PAPER

Frank-Wolfe Algorithm for Learning SVM-Type Multi-Category Classifiers

Kenya TAJIMA[†], Yoshihiro HIROHASHI, Esmeraldo ZARA[†], *Nonmembers*, and Tsuyoshi KATO^{†a)}, *Member*

SUMMARY The multi-category support vector machine (MC-SVM) is one of the most popular machine learning algorithms. There are numerous MC-SVM variants, although different optimization algorithms were developed for diverse learning machines. In this study, we developed a new optimization algorithm that can be applied to several MC-SVM variants. The algorithm is based on the Frank-Wolfe framework that requires two subproblems, direction-finding and line search, in each iteration. The contribution of this study is the discovery that both subproblems have a closed form solution if the Frank-Wolfe framework is applied to the dual problem. Additionally, the closed form solutions on both the direction-finding and line search exist even for the Moreau envelopes of the loss functions. We used several large datasets to demonstrate that the proposed optimization algorithm rapidly converges and thereby improves the pattern recognition performance.

key words: SVM, convex optimization, Frank-Wolfe, dual problem, multi-category classification

1. Introduction

Multi-category classification is a task that assigns an input object to one of the pre-defined categories. Numerous supervised learning problems are reduced to the multi-category classification, although in the field of pattern recognition, the focus of various research and theoretical analyses has been a simpler task, the binary classification, yielding the most successful machine learning algorithm - the support vector machine (SVM). In the 1990's, the so-called one-versus-rest approach was employed to apply SVM to multi-category classification tasks. In the one-versus-rest approach, the learning task is divided into several independent optimization problems, and SVM is applied to each of those problems. A drawback of the one-versus-rest approach is the inability for learning correlation among the categories. Crammer and Singer [3] proposed an alternative method, which formulates the learning problem with a single optimization problem. This method is called the multi-category SVM (MC-SVM). Since the emergence of Crammer and Singer's MC-SVM, several variants, such as the structured SVM [18], SVM_{multi} [9], and top-k SVM [12] have been developed. Structured SVM expanded the applicability of machine learning to a wide range, including natural language parsing [6], the deformable part model for image analysis [4], and the biological sequence alignment [18].

The SVM_{multi} provides a framework that directly learns the performance measures such as F1-score and precision/recall breakeven point, precision at k, and ROC score [9]. Top-k SVM is trained by minimizing the empirical risk based on top-k error [12].

The learning machine cannot be practical without an efficient and stable optimization algorithm. The above-mentioned MC-SVM variants are learned with different optimization algorithms, each of which is specialized to the corresponding learning machine. For example, cutting plane methods [10] were developed for learning the structured SVM and SVM_{multi}. Optimization algorithms for learning top-k SVM were proposed by two research groups [2], [12], and both algorithms were based on the stochastic dual coordinate ascent (SDCA) method [15]. However, the algorithms were derived from an incorrect theory, making both algorithms fail to attain an optimum [11]. Kato and Hirohashi [11] considered applying the Frank-Wolfe method [5] to the dual problem of top-k SVM. The Frank-Wolfe method is an iterative framework for convex optimization over a polyhedron and each iteration consists of a direction-finding step and line search step. Sub-linear convergence to the optimum is guaranteed if both steps are performed exactly [8]. Kato and Hirohashi [11] found that both the direction-finding and line search steps can be provided in a closed form, and the computational time is within $O(mn^2)$.

A primary contribution of this study is the finding that both the direction-finding and line search steps of the Frank-Wolfe method are expressed in a closed form not only for top-k SVM but also a wide range of the MC-SVM variants. In this study, a condition for expressing the two steps in a closed form is clarified. Compared to gradient methods that are often employed for machine learning, the proposed Frank-Wolfe algorithm has no hyper-parameter such as a step size, often requiring a manual tuning for optimization and guarantees the resulting solution's accuracy. Owing to this study's findings, an optimization algorithm that does not require a step size and can be terminated with a pre-defined accuracy becomes available for learning a variety of MC-SVM variants.

In addition, we extended our analysis to the Moreau envelope [1] of the loss function. The Moreau envelope is a trick that is widely used in the machine learning field. Application of the Moreau envelope makes the loss functions smooth and thereby generally accelerates optimization [13], [20]. Kato and Hirohashi [11] found that each step of the Frank-Wolfe method can be expressed in a closed

Manuscript received January 30, 2021.

Manuscript revised July 8, 2021.

Manuscript publicized August 11, 2021.

[†]The authors are with Graduate School of Science and Technology, Gunma University, Kiryu-shi, 376-8515 Japan.

a) E-mail: katotsu@cs.gunma-u.ac.jp

DOI: 10.1587/transinf.2021EDP7025

form even when the Moreau envelope of the unweighted top- k loss function is employed. In this study, we confirmed that the same technique can be applied to a larger class of loss functions.

This is the journal version of our conference paper [17]. Proofs and additional experimental results of this paper are provided in [16].

Notation: We shall use the notation $\pi(j; s) \in [m]$ which is the index of the j -th largest component in a vector $s \in \mathbb{R}^m$. When using this notation, the vector s is omitted if there is no danger of confusion. Namely, for a vector $s \in \mathbb{R}^m$, we can write $s_{\pi(1)} \geq s_{\pi(2)} \geq \dots \geq s_{\pi(m)}$. Let us define $\pi(s) := [\pi(1; s), \dots, \pi(m; s)]^\top$ and introduce a notation for a vector with permuted components as $s_{\pi(s)} := [s_{\pi(1)}, \dots, s_{\pi(m)}]^\top$.

