IEICE TRANSACTIONS

on Information and Systems

DOI:10.1587/transinf.2023EDP7139

Publicized:2024/08/08

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Stochastic Dual Coordinate Ascent for Learning Sign Constrained Linear Predictors

Yuya TAKADA[†], **Rikuto MOCHIDA**[†], **Miya NAKAJIMA**[†], **Syun-suke KADOYA**[††], **Daisuke SANO**[†††], *and* **Tsuyoshi KATO**[†], *Nonmembers*

SUMMARY *Sign constraints* are a handy representation of domain-specific prior knowledge that can be incorporated to machine learning. This paper presents new stochastic dual coordinate ascent (SDCA) algorithms that find the minimizer of the empirical risk under the sign constraints. Generic surrogate loss functions can be plugged into the proposed algorithm with the strong convergence guarantee inherited from the vanilla SDCA. The prediction performance is demonstrated on the classification task for microbiological water quality analysis.

key words: *sign constraints, convex optimization, stochastic dual coordinate ascent, empirical risk minimization, microbiological water quality analysis.*

1. Introduction

Machine learning problems for linear prediction are often formulated as an *empirical risk minimization* (ERM) problem [9]. Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be input vectors in $\mathbb{R}^d$, let $\phi_1, \ldots, \phi_n : \mathbb{R} \to \mathbb{R}$ be convex loss functions, and let λ be a positive regularization constant. The ERM problem discussed in this paper is described as follows:

$$\begin{aligned} &\min \quad P(\boldsymbol{w}) \quad \text{wrt } \boldsymbol{w} \in \mathbb{R}^d, \\ &\text{where} \quad P(\boldsymbol{w}) := \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \phi_i(\langle \boldsymbol{w}, \boldsymbol{x}_i \rangle). \end{aligned} \tag{1}$$

Support vector machines (SVM) are recovered if we set the loss functions to the hinge loss $\phi_i(s) = \max(0, 1 - y_i s)$ where $y_i \in \{\pm 1\}$ are the class labels. Setting the loss functions to the log loss $\phi_i(s) = \log(1 + \exp(-y_i s))$, logistic regression is obtained. With y_i continuous labels, setting the square error loss function $\phi_i(s) = \frac{1}{2}(y_i - s)^2$ yields the ridge regression.

Recently, Tajima et al. [29] constrained the signs of the weights $\boldsymbol{w}$ to the linear SVM algorithm, and demonstrated the effectiveness of the *sign constraints* in the application to a biological sequence classification. The sign constraints are given to some of coefficients in the weight vector $\boldsymbol{w} = [w_1, \ldots, w_d]^\top$. For some pre-defined subset of indices $\mathcal{I}_{\geq} \subseteq [n]$, where $[n] := \{1, \ldots, n\}$, the non-negative constraints $w_h \geq 0$ are given for every $h \in \mathcal{I}_{\geq}$, and for another pre-defined subset $\mathcal{I}_{\leq} \subseteq ([n] \setminus \mathcal{I}_{\geq})$, the non-positive constraints $w_h \leq 0$ are given for every $h \in \mathcal{I}_{\leq}$.

The sign constraints explicitly avoid violation of the prior knowledge for the directions of correlations between features and class labels. Negative weight coefficients w_h are undesired if positive correlation between the hth features and the class label is known in advance. Nevertheless, without the sign constraints, a portion of coefficients w_h can be negative, which degrades the generalization performance. Similarly, positive weight coefficients are unfavorable if negative correlation to the class label is known in advance. Posing the sign constraints prevent the coefficients from falling into such an unfavorable region.

In this paper, we present new optimization algorithms for the sign-constrained ERM problems. The proposed algorithms solve a dual problem instead of minimizing the primal objective directly, which enables us to use a clear termination criterion which is the difference between the primal objective and the dual objective values. When the difference between the primal objective and the dual objective values is below a threshold, the primal objective gap is ensured to be smaller than the threshold. Tajima et al. employed the Frank-Wolfe algorithm [13] for a slightly different problem in which their algorithm is specialized to the sign-constrained ERM based on the classical non-smooth hinge loss function. The proposed algorithms are based on the stochastic dual coordinate ascent (SDCA) framework [27] to solve the sign-constrained ERM formulated with smooth loss functions, where being smooth means having a Lipschitz-continuous gradient. An attractive property of the proposed algorithms is a theoretical guarantee that ensures the exponential convergence [24] upper-bounding the number of iterations to attain a sufficiently small sub-optimality.

Besides the aforementioned work reported by Tajima et al., a large potential of this kind of prior knowledge suitable to the sign constraints may exist in many applications but may not have been discovered so far. For example, in the domain of water engineering, numerous prior studies, excluding [16], have overlooked this valuable reservoir of knowledge, despite the well-established associations between various water quality metrics and microbiological concentrations. Microbiological water quality datasets often exhibit limited size due to the considerable expenses associated with data collection. It has been observed that typical water quality metrics utilized in previous studies (e.g. [16]) are indeed associated

†The authors are with the Graduate School of Science and Technology, Gunma University, 1–5–1 Tenjin-chou, Kiryu, Gunma 376–8515, Japan

†††The author is with Dept. Civil. Environ. Eng., Tohoku University, 6–6–06 Aramaki-Aza-Aoba, Sendai, Miyagi 980–8579, Japan

††The author is with the University of Tokyo, 7–3–1 Hongo, Bunkyo-ku, Tokyo 113–8656, Japan

with microbiological concentrations in water. Nevertheless, these associations tend to be weak, which can result in contrasting correlations within a small dataset. Kato et al. [16] reported the effectiveness of incorporating sign constraints in regression tasks. In this study, our algorithm was applied to binary classification tasks for microbiological water quality analysis to demonstrate the power of the sign constraints.

This paper is organized as follows. Related work is discussed in the next section. In Section 3, the learning problem with the sign constraints is formulated and its dual problem is described. After the general SDCA framework is introduced in Section 4, the implementations of SDCA iterations for the sign constrained learning problem are presented in Section 5. The experimental results for runtime comparison and the application to microbiological water quality analysis are reported in Section 6, followed by the last section concluding this paper.

2. Related work

The sign constraints have been used widely in regression and classification. Readers familiar with machine learning may recognize the sign constrained regression as one of the important components of the non-negative matrix factorization [5], [8], [18], [21], [30]. Besides it, the sign constrained least square estimation is applied to widespread applications including non-negative image restoration [11], [19], [28], [31], face representation [10], [14], microbial analysis [3], pathogenic water quality analysis [16], image super-resolution [6], spectral analysis [33], tomographic imaging [23], and sound source localization [22]. For classification, Tajima et al. [29] developed the sign-constrained support vector machines. Fernandes et al. [7] studied other loss functions in a different formulation, which penalizes the weights violating prior knowledge instead of posing sign constraints. For the square error loss function, computationally stable and fast optimization algorithms are available [2], [17], [20]. For the hinge loss function, Tajima et al. developed a Frank-Wolfe optimization algorithm [13]. Meanwhile, without sign constraints, there are many stable optimization algorithms for generic empirical risk minimization [4], [15], [25], [26], [32]. However, to the best of our knowledge, algorithms for optimizing with generic loss functions under sign constraints have not been studied well so far.

3. Primal and dual problems

The goal of this work is to develop an optimization algorithm for the following constrained ERM problem:

$$\begin{aligned} \min \quad & P(\boldsymbol{w}) \quad \text{wrt } \boldsymbol{w} \in \mathbb{R}^d, \\ \text{subject to} \quad & \forall h \in \mathcal{I}_{\geq}, \quad w_h \geq 0, \\ & \forall h \in \mathcal{I}_{\leq}, \quad w_h \leq 0, \end{aligned} \tag{2}$$

The index sets $\mathcal{I}_{\geq}$ and $\mathcal{I}_{\leq}$ are assumed to be designed so that

$$\mathcal{I}_{\geq} \cup \mathcal{I}_{\leq} \subseteq [d] \quad \text{and} \quad \mathcal{I}_{\geq} \cap \mathcal{I}_{\leq} = \emptyset. \tag{3}$$

The remaining index set $\mathcal{I}_0 := [d] \setminus (\mathcal{I}_{\geq} \cup \mathcal{I}_{\leq})$ may be non-empty. Typically, the index of the bias term is included in $\mathcal{I}_0$. As previously discussed in Section 1, the sign constraints can be tailored based on prior knowledge. If we have advance knowledge that the hth feature x_h in the positive class tends to be larger than in the negative class, we can apply a non-negative constraint to w_h, and conversely, if the opposite relationship holds.

Hereinafter, we do not assume any non-positive constraints within the algorithmic description, as non-positive constraints can be effectively converted into non-negative constraints. This transformation involves negating the features $x_{h,i}$ for $h \in \mathcal{I}_{\leq}$ (i.e., $x_{h,i} \leftarrow -x_{h,i}$), where $x_{h,i}$ represents the hth component in the ith input vector for training, denoted as $\boldsymbol{x}_i \in \mathbb{R}^d$. After the learning process, the corresponding weight w_h is negated to reestablish weights that satisfy $w_h \leq 0$.

