Definition 4.1(iii)] if $\|Q(x) - y\| \leq \|x - y\|$ for all $x \in \mathbb{R}^N$ and for all $y \in \mathrm{Fix}(Q)$. When a quasi-nonexpansive

mapping has one fixed point, its fixed point set is closed and convex [3, Proposition 2.6]. $R\colon \mathbb{R}^N \to \mathbb{R}^N$ is called a *quasi-firmly nonexpansive* mapping [2, Section 3] if a quasi-nonexpansive mapping $Q\colon \mathbb{R}^N \to \mathbb{R}^N$ exists such that $R = (1/2)(\mathrm{Id} + Q)$. Q is *nonexpansive* if $\|Q(x) - Q(y)\| \leq \|x - y\|$ for all $x, y \in \mathbb{R}^N$. R is said to be *firmly nonexpansive* if there exists a nonexpansive mapping Q such that $R = (1/2)(\mathrm{Id} + Q)$.

Let $f_0\colon \mathbb{R}^N \to \mathbb{R}$ be a convex function with $\mathrm{lev}_{\leq 0} f_0 := \{x \in \mathbb{R}^N\colon f_0(x) \leq 0\} \neq \emptyset$. Then $\partial f_0(x)$ $(x \in \mathbb{R}^N)$ has a point, and the *subgradient* of f_0 at x can be denoted by $f_0'(x) \in \partial f_0(x)$. The *subgradient projection relative to* f_0 [3, Proposition 2.3], [22, Subchapter 4.3] is defined for all $x \in \mathbb{R}^N$ by

$$Q_{\mathrm{sp}}^{f_0}(x) := \begin{cases} x - \dfrac{f_0(x)}{\|f_0'(x)\|^2} f_0'(x) & \text{if } f_0(x) > 0, \\ x & \text{otherwise.} \end{cases}$$

The *metric projection* onto a nonempty, closed convex subset C of $\mathbb{R}^N$, denoted by P_C, is defined by $P_C(x) \in C$ and $\|x - P_C(x)\| = \inf_{y \in C} \|x - y\|$.

$Q\colon \mathbb{R}^N \to \mathbb{R}^N$ is said to be *fixed-point closed* [2, Lemma 2.1] if $x \in \mathrm{Fix}(Q)$ whenever $(x_n)_{n\in\mathbb{N}} \subset \mathbb{R}^N$ is such that $\lim_{n\to\infty} x_n = x \in \mathbb{R}^N$ and $\lim_{n\to\infty} \|x_n - Q(x_n)\| = 0$.

Proposition 2.1. *Suppose that* $f_0\colon \mathbb{R}^N \to \mathbb{R}$ *is convex with* $\mathrm{lev}_{\leq 0} f_0 \neq \emptyset$ *and* $C \subset \mathbb{R}^N$ *is nonempty, closed, and convex. Then, the following hold.*

(i) [2, Lemma 3.1], [3, Proposition 2.3], [22, Subchapter 4.3] $Q_{\mathrm{sp}}^{f_0}$ *satisfies the quasi-firm nonexpansivity and fixed-point closedness conditions and* $\mathrm{Fix}(Q_{\mathrm{sp}}^{f_0}) = \mathrm{lev}_{\leq 0} f_0$.

(ii) [4, (4.8), Proposition 4.8] P_C *satisfies the firm nonexpansivity condition and* $\mathrm{Fix}(P_C) = C$.

Proposition 2.1(ii) ensures that P_C is nonexpansive. Accordingly, if $(x_n)_{n\in\mathbb{N}}$ converges to $x \in \mathbb{R}^N$, we have $\limsup_{n\to\infty} \|P_C(x_n) - P_C(x)\| \leq \lim_{n\to\infty} \|x_n - x\| = 0$; i.e., P_C is continuous.

2.3. Main problem

First, we outline the mathematical model for a classifier ensemble [23, Section 2], [24, Section 3]. In ensemble learning for classification problems, each instance a is associated with a label y. It is assumed that there are

N different classifiers $(h^n)_{n=1}^N$ for classifying a into K classes. For a process instance a, each classifier h^n $(n = 1, 2, \ldots, N)$ outputs a discriminant measure $x^n := h^n(a)$; i.e., we have $x := (x^n)_{n=1}^N \in \mathbb{R}^N$. The final classification for a is determined with the classifier ensemble after the outputs of multiple classifiers are fused to give the combined class similarity measures. A weighted measure [23, (1)], [24, (1)] is computed for each instance a by $H(a) := \sum_{n=1}^N w^n x^n = \langle w, x \rangle$, where $w^n \in \mathbb{R}$ $(n = 1, 2, \ldots, N)$ is the weight for the nth classifier and $w := (w^n)_{n=1}^N \in \mathbb{R}^N$.