We use e_i to denote a unit vector where i -th entry is one. The n -dimensional vector, all of whose entries are one is denoted by $\mathbf{1}_n$. We use an operator $\|\cdot\|_F$ to denote the Frobenius norm.

2. MC-SVM and its Variants

In this section, we review MC-SVM and its variants. Let us denote the discrete output space by $\mathcal{Y} := \{1, \dots, m\}$ where m is the number of categories. In the multi-category classification scenario, prediction is the assignment of an input $x \in \mathcal{X}$ to one of the elements in $\mathcal{Y}$, where $\mathcal{X}$ is the input space. Feature vectors are extracted not only from an input but also from a candidate of categories. Let $\psi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^d$ be the feature extractor. A typical feature extractor implementation is, $\psi(x, j) := e_j \otimes x$, where $\otimes$ is the operator for the Kronecker product and e_j is an m -dimensional unit vector. Using the model parameter $w \in \mathbb{R}^d$, the prediction score for the category j is given by the inner product between the feature and parameter vectors, i.e., $\langle \psi(x, j), w \rangle$. Prediction of an input $x \in \mathcal{X}$ is accomplished by computing the prediction score for each category j , for instance $\langle \psi(x, j), w \rangle$, and finding the maximal score among m prediction scores. The corresponding category is the prediction result.

We used n training examples, $(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$, to determine the value of the model parameter $w \in \mathbb{R}^d$. The MC-SVM attempts to find the minimizer of the regularized empirical risk defined as

$$P(w) := \frac{\lambda}{2} \|w\|^2 + \frac{1}{n} \sum_{i=1}^n \Phi(\Psi(x_i)^\top w; y_i), \quad (1)$$

where $\Psi(x_i)$ is the horizontal concatenation of m feature vectors (i.e., $\Psi(x) := [\psi(x, 1), \dots, \psi(x, m)] \in \mathbb{R}^{d \times m}$); λ is a positive constant called the *regularized parameter*; $\Phi(\cdot; y) : \mathbb{R}^m \rightarrow \mathbb{R}$ is a loss function. For MC-SVM, the *max hinge loss* $\Phi_{mh}(\cdot; y)$ is adopted for $\Phi(\cdot; y)$. Using the Kronecker delta $\delta_{\cdot, \cdot}$, the max hinge loss is defined as

$$\Phi_{mh}(s; y) := \max_{j \in [m]} (s_j - s_y + 1 - \delta_{j,y}). \quad (2)$$

Fenchel dual: Function $D : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ is defined as

$$D(A) := -\frac{\lambda}{2} \|w(A)\|^2 - \frac{1}{n} \sum_{i=1}^n \Phi^*(-\alpha_i; y_i) \quad (3)$$

which is a Fenchel dual to the regularized empirical risk $P : \mathbb{R}^d \rightarrow \mathbb{R}$, where $A := [\alpha_1, \dots, \alpha_n] \in \mathbb{R}^{m \times n}$; $w(A) := \frac{1}{\lambda n} \sum_{i=1}^n \Psi(x_i) \alpha_i$; $\Phi^*(\cdot; y)$ is the *convex conjugate* of $\Phi(\cdot; y)$. The optimal solution of the primal variable $w_\star$ is obtained by $w_\star := w(A_\star)$, where $A_\star$ is the maximizer of the dual objective $D(A)$. The gap $P(w(A)) - D(A)$ is non-negative for any $A \in \mathbb{R}^{m \times n}$ and vanishes at the optimum $A = A_\star$. From this fact, we can terminate the iterations for optimization when $P(w(A)) - D(A) \leq \epsilon$ with the pre-defined small positive constant ϵ . Then, the primal error $P(w(A)) - P(w_\star)$ is guaranteed to not be over ϵ .

Structured SVM: In the structured SVM [18], a non-negative loss $\Delta_y^{(i)}$ for the i -th training example is arbitrarily designed for the case that the i -th training example is predicted as $\hat{y} \in \mathcal{Y}$ (i.e. $\arg\max_{j \in [m]} \langle w_j, \psi(x_i, j) \rangle = \hat{y}$). This is in contrast to Crammer and Singer's MC-SVM that adopts the *convex surrogate* of the 0/1 loss that always suffers a unit loss for a mistake. Structured SVM employs the convex surrogate of $\Delta_y^{(i)}$, defined as

$$\Phi_{i,sh}(s; y) := \max_{j \in [m]} (s_j - s_y + \Delta_y^{(i)}). \quad (4)$$

Unweighted Top- k SVM: The unweighted top- k SVM is a variant of MC-SVM. While MC-SVM assumes that a single category is assigned to an input, the unweighted top- k SVM assigns k categories to an input. Prediction results are interpreted so that one of the predicted k categories will be the input category. Given an input $x \in \mathcal{X}$, the set of k categories are chosen as $\{\pi(1; s), \dots, \pi(k; s)\}$, where s is the prediction score vector (i.e. $s = [s_1, \dots, s_m]^\top := \Psi(x)^\top w$). For training such a classifier, the loss function is designed as

$$\Phi_{utk}(s; y) := \max \left\{ 0, \frac{1}{k} \sum_{j=1}^k (s - s_y \mathbf{1} + \mathbf{1} - e_y)_{\pi(j)} \right\}. \quad (5)$$

This is called the *unweighted top- k hinge loss*.

