Electronic Preprint for Journal of Information Processing Vol.28

Regular Paper

Learning Weighted Top-k Support Vector Machine

Yoshihiro Hirohashi[1,a)] Tsuyoshi Kato[2,†1,b)]

Received: October 19, 2019, Accepted: April 8, 2020

Abstract: *Top-k error ratio* is a popular performance measure for multi-category classification in which the number of categories is large. With the aim of obtaining the multi-category classifier minimizing the top-k error, Lapin et al. has developed the *top-k support vector machine* (top-k SVM) which is trained with the *top-k hinge loss*. Although top-k hinge is designed to be suitable for the top-k error, another loss or the top-k' hinge loss with $k' \neq k$ often yields a smaller top-k error ratio than the top-k hinge loss. This suggests that the top-k hinge loss is not always the optimal choice for the top-k error, which motivates us to explore variants of the top-k hinge loss. In this paper, we studied a weighted variant of the top-k hinge loss, and refer to the learning machine as the *weighted top-k SVM*. We developed a new optimization algorithm based on the *Frank-Wolfe algorithm* that requires no step size, enjoys the clear stopping criterion, and is never solicitous for computational instability. The Frank-Wolfe algorithm repeats the *direction finding step* and the *line search step*. The discoveries in this study are that both the steps can be given in a closed form. By smoothing the loss function, geometrical convergence can be achieved. Experimental results reveal that the weighted top-k SVM often achieved the better pattern recognition performance compared to the unweighted top-k SVM.

Keywords: top-k SVM, empirical risk minimization, convex optimization, Frank-Wolfe algorithm

1. Introduction

Top-k error ratio is a popular performance measure for multi-category classification in many fields including computer vision and natural language processing in which the number of categories is large. When using the performance measure, top-k error, the multi-category classifier of interest is supposed to give *top-k outputs* for an unknown output. In evaluation with the top-k error ratio, the testing examples are counted if the set of the top-k outputs does not contain the true category. The criterion is more suitable when the number of categories is larger which makes heavier the ambiguity of the boundaries among categories.

With the aim of obtaining the multi-category classifier minimizing the top-k error, Lapin et al. [10] has developed the *top-k support vector machine* (top-k SVM) which is trained with the *top-k hinge loss*. Although top-k hinge is designed to be suitable for the top-k error, another loss or the top-k' hinge loss with $k' \neq k$ often yields a smaller top-k error ratio than the top-k hinge loss, as reported in Ref. [10]. This suggests that the top-k hinge loss is not always the optimal choice for the top-k error, which motivates us to explore variants of the top-k hinge loss [1], [3], [16], [17].

In most of modern machine learning methods, the values of model parameters are determined by *empirical risk minimization* (ERM). A shortcoming that many algorithms for ERM suffer is that insufficient manual tuning of parameters for optimization often induces an optimization failure. For example, the number of *epochs* and a *step size* have to be chosen carefully by monitoring the learning curve in training deep neural networks. Meanwhile, the framework of *stochastic dual coordinate ascent* (SDCA) algorithm [15] does not entail any manual tuning. At each iteration of SDCA, an upper bound of the *objective gap*, which is the difference between the current primal objective value and the minimum, can be computed, making the accuracy of the solution guaranteed by stopping iterations when the upper bound is small enough. Furthermore, SDCA works without a step size. In SDCA, a set of the model parameters is divided into many blocks. At each iteration, one of the blocks is chosen randomly, and the rest of the blocks are fixed whereas the chosen block is optimized. Lapin et al. [10] have employed SDCA to train the top-k SVM. They have attempted to develop a projection algorithm to solve the sub-problem for optimization of a block of variables in each iteration of SDCA. Chu et al. [4] have developed a Newton-based method for SDCA update, and demonstrated that their algorithm was faster than the projection algorithm in their numerical experiments. Both the algorithms are specialized to the top-k hinge loss, forcing the applicability to variants of the top-k hinge to be limited. This is one of the reasons for developing a new optimization algorithm that can also be applied to a wide class of extensions of the top-k hinge loss function.

In this paper, we consider a weighted variant of the top-k hinge loss, and refer to the learning machine as the *weighted top-k support vector machine*. The weighted variant is a special case of the robust top-k hinge loss presented by Chang et al. [3] who have provided a difference of convex algorithm for learning the robust top-k SVM. Their algorithm requires careful adjustment of step size and sometimes fails to converge to the optimum. The new optimization algorithm developed in this study is based on the

[1] DENSO CORPORATION, Chuo, Tokyo 103–6015, Japan
[2] Graduate School of Science and Technology, Gunma University, Kiryu, Gunma 376–8515, Japan
[†1] Presently with Center for Research on Adoption of NextGen Transportation Systems, Gunma University
[a)] udpip.nnct@gmail.com
[b)] katotsu@cs.gunma-u.ac.jp

Frank-Wolfe algorithm [5] that requires no step size, enjoys the clear stopping criterion, and is never solicitous for computational instability. Frank-Wolfe algorithm repeats the *direction finding step* and the *line search step*. One of the discoveries in this study is that both the steps can be given in a closed form, which shall be presented in Section 5. The proposed algorithm can be applied not only to the original top-k SVM but also to the weighted variant, in spite of a much more complicated effective domain than that for the original top-k hinge loss (Section 4). By smoothing the loss function, some variants of the Frank-Wolfe algorithm can converge geometrically [8]. The proposed algorithm can be applied even when smoothing the weighted top-k hinge, which is described in Section 6. Experimental results reveal that the weighted top-k SVM outperforms the multi-category classifiers trained with Lapin et al. [10]'s and Chu et al. [4]'s learning methods (Section 7). Several fundamental techniques related to convex analysis are used throughout this paper. Readers unfamiliar to convex optimization may refer to some textbooks such as Ref. [2]. This paper is a journal version of our conference paper [7]. All the derivations of the proofs are referred to the supplemental materials of the conference paper [7].