For a sample set $((a_m, y_m))_{m=1}^M$, with M samples and N different classifiers, we have $((x_m, y_m))_{m=1}^M$, where $x_m := (x_m^n)_{n=1}^N \in \mathbb{R}^N$ and $x_m^n \in \mathbb{R}$ $(n = 1, 2, \ldots, N, m = 1, 2, \ldots, M)$ is the measure for the nth classifier in an ensemble and the mth sample in the sample set. The main objective of the basic learning algorithm for a classifier ensemble is to find the classifier weights $w = (w^n)_{n=1}^N \in \mathbb{R}^N$ that produce the smallest empirical loss. Accordingly, the objective of the general optimization problem for obtaining the classifier weights w is to minimize the least squares loss defined by $f(w) := (1/2) \sum_{m=1}^M (\langle w, x_m \rangle - y_m)^2$ $(w \in \mathbb{R}^N)$ over $C_1 := \mathbb{R}_+^N$ [23, (2), (3)], [24, (2), (4)].

The aim of *sparsity learning* [23, Subsection 2.2.2], [24, Subsection 3.2.2] for combining multiple classifiers is to minimize the function f over the intersection of $C_1 := \mathbb{R}_+^N$ and $C_2 := \{w := (w^n)_{n=1}^N \in \mathbb{R}^N \colon \|w\|_1 := \sum_{n=1}^N |w^n| \leq t_1\}$, where t_1 is the sparsity control parameter. This implies the classifier weights for sparsity learning are learned by incorporating the l_1-norm $\|\cdot\|_1$. Meanwhile, the aim of *diversity learning* [24, Subsections 3.2.3 and 4.1] is to minimize the function f over the intersection of $C_1 := \mathbb{R}_+^N$ and $C_3 := \{w \in \mathbb{R}^N \colon f_{\mathrm{div}}(w) := \sum_{m=1}^M \{\langle [x_m], w \rangle - \langle x_m, w \rangle^2\} \geq t_2\}$, where $[x_m] := ((x_m^n)^2)_{n=1}^N = ((x_m^1)^2, (x_m^2)^2, \ldots, (x_m^N)^2)^{\mathrm{T}} \in \mathbb{R}^N$ and t_2 is the diversity control parameter. See (12)–(14) in [24] for a detailed derivation of the ensemble diversity measure f_{div}.

From the above discussion, this paper considers the following classifier ensemble problem with sparsity and diversity learning [23, (10)], [24, (15)].

Problem 2.1. *Given vectors* $x_m := (x_m^n)_{n=1}^N = (x_m^1, x_m^2, \ldots, x_m^N)^{\mathrm{T}} \in \mathbb{R}^N$ *with* $x_m^n \neq 0$ $(n = 1, 2, \ldots, N, m = 1, 2, \ldots, M)$ *and* $y_m \in \mathbb{R}$ $(m = 1, 2, \ldots, M)$, *and* $t_i > 0$ $(i = 1, 2)$, *let* $[x_m] := ((x_m^n)^2)_{n=1}^N = ((x_m^1)^2, (x_m^2)^2, \ldots, (x_m^N)^2)^{\mathrm{T}} \in$

$\mathbb{R}^N$. *Then,*

$$\textit{minimize } f(w) := \frac{1}{2}\sum_{m=1}^{M}\left(\langle w, x_m\rangle - y_m\right)^2 \textit{ subject to } w \in C_1 \cap C_2 \cap C_3,$$

where each C_i $(i = 1, 2, 3)$ *is a nonempty, closed convex set defined as follows.*

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

$$C_2 := \left\{ w := (w^n)_{n=1}^N \in \mathbb{R}^N \colon \|w\|_1 := \sum_{n=1}^{N} |w^n| \leq t_1 \right\},$$

$$C_3 := \left\{ w \in \mathbb{R}^N \colon \sum_{m=1}^{M} \left\{ \langle [x_m], w\rangle - \langle x_m, w\rangle^2 \right\} \geq t_2 \right\}.$$

The existing approach [23, 24] for the classifier ensemble problem is to formulate Problem 2.1 as the following relaxation problem [24, (16)],

$$\text{minimize } f(w) + \bar{\alpha}\|w\|_1 - \bar{\beta}\sum_{m=1}^{M}\left\{\langle [x_m], w\rangle - \langle x_m, w\rangle^2\right\} \text{ subject to } w \in C_1, \tag{2.1}$$

where $\bar{\alpha}$ and $\bar{\beta}$ are control parameters for sparsity regularization and diversity calculation, and then solve the above problem using the existing methods for convex quadratic programming or by computing closed-form solutions. From [24, (18)], the closed-form solution w^* to the relaxation problem (2.1) is

$$w^{*\top} := \frac{1}{1+2\bar{\beta}}\left(\sum_{m=1}^{M}\left(y_m x_m + \bar{\beta}[x_m]\right) - \bar{\alpha} I\right)^{\top}\left(\sum_{m=1}^{M}\left(x_m x_m^{\top}\right)\right)^{-1}, \tag{2.2}$$

where $I := (1, 1, \ldots, 1)^{\top} \in \mathbb{R}^N$. In this paper, we propose fixed point optimization algorithms for solving Problem 2.1 without any recourse to relaxation problems and their associated methods.