Unweighted Usunier SVM: Similar to the unweighted top- k SVM, the unweighted Usunier SVM trains the classifier that performs top- k prediction, but the loss function is slightly different. The empirical risk for learning the unweighted Usunier SVM consists of the following loss function:

$$\Phi_{uu}(s; y) := \frac{1}{k} \sum_{j=1}^k \max \left\{ 0, (s - s_y \mathbf{1} + \mathbf{1} - e_y)_{\pi(j; s - e_y)} \right\}. \quad (6)$$

This loss function is called the Usunier loss. The original loss function developed by Usunier et al. [19] is devised for ranking prediction. Lapin et al. [12] redesigned their loss

function for top- k prediction.

Weighted Top- k SVM: Using a constant weight vector $\rho = [\rho_1, \dots, \rho_m]^\top \in \mathbb{R}^m$ such as $\rho_1 \geq \dots \geq \rho_{m-1} \geq \rho_m = 0$, Kato and Hirohashi [11] extended the unweighted top- k hinge loss to the weighted version:

$$\Phi_{\text{wtk}}(s; y) := \max \left\{ 0, \sum_{j=1}^m \left(\mathbf{1}_m - \mathbf{e}_y + s - s_y \mathbf{1}_m \right)_{\pi(j)} \rho_j \right\}. \quad (7)$$

They called this function (7) the *weighted top- k hinge loss*.

Weighted Usunier SVM: The weighted version of the Usunier loss can be considered. The *weighted Usunier loss* function is defined as

$$\Phi_{\text{wu}}(s; y) := \sum_{j=1}^m \max \left\{ 0, \left(\mathbf{1}_m - \mathbf{e}_y + s - s_y \mathbf{1}_m \right)_{\pi(j)} \right\} \rho_j, \quad (8)$$

where $\rho = [\rho_1, \dots, \rho_m]^\top \in \mathbb{R}^m$ is a constant vector such as $\rho_1 \geq \dots \geq \rho_{m-1} \geq \rho_m = 0$.

3. Max Dot Over Simplex-Type Loss Functions

In this section, learning machines targeted by our learning algorithm are formulated. The learning machine trains a classifier by minimizing the regularized empirical risk given in (1). In the learning algorithm presented in the next section, the loss function appearing in the expression of the regularized empirical risk is assumed to be the *max dot over simplex-type (mdos-type)* defined below,

Definition 3.1: Function $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}$ is said to be the mdos-type if there exists a simplex $\mathcal{B}$ such that $\forall y \in [m]$, $\forall s \in \mathbb{R}^m$

$$\Phi(s; y) = \max_{\beta \in \mathcal{B}} \langle \beta, \mathbf{1} - \mathbf{e}_y + s - s_y \mathbf{1} \rangle. \quad (9)$$

In the previous section, six loss functions: the max hinge loss, structured hinge loss, unweighted top- k hinge loss, unweighted Usunier loss, weighted top- k hinge loss, and weighted Usunier loss, were described. It can be shown that all six loss functions are the mdos-type. In the following section, the corresponding simplexes are presented.

- The simplex for the max hinge loss is $\mathcal{B}_{\text{sh}} := \Delta(1)$, where

$$\Delta(r) := \left\{ \beta \in \mathbb{R}_+^m \mid \|\beta\|_1 \leq r \right\}. \quad (10)$$

- The simplex for the unweighted top- k loss is $\mathcal{B}_{\text{tk}} := \Delta_{\text{tk}}(k, 1)$, where

$$\Delta_{\text{tk}}(k, r) := \left\{ \beta \in \Delta(r) \mid \beta \leq \frac{\|\beta\|_1}{k} \mathbf{1} \right\}. \quad (11)$$

- The simplex for the unweighted Usunier loss (6) is $\mathcal{B}_{\text{uu}} := \Delta_{\text{u}}(k, 1)$, where

$$\Delta_{\text{u}}(k, r) := \left\{ \beta \in \Delta(r) \mid \beta \leq \frac{1}{k} \mathbf{1} \right\}. \quad (12)$$

- The simplex for the weighted top- k hinge loss (7) is

$$\begin{aligned} \mathcal{B}_{\text{wtk}} &:= \left\{ \beta \in \mathbb{R}^m \mid \exists \zeta \in \mathbb{R}, \forall \ell \in [L], \right. \\ &\quad \left. \exists \lambda_\ell \in \Delta_{\text{tk}}(k_\ell, \rho'_\ell k_\ell), \zeta = \frac{\langle \mathbf{1}, \lambda_\ell \rangle}{k_\ell \rho'_\ell}, \beta = \sum_{\ell=1}^L \lambda_\ell \right\}. \end{aligned} \quad (13)$$

- The simplex of the weighted Usunier loss (8) is

$$\begin{aligned} \mathcal{B}_{\text{wu}} &:= \left\{ \beta \in \mathbb{R}_+^m \mid \forall \ell \in [L], \right. \\ &\quad \left. \exists \lambda_\ell \in \Delta_{\text{u}}(1/\rho'_\ell, k_\ell \rho'_\ell), \beta = \sum_{\ell=1}^L \lambda_\ell \right\}. \end{aligned} \quad (14)$$

Therein, the variables L and $\rho'_1, \dots, \rho'_L$ used in (13) and (14) are defined as follows. The variable L takes a natural number representing the cardinality of the set

$$\mathcal{K} := \{k \in [m] \mid \rho_k > \rho_{k+1}\}. \quad (15)$$

Denote by $k_1, \dots, k_L$ the entries in $\mathcal{K}$ sorted as $1 \leq k_1 < \dots < k_L < m$. The remaining variables $\rho'_1, \dots, \rho'_L$ are defined as $\rho'_\ell := \rho_{k_\ell}$ for $\ell = 1, \dots, L$. The above results are summarized in the following theorem.