Denoting the feasible region by $\mathcal{S}$, the constrained problem in (2) can be rewritten as

$$\min \quad P(\boldsymbol{w}) \quad \text{wrt } \boldsymbol{w} \in \mathcal{S} \subseteq \mathbb{R}^d. \tag{4}$$

To express $\mathcal{S}$ simply, we use $\boldsymbol{\sigma} \in \{1, 0\}^d$ where its hth entry is given by $\sigma_h = 1$ for $h \in \mathcal{I}_{\geq}$ and $\sigma_h = 0$ for $h \in \mathcal{I}_0$. Then, the primal feasible region can be re-expressed as

$$\mathcal{S} := \{\boldsymbol{w} \in \mathbb{R}^d \mid \forall h \in [d],\ \sigma_h w_h \geq 0\}. \tag{5}$$

The optimization algorithm is based on SDCA framework that maximizes the *Fenchel dual* of the primal objective function. The Fenchel dual [1], say $D : \mathbb{R}^n \to \bar{\mathbb{R}}$, where $\bar{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$, is expressed as

$$D(\boldsymbol{\alpha}) := -\frac{1}{2\lambda n^2} \left\| \boldsymbol{\pi} \left(\sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i \right) \right\|^2 - \frac{1}{n} \sum_{i=1}^{n} \phi_i^*(-\alpha_i), \tag{6}$$

where $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_n]^\top \in \mathbb{R}^n$ is a dual variable vector, $\phi_i^* : \mathbb{R} \to \bar{\mathbb{R}}$ is the convex conjugate of ϕ_i, and $\boldsymbol{\pi}(\boldsymbol{v}) := \boldsymbol{v} - \max(\boldsymbol{0}, -\boldsymbol{\sigma} \odot \boldsymbol{v})$. Therein, the operator $\odot$ represents the Hadamard product. The vector-valued function $\boldsymbol{\pi} : \mathbb{R}^d \to \mathbb{R}^d$ is the projection operator onto $\mathcal{S}$. If we denote by $\pi_h(\boldsymbol{v})$ the hth entry of $\boldsymbol{\pi}(\boldsymbol{v})$, it holds that $\pi_h(\boldsymbol{v}) = 0$ if $h \in \mathcal{I}_{\geq}$ and $v_h \leq 0$; otherwise $\pi_h(\boldsymbol{v}) = v_h$. The derivation of the dual function $D(\boldsymbol{\alpha})$ is given in Appendix A.

Once the maximizer of $D(\boldsymbol{\alpha})$, denoted by $\boldsymbol{\alpha}_\star := \left[\alpha_1^\star, \ldots, \alpha_n^\star\right]^\top$, is found, the optimal solution to the primal problem (4) can be recovered by

$$\boldsymbol{w}_\star = \frac{1}{\lambda n} \boldsymbol{\pi} \left[\sum_{i=1}^{n} \alpha_i^\star \boldsymbol{x}_i \right]. \tag{7}$$

The loss function ϕ_i is assumed to be $1/\gamma$-smooth (i.e. $\nabla\phi_i$ is $1/\gamma$-Lipschitz continuous). For example, the log loss is 0.25-smooth. The quadratic hinge loss defined as

$$\phi_i(s) := \frac{1}{2}(\max\{0, 1 - y_i s\})^2 \tag{8}$$

and the smoothed hinge loss defined as

$$\phi_i(s) := \begin{cases} \frac{1-2y_i s}{2} & \text{for } s < 0, \\ \frac{1}{2}(\max\{0, 1 - y_i s\})^2 & \text{for } s \geq 0 \end{cases} \tag{9}$$

are both 1-smooth. The convex conjugates of $(1/\gamma)$-smooth convex functions are a γ-strongly convex function [12]. That is, $\forall \eta \in [0, 1]$,

$$\begin{aligned} &\eta\phi_i^*(-u) + (1-\eta)\phi_i^*(-\alpha) \\ &\geq \phi_i^*(-\eta u - (1-\eta)\alpha) + \frac{\gamma}{2}(u-\alpha)^2(1-\eta)\eta. \end{aligned} \tag{10}$$

The SDCA framework uses the above inequality rearranged as

$$\begin{aligned} &\phi_i^*(-\alpha) - \phi_i^*(-\alpha - \eta(u-\alpha)) \\ &\geq (\phi_i^*(-\alpha) - \phi_i^*(-u))\eta + \frac{\gamma}{2}(u-\alpha)^2(1-\eta)\eta. \end{aligned} \tag{11}$$

4. SDCA framework

SDCA updates only one randomly selected entry in the dual variable vector $\boldsymbol{\alpha}$ at every iteration. Let i be the index of the selected entry in $\boldsymbol{\alpha}$. Denote by $\Delta\alpha$ the difference of the randomly selected entry from the previous value: $\alpha_i^{(t)} := \alpha_i^{(t-1)} + \Delta\alpha$. For $i' \in [n] \setminus \{i\}$, the values of the dual variables are unchanged (i.e. $\alpha_{i'}^{(t)} := \alpha_{i'}^{(t-1)}$). Let

$$\begin{aligned} &\bar{\boldsymbol{w}}^{(t)} := \frac{1}{\lambda n}\sum_{i'=1}^{n} \alpha_{i'}^{(t)} \boldsymbol{x}_{i'}. \quad \text{and} \\ &\boldsymbol{w}^{(t)} := \boldsymbol{\pi}\left[\bar{\boldsymbol{w}}^{(t)}\right]. \end{aligned} \tag{12}$$

Once $\Delta\alpha$ is determined in each iteration, this vector $\bar{\boldsymbol{w}}^{(t)}$ can be updated with $O(d)$ costs, which can be seen by

$$\begin{aligned} \bar{\boldsymbol{w}}^{(t)} &= \frac{\alpha_i^{(t-1)} + \Delta\alpha}{\lambda n}\boldsymbol{x}_i + \frac{1}{\lambda n}\sum_{i' \in [n]\setminus\{i\}}^{n} \alpha_{i'}^{(t-1)} \boldsymbol{x}_{i'} \\ &= \bar{\boldsymbol{w}}^{(t-1)} + \frac{\Delta\alpha}{\lambda n}\boldsymbol{x}_i. \end{aligned} \tag{13}$$

For the simplicity of notation, we here shall drop the superscript $(t-1)$, to denote

$$\boldsymbol{w} := \boldsymbol{w}^{(t-1)}, \quad \boldsymbol{v}_0 := \bar{\boldsymbol{w}}^{(t-1)}, \quad \text{and} \quad \boldsymbol{\alpha} := \boldsymbol{\alpha}^{(t-1)}. \tag{14}$$

It is ideal to choose the maximizer of the function:

$$\begin{aligned} J_t^0(\Delta\alpha) &:= D(\boldsymbol{\alpha} + \Delta\alpha \boldsymbol{e}_i) - D(\boldsymbol{\alpha}) \\ &= \frac{\lambda}{2}\left\|\boldsymbol{\pi}\left[\boldsymbol{v}_0\right]\right\|^2 - \frac{\lambda}{2}\left\|\boldsymbol{\pi}\left[\boldsymbol{v}_0 + \frac{\Delta\alpha}{\lambda n}\boldsymbol{x}_i\right]\right\|^2 \\ &\quad + \frac{1}{n}\left(\phi_i^*(-\alpha) - \phi_i^*(-\alpha - \Delta\alpha)\right) \end{aligned} \tag{15}$$

where $\boldsymbol{e}_i$ is the unit vector with ith entry one. Since $\Delta\alpha$ is still in the argument of ϕ_i^*, finding the optimal $\Delta\alpha$ is complicated in general. To obtain a closed-form update rule, the range of $\Delta\alpha$ is restricted such that

Algorithm 1: SDCA algorithm for maximizing $D(\boldsymbol{\alpha})$.

```
begin
    Choose α^(0) s.t. α^(0) ∈ dom(−D);
    for t := 1 to T do
        Pick i randomly from {1, ..., n};
        η_t ∈ argmax_{η∈[0,1]} J_t^1(η);
        α^(t) := α^(t−1) − (∇φ(⟨w^(t−1), x_i⟩) + α_i^(t−1)) η_t e_i;
        Compute w̄^(t) and w^(t);
    end
end
```

$$\eta := -\frac{\Delta\alpha}{\alpha_i + \nabla\phi(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle)} \in [0, 1] \tag{16}$$

if $\alpha_i + \nabla\phi(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle) \neq 0$; otherwise $\Delta\alpha := 0$. Hereinafter, we discuss only the non-trivial case of $u := -\nabla\phi(\langle \boldsymbol{w}, \boldsymbol{x}_i\rangle) \neq \alpha_i$, where α_i is the ith entry in $\boldsymbol{\alpha}^{(t-1)}$. Then, $\Delta\alpha = q\eta$ where $q := u - \alpha_i$. Let

$$\boldsymbol{v}_q := \frac{q}{\lambda n}\boldsymbol{x}_i. \tag{17}$$

By applying the inequality (11), $J_t^0(q\eta)$ is bounded from below as

$$\begin{aligned} J_t^0(q\eta) &= \frac{\lambda}{2}\left\|\boldsymbol{\pi}\left[\boldsymbol{v}_0\right]\right\|^2 - \frac{\lambda}{2}\left\|\boldsymbol{\pi}\left[\boldsymbol{v}_0 + \eta\boldsymbol{v}_q\right]\right\|^2 \\ &\quad + \frac{1}{n}\left(\phi^*(-\alpha_i) - \phi^*(-\alpha_i - q\eta)\right) \\ &\geq \frac{\lambda}{2}\left\|\boldsymbol{\pi}\left[\boldsymbol{v}_0\right]\right\|^2 - \frac{\lambda}{2}\left\|\boldsymbol{\pi}\left[\boldsymbol{v}_0 + \eta\boldsymbol{v}_q\right]\right\|^2 \\ &\quad + a_{\text{offs}}\eta^2 + b_{\text{offs}}\eta =: J_t^1(\eta) \end{aligned} \tag{18}$$

where

$$\begin{aligned} &a_{\text{offs}} := -\frac{q^2\gamma}{2n}, \quad \text{and} \\ &b_{\text{offs}} := \frac{\phi^*(-\alpha) - \phi^*(-u) + 0.5q^2\gamma}{n}. \end{aligned} \tag{19}$$

The lower bound J_t^1 is more amenable than J_t^0 because no loss function appears in J_t^1 any more. The SDCA for learning under sign constraints is summarized in Algorithm 1. The exponential convergence of SDCA is still guaranteed even if J_t^1 is maximized instead of J_t^0 [27].