Here, let us define a function $\bar{f}_m \colon \mathbb{R}^N \to \mathbb{R}$ $(m = 1, 2, \ldots, M)$ for all $w \in \mathbb{R}^N$ by $\bar{f}_m(w) := (\langle x_m, w\rangle - y_m)^2$. Then, $\nabla \bar{f}_m(w) = 2(\langle x_m, w\rangle - y_m)x_m$ $(m = 1, 2, \ldots, M)$. Accordingly, for all $w_1, w_2 \in \mathbb{R}^N$ where $w_1 \neq w_2$,

$$\begin{aligned}
&\left\langle w_1 - w_2, \nabla \bar{f}_m(w_1) - \nabla \bar{f}_m(w_2)\right\rangle \\
&= 2\left\langle w_1 - w_2, (\langle x_m, w_1\rangle - y_m)x_m - (\langle x_m, w_2\rangle - y_m)x_m\right\rangle \\
&= 2\left\langle w_1 - w_2, \langle w_1 - w_2, x_m\rangle\, x_m\right\rangle \\
&= 2\left\langle w_1 - w_2, x_m\right\rangle^2,
\end{aligned}$$

which, together with $x_m^n \neq 0$ $(n = 1, 2, \ldots, N)$, implies that $\nabla \bar{f}_m$ $(m = 1, 2, \ldots, M)$ satisfies the strict monotonicity condition; i.e.,

$$f(\cdot) := \frac{1}{2} \sum_{m=1}^{M} \left(\langle \cdot, x_m \rangle - y_m \right)^2 = \frac{1}{2} \sum_{m=1}^{M} \bar{f}_m \text{ is strictly convex.} \tag{2.3}$$

Next, we show that each C_i $(i = 1, 2, 3)$ can be expressed as the fixed point of a certain nonexpansive mapping. From Proposition 2.1(ii),

$$P_{C_1} \text{ is firmly nonexpansive and continuous with } \operatorname{Fix}(P_{C_1}) = C_1. \tag{2.4}$$

It is obvious from the definitions of firmly nonexpansive and quasi-firmly nonexpansive mappings that P_{C_1} is quasi-firmly nonexpansive. P_{C_1} can be easily computed within a finite number of arithmetic operations [4, Subchapter 28.3]. Define $f_0(w) := \|w\|_1 - t_1$ $(w \in \mathbb{R}^N)$, then f_0 satisfies the convexity and nonsmoothness conditions and $C_2 = \operatorname{lev}_{\leq 0} f_0 \neq \emptyset$. Proposition 2.1(i) thus implies that

$$Q_{\mathrm{sp}}^{f_0} \text{ is quasi-firmly nonexpansive and fixed-point closed with } \operatorname{Fix}\left(Q_{\mathrm{sp}}^{f_0}\right) = C_2. \tag{2.5}$$

Since the subgradient of $f_0 := \|\cdot\|_1 - t_1$ at any point in $\mathbb{R}^N$ can be efficiently calculated [4, Example 16.25], it is easy to compute $Q_{\mathrm{sp}}^{f_0}$. We define a function $g_0 \colon \mathbb{R}^N \to \mathbb{R}$ for all $w \in \mathbb{R}^N$ by $g_0(w) := t_2 - \sum_{m=1}^{M} \{ \langle [x_m], w \rangle - \langle x_m, w \rangle^2 \}$. Then, g_0 is convex and differentiable with $C_3 = \operatorname{lev}_{\leq 0} g_0$. Hence,

$$Q_{\mathrm{sp}}^{g_0} \text{ is quasi-firmly nonexpansive and fixed-point closed with } \operatorname{Fix}\left(Q_{\mathrm{sp}}^{g_0}\right) = C_3. \tag{2.6}$$

$Q_{\mathrm{sp}}^{g_0}$ can then be computed from the closed-form expression of ∇g_0.

We now discuss the following convex optimization problem over the intersection of fixed point sets of quasi-nonexpansive mappings.

Problem 2.2. *Suppose that $f_i \colon \mathbb{R}^N \to \mathbb{R}$ is strictly convex and $Q_i \colon \mathbb{R}^N \to \mathbb{R}^N$ $(i \in \mathcal{I} := \{1, 2, \ldots, I\})$ is quasi-firmly nonexpansive and fixed-point closed. Our objective is to*

$$\textit{find } w^\star \in W^\star := \left\{ w^\star \in W := \bigcap_{i \in \mathcal{I}} \operatorname{Fix}(Q_i) \colon \sum_{i \in \mathcal{I}} f_i(w^\star) = \inf_{w \in W} \sum_{i \in \mathcal{I}} f_i(w) \right\},$$

where one assumes that $W^\star \neq \emptyset$.

Under $W^\star \neq \emptyset$, the strict convexity of f leads to the uniqueness of the solution to Problem 2.2 [25, Corollary 25.15].