Theorem 3.1: The five loss functions, $\Phi_{\text{mh}}(\cdot; y)$, $\Phi_{\text{utk}}(\cdot; y)$, $\Phi_{\text{uu}}(\cdot; y)$, $\Phi_{\text{wtk}}(\cdot; y)$, and $\Phi_{\text{wu}}(\cdot; y)$, are the mdos-type. Loss function $\Phi_{i,\text{sh}}(\cdot; y)$ is the mdos-type when $\Delta_j^{(i)} = 1 - \delta_{y,j}$ where $\delta_{\cdot,\cdot}$ is the Kronecker delta.

The detail of the proof for Theorem 3.1 is given in Sect. A of [16].

4. Frank-Wolfe Algorithm

In this section, an optimization algorithm for learning MC-SVM is presented. Here, the loss function Φ appearing in the regularized empirical loss is supposed to be the mdos-type. The optimization algorithm presented here is the Frank-Wolfe method maximizing the dual objective $D(\mathbf{A})$. Each iteration of the Frank-Wolfe method consists of the direction-finding and line search steps. The dual variable at the t -th iteration is denoted by $\mathbf{A}^{(t)} = [\alpha_1^{(t)}, \dots, \alpha_n^{(t)}]$. The direction-finding step solves the following linear program:

$$\mathbf{U}^{(t-1)} \in \underset{\mathbf{U} \in \text{dom}(-D)}{\text{argmax}} \left\langle \nabla D(\mathbf{A}^{(t-1)}), \mathbf{U} \right\rangle. \quad (16)$$

At the line search step, a solution maximizing $D(\mathbf{A})$ over the line segment between two points, $\mathbf{A}^{(t-1)}$ and $\mathbf{U}^{(t-1)}$, is found

$$\gamma^{(t-1)} := \underset{\gamma \in [0,1]}{\text{argmax}} D\left((1-\gamma)\mathbf{A}^{(t-1)} + \gamma\mathbf{U}^{(t-1)}\right). \quad (17)$$

Using the solutions to the two subproblems, such as $\mathbf{U}^{(t-1)}$ and $\gamma^{(t-1)}$, the dual variable is updated as

$$\mathbf{A}^{(t)} := (1 - \gamma^{(t-1)})\mathbf{A}^{(t-1)} + \gamma^{(t-1)}\mathbf{U}^{(t-1)}. \quad (18)$$

As long as the two subproblems are solved exactly at each iteration, the sublinear convergence is guaranteed. However, the algorithm would be impractical if each step could not be efficiently solved.

We first discuss how we can perform the direction-finding step. Under the assumption that the loss function is the mdsos-type, the effective domain $\text{dom}(-D)$ is a polyhedron. Hence, a general-purpose solver for linear programs can be used for the direction-finding step, it takes a prohibitive computational cost if resorting to a general-purpose solver at every iteration. In this study, we have the following theorem.

Theorem 4.1: Consider applying the Frank-Wolfe algorithm to the problem for maximizing $D(\mathbf{A})$ with respect to $\mathbf{A}$. Assume the loss function $\Phi(\cdot; y)$ to be the mdsos-type. Then, both the direction-finding and line search steps are expressed in a closed form.

See Sect. B in [16] for the proof of Theorem 4.1. The concrete solution to the subproblem (16) for the direction-finding step is given as follows. The i -th column in the matrix $\mathbf{U}^{(t-1)}$, say $\mathbf{u}_i^{(t-1)}$, is set to

$$\mathbf{u}_i^{(t-1)} \in -\partial\Phi(\Psi(\mathbf{x}_i)^\top \mathbf{w}(\mathbf{A}^{(t-1)}); y_i), \quad (19)$$

where $\partial\Phi(\cdot; y)$ is the subdifferential of $\Phi(\cdot; y)$. An arbitrary subgradient can be taken if the set $\partial\Phi(\cdot; y)$ has multiple elements.

The subproblem for the line search step (17) is also solved in a closed-form solution as $\gamma^{(t-1)} = \max(0, \min(1, \hat{\gamma}^{(t-1)}))$ where

$$\hat{\gamma}^{(t-1)} := \frac{\sum_{i=1}^n \langle \Delta\alpha_i^{(t-1)}, \mathbf{e}_{y_i} - \Psi(\mathbf{x}_i)^\top \mathbf{w}(\mathbf{A}^{(t-1)}) \rangle}{\lambda n \|\mathbf{w}(\Delta\mathbf{A}^{(t-1)})\|^2} \quad (20)$$

where $\Delta\alpha_i^{(t-1)}$ is the i -th column of the $m \times n$ matrix $\Delta\mathbf{A}^{(t-1)} := \mathbf{U}^{(t-1)} - \mathbf{A}^{(t-1)}$.