Theorem 1: Let $R := \max_{i\in[n]}\|\boldsymbol{x}_i\|$, $h_{\text{P}}^{(t)} := P(\boldsymbol{w}^{(t)}) - P(\boldsymbol{w}_\star)$ and $h_{\text{D}}^{(t)} := D(\boldsymbol{\alpha}_\star) - D(\boldsymbol{\alpha}^{(t)})$. For any $\epsilon_{\text{P}} > 0$, it holds that $\mathbb{E}\left[h_{\text{P}}^{(t)}\right] \leq \epsilon_{\text{P}}$ if Algorithm 1 is run for

$$t \geq \frac{\lambda n\gamma + R^2}{\lambda\gamma}\log\left(\frac{h_{\text{D}}^{(0)}}{\epsilon_{\text{P}}}\frac{\lambda n\gamma + R^2}{\lambda\gamma}\right). \tag{20}$$

The proof of this theorem is given in Appendix B. The bound of the vanilla SDCA algorithm is essentially same as

the above bound. In the original bound presented in [27], $h_D^{(0)}$ is replaced to 1 by assuming that $\boldsymbol{\alpha}^{(0)} = \mathbf{0}$ and $\phi_i(0) \leq 1$. In [27], an idea for using the hot-starting is discussed. This idea can also be applied to the proposed algorithm. In this case, the initial dual variables may be non-zero with high probability.

In the next section, how to implement Line 5 in Algorithm 1 shall be discussed.

5. Implementations for SDCA iteration

In this section, an algorithm for finding the maximizer of $J_t^1(\eta)$ is presented. A key ingredient found in this study is the fact that J_t^1 is a piecewise concave quadratic function. This finding enabled us to develop efficient algorithms for the update rule. Below, an explicit form of the piecewise quadratic function shall be presented (Subsection 5.1), followed by descriptions of two algorithms to find the maximizer of $J_t^1(\eta)$ (Subsections 5.2 and 5.3).

5.1 Piecewise quadratic form

Denote by $v_{h,0}$ and $v_{h,q}$ the hth entries of $\boldsymbol{v}_0$ and $\boldsymbol{v}_q$, respectively. Let $\mathcal{I}_0 := \{h \in [d] \,|\, \sigma_h = 0\}$. Define $\boldsymbol{\theta} := \left[\theta_1, \ldots, \theta_{d_t}, \theta_{d_t+1}\right]^\top$ such that $0 = \theta_1 < \cdots < \theta_{d_t+1} = 1$ where $\theta_1, \ldots, \theta_{d_t}, \theta_{d_t+1}$ are the elements of a set $\Theta \subset \mathbb{R}$ such as $\mathrm{Card}[\Theta] = d_t + 1$ defined as

$$\begin{aligned}\Theta :=& \{0, 1\} \cup \{\theta \in (0, 1) \\ & \,|\, \exists h \in \mathcal{I}_{\geq} \text{ s.t. } v_{h,0} = -\theta v_{h,q} \neq 0\}.\end{aligned} \tag{21}$$

The element θ_k for $k \in \{2, \ldots, d_t\}$ is the position at which for some $h \in \mathcal{I}_{\geq}$ the affine function $\eta \mapsto v_{h,0} + \eta v_{h,q}$ crosses the horizontal axis. Figure 1 shows a numerical example of the affine functions where $d = 3$, $\mathcal{I}_{\geq} = \{1, 2, 3\}$, $\boldsymbol{v}_0 = [0.5, 0.75, -0.5]^\top$, and $\boldsymbol{v}_q = [0.5, -1, 1]^\top$. From the definition of Θ, we have $d_t = 3$, $\theta_1 = 0$, $\theta_2 = 0.5$, $\theta_3 = 0.75$, and $\theta_4 = 1$. It is observed that the affine function $\eta \mapsto v_{3,0} + \eta v_{3,q}$ crosses the horizontal axis at $\eta = \theta_2$, and the affine function $\eta \mapsto v_{2,0} + \eta v_{2,q}$ crosses the horizontal axis at $\eta = \theta_3$.

Let us define index sets, for $k \in [d_t]$,

$$\begin{aligned}\mathcal{H}_k := \mathcal{I}_0 \cup \{h \in \mathcal{I}_{\geq} \,|\, \\ 2v_{h,0} + (\theta_k + \theta_{k+1}) v_{h,q} > 0\}.\end{aligned} \tag{22}$$

In the case of the example depicted in Figure 1, the index sets are $\mathcal{H}_1 = \{1, 2\}$, $\mathcal{H}_2 = \{1, 2, 3\}$, and $\mathcal{H}_3 = \{1, 3\}$, from the definition (22). For $h \in \mathcal{H}_k \cap \mathcal{I}_{\geq}$, the affine functions $\eta \mapsto v_{h,0} + \eta v_{h,q}$ are over the horizontal axis. Namely, it holds that

$$\forall \eta \in (\theta_k, \theta_{k+1}), \quad \forall h \in \mathcal{H}_k, \quad v_{h,0} + \eta v_{h,q} > 0, \tag{23}$$

which leads to $\forall \eta \in (\theta_k, \theta_{k+1})$,

$$[\boldsymbol{\pi}(\boldsymbol{v}_0 + \eta \boldsymbol{v}_q)]_h = \begin{cases} v_{h,0} + \eta v_{h,q} & \text{for } h \in \mathcal{H}_k, \\ 0 & \text{for } h \notin \mathcal{H}_k \end{cases} \tag{24}$$

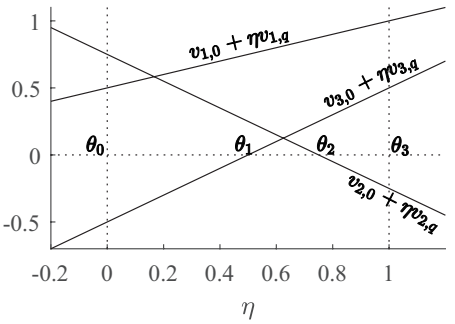


Fig. 1 Example of affine functions $\eta \mapsto v_{h,0} + \eta v_{h,q}$.

where $[\boldsymbol{\pi}(\boldsymbol{v}_0 + \eta \boldsymbol{v}_q)]_h$ is the hth entry in the d-dimenstional vector $\boldsymbol{\pi}(\boldsymbol{v}_0 + \eta \boldsymbol{v}_q)$. The vector $\boldsymbol{\theta}$ and the sets $\mathcal{H}_k$ for $k \in [d_t]$ result in a piecewise quadratic expression for the function J_t^1:

$$\forall \eta \in [\theta_k, \theta_{k+1}], \quad J_t^1(\eta) = a_k \eta^2 + b_k \eta \tag{25}$$

where a_k and b_k are given by

$$\begin{aligned} a_k &= a_{\text{offs}} - \frac{\lambda}{2} \sum_{h \in \mathcal{H}_k} v_{h,q}^2, \quad \text{and} \\ b_k &= b_{\text{offs}} - \lambda \sum_{h \in \mathcal{H}_k} v_{h,q} v_{h,0}. \end{aligned} \tag{26}$$

5.2 $O(d^2)$ implementation

Due to the concavity and the differentiability of J_t^1, one of the maximizers of $J_t^1(\eta)$, denoted by $\eta_\star$, can be found as follows.

- If $\nabla J_t^1(0) = b_1 \leq 0$, then $\eta_\star = 0$;
- if $\nabla J_t^1(1) = 2a_{d_t+1} + b_{d_t+1} \geq 0$, then $\eta_\star = 1$;
- otherwise, there exists $k_\star \in [d_t]$ such that the interval $[\theta_{k_\star}, \theta_{k_\star+1}]$ contains a maximizer $\eta_\star = -0.5 b_{k_\star} / a_{k_\star}$.

The interval index $k_\star$ in the third case (i.e. $2a_{d_t+1} + b_{d_t+1} < 0 < b_1$) can be found by checking every interval, because it holds that $\nabla J_t^1(\theta_{k_\star}) \geq 0 \geq \nabla J_t^1(\theta_{k_\star+1})$ due to the differentiability of J_t^1. Combining this discussion and the aforementioned observations, each iteration of SDCA can be implemented as follows.

1. Pick $i \in [n]$ at random;	$O(1)$.
2. Compute $\boldsymbol{v}_0$ and $\boldsymbol{v}_q$;	$O(d)$.
3. Determine Θ;	$O(d)$.
4. Sort the elements in Θ;	$O(d \log d)$.
5. Compute $\mathcal{H}_k$ for $k \in [d_t]$;	$O(d^2)$.
6. Compute coefficients (a_k, b_k) for $k \in [d_t]$;	$O(d^2)$.
7. Find the maximizer $\eta_\star$;	$O(d)$.
8. $\Delta\alpha = q\eta_\star$;	$O(1)$.
9. Compute $\bar{\boldsymbol{w}}^{(t)}$ by (13);	$O(d)$.