Here, we show that Problem 2.1 is contained within Problem 2.2. From (2.4), (2.5), and (2.6), $Q_1 := P_{C_1}$, $Q_2 := Q_{\mathrm{sp}}^{f_0}$, and $Q_3 := Q_{\mathrm{sp}}^{g_0}$ are quasi-firmly nonexpansive and fixed-point closed with $\bigcap_{i=1}^3 C_i = \bigcap_{i=1}^3 \mathrm{Fix}(Q_i)$. Define $\bar{Q}\colon \mathbb{R}^N \to \mathbb{R}^N$ and $Q\colon \mathbb{R}^N \to \mathbb{R}^N$ by

$$\bar{Q} := \sum_{i=1}^{3} \omega_i Q_i \text{ and } Q := \frac{1}{2}\left(\mathrm{Id} + \bar{Q}\right), \tag{2.7}$$

where $\omega_i > 0$, $(i = 1, 2, 3)$ satisfies $\sum_{i=1}^3 \omega_i = 1$. Then, $\bar{Q}$ is quasi-nonexpansive and $Q := (1/2)(\mathrm{Id}+\bar{Q})$ is quasi-firmly nonexpansive with $\mathrm{Fix}(Q) = \mathrm{Fix}(\bar{Q}) = \bigcap_{i=1}^3 \mathrm{Fix}(Q_i)$ [4, Exercise 4.11]. Furthermore, Q satisfies the fixed-point closedness condition. Therefore, we can conclude that Problem 2.1 is equivalent to the problem of minimizing the strictly convex function defined by (2.3) over the fixed point set of the quasi-firmly nonexpansive mapping Q defined by (2.7); i.e.,

$$\text{Problem 2.1 is a special case of Problem 2.2 when } I = 1. \tag{2.8}$$

Let us divide f defined by (2.3) into

$$f = \sum_{i=1}^{3} \omega_i f = \sum_{i=1}^{3} f_i. \tag{2.9}$$

Since f defined by (2.3) is strictly convex, $f_i := \omega_i f$ $(i = 1, 2, 3)$ is also strictly convex. Therefore, we can conclude that Problem 2.1 is equivalent to the problem of minimizing the strictly convex function defined by (2.9) over the intersection of the fixed point sets of the quasi-firmly nonexpansive mappings $Q_1 := P_{C_1}$, $Q_2 := Q_{\mathrm{sp}}^{f_0}$, and $Q_3 := Q_{\mathrm{sp}}^{g_0}$ defined by (2.4), (2.5), and (2.6); i.e.,

$$\text{Problem 2.1 is a special case of Problem 2.2 when } I = 3. \tag{2.10}$$

3. Fixed Point Optimization Algorithms

3.1. Convergence analysis of the algorithm for Problem 2.2 when $I = 1$

Let us consider Problem 2.2 when $I := 1$, i.e., the problem of minimizing a strictly convex function f over the fixed point set of a quasi-firmly nonexpansive and fixed-point closed mapping Q. The relationship (2.8) between

Problems 2.1 and 2.2 guarantees that Problem 2.2, when $I := 1$, includes Problem 2.1.

The following algorithm [14, Algorithm 3.1] is for solving Problem 2.2 when $I := 1$.

Algorithm 3.1.

Step 0. *Choose $w_0 \in \mathbb{R}^N$ arbitrarily, set $\alpha \in (0,1)$ and $(\lambda_n)_{n\in\mathbb{N}} \subset (0,\infty)$, and define $Q_\alpha := \alpha \mathrm{Id} + (1-\alpha)Q$.*

Step 1. *Compute $w_{n+1} \in \mathbb{R}^N$ by*

$$w_{n+1} := Q_\alpha\left(w_n\right) - \lambda_n g_n, \ \textit{where } g_n \in \partial f\left(Q_\alpha(w_n)\right).$$

Set $n := n+1$ and return to Step 1.

The following is a convergence analysis of Algorithm 3.1. Proposition 3.1 can be proved from [14, Theorem 3.2] (proof omitted).

Proposition 3.1. *Suppose that $(w_n)_{n\in\mathbb{N}}$ generated by Algorithm 3.1 is bounded and $(\lambda_n)_{n\in\mathbb{N}}$ satisfies $\lim_{n\to\infty}\lambda_n = 0$ and $\sum_{n=0}^{\infty}\lambda_n = \infty$. Then, $(w_n)_{n\in\mathbb{N}}$ converges to the solution of Problem 2.2 when $I := 1$.*

Let us apply Algorithm 3.1 to Problem 2.1. Since f in Problem 2.1 is strictly convex and differentiable (see also (2.3)) and Q in Problem 2.1 is defined by (2.7), Algorithm 3.1 can be applied to Problem 2.1 in the form

$$\begin{aligned} &Q_\alpha(w_n) = \alpha w_n + (1-\alpha)\frac{1}{2}\left(w_n + \omega_1 P_{\mathbb{R}^N_+}(w_n) + \omega_2 Q^{f_0}_{\mathrm{sp}}(w_n) + \omega_3 Q^{g_0}_{\mathrm{sp}}(w_n)\right), \\ &w_{n+1} := Q_\alpha(w_n) - \lambda_n \nabla f\left(Q_\alpha(w_n)\right) \ (n \in \mathbb{N}). \end{aligned} \tag{3.1}$$