5. Extension to Moreau Envelope

The Moreau envelope [1] is a trick often used for transforming a non-differentiable convex function into a smoothed function. For example, the Huber loss [7] and the smoothed hinge loss [15], widely used in machine learning, are, respectively, the Moreau envelopes of the absolute error and the hinge loss. Since it tends to take less time to minimize a smooth objective function, the Moreau envelope is a useful technique to make machine learning efficient. The Moreau envelope of a convex loss function $\Phi(\cdot; y) : \mathbb{R}^m \rightarrow \mathbb{R}$ is defined as

$$\Phi_m(s; y) := \left(\Phi^*(\cdot; y) + \frac{\gamma_{sm}}{2} \|\cdot\|^2 \right)^* (s; y) \quad (21)$$

where γ_{sm} is a non-negative constant called the *smoothness parameter*. As long as $\Phi(\cdot; y)$ is a convex function,

Algorithm 1: Frank-Wolfe algorithm for minimizing a risk based on the Moreau envelope.

```

1 begin
2    $\mathbf{A}^{(0)} \in \text{dom}(-D)$ ;
3   for  $t := 1$  to  $T$  do
4      $\mathbf{w}^{(t-1)} := \frac{1}{\lambda n} \sum_{i=1}^n \Psi(\mathbf{x}_i) \alpha_i^{(t-1)}$ ;
5     for  $i \in [n]$  do
6        $\tilde{s}_i^{(t-1)} := \Psi(\mathbf{x}_i)^\top \mathbf{w}^{(t-1)} - \gamma_{sm} \alpha_i^{(t-1)}$ ;
7        $\mathbf{u}_i^{(t-1)} \in -\partial\Phi(\tilde{s}_i^{(t-1)}; y_i)$ ;
8        $\Delta\alpha_i^{(t-1)} := \mathbf{u}_i^{(t-1)} - \alpha_i^{(t-1)}$ ;
9     end
10     $\Delta\mathbf{w}^{(t-1)} := \frac{1}{\lambda n} \sum_{i=1}^n \Psi(\mathbf{x}_i) \Delta\alpha_i^{(t-1)}$ ;
11     $\hat{\gamma}^{(t-1)} := \frac{\sum_{i=1}^n \langle \Delta\alpha_i^{(t-1)}, \mathbf{e}_{y_i} - \tilde{s}_i^{(t-1)} \rangle}{\lambda n \|\Delta\mathbf{w}^{(t-1)}\|^2 + \frac{\gamma_{sm}}{\lambda n} \|\Delta\mathbf{A}^{(t-1)}\|^2}$ ;
12     $\gamma^{(t-1)} := \max(0, \min(1, \hat{\gamma}^{(t-1)}))$ ;
13     $\mathbf{A}^{(t)} := \mathbf{A}^{(t-1)} + \gamma^{(t-1)} \Delta\mathbf{A}^{(t-1)}$ ;
14  end
15 end
```

Table 1 Computational times for one iteration.

	PG	StdFW	LSFW
Caltech101	0.453 sec	0.589 sec	0.814 sec
CUB200	2.570 sec	2.458 sec	3.324 sec
Flower102	0.661 sec	0.649 sec	0.903 sec
Indoor67	3.040 sec	2.855 sec	3.919 sec
News20	0.548 sec	0.436 sec	0.531 sec

its Moreau envelope $\Phi_m(\cdot; y)$ is ensured to be $(1/\gamma_{sm})$ -smooth if $\gamma_{sm} > 0$. The definition immediately implies that $\Phi_m(\cdot; y) = \Phi^*(\cdot; y)$ if $\gamma_{sm} = 0$. We now discuss only the case of $\gamma_{sm} > 0$. To minimize the regularized empirical risk

$$P_m(\mathbf{w}) := \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{n} \sum_{i=1}^n \Phi_m(\Psi(\mathbf{x}_i)^\top \mathbf{w}; y_i) \quad (22)$$

we now consider again applying the Frank-Wolfe method for maximization of the Fenchel dual

$$D_m(\mathbf{A}) := -\frac{\lambda}{2} \|\mathbf{w}(\mathbf{A})\|^2 - \frac{1}{n} \sum_{i=1}^n \Phi_m^*(-\alpha_i; y_i). \quad (23)$$

Notice that the Moreau envelope of the loss function $\Phi(\cdot; y)$ is no longer the mdsos-type even if $\Phi(\cdot; y)$ is the mdsos-type, implying that the optimization algorithm presented in the previous section cannot be directly applied to the Moreau envelope. We obtained the following result:

Theorem 5.1: Consider applying the Frank-Wolfe algorithm to the problem for maximizing $D_m(\mathbf{A})$ with respect to $\mathbf{A}$. Assume that the loss function $\Phi(\cdot; y)$ to be the mdsos-type. Then, both the direction-finding and line search steps are expressed in a closed form even if $\gamma_{sm} > 0$.

See Sect. C in [16] for the proof of Theorem 5.1. The update rules of the direction-finding and line search steps are described below. Let

$$\tilde{s}_i^{(t-1)} := \Psi(\mathbf{x}_i)^\top \mathbf{w}(\mathbf{A}^{(t-1)}) - \gamma_{sm} \alpha_i^{(t-1)}. \quad (24)$$

Table 2 Time complexity of each line in Algorithm 1.