This implementation enables each iteration to run within $O(d^2)$ computational cost. The most heavy steps in this

implementation are the step computing index sets $\mathcal{H}_k$ (i.e. Step 5) and the step computing coefficients a_k, b_k (i.e. Step 6), both of which pays $O(d^2)$ cost. These time complexities are derived as follows. Observe that the number of pieces of the piecewise quadratic function is bounded as $d_t \leq \mathrm{Card}(\mathcal{I}_{\geq}) + 2 \leq d + 2 = O(d)$. For $k \in [d_t]$, each $\mathcal{H}_k$ is computed with $O(d)$ time since $\mathcal{H}_k \subseteq [d]$. Hence, it is proved that the time complexity of Step 5 is $O(d^2)$. Since $\mathrm{Card}(\mathcal{H}_k) = O(d)$, computation of $2d_t (= O(d))$ coefficients, $a_1, b_1, \ldots, a_{d_t}, b_{d_t}$, using (26) consumes $O(d^2)$ cost in total.

Meanwhile, we found another algorithm that cut down the time complexity to a linear cost if ignoring the logarithmic term. The linear-time algorithm shall be presented below.

5.3 $O(d \log d)$ implementation

Here, another algorithm that exactly maximizes $J_t^1(\eta)$ with respect to $\eta \in [0, 1]$ is presented. The theoretical time complexity of the algorithm given in Subsection 5.2 is $O(d^2)$, whereas the time complexity of the algorithm presented below is reduced to $O(d \log d)$. Defining

$$\mathcal{H}_{k,\mathrm{in}} := \mathcal{H}_k \setminus \mathcal{H}_{k-1} \quad \text{and} \quad \mathcal{H}_{k,\mathrm{out}} := \mathcal{H}_{k-1} \setminus \mathcal{H}_k \tag{27}$$

allows us to recursively express the coefficients of the piecewise quadratic functions as $\forall k \geq 2$,

$$\begin{aligned} a_k &:= a_{k-1} - \frac{\lambda}{2} \sum_{h \in \mathcal{H}_{k,\mathrm{out}}} v_{h,q}^2 + \frac{\lambda}{2} \sum_{h \in \mathcal{H}_{k,\mathrm{in}}} v_{h,q}^2, \\ b_k &:= b_{k-1} - \lambda \sum_{h \in \mathcal{H}_{k,\mathrm{out}}} v_{h,q} v_{h,0} + \lambda \sum_{h \in \mathcal{H}_{k,\mathrm{in}}} v_{h,q} v_{h,0}, \end{aligned} \tag{28}$$

To use (28) to compute a_k and b_k, the sets $\mathcal{H}_{k,\mathrm{in}}$ and $\mathcal{H}_{k,\mathrm{out}}$ as well as $\mathcal{H}_1$ are required beforehand. The set $\mathcal{H}_1$ can be obtained within $O(d)$ by checking whether one of the following three conditions is satisfied:

$$\begin{aligned} &\text{(i) } h \in \mathcal{I}_0; \quad \text{(ii) } v_{h,0} > 0; \\ &\text{(iii) } v_{h,0} = 0 \quad \text{and} \quad v_{h,q} > 0. \end{aligned} \tag{29}$$

Namely, if $h \in [d]$ satisfies one of the three above conditions, then $h \in \mathcal{H}_1$; otherwise $h \notin \mathcal{H}_1$. We now discuss how to compute $\mathcal{H}_{k,\mathrm{in}}$ and $\mathcal{H}_{k,\mathrm{out}}$. To this end, we first compute $\tilde{\theta}_h^\circ := -\frac{v_{h,0}}{v_{h,q}}$ for all $h \in \mathcal{I}_{\geq}$. The $(d_t - 1)$ end points $\theta_2, \ldots, \theta_{d_t}$ are then obtained by sorting the values of $\tilde{\theta}_h^\circ$, eliminating the values outside the open interval $(0, 1)$, and excluding duplicate values. During the process for computing $\boldsymbol{\theta}$, the sets $\mathcal{H}_{k,\mathrm{in}}$ and $\mathcal{H}_{k,\mathrm{out}}$ for $k \in \{2, \ldots, d_t\}$ can be computed simultaneously as

$$\begin{aligned} \mathcal{H}_{k,\mathrm{in}} &= \left\{ h \in \mathcal{I}_{\geq} \,\middle|\, \theta_k = \tilde{\theta}_h^\circ,\ v_{h,q} > 0 \right\}, \quad \text{and} \\ \mathcal{H}_{k,\mathrm{out}} &= \left\{ h \in \mathcal{I}_{\geq} \,\middle|\, \theta_k = \tilde{\theta}_h^\circ,\ v_{h,q} < 0 \right\}. \end{aligned} \tag{30}$$

From these discussions, the $O(d^2)$ implementation given in Subsection 5.2 can be modified as follows.

Table 1 Features and sign constraints for four datasets. Check-marked features are contained in the corresponding dataset. Sign constraints were given as described in the rightmost column where '≥ 0' and '≤ 0' means the non-negative and non-positive constraints, respectively.

Feature	Sapporo	NY top	NY bottom	Indian	Constraint
WT	✓				≥ 0
pH	✓	✓	✓		≤ 0
EC	✓	✓	✓	✓	
DO	✓	✓	✓	✓	≤ 0
SS	✓				≥ 0
BOD	✓	✓	✓	✓	
TN	✓				≥ 0
TP	✓				≥ 0
FR	✓				≤ 0
TC		✓	✓	✓	≥ 0
Nitro				✓	≥ 0

1. Pick $i \in [n]$ at random;	$O(1)$.
2. Compute $\boldsymbol{v}_0$ and $\boldsymbol{v}_q$;	$O(d)$.
3. Compute $\mathcal{H}_1$;	$O(d)$.
4. Compute $\tilde{\theta}_h^\circ$ for $h \in \mathcal{I}_{\geq}$;	$O(d)$.
5. Compute $(\mathcal{H}_{k,\mathrm{in}}, \mathcal{H}_{k,\mathrm{in}})$ and θ_k for $k \in [d_t]$;	$O(d \log d)$.
6. Compute coefficients (a_k, b_k) for $k \in [d_t]$;	$O(d)$.
7. Find the maximizer $\eta_\star$;	$O(d)$.
8. $\Delta\alpha = q\eta_\star$;	$O(1)$.
9. Compute $\bar{\boldsymbol{w}}^{(t)}$ by (13);	$O(d)$.

Step 5 takes $O(d \log d)$ cost for sorting $\tilde{\theta}_h^\circ$ because the number of values to be sorted is $\mathrm{Card}(\mathcal{I}_{\geq}) = O(d)$. The computational cost for Line 6 is $O(d)$ since the relationship

$$\bigcup_{k=2}^{d_t} \mathcal{H}_{k,\mathrm{in}} \subseteq \mathcal{I}_+ \subseteq [d] \quad \text{and} \quad \bigcup_{k=2}^{d_t} \mathcal{H}_{k,\mathrm{out}} \subseteq \mathcal{I}_+ \subseteq [d] \tag{31}$$

leads to the fact that an upper bound of the number of the total terms in (28) for all $k \in \{2, \ldots, d_t\}$ is $4d$. Thus, it can be shown that each iteration of SDCA can be done within $O(d \log d)$ computation.

6. Experiments

6.1 Pattern recognition performance

We conducted experiments to demonstrate the effects of the sign constraints on the pattern recognition performance. For a pattern recognition task, we selected the microbiological water quality analysis. We used four water quality datasets named *Sapporo*, *NY top*, *NY bottom*, and *Indian*. The dataset Sapporo was provided in the supplement of [16], and contained $n_{\mathrm{tot}} := 177$ examples, each consisting of a target variate E.coli and nine feature variates WT, pH, EC, DO, SS, BOD, TN, TP, and FR, where the abbreviations are referred to [16]. The task was to predict whether E.coli exceeds 500 MPN or not. Then, 88 positive examples and 89 negative examples were obtained. NY top, NY bottom and Indian

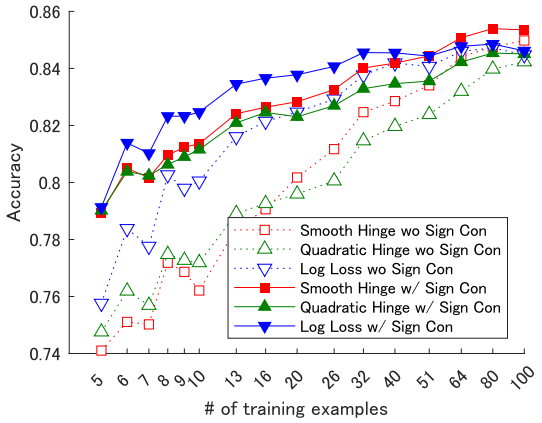


Fig. 2 Prediction performance on Sapporo dataset. The solid curves indicate the accuracies of predictors optimized under sign constraints, and the dashed curves are those the accuracies when sign constraints are not employed. The markers of the squares, the upward-pointing triangles, and the downward-pointing triangles represents the smoothed hinge loss, the quadratic hinge loss, and the log loss, respectively.