To ensure the boundedness of $(w_n)_{n\in\mathbb{N}}$ generated by Algorithm 3.1, the algorithm can be modified as follows.

$$w_{n+1} := P_K\left[Q_\alpha(w_n) - \lambda_n \nabla f\left(Q_\alpha(w_n)\right)\right] \ (n \in \mathbb{N}), \tag{3.2}$$

where $K \subset \mathbb{R}^N$ is bounded, closed, and convex and P_K can be easily computed. Since C_2 is bounded, we can choose a closed ball K ($\supset C_2$) with a large enough radius (see [14, Assumption 3.2] for the details of the modification of Algorithm 3.1). Therefore, Proposition 3.1 guarantees that the

sequence $(w_n)_{n\in\mathbb{N}}$ generated by Algorithm (3.2) converges to the solution of Problem 2.1.

The following proposition is required to determine the rate of convergence of Algorithm 3.1 (Algorithms (3.1) and (3.2)). We omit the details but Proposition 3.2 can be proved from [14, Corollary 3.2].

Proposition 3.2. *Suppose that the assumptions in Proposition 3.1 hold, $w^\star \in W^\star$ is the unique solution to Problem 2.2, and $\lambda_n := 1/(n+1)$ for all $n \in \mathbb{N}$. If there exists $\beta > 0$ such that $\alpha > \beta^2/(\beta^2+2)$ and $\mathrm{d}(w_n, W) := \|w_n - P_W(w_n)\| \leq \beta\|w_n - Q_\alpha(w_n)\|$ for all $n \in \mathbb{N}$ and if $(\|w_n - Q(w_n)\|)_{n\in\mathbb{N}}$ is monotone decreasing,*[1] *Then, for $n \in \mathbb{N}$,*

$$\|w_n - Q(w_n)\| = \mathcal{O}\left(\frac{1}{\sqrt{n+1}}\right) \text{ and } f(w_n) - f(w^\star) = \mathcal{O}\left(\frac{1}{\sqrt{n+1}}\right).$$

3.2. Convergence analysis of the algorithm for Problem 2.2 when $I > 1$

Let us consider Problem 2.2 when $I > 1$. The expression (2.10) relating Problems 2.1 and 2.2 guarantees that Problem 2.1 is contained within Problem 2.2 when $I := 3$.

We present an incremental subgradient algorithm [14, Algorithm 4.1] for solving Problem 2.2 when $I > 1$.

Algorithm 3.2.

Step 0. *Choose $w_0 := w_{0,0} \in \mathbb{R}^N$ arbitrarily, set $\alpha_i \in (0,1)$ $(i \in \mathcal{I})$ and $(\lambda_n)_{n\in\mathbb{N}} \subset (0,\infty)$, and define $Q_{\alpha_i} := \alpha_i \mathrm{Id} + (1-\alpha_i)Q_i$ $(i \in \mathcal{I})$.*

Step 1. *Compute $w_{n,i} \in \mathbb{R}^N$ $(i \in \mathcal{I})$ by*

$$w_{n,i} := Q_{\alpha_i}(w_{n,i-1}) - \lambda_n g_{n,i}, \text{ where } g_{n,i} \in \partial f_i(Q_{\alpha_i}(w_{n,i-1})).$$

Step 2. *Set $w_{n+1} := w_{n+1,0} := w_{n,I}$. The algorithm sets $n := n+1$ and returns to Step 1.*

The following proposition shows that Algorithm 3.2 converges to the solution of the main problem (see [14, Theorem 4.2] for a proof).

[1] In the case where $\alpha := 1/2$ and $Q := (1/(1-\alpha))(P_W - \alpha\mathrm{Id})$, $\beta = 1$ can be chosen [14, p.523]. Proposition 3.1 guarantees that $(w_n)_{n\in\mathbb{N}}$ in Algorithm 3.1 satisfies $\lim_{n\to\infty}\|w_n - Q(w_n)\| = 0$.

Proposition 3.3. *Suppose that* $(w_{n,i})_{n\in\mathbb{N}}$ $(i \in \mathcal{I})$ *generated by Algorithm 3.2 is bounded and* $(\lambda_n)_{n\in\mathbb{N}}$ *satisfies* $\lim_{n\to\infty} \lambda_n = 0$ *and* $\sum_{n=0}^{\infty} \lambda_n = \infty$. *Then,* $(w_{n,i})_{n\in\mathbb{N}}$ $(i \in \mathcal{I})$ *converges to the solution of Problem 2.2.*