Line number	General case	A special case
Line 4	$O(mnd)$	$O(nd)$
Line 6	$O(md) \times n$	$O(d) \times n$
Line 7	Depends on Φ	Depends on Φ
Line 8	$O(m) \times n$	$O(m) \times n$
Line 10	$O(mnd)$	$O(nd)$
Line 11	$O(mn + d)$	$O(mn + d)$
Line 13	$O(mn)$	$O(mn)$

The update rule of the direction-finding step (19) is replaced with:

$$\mathbf{u}_i^{(t-1)} \in -\partial\Phi(\tilde{\mathbf{s}}_i^{(t-1)}; y_i) \quad (25)$$

Note that $\Phi(\cdot; y)$ in (25) is the mdos-type loss function, not its Moreau envelope. The expression of $\hat{\gamma}^{(t-1)}$ for the line search step $\gamma^{(t-1)} = \max(0, \min(1, \hat{\gamma}^{(t-1)}))$ is replaced with:

$$\hat{\gamma}^{(t-1)} := \frac{\sum_{i=1}^n \langle \Delta \alpha_i^{(t-1)}, \mathbf{e}_{y_i} - \tilde{\mathbf{s}}_i^{(t-1)} \rangle}{\lambda n \|\mathbf{w}(\Delta \mathbf{A}^{(t-1)})\|^2 + \frac{\gamma_{\text{sm}}}{\lambda n} \|\Delta \mathbf{A}^{(t-1)}\|^2}. \quad (26)$$

The detailed derivations are described in the proof of Theorem 5.1. The procedure is summarized in Algorithm 1.

In summary, the results show that the two steps in each iteration of the Frank-Wolfe method are expressed in a closed form not only for mdos-type loss functions but also for their Moreau envelopes.

5.1 Time Complexity

We analyze the time complexity of the Frank-Wolfe algorithm presented in Algorithm 1. The middle column in Table 2 shows the time complexity consumed in each line. Line 6 and Line 8 require $O(md)$ and $O(m)$, respectively, for each $i \in [n]$, which take $O(md) \times n$ and $O(m) \times n$ to compute n vectors of $\tilde{\mathbf{s}}_i^{(t-1)}$ and $\Delta \alpha_i^{(t-1)}$, respectively. Line 7 contains the computation of a subgradient of the loss function. The time complexity for this line depends on the loss function definition. The max hinge loss requires $O(m)$ computation for Line 7. The top- k hinge loss, the Usunier loss, and their weighted generalizations consume $O(m \log m)$ computation for the line.

In the case where the feature extractor is $\psi(\mathbf{x}, j) = \mathbf{e}_j \otimes \mathbf{x}$ with $\mathcal{X} := \mathbb{R}^{d_0}$, the time complexity is improved compared to the general case. In this case, $d = md_0$. The update rule of the primal variable in Line 4 in Algorithm 1 is rewritten as

$$\mathbf{w}^{(t-1)} = \frac{1}{\lambda n} \sum_{i=1}^n \alpha_i^{(t-1)} \otimes \mathbf{x}_i. \quad (27)$$

Now, we discuss how to update $\tilde{\mathbf{s}}_i^{(t-1)}$ in Line 6. To compute the j th entry in the m -dimensional vector $\Psi(\mathbf{x}_i)^\top \mathbf{w}^{(t-1)}$, we extract a sub-vector $(\mathbf{e}_j^\top \otimes \mathbf{I}_{d_0})\mathbf{w}$ and take the inner-product between it and the input vector $\mathbf{x}_i$. Both extraction of m sub-vectors and computation of m inner-products takes $O(m)$ computational cost. These discussions are summarized in

the third column of Table 2.

6. Experiments

In this section, we demonstrate the performance of the proposed Frank-Wolfe algorithm in terms of the convergence speed and the pattern recognition performance.

6.1 Convergence Speed

To illustrate how rapidly the proposed optimization algorithm for empirical risk minimization is converged, we used four image datasets: Caltech101 Silhouettes, CUB200, Flower102, and Indoor67 containing $n = 6,339, 6,033, 2,040$, and $15,607$ images, respectively, and a text dataset: News20 containing $15,935$ texts. The images or texts in each dataset are classified into $m = 102, 200, 102, 67$, and 20 categories, respectively.

The deep neural structure named VGG16 was used to extract an input vector $\mathbf{x} \in \mathbb{R}^d$ from each image for CUB200, Flower102, and Indoor67. We extract $d = 4,096$ features from the deep networks for the three image datasets, respectively. Features for Caltech101 were the vectorization of the pixel intensities of 16×16 gray scaled images. Singular value decomposition was performed to reduce $15,935$ word counts in News20 to $d = 1,024$ features. For the loss function, (8) was chosen with $\rho_j := \max(0, 6 - j)/15$. The regularization parameter is set to $\lambda = 1/n$. The proposed optimization algorithm was compared with two methods: PG and StdFW. The method PG is the projected gradient algorithm [14]. In each iteration, PG updates the primal variable to the descent direction and projects it onto a ball to suppress the norm of the gradient vector. The sublinear convergence is ensured for the Lipschitz continuous loss functions. The method StdFW is an alternative to the proposed Frank-Wolfe algorithm. In every iteration of the proposed algorithm, the direction-finding step is followed by the line search step that finds the optimal ratio, such as γ_t , for mixing the previous point with the new point computed at the direction-finding step. Theoretically, the sublinear convergence is guaranteed even if the ratio is pre-scheduled with $\gamma_t = 2/(t+1)$. The method StdFW denotes the Frank-Wolfe using the pre-scheduled γ_t , while the proposed Frank-Wolfe is referred to as the line search Frank-Wolfe abbreviated with LSFw.