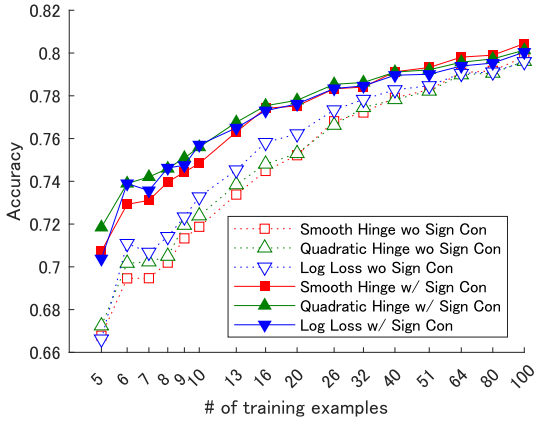


Fig. 3 Prediction performance on NY top dataset.

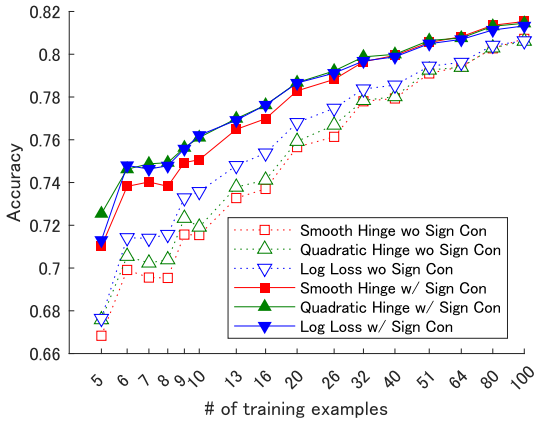


Fig. 4 Prediction performance on NY bottom dataset.

were provided by kaggle.com. The three datasets contain 534, 523, and 896 examples, respectively. Each example has a target variate FC and five feature variates. The positive and negative class variates, respectively, were given to FC

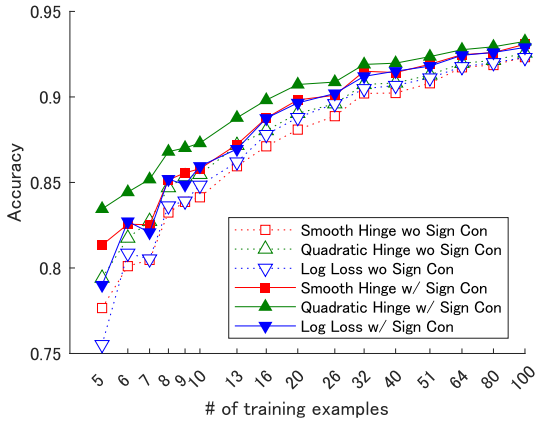


Fig. 5 Prediction performance on Indian dataset.

Table 2 Average number of pieces in the piecewise quadratic function $J_t^1 : [0, 1] \to \mathbb{R}$.

d	1st epoch	2nd epoch	3rd epoch	4th epoch	5th epoch
1,000	10.21	2.49	1.78	1.69	1.48
3,163	31.88	2.44	1.63	1.08	0
10,000	97.83	1.28	0.2	0	0
31,623	293.44	1	0	0	0
100,000	960.93	1	0	0	0

over and below the median, to pose a binary classification problem. The sign constraints were imposed as described in Table 1. Three loss functions, the *smoothed hinge loss*, the *quadratic hinge loss*, and the *log loss*, were examined. For each loss function, the conventional learning and the sign-constrained learning were performed. Then, six linear predictors were obtained in total. Accuracy (i.e. the sum of true positives and true negatives over the size of testing dataset) was used for the performance criterion. The number of training examples, n, was varied from 5 to 100. The n examples were picked at random from each dataset in a stratified manner. The n examples were fed to the six learning methods to get six predictors. The remaining $(n_{\mathrm{tot}} - n)$ examples were used for testing. This procedure was repeated 200 times.

The averages of the 200 accuracies obtained for the four water quality datasets were plotted against the size of the training dataset, say n, in Figure 2, Figure 3, Figure 4, and Figure 5, respectively. For all four datasets and all three loss functions, the sign constraints improved the prediction performance. In particular, the improvement was more significant when training examples were fewer. Sign constraints represent a sort of the domain-specific prior knowledge, and explicitly prevent the learning from violating the prior knowledge. Without sign constraints, when the sample size is small, the signs of weights in linear predictors may often be flipped from the true signs of correlations between the features and the class label. The improvement of the generalization performance must be the effect of the sign constraints that avoided the inversion of the weight signs.

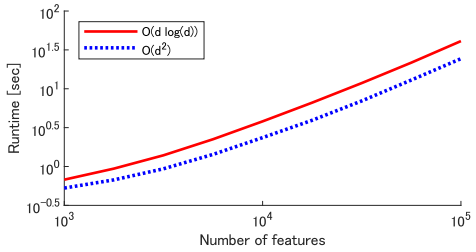


Fig. 6 Runtime for one epoch.

6.2 Runtimes of $O(d \log d)$ and $O(d^2)$ algorithms

The two algorithms, presented in Subsections 5.2 and 5.3, were implemented with Cython 3.0.0a11 and run on a Linux machine equipped with Core i7-12700K and two 16GB DDR4 SDRAM. Feature vectors were generated with uniform distribution $U(-1, 1)$ and normalized. Binary class labels were generated at random with equal probabilities. The size of training examples was fixed to $n = 500$. The number of features was varied from $d = 10^3$ to 10^5.

Figure 6 shows the runtimes consumed in one epoch. In conflict with the theoretical analysis, the two algorithms seemed to have the same time complexity, and the $O(d^2)$ algorithm always ran faster than the $O(d \log d)$ algorithm. To analyze why the inconsistency between the theory and the actual runtime happened, the numbers of pieces in the piecewise quadratic functions $J_t^1(\eta)$, say d_t, were counted, where we set $d_t = 0$ after the convergence $(P(\boldsymbol{w}(\boldsymbol{\alpha}^{(t)})) - D(\boldsymbol{\alpha}^{(t)}) < 10^{-6})$. The average numbers within each epoch were reported in Table 2. It was observed that the average numbers of pieces were around 1% of the number of features at the first epoch, and were less than 2.5 after the first epoch. It suggested that the actual number of pieces was much smaller than the number of features. Nevertheless, in our theoretical analysis, we used $d_t = O(d)$ which is based on a loose bound $d_t \leq \mathrm{Card}(\mathcal{I}_{\geq}) \leq d$, resulting in the theoretical time complexity $O(d^2)$ for the implementation presented in Subsection 5.2. The difference of the upper bound from the actual number of pieces yielded the inconsistency between the theory and the actual runtime.

6.3 Convergence performance

To assess the rapid convergence of the proposed SDCA algorithm, we conducted convergence analysis on three distinct datasets: `Magic04`, `Segment`, and `Waveform`. We then compared the performance of our algorithm with projected stochastic gradient descent (PSGD), updating the solution using the equation $\boldsymbol{w}^{(t+1)} := \boldsymbol{\pi}(\boldsymbol{w}^{(t)} - \eta \nabla P(\boldsymbol{w}^{(t)}))$, where η represents the step size.

The dataset characteristics are as follows: `Magic04` contains 19,024 examples with 10 features, `Segment` has 2,310 examples with 19 features, and `Waveform` comprises 5,000 examples with 21 features. We experimented with various step sizes for PSGD, specifically $\eta = 10^0$, $\eta = 10^{-1}$, $\eta = 10^{-2}$, and $\eta = 10^{-3}$. The regularization constant was set to $\lambda = 1/n$. In our optimization process, we employed the log-loss function ϕ_i. We imposed non-negative constraints on half of the randomly selected features and non-positive constraints on the remaining half. Objective errors were monitored at each iteration. It is worth noting that assessing the true primal objective error $P(\boldsymbol{w}_\star)$ is often impractical due to its unknown nature. To approximate this error, we executed the proposed algorithm and evaluated the dual objective value at the 1000th iteration, denoted as T'. While $D(\boldsymbol{\alpha}^{(T')})$ may slightly underestimate the true minimal primal objective value, we considered the quantity $P(\boldsymbol{w}) - D(\boldsymbol{\alpha}^{(T')})$ for assessing the primal objective error. In all the experiments we report, we consistently observed that $P(\boldsymbol{w}^{(T')}) - D(\boldsymbol{\alpha}^{(T')})$ remained below 10^{-7} when using the proposed algorithm, ensuring that the absolute difference between the true and approximate primal objective errors was bounded by 10^{-7}.

In Figure 7, we present three panels that illustrate the evolution of primal objective errors throughout multiple epochs for the datasets `Magic04`, `Segment`, and `Waveform`. The proposed algorithm achieved an accuracy of 10^{-5} after approximately 1.9, 2.7, and 3.7 epochs for the respective datasets. In contrast, PSGD did not reach this level of accuracy when using step sizes of $\eta = 10^0$ and $\eta = 10^{-3}$ for any of the datasets, due to step sizes being excessively large and small, respectively. With a step size of $\eta = 10^{-2}$, PSGD eventually attained the 10^{-5} accuracy level, but only after a considerable number of epochs —– 429, 306, and 536 for the three datasets, respectively. These numbers of epochs were over 100 times greater than those needed by the proposed algorithm. Moreover, with a step size of $\eta = 10^{-1}$, PSGD reached the 10^{-5} accuracy level after 79 and 109 epochs for `Magic04` and `Segment`, respectively. However, for `Waveform`, PSGD did not achieve this accuracy level even after more than 10^3 epochs. These findings strongly support the conclusion that the proposed algorithm converges to the optimal solution significantly faster than PSGD.