Algorithm 3.2 for Problem 2.1 is written as

$$
\begin{aligned}
&w_n := w_{n,0},\\
&w_{n,1} := P_K\left[P_{\mathbb{R}^N_+}\left(w_{n,0}\right) - \omega_1\lambda_n\nabla f\left(P_{\mathbb{R}^N_+}(w_{n,0})\right)\right],\\
&Q_{\alpha_2} := \alpha_2\mathrm{Id} + (1-\alpha_2)\, Q_{\mathrm{sp}}^{f_0},\\
&w_{n,2} := P_K\left[Q_{\alpha_2}\left(w_{n,1}\right) - \omega_2\lambda_n\nabla f\left(Q_{\alpha_2}\left(w_{n,1}\right)\right)\right],\\
&Q_{\alpha_3} := \alpha_3\mathrm{Id} + (1-\alpha_3)\, Q_{\mathrm{sp}}^{g_0},\\
&w_{n+1} = w_{n,3} := P_K\left[Q_{\alpha_3}\left(w_{n,2}\right) - \omega_3\lambda_n\nabla f\left(Q_{\alpha_3}\left(w_{n,2}\right)\right)\right]\ (n\in\mathbb{N}),
\end{aligned}
\tag{3.3}
$$

where K is a closed ball with a radius large enough to satisfy $K \supset C_2$. The same discussion as in Subsection 3.1 describing the existence of a simple, bounded, closed, and convex set K leads to the boundedness of $(w_{n,i})_{n\in\mathbb{N}}$ $(i \in \mathcal{I})$. Proposition 3.3 thus guarantees that the sequence $(w_{n,i})_{n\in\mathbb{N}}$ $(i \in \mathcal{I})$ generated by Algorithm (3.3) converges to the solution of Problem 2.1.

Moreover, Algorithm 3.2 can be applied to Problem 2.2 under the following conditions: for all $m \in \mathcal{I} = \{1, 2, \ldots, M\}$,

$$
\begin{aligned}
&f_m(w) := \bar{f}_m(w) = \left(\langle x_m, w\rangle - y_m\right)^2\ (w\in\mathbb{R}^N),\\
&Q_m \text{ is defined by one of } P_{\mathbb{R}^N_+},\ Q_{\mathrm{sp}}^{f_0},\ \text{and } Q_{\mathrm{sp}}^{g_0}.
\end{aligned}
$$

This implies that Algorithm 3.2 is suitable for use when Problem 2.1 cannot be solved under centralized control.

The following proposition can be proved from [14, Corollary 4.2] and establishes the rate of convergence of Algorithm 3.2 (Algorithm (3.3)). See [14, Subsection 4.3] for the details of the convergence rate analysis for Algorithm 3.2.

Proposition 3.4. *Suppose that the assumptions in Proposition 3.3 hold,* $w^\star \in W^\star$ *is the unique solution to Problem 2.2, and* $\lambda_n := 1/(n+1)$ *for all* $n \in \mathbb{N}$. *If there exists* $\beta_i > 0$ *such that* $\alpha_i > \beta_i^2/(\beta_i^2+2)$ *and* $\mathrm{d}(w_{n,i-1}, W) \le \beta_i\|w_{n,i-1} - Q_{\alpha_i}(w_{n,i-1})\|$ *for all* $n \in \mathbb{N}$ *and if* $(\|w_{n,i-1} - Q_i(w_{n,i-1})\|)_{n\in\mathbb{N}}$ *is monotone decreasing for all* $i \in \mathcal{I}$, *then, for* $n \in \mathbb{N}$ *and for all* $i \in \mathcal{I}$,

$$
\|w_{n,i-1} - Q_i\left(w_{n,i-1}\right)\| = \mathcal{O}\left(\frac{1}{\sqrt{n+1}}\right)\ \text{and}\ f\left(w_n\right) - f\left(w^\star\right) = \mathcal{O}\left(\frac{1}{\sqrt{n+1}}\right).
$$

4. Numerical experiments

This section compares the proposed sparsity and diversity learning methods with the learning method in [24]. The experiments use the "Adult," "Arrhythmia," "Phishing," "Sonar," and "Heart" data sets from the UCI machine learning repository [9, 17] and the LIBSVM [6]. Information for the data sets is shown in Table 1. In the experiments, 3-fold cross validation for the data sets was performed. The base learner was the support vector machine classifier, and 100 ensembles were constructed by bagging. The experiments used a 13-inch MacBook Air with an Intel(R) Core(TM) i7-5650U CPU processor, two 4 GB 1600 MHz DDR3 memory modules (the total RAM is 8 GB), and Mac OS X El Capitan (Version 10.11.6) operating system. The algorithms used in the experiments were coded in Python 3.5.3.