Figure 1 (a) has two sub-panels that show the convergence behaviors for fitting to Caltech101 dataset. For the other four datasets, see Fig. 1 in [16]. The upper and lower sub-panels show the objective errors and the duality gaps against the number of iterations, respectively, where the objective error and the duality gap at the t -th iteration are defined as $P(\mathbf{w}^{(t)}) - P(\mathbf{w}_\star)$ and $P(\mathbf{w}^{(t)}) - D(\mathbf{A}^{(t)})$ where $\mathbf{w}^{(t)}$ and $\mathbf{A}^{(t)}$ are the values of the primal and dual variables at t -th iteration, respectively. For StdFW and LSFw, the value of the primal variable is recovered by $\mathbf{w}^{(t)} := \mathbf{w}(\mathbf{A}^{(t)})$ for $t \in \mathbb{N}$. Because it is impossible to know the exact value of the optimal solution $\mathbf{w}_\star$, the value of $\mathbf{w}_\star$ is approximated by $\mathbf{w}_\star \approx \mathbf{w}(\mathbf{A}^{(t)})$ in these experiments, where $\mathbf{A}^{(t)}$ is obtained by it-

Fig. 1 Convergence behaviors for minimizing the empirical risk without the Moreau envelope (a) and with the Moreau envelope (b). Black line: LSFW; dashed red line: StdFW; dashed black line: Pegasos.

erating the Frank-Wolfe until reaching $P(w^{(t)}) - D(A^{(t)}) < 10^{-5}$. For CUB200, Flower102, and Indoor67, the convergence of LSFW was much faster than those of the other two methods, whereas the convergence speed among three methods were similar for Caltech101 and News20. A property of the three datasets, CUB200, Flower102, and Indoor67, differs from that of the two datasets, Caltech101 and News20. The property is the number of feature vector dimensions. Feature vectors in CUB200, Flower102, and Indoor67 have a higher dimension than those in Caltech101 and News20. High-dimensional features tend to make the dual objective function more strongly concave. The authors conjecture that the difference in the number of dimensions yields the convergence behavior differences.

We next applied the Moreau envelope to the loss function with $\gamma_{sm} = 0.01$. The convergence behaviors were changed as shown in Fig. 1 (b) of this paper and in Fig. 2 of [16], where the negative dual function is strongly convex with the coefficient γ_{sm}/n whatever training data are given. Indeed, the Moreau envelope makes the convergence of LSFW for Caltech101 and News20 faster, although the convergence for the other three datasets were not accelerated. An explanation of this phenomenon may be that the negative dual objectives for CUB200, Flower102, and Indoor67 are already strongly convex even without the Moreau envelope.

Table 1 displays the computational times for each iteration of the three optimization algorithms. The running times for PG and StdFW are similar. Compared to StdFW, LSFW requires more computation to perform the line search. The computational times of LSFW do not exceed 1.5 times that of the StdFW for one iteration. By combining the experimental results for the running time of one iteration with the objective errors against the number of iterations, it can be concluded that LSFW can achieve accurate solutions with much lower computational times.

6.2 Pattern Recognition Performance

We examined the pattern recognition performance on the five datasets used for the convergence experiments. For each dataset, 50% of data were randomly picked. Each of the three optimization algorithms was applied to the picked data

Table 3 Pattern recognition performance.

	Top-1	Top-3	Top-5
PG	0.469(0.008)	0.307(0.005)	0.257(0.005)
StdFW	0.481(0.022)	0.317(0.032)	0.266(0.032)
LSFW	0.467 (0.048)	0.300 (0.010)	0.244 (0.005)

to train a multi-category classifier. The remaining data were used for testing the generalization performance of pattern recognition. Each optimization algorithm was implemented for 1,000 iterations. Cross-validation was performed to determine the value of the regularization constant λ . Top-1, top-3, and top-5 error ratios were used for assessing the generalization performance. A lower value indicates better performance. The above procedures were performed 20 times, and the averages and the standard deviations of the performance measures across the 20 trials is shown in Table 3 of the paper for Caltech101, and the performances for the other datasets were listed in Table 1 of [16]. The bold-faced values indicate the best performance. The underlined values have no significant difference from the best performance, where the significance is based on the one-sample t-test. For all datasets, LSFW achieved the smallest top- k error. Most of the error ratios of LSFW were significantly smaller than those of StdFW and PG. That may be because LSFW successfully produces sufficiently accurate solutions for training within 1,000 iterations, whereas the other two could not.