In our implementation, the average time per epoch for the proposed algorithm across the three datasets is 0.488, 0.0556, and 0.146 seconds, respectively. In comparison, PSGD averages 0.141, 0.0231, and 0.0428 seconds per epoch. While the per-epoch computation time of our algorithm exceeds that of PSGD, it compensates by requiring significantly fewer epochs to achieve an accurate solution. This efficiency results in a reduced overall time to reach an accurate solution compared to PSGD.

7. Conclusions

In this paper, new algorithms for ERM under the sign constraints were presented. Tajima et al. developed the Frank-Wolfe optimization algorithm for learning SVM under sign constraints. The algorithm developed in this study extends the class of ERM problems so that an arbitrary smooth and convex loss function can be employed. The optimization algorithm is based on the SDCA framework, which inherits

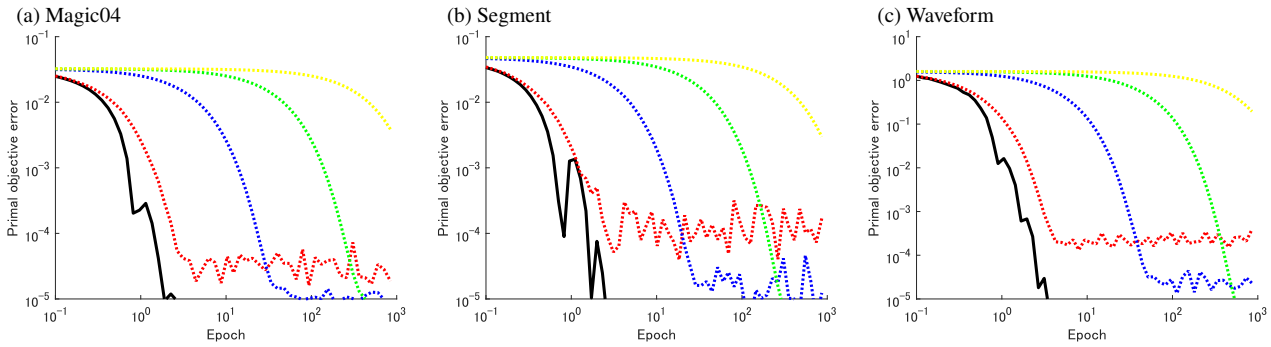


Fig. 7 Convergence behaviors on three datasets. Black solid curve: proposed algorithm, dashed curves: PSGD. Red, blue, green and yellow dash curves are obtained with step sizes 10^0, 10^{-1}, 10^{-2}, 10^{-3}, respectively.

a favorable property that guarantees the exponential convergence which is superior to the convergence rate of the Frank-Wolfe algorithm. The effects of the sign constraints on the pattern recognition were demonstrated with simulation experiments on microbiological water quality analysis using real-world data. Actual runtimes of the two SDCA algorithms developed in this study were compared, which suggested that the simpler $O(d^2)$ algorithm runs fast enough compared to the $O(d \log d)$ algorithm.

We expect that a significant untapped reservoir of pre-existing knowledge, amenable to sign constraints, holds promise for numerous applications but remains yet to be fully explored. Tajima et al. identified a viable application in the realm of bioinformatics [29], while Kato et al. discovered its utility in the field of water engineering [16]. In this paper, we revisited the water engineering problem to exemplify the efficacy of sign constraints. Subsequent research endeavors will involve exploring additional domains amenable to sign constraints.

Appendix A: Deriving the dual function

Lemma 1: $\forall \boldsymbol{v} := [v_1, \ldots, v_d]^\top \in \mathbb{R}^d$,

$$\inf_{\boldsymbol{w} \in \mathcal{S}} \|\boldsymbol{w} - \boldsymbol{v}\|^2 - \|\boldsymbol{v}\|^2 = -\|\boldsymbol{\pi}(\boldsymbol{v})\|^2. \tag{A·1}$$

Proof of Lemma 1: Denote by $\mathbb{R}_\geq$ the set of non-negative real numbers. We have

$$\min_{w_h \in \mathbb{R}} (w_h - v_h)^2 - v_h^2 = -v_h^2 \tag{A·2}$$

for $h \in \mathcal{I}_0$, and

$$\min_{w_h \in \mathbb{R}_\geq} (w_h - v_h)^2 - v_h^2 = -(v_h - \max\{0, -v_h\})^2 \tag{A·3}$$

for $h \in \mathcal{I}_\geq$. Hence,

$$\begin{aligned}
\text{LHS of (A·1)} &= \sum_{h \in \mathcal{I}_0} \min_{w_h \in \mathbb{R}} (w_h - v_h)^2 - v_h^2 \\
&\quad + \sum_{h \in \mathcal{I}_\geq} \min_{w_h \in \mathbb{R}_\geq} (w_h - v_h)^2 - v_h^2 \\
&= -\sum_{h \in \mathcal{I}_0} v_h^2 - \sum_{h \in \mathcal{I}_\geq} (v_h - \max\{0, -v_h\})^2 \\
&= -\|\boldsymbol{\pi}(\boldsymbol{v})\|^2 = \text{RHS of (A·1)}
\end{aligned} \tag{A·4}$$

where LHS and RHS are the abbreviations of left and right hand sides, respectively.

q.e.d. of Lemma 1

We now derive the dual function in (6). The primal optimization problem (4) is equivalent to the following constrained problem:

$$\begin{aligned}
\min \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^n \phi_i(z_i) \\
\text{wrt} \quad & \boldsymbol{w} \in \mathcal{S},\ \boldsymbol{z} := [z_1, \ldots, z_n]^\top \in \mathbb{R}^n \\
\text{subject to} \quad & \forall i \in [n], \quad z_i = \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle .
\end{aligned} \tag{A·5}$$

Letting $\bar{\boldsymbol{w}}(\boldsymbol{\alpha}) := (\lambda n)^{-1} \sum_{i=1}^n \boldsymbol{x}_i \alpha_i$, the Lagrangian function is

$$\begin{aligned}
&L(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\alpha}) \\
&:= \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{n}\sum_{i=1}^n \phi_i(z_i) + \frac{1}{n}\sum_{i=1}^n (z_i - \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle)\alpha_i \\
&= \frac{\lambda}{2}\|\boldsymbol{w}\|^2 - \lambda \langle \boldsymbol{w}, \bar{\boldsymbol{w}}(\boldsymbol{\alpha}) \rangle + \frac{1}{n}\sum_{i=1}^n (\phi_i(z_i) + \alpha_i z_i) \\
&= \frac{\lambda}{2}\left(\|\boldsymbol{w} - \bar{\boldsymbol{w}}(\boldsymbol{\alpha})\|^2 - \|\bar{\boldsymbol{w}}(\boldsymbol{\alpha})\|^2 \right) + \frac{1}{n}\sum_{i=1}^n (\phi_i(z_i) + \alpha_i z_i) .
\end{aligned} \tag{A·6}$$

For the convex conjugate of the loss functions ϕ_i, we have

$$\begin{aligned}
-\phi_i^*(-\alpha_i) &= -\sup_{z_i \in \mathbb{R}} (-\alpha_i z_i - \phi_i(z_i)) \\
&= \inf_{z_i \in \mathbb{R}} (\alpha_i z_i + \phi_i(z_i)) .
\end{aligned} \tag{A·7}$$

The dual function is then obtained as

$$\begin{aligned} D(\boldsymbol{\alpha}) &= \inf_{\boldsymbol{w}\in\mathcal{S}} \inf_{\boldsymbol{z}\in\mathbb{R}^n} L(\boldsymbol{w},\boldsymbol{z},\boldsymbol{\alpha}) \\ &= \frac{\lambda}{2} \inf_{\boldsymbol{w}\in\mathcal{S}} \left(\|\boldsymbol{w}-\bar{\boldsymbol{w}}(\boldsymbol{\alpha})\|^2 - \|\bar{\boldsymbol{w}}(\boldsymbol{\alpha})\|^2 \right) \\ &\quad + \frac{1}{n}\sum_{i=1}^n \inf_{z_i\in\mathbb{R}} (\phi_i(z_i)+\alpha_i z_i) \\ &= -\frac{\lambda}{2}\|\boldsymbol{\pi}(\bar{\boldsymbol{w}}(\boldsymbol{\alpha}))\|^2 - \frac{1}{n}\sum_{i=1}^n \phi_i^*(-\alpha_i) \\ &= -\frac{1}{2\lambda n^2}\left\| \boldsymbol{\pi}\left(\sum_{i=1}^n \alpha_i \boldsymbol{x}_i \right) \right\|^2 - \frac{1}{n}\sum_{i=1}^n \phi_i^*(-\alpha_i), \end{aligned} \tag{A·8}$$

where the third equality follows from Lemma 1 and (A·7).

Appendix B: Proof of Theorem 1

The proof of Theorem 1 given here mainly follows the proof technique used for the vanilla SDCA in [27]. The following lemmas are important ingredients.