Table 1: Data sets used for classification

Data set	Instances	Attributes	Classes
Adult	32561	123	2
Arrhythmia	420	278	2
Phishing	11055	68	2
Sonar	208	60	2
Heart	270	13	2

The existing sparsity and diversity learning (SDL) method [24] uses the closed-form solution w^* defined by (2.2) for the relaxation problem (2.1), i.e.,

$$w^{*\top} := \frac{1}{1+2\bar{\beta}} \left(\sum_{m=1}^{M} \left(y_m x_m + \bar{\beta}[x_m] \right) - \bar{\alpha} I \right)^{\top} \left(\sum_{m=1}^{M} \left(x_m x_m^{\top} \right) \right)^{-1}, \quad (4.1)$$

where $\bar{\alpha}$ and $\bar{\beta}$ are computed by the grid search algorithm in [24, Figure 2]. When the matrix $\sum_{m=1}^{M} \left(x_m x_m^{\top} \right)$ is singular, $(\sum_{m=1}^{M} \left(x_m x_m^{\top} \right))^{-1}$ in (4.1) is replaced with the pseudo inverse matrix of $\sum_{m=1}^{M} \left(x_m x_m^{\top} \right)$. Meanwhile, the proposed SDL methods use the solution $w^\star$ to Problem 2.1 that can be computed by Algorithms (3.2) and (3.3). In the experiments, Algorithms (3.2) and (3.3) were implemented using an initial estimate $w_0 = 0$, $\alpha = \alpha_i = 1/2$ $(i = 1, 2, 3)$, and $\lambda_n := 10^{-3}/(n+1)$ $(n \in \mathbb{N})$.[2] The stopping condition for Algorithms (3.2) and (3.3) was $n = 1000$ or $\|w_n - w_{n-1}\| < 10^{-2}$.

[2]Numerical results in [14, 15] indicate that fixed point algorithms with small step sizes

Table 2 shows the classification accuracies for the existing SDL method [24] and the proposed SDL methods using Algorithms (3.2) and (3.3). The results indicate that the proposed methods performed better than the existing method when using (4.1). In particular, compared with the SDL using (4.1), the classification accuracies were improved by the proposed SDL method when using Algorithm (3.2) (resp. Algorithm (3.3)) with 50.63%, 12.47%, 1.25%, 14.71%, and 0.75% (resp., 50.61%, 12.47%, 0.54%, 8.82%, and 0.75%) for the "Adult," "Arrhythmia," "Phishing," "Sonar," and "Heart" data sets. The average accuracies of the SDL methods using Algorithms (4.1), (3.2), and (3.3) were 58.51%, 74.47%, and 73.15%, respectively. Here, we verify whether the performances of the existing and proposed SDL methods were different with a t-test. The value of the T.TEST function[3] in Microsoft Excel for Algorithms (4.1) and (3.2) (resp. Algorithms (4.1) and (3.3)) was 0.154975719 (resp. 0.190029206). This implies that the average performance of the existing SDL method was different from that of the proposed SDL methods. Therefore, from the results of the t-tests and Table 2, we can conclude that the proposed SDL methods using Algorithms (3.2) and (3.3) are superior for solving the classifier ensemble problem with sparsity and diversity learning.

Table 2: Classification accuracies (%) and elapsed time (sec) for the SDL methods applied to the data in Table 1.

	Alg.(4.1) [24]		Alg.(3.2)		Alg.(3.3)	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
Adult	25.31	4.81	75.94	10.08	75.92	16.58
Arrhythmia	43.65	2.01	56.12	0.14	56.12	0.14
Phishing	90.25	0.83	91.50	1.72	90.79	2.68
Sonar	52.45	0.09	67.16	0.05	61.27	0.07
Heart	80.90	0.02	81.65	0.07	81.65	0.09

When the "Adult," "Phishing," and "Heart" data sets were used, the elapsed time for the SDL method using Algorithm (4.1) was shorter than the elapsed time for the SDL method using Algorithms (3.2) and (3.3). This

have faster convergence. Hence, the experiments described in this section used $\lambda_n := \mu/(n+1)$ with a small positive constant μ.

[3]https://support.office.com/en-us/article/T-TEST-function-d4e08ec3-c545-485f-962e-276f7cbed055

is because the "Adult," "Phishing," and "Heart" data sets have smaller attributes so that $(\sum_{m=1}^{M}(x_m x_m^\top))^{-1}$ in Algorithm (4.1) was efficiently computed. Next, we consider the "Arrhythmia" data set which has larger attributes than the "Adult," "Phishing," and "Heart" data sets. The SDL method using Algorithm (4.1) is time-consuming because Algorithm (4.1) requires us to compute the inverse of a large matrix. Accordingly, the SDL methods using Algorithms (3.2) and (3.3) were completed faster than the one using Algorithm (4.1).

5. Conclusion

Classifier ensembles with sparsity and diversity learning are an area of active research, from both practical and theoretical viewpoints. The main contributions of this paper are summarized as follows. We first show that the classifier ensemble problem can be expressed as a minimization problem for the sum of specified convex functions over the intersection of fixed point sets of certain quasi-nonexpansive mappings. Next, we propose fixed point optimization algorithms and show that they converge to the solution of the minimization problem. The results suggest that the proposed algorithm can directly solve the classifier ensemble problem with sparsity and diversity learning. Finally, we compare numerically the proposed sparsity and diversity learning methods with an existing sparsity and diversity learning method [24]. The numerical results show that the proposed methods can provide a more than 50% increase in the classification accuracy, compared to the existing method.