7. Conclusions

In this study, we presented a new Frank-Wolfe algorithm that can be applied to the mdos-type learning machines. The mdos-type is a new class that has been introduced in this study to analyze, in a unified fashion, a wide variety of loss functions originating from the max-hinge loss. The sublinear convergence of the Frank-Wolfe algorithm is ensured if both the direction-finding and line search steps are implemented accurately. We found that if the Frank-Wolfe algorithm is applied to the Fenchel dual of the regularized empirical risk function, closed form solutions exist for both steps. If the loss function is replaced to its Moreau envelope, the replaced loss function is no longer the mdos-type, implying that the proposed Frank-Wolfe cannot be directly applied. We showed that the proposed Frank-Wolfe can be applied again with closed-form updates by reusing the technique discovered by Kato and Hirohashi [11]. We carried out experiments to empirically show that our algorithm converges faster and achieves an improved pattern recognition performance compared to the existing methods.

Acknowledgements

This research was performed by the Environment Research and Technology Development Fund JPMEERF20205006 of the Environmental Restoration and Conservation Agency of Japan and by JSPS KAKENHI Grant Number 19K04661.

References

- [1] D.P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.
- [2] D. Chu, R. Lu, J. Li, X. Yu, C. Zhang, and Q. Tao, "Optimizing top- k multiclass SVM via semismooth newton algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol.29, no.12, pp.6264–6275, Dec. 2018.
- [3] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol.2, pp.265–292, March 2002.
- [4] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.32, no.9, pp.1627–1645, Sept. 2010.
- [5] M. Frank and P. Wolfe, "An algorithm for quadratic programming," *Naval Research Logistics Quarterly*, vol.3, no.1-2, pp.95–110, March 1956. doi:10.1002/nav.3800030109
- [6] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S.L. Hicks, and P.H.S. Torr, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.38, no.10, pp.2096–2109, Oct. 2016.
- [7] P.J. Huber, "Robust estimation of a location parameter," *The Annals of Mathematical Statistics*, vol.35, no.1, pp.73–101, March 1964.
- [8] M. 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, Proceedings of Machine Learning Research*, vol.28, no.1, pp.427–435, Atlanta, Georgia, USA, 17–19 June 2013. PMLR.
- [9] T. Joachims, "A support vector method for multivariate performance measures," *Proceedings of the 22nd international conference on Machine learning - ICML '05*, ACM Press, pp.377–384, 2005.
- [10] T. Joachims, T. Finley, and C.-N. John Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol.77, no.1, pp.27–59, May 2009.
- [11] T. Kato and Y. Hirohashi, "Learning weighted top- k support vector machine," In W.S. Lee and T. Suzuki, editors, *Proceedings of The Eleventh Asian Conference on Machine Learning*, volume 101 of *Proceedings of Machine Learning Research*, pp.774–789, Nagoya, Japan, 17–19 Nov. 2019. PMLR.
- [12] M. Lapin, M. Hein, and B. Schiele, "Top- k multiclass svm," *Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15*, vol.1, pp.325–333, Cambridge, MA, USA, 2015. MIT Press.
- [13] J. Rennie and N. "Srebro, Loss functions for preference levels: Regression with discrete ordered labels," *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, 01 2005.
- [14] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: primal estimated sub-gradient solver for SVM," *Math. Program.*, vol.127, no.1, pp.3–30, 2011.
- [15] S. Shalev-Shwartz and T. Zhang, "Stochastic dual coordinate ascent methods for regularized loss," *J. Mach. Learn. Res.*, vol.14, no.1, pp.567–599, Feb. 2013.
- [16] K. Tajima, Y. Hirohashi, E.R. Rey Zara, and T. Kato, Frank-wolfe algorithm for learning svm-type multi-category classifiers, 2020. <https://arxiv.org/abs/2008.08894>.
- [17] K. Tajima, Y. Hirohashi, E.R. Rey Zara, and T. Kato, "Frank-wolfe algorithm for learning svm-type multi-category classifiers," In, editor, *Proceedings of SIAM International Conference on Data Mining (SDM21)*, pp.432–440, Virginia, USA, April 2021. SIAM.
- [18] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol.6, pp.1453–1484, 2005.
- [19] N. Usunier, D. Buffoni, and P. Gallinari, "Ranking with ordered weighted pairwise classification," *Proceedings of the 26th Annual International Conference on Machine Learning - ICML09*, ACM Press, 2009.
- [20] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," *Twenty-first international conference on Machine learning - ICML04*, ACM Press, 2004.

Keyna Tajima received his B.S. degree from Gunma University, Japan in 2020. His research interests include machine learning and data science.

Yoshihiro Hirohashi received his B.E., M.E. degrees from Gunma University, Gunma, Japan in 2013, from Tohoku University, Sendai, Japan in 2015, respectively. From 2015 to 2017, he was a research engineer at Toshiba Corporation. From 2017 to currently, he works at DENSO Corporation as an engineer for image recognition products. His research interests include computer vision, machine learning, and optimization theory.

Esmeraldo Zara received his B.S. and M.S. degrees from the University of the Philippines in 2009 and 2012 respectively and worked as an instructor till 2013. From 2014 to 2019, he was a IB Physics/Math teacher in Ritsumeikan Uji. He is now a doctoral student at the Graduate School of Science and Technology, Gunma University. His research interests include image processing, machine learning and optimization.

Tsuyoshi Kato received his B.E., M.E., and PhD degrees from Tohoku University, Sendai, Japan, in 1998, 2000, and 2003, respectively. He is now an associate professor at the Graduate School of Science and Technology, Gunma University. His current scientific interests include pattern recognition, computer vision, water engineering and bioinformatics. He is a member of IPSJ and IEICEJ.