Lemma 2: For any $\boldsymbol{v}_0, \boldsymbol{v}_q \in \mathbb{R}^d$ and $\eta \in [0,1]$, it holds that

$$\begin{aligned} &\|\boldsymbol{\pi}[\boldsymbol{v}_0]\|^2 - \|\boldsymbol{\pi}[\boldsymbol{v}_0+\eta\boldsymbol{v}_q]\|^2 \\ &\quad + 2\eta\left\langle \boldsymbol{\pi}[\boldsymbol{v}_0], \boldsymbol{v}_q \right\rangle + \eta^2\|\boldsymbol{v}_q\|^2 \geq 0. \end{aligned} \tag{A·9}$$

Lemma 3: Let $F_i^{(t-1)} := \phi_i(z_i^{(t-1)}) + \phi_i^*(-\alpha_i^{(t-1)}) + z_i^{(t-1)}\alpha_i^{(t-1)}$. It then holds that

$$h_{\mathrm{P}}^{(t-1)} + h_{\mathrm{D}}^{(t-1)} \leq \frac{1}{n}\sum_{j=1}^n F_j^{(t-1)}. \tag{A·10}$$

Proof of Lemma 2: It suffices to show that $\forall h \in [n]$,

$$\begin{aligned} &\pi_h[v_{h,0}]^2 - \pi_h[v_{h,0}+\eta v_{h,q}]^2 \\ &\quad + 2\eta\pi[v_{h,0}]v_{h,q} + \eta^2 v_{h,q}^2 \geq 0. \end{aligned} \tag{A·11}$$

where $\pi_h[v_h] := v_h - \max\{0, -\sigma_h v_h\}$ for $v_h \in \mathbb{R}$, because the sum of LHS of (A·11) over $h \in [n]$ is equal to LHS of (A·9).

For $h \in \mathcal{I}_0$ and $h \in \mathcal{I}_{\geq}$ such that $v_{h,0} + \eta v_{h,q} \geq 0$ and $v_{h,0} \geq 0$, LHS of (A·11) can be rearranged as

$$v_{h,0}^2 - (v_{h,0}+\eta v_{h,q})^2 + 2\eta v_{h,0}v_{h,q} + \eta^2 v_{h,q}^2 = 0. \tag{A·12}$$

For $h \in \mathcal{I}_{\geq}$ such that $v_{h,0} + \eta v_{h,q} \leq 0$ and $v_{h,0} \geq 0$, LHS of (A·11) can be rearranged as

$$v_{h,0}^2 - 0 + 2\eta v_{h,0}v_{h,q} + \eta^2 v_{h,q}^2 = (v_{h,0}+\eta v_{h,q})^2 \geq 0. \tag{A·13}$$

For $h \in \mathcal{I}_{\geq}$ such that $v_{h,0} + \eta v_{h,q} \geq 0$ and $v_{h,0} \leq 0$, LHS of (A·11) can be rearranged as

$$0 - (v_{h,0}+\eta v_{h,q})^2 + 0 + \eta^2 v_{h,q}^2 \geq 0. \tag{A·14}$$

since $0 \leq v_{h,0} + \eta v_{h,q} \leq \eta v_{h,q}$.

q.e.d. of Lemma 2

Proof of Lemma 3: Observe that $\forall \boldsymbol{v} := [v_1, \ldots, v_d]^\top \in \mathbb{R}^d$,

$$\begin{aligned} \|\boldsymbol{\pi}[\boldsymbol{v}]\|^2 &= \sum_{h\in\mathcal{I}_0} v_h^2 + \sum_{h\in\mathcal{I}_{\geq}} \pi_h[v_h]^2 \\ &\leq \sum_{h\in\mathcal{I}_0} v_h^2 + \sum_{h\in\mathcal{I}_{\geq}} v_h^2 = \|\boldsymbol{v}\|^2 \end{aligned} \tag{A·15}$$

and

$$\begin{aligned} \|\bar{\boldsymbol{w}}^{(t-1)}\|^2 &= \left\langle \bar{\boldsymbol{w}}^{(t-1)}, \frac{1}{\lambda n}\sum_{j=1}^n \boldsymbol{x}_j \alpha_j^{(t-1)} \right\rangle \\ &= \frac{1}{\lambda n}\sum_{j=1}^n z_j^{(t-1)}\alpha_j^{(t-1)}. \end{aligned} \tag{A·16}$$

Then, these inequalities lead to

$$\begin{aligned} &h_{\mathrm{P}}^{(t-1)} + h_{\mathrm{D}}^{(t-1)} = P(\boldsymbol{\pi}[\boldsymbol{\alpha}^{(t-1)}]) - D(\boldsymbol{\alpha}^{(t-1)}) \\ &= \lambda\|\boldsymbol{\pi}[\bar{\boldsymbol{w}}^{(t-1)}]\|^2 + \frac{1}{n}\sum_{j=1}^n \phi_j(z_j^{(t-1)}) + \phi_j^*(-\alpha_j^{(t-1)}) \\ &\leq \lambda\|\bar{\boldsymbol{w}}^{(t-1)}\|^2 + \frac{1}{n}\sum_{j=1}^n \phi_j(z_j^{(t-1)}) + \phi_j^*(-\alpha_j^{(t-1)}) \\ &= \frac{1}{n}\sum_{j=1}^n F_j^{(t-1)}. \end{aligned} \tag{A·17}$$

q.e.d. of Lemma 3

We are now ready to prove Theorem 1. Let $z_i^{(t-1)} := \left\langle \boldsymbol{w}^{(t-1)}, \boldsymbol{x}_i \right\rangle$, $u_i^{(t-1)} := -\nabla\phi_i(z_i^{(t-1)})$, $q_i^{(t-1)} := u_i^{(t-1)} - \alpha_i^{(t-1)}$, and $\bar{\beta} := \lambda\gamma/(\lambda n\gamma + R^2)$. Then,

$$\begin{aligned} &h_{\mathrm{D}}^{(t-1)} - h_{\mathrm{D}}^{(t)} \geq J_t^1(\eta_t) \geq J_t^1(n\bar{\beta}) \\ &\geq -2\lambda n\bar{\beta}\left\langle \boldsymbol{w}^{(t-1)}, \frac{\boldsymbol{x}_i q_i^{(t-1)}}{\lambda n} \right\rangle - \lambda n^2\bar{\beta}^2 \left\| \frac{\boldsymbol{x}_i q_i^{(t-1)}}{\lambda n} \right\|^2 \\ &\quad + a_{\mathrm{offs}} n^2\bar{\beta}^2 + b_{\mathrm{offs}} n\bar{\beta} \\ &= \bar{\beta}F_i^{(t-1)} + \frac{\gamma\bar{\beta}(q_i^{(t-1)})^2}{2}\left(1 - \frac{(\lambda n\gamma + \|\boldsymbol{x}_i\|^2)\bar{\beta}}{\lambda\gamma}\right) \\ &\geq \bar{\beta}F_i^{(t-1)} + \frac{\gamma\bar{\beta}(q_i^{(t-1)})^2}{2}\left(1 - \frac{(\lambda n\gamma + R^2)\bar{\beta}}{\lambda\gamma}\right) = \bar{\beta}F_i^{(t-1)} \end{aligned} \tag{A·18}$$

where we have used Lemma 2 to get the third inequality. Taking the expectation with respect to the randomness for selection of $i \in [n]$ at tth iteration, we obtain

$$\begin{aligned} &h_{\mathrm{D}}^{(t-1)} - \mathbb{E}\left[h_{\mathrm{D}}^{(t)}\right] \geq \frac{\bar{\beta}}{n}\sum_{i=1}^n F_i^{(t-1)} \\ &\geq \bar{\beta}\cdot\left(h_{\mathrm{P}}^{(t-1)} + h_{\mathrm{D}}^{(t-1)}\right) \geq \bar{\beta}\cdot\max\left\{h_{\mathrm{P}}^{(t-1)}, h_{\mathrm{D}}^{(t-1)}\right\} \end{aligned}$$

$$(\mathrm{A}\cdot 19)$$

where the second inequality follows from Lemma 3. This leads to the bound of the expected primal objective error with respect to randomness at previous iterations:

$$\begin{aligned}\mathbb{E}[h_{\mathrm{P}}^{(t)}] &\leq \bar{\beta}^{-1}\mathbb{E}\left[h_{\mathrm{D}}^{(t)} - h_{\mathrm{D}}^{(t+1)}\right] \leq \bar{\beta}^{-1}\mathbb{E}\left[h_{\mathrm{D}}^{(t)}\right] \\ &\leq \bar{\beta}^{-1}\mathbb{E}\left[h_{\mathrm{D}}^{(t-1)}\right](1-\bar{\beta}) \\ &\leq \bar{\beta}^{-1}h_{\mathrm{D}}^{(0)}\cdot(1-\bar{\beta})^{t} \leq \bar{\beta}^{-1}h_{\mathrm{D}}^{(0)}\exp\left(-\bar{\beta}t\right).\end{aligned} \tag{A·20}$$

Hence, it holds that $\mathbb{E}[h_{\mathrm{P}}^{(t)}] \leq \epsilon_{\mathrm{P}}$ conditioned on $\bar{\beta}^{-1}h_{\mathrm{D}}^{(0)}\exp\left(-\bar{\beta}t\right) \leq \epsilon_{\mathrm{P}}$. This condition can be rearranged as

$$t \geq \frac{1}{\bar{\beta}}\log\left(\frac{h_{\mathrm{D}}^{(0)}}{\epsilon_{\mathrm{P}}\bar{\beta}}\right). \tag{A·21}$$

q.e.d. of Theorem 1

Acknowledgments

This work was supported by the Environment Research and Technology Development Fund (5-2006) of the Environmental Restoration and Conservation Agency of Japan and JSPS KAKENHI Grant Number 22K04372.