The present algorithms have a different optimization structure to the method in Yin et al. [24] with its diversity and sparsity notions. Our results contribute to the field of ensemble selection by providing better analytical solutions for diversity and sparsity using their respective objective function and constraints.

Acknowledgements

We are sincerely grateful to the editors-in-chief, Zidong Wang and Steven Hoi, the anonymous associate editor, and the two anonymous reviewers for insightful comments. We also thank Kazuhiro Hishinuma for his input on the numerical examples.

[1] Bauschke, H.H., Borwein, J.M.: On projection algorithms for solving convex feasibility problems. SIAM Review **38**, 367–426 (1996)

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

[3] Bauschke, H.H., Combettes, P.L.: A weak-to-strong convergence principle for Fejér-monotone methods in Hilbert space. Mathematics of Operations Research **26**, 248–264 (2001)

[4] Bauschke, H.H., Combettes, P.L.: Convex Analysis and Monotone Operator Theory in Hilbert Spaces. Springer (2011)

[5] Breiman, L.: Bagging predictors. Machine Learning **24**, 123–140 (1996)

[6] Chang, C.C., Lin, C.J.: LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2**, 1–27 (2011) URL `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[7] Chen, H., Tino, P., Yao, X.: Predictive ensemble pruning by expectation propagation. IEEE Transactions on Knowledge and Data Engineering **21**, 999–1013 (2009)

[8] Chen, H., Yao, X.: Multiobjective neural network ensembles based on regularized negative correlation learning. IEEE Transactions on Knowledge and Data Engineering **22**, 1751–1783 (2009)

[9] Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Science (2010) URL `http://archive.ics.uci.edu/ml/`

[10] Goebel, K., Kirk, W.A.: Topics in Metric Fixed Point Theory. Cambridge Studies in Advanced Mathematics. Cambridge University Press (1990)

[11] Goebel, K., Reich, S.: Uniform Convexity, Hyperbolic Geometry, and Nonexpansive Mappings. Dekker (1984)

[12] Iiduka, H.: Convex optimization over fixed point sets of quasi-nonexpansive and nonexpansive mappings in utility-based bandwidth

allocation problems with operational constraints. Journal of Computational and Applied Mathematics **282**, 225–236 (2015)

[13] Iiduka, H.: Parallel optimization algorithm for smooth convex optimization over fixed point sets of quasi-nonexpansive mappings. Journal of the Operations Research Society of Japan **58**, 330–352 (2015)

[14] Iiduka, H.: Convergence analysis of iterative methods for nonsmooth convex optimization over fixed point sets of quasi-nonexpansive mappings. Mathematical Programming **159**, 509–538 (2016)

[15] Iiduka, H.: Proximal point algorithms for nonsmooth convex optimization with fixed point constraints. European Journal of Operational Research **253**, 503–513 (2016)

[16] Kim, Y., Street, N.W., Menczer, F.: Meta-evolutionary ensembles. IEEE International Joint Conference on Neural Networks **123**, 2791–2796 (2002)

[17] The MathWorks, Inc.: Sample Data Sets-MATLAB. URL `https://www.mathworks.com/help/stats/_bq9uxn4.html`

[18] Martinez-Munoz, G., Hernandez-Lobato, D., Suarez, A.: An analysis of ensemble pruning techniques based on ordered aggregation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**, 245–259 (2009)

[19] Özöğür-Akyü, S., Windeatt, T., Smith, R.: Pruning of error correcting output codes by optimization of accuracy-diversity trade off. Machine Learning **22**, 1751–1783 (2015)

[20] Rockafellar, R.T.: Convex Analysis. Princeton University Press (1970)

[21] Schapire, R.E.: A brief introduction to boosting. pp. 1401–1406. 16th International Joint Conference on Artificial Intelligence, Burlington: Morgan Kauffman (1999)

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

[23] 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)

[24] Yin, X.C., Huang, K., Yang, C., Hao, H.W.: Convex ensemble learning with sparsity and diversity. Information Fusion **20**, 49–58 (2014)

[25] Zeidler, E.: Nonlinear Functional Analysis and Its Applications II/B. Nonlinear Monotone Operators. Springer (1985)

[26] Zhang, L., Zhou, W.D.: Sparse ensemble using weighted combination methods based on linear programming. Pattern Recognition **44**, 97–106 (2011)

[27] Zhang, Y., Burer, S., Street, W.N.: Ensemble pruning via semi-definite programming. Journal of Machine Learning Research **7**, 1315–1338 (2006)

[28] Zhou, Z.H., Wu, J.W., Tang, W.: Ensembling neural networks: many could be better than all? Artificial Intelligence **137**, 239–263 (2002)

[29] Zor, C., Windeatt, T., Kittler, J.: Ecoc matrix pruning using accuracy information. In: Multiple Classifier Systems, vol. 7872. 11th International Workshop, MCS 2013, Nanjing, China, May 15–17, 2013 Proceedings (2013)