References

[1] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[2] M. Bierlaire, Ph.L. Toint, and D. Tuyttens. On iterative algorithms for linear least squares problems with bound constraints. *Linear Algebra and its Applications*, 143:111–143, jan 1991. doi: 10.1016/0024-3795(91)90009-l.

[3] Y. Cai, H. Gu, and T. Kenney. Learning microbial community structures with supervised and unsupervised non-negative matrix factorization. *Microbiome*, 5(1):110, Aug 2017.

[4] Aaron Defazio, Francis Bach, and Simon Lacoste-julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N.d. Lawrence, and K.q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1646–1654. Curran Associates, Inc., 2014.

[5] Chris Ding, Tao Li, Wei Peng, and Haesun Park. Orthogonal nonnegative matrix tri-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD06*. ACM Press, 2006. doi:10.1145/1150402.1150420.

[6] W. Dong, F. Fu, G. Shi, X. Cao, J. Wu, G. Li, and G. Li. Hyperspectral image super-resolution via non-negative structured sparse representation. *IEEE Trans Image Process*, 25(5):2337–52, May 2016.

[7] Kelwin Fernandes and Jaime S. Cardoso. Hypothesis transfer learning based on structural model similarity. *Neural Computing and Applications*, nov 2017. doi:10.1007/s00521-017-3281-4.

[8] Cédric Févotte and Jérôme Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456, sep 2011. doi:10.1162/neco_a_00168.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning – Data Mining, Inference, and Prediction*. Springer, 2nd edition, 2009.

[10] R. He, W. S. Zheng, B. G. Hu, and X. W. Kong. Two-stage nonnegative sparse representation for large-scale face recognition. *IEEE Trans Neural Netw Learn Syst*, 24(1):35–46, Jan 2013.

[11] Simon Henrot, Said Moussaoui, Charles Soussen, and David Brie. Edge-preserving nonnegative hyperspectral image restoration. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, may 2013. doi: 10.1109/icassp.2013.6637926.

[12] Jean-Baptiste Hiriart-Urruty. *Fundamentals of Convex Analysis*. Springer, 2001.

[13] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 427–435, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

[14] Yangfeng Ji, Tong Lin, and Hongbin Zha. Mahalanobis distance based non-negative sparse representation for face recognition. In *2009 International Conference on Machine Learning and Applications*. IEEE, dec 2009. doi:10.1109/icmla.2009.50.

[15] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26: Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 315–323, 2013.

[16] T. Kato, A. Kobayashi, W. Oishi, S. S. Kadoya, S. Okabe, N. Ohta, M. Amarasiri, and D. Sano. Sign-constrained linear regression for prediction of microbe concentration based on water quality datasets. *J Water Health*, 17(3):404–415, Jun 2019.

[17] Dongmin Kim, Suvrit Sra, and Inderjit S. Dhillon. Tackling box-constrained optimization via a new projected quasi-newton approach. *SIAM Journal on Scientific Computing*, 32(6):3548–3563, jan 2010. doi:10.1137/08073812x.

[18] Keigo Kimura, Mineichi Kudo, and Yuzuru Tanaka. A column-wise update algorithm for nonnegative matrix factorization in bregman divergence with an orthogonal constraint. *Machine Learning*, 103(2):285–306, mar 2016. doi:10.1007/s10994-016-5553-0.

[19] Germana Landi and Elena Loli Piccolomini. NPTool: a Matlab software for nonnegative image restoration with Newton projection methods. *Numerical Algorithms*, 62(3):487–504, jun 2012. doi: 10.1007/s11075-012-9602-x.

[20] Charles L. Lawson and Richard J. Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, jan 1995. doi:10.1137/1.9781611971217.

[21] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[22] Yuanqing Lin, D.D. Lee, and L.K. Saul. Nonnegative deconvolution for time of arrival estimation. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2004. doi:10.1109/icassp.2004.1326273.

[23] Jun Ma. Algorithms for non-negatively constrained maximum penalized likelihood reconstruction in tomographic imaging. *Algorithms*, 6(1):136–160, mar 2013. doi: 10.3390/a6010136.

[24] Yurii E. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.

[25] Nicolas L. Roux, Mark Schmidt, and Francis R. Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc., 2012.

[26] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112, jun 2016. doi:10.1007/s10107-016-1030-6.

[27] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *J. Mach. Learn. Res.*, 14(1):567–599, February 2013.

[28] Amnon Shashua and Tamir Hazan. Non-negative tensor factorization

with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning - ICML'05*. ACM Press, 2005. doi:10.1145/1102351.1102451.

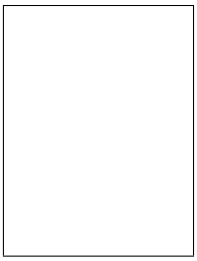
[29] Kenya Tajima, Kohei Tsuchida, Esmeraldo Ronnie R. Zara, Naoya Ohta, and Tsuyoshi Kato. Learning sign-constrained support vector machines. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, January 2021. doi:10.1109/icpr48806.2021.9412786.

[30] J. Wang, F. Tian, H. Yu, C. H. Liu, K. Zhan, and X. Wang. Diverse non-negative matrix factorization for multiview data representation. *IEEE Trans Cybern*, -(-):–, Sep 2017.

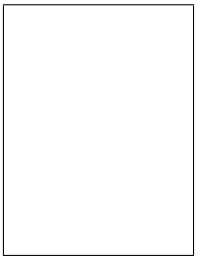
[31] Yanfei Wang and Shiqian Ma. Projected barzilai–borwein method for large-scale nonnegative image restoration. *Inverse Problems in Science and Engineering*, 15(6):559–583, sep 2007. doi: 10.1080/17415970600881897.

[32] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, jan 2014. doi:10.1137/140961791.

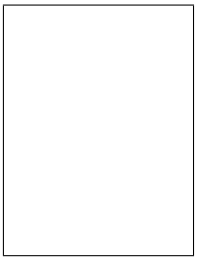
[33] Qiang Zhang, Han Wang, Robert Plemmons, and V. Paul Pauca. Spectral unmixing using nonnegative tensor factorization. In *Proceedings of the 45th annual southeast regional conference on ACM-SE 45*. ACM Press, 2007. doi: 10.1145/1233341.1233449.



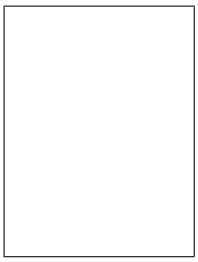
Yuya Takada received his B.E. degree from Gunma University, Japan in 2022. He is now a master's student at the Graduate School of Science & Technology, Gunma University. His research interests include machine learning and data science. He is a member of IPSJ.



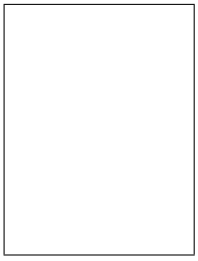
Rikuto Mochida received his B.E. degree from Gunma University, Japan in 2023. He is now a master's student at the Graduate School of Science & Technology, Gunma University. His research interests include machine learning and data science. He is a member of IPSJ.



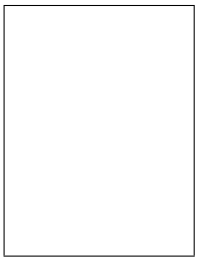
Miya Nakajima received his B.E. degree from Gunma University, Japan in 2022. He is now a master's student at the Graduate School of Science & Technology, Gunma University. His research interests include machine learning and data science. He is a member of IPSJ and JSNDI.



Syun-suke Kadoya received his B.E., and M.E. degrees from Hokkaido University, Sapporo, Japan in 2016 and 2018, respectively. He received his PhD degree from Tohoku University, Sendai, Japan in 2021. He is now a post-doc at the University of Tokyo. He is currently interested in time series analysis, infectious disease modeling, virus population genetics and bioinformatics. He is a member of JSCE and JSWE.



Daisuke Sano received his B.E., M.E., and PhD degrees from Tohoku University, Sendai, Japan, in 1998, 2000, and 2003, respectively. He took a Post-Doctoral Fellowship at Tohoku University (2003-2007) and University of Barcelona, Spain (2007-2009), and then, he got a tenure position (Associate Professor) at Hokkaido University, Sapporo, Japan, since 2009 and was managing a team of water and public health study. He moved back to Tohoku University in 2017 and is responsible for the research and supervision of international and domestic projects in the field of water and public health. He is currently interested in the statistical modeling of enteric pathogens disinfection/removal, antibiotic resistance and water, and the impact of chemical contaminants on ecosystems. He is a member of JSCE, JSWE, IWA, ASM and ACE.



Tsuyoshi Kato received his B.E., M.E., and PhD degrees from Tohoku University, Sendai, Japan, in 1998, 2000, and 2003, respectively. From 2003 to 2005, he was a postdoctoral fellow at the National Institute of Advanced Industrial Science Technology (AIST) in Tokyo. From 2005 to 2008, he was an assistant professor at the University of Tokyo. He is now a full professor at the Faculty of Informatics, Gunma University. His current scientific interests include pattern recognition, non-destructive testing, water engineering and bioinformatics. He is a member of JSNDI, JSWE, and IPSJ.