Notation: We denote vectors by bold-faced lower-case letters and matrices by bold-faced upper-case letters. Entries of vectors and matrices are not bold-faced. The transposition of a matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^\top$, and the inverse of $\boldsymbol{A}$ is by $\boldsymbol{A}^{-1}$. We use $\boldsymbol{e}_i$ to denote a unit vector in which (i)-th entry is one and all the others are zero. The size of $\boldsymbol{e}_i$ depends on the context. The n-dimensional vector all of whose entries are one is denoted by $\mathbf{1}_n$. We use $\mathbb{R}$ and $\mathbb{N}$ to denote the set of real numbers and positive integers. Positive integers are referred to as *natural numbers* in this article. Note that $0 \notin \mathbb{N}$ in this definition. $\mathbb{R}^n$ and $\mathbb{N}^n$ to denote the set of n-dimensional real and natural vectors, and $\mathbb{R}^{m\times n}$ to denote the set of $m \times n$ real matrices. For any $n \in \mathbb{N}$, we use $[n]$ to denote the set of natural numbers less than or equal to n. The set of real nonnegative numbers is denoted by $\mathbb{R}_+$. For any $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, define $\langle \boldsymbol{x}, \boldsymbol{y}\rangle := \sum_{i=1}^n x_i y_i$ where x_i and y_i is the i-th entry of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. For any $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{m\times n}$, define $\langle \boldsymbol{X}, \boldsymbol{Y}\rangle := \sum_{i=1}^m \sum_{j=1}^n X_{i,j} Y_{i,j}$ where $X_{i,j}$ and $Y_{i,j}$ is the (i, j)-th entry of $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively. The notation, given so far, is standard and used in many literature.

We shall introduce the notation $\pi(j\,;\,\boldsymbol{s}) \in [m]$ which is the index of the j-th largest component in a vector $\boldsymbol{s} \in \mathbb{R}^m$. When using this notation, the vector $\boldsymbol{s}$ is omitted if there is no danger of confusion. Namely, for a vector $\boldsymbol{s} \in \mathbb{R}^m$, we can write $s_{\pi(1)} \geq s_{\pi(2)} \geq \cdots \geq s_{\pi(m)}$.

2. Empirical Risk Minimization

The linear multi-category classifier discussed in this paper has a parameter $\boldsymbol{W} := [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m] \in \mathbb{R}^{d\times m}$, where the number of categories is m, to predict the category label of an unknown input $\boldsymbol{x} \in \mathbb{R}^d$ by choosing the largest one from m prediction scores

$$\langle \boldsymbol{w}_1, \boldsymbol{x}\rangle, \ldots, \langle \boldsymbol{w}_m, \boldsymbol{x}\rangle. \tag{1}$$

In order to determine the value of the parameter $\boldsymbol{W}$, suppose that we are given n training examples,

$$(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times [m]. \tag{2}$$

Typical approach is the *empirical risk minimization* (ERM), in which the parameter $\boldsymbol{W}$ is set to the value that minimizes the regularized empirical risk defined as

$$P(\boldsymbol{W}) := \frac{\lambda}{2} \|\boldsymbol{W}\|_{\mathrm{F}}^2 + \frac{1}{n} \sum_{i=1}^n \Phi(\boldsymbol{W}^\top \boldsymbol{x}_i\,;\, y_i) \tag{3}$$

where $\lambda > 0$ is a regularization constant and $\Phi(\cdot\,;\, y) : \mathbb{R}^m \to \mathbb{R}$ is a convex loss function for a true class $y \in [m]$.

Dual methods have been adopted by several studies to find the minimizer of the regularized empirical risk [4], [6], [9], [10], [15]. The dual methods attempt to find the maximizer of the *Fenchel dual function* [2] given by

$$D(\boldsymbol{A}) := -\frac{\lambda}{2} \|\boldsymbol{W}(\boldsymbol{A})\|_{\mathrm{F}}^2 - \frac{1}{n} \sum_{i=1}^n \Phi^*(-\boldsymbol{\alpha}_i\,;\, y_i) \tag{4}$$

where $\boldsymbol{\alpha}_i$ is the i-th column in the $m \times n$ matrix $\boldsymbol{A}$ which is the dual variable; function $\Phi^*(\cdot\,;\, y_i) : \mathbb{R}^m \to \bar{\mathbb{R}}$ is the convex conjugate of the loss function $\Phi(\cdot\,;\, y_i)$ where $\bar{\mathbb{R}} := \mathbb{R} \cup \{+\infty\}$; function $\boldsymbol{W}(\cdot)$ is defined as

$$\boldsymbol{W}(\boldsymbol{A}) := \frac{1}{\lambda n} \boldsymbol{X} \boldsymbol{A}^\top \tag{5}$$

where $\boldsymbol{X} := [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$. One of the strong advantages of dual methods is that, during the iterations, the *duality gap* $P(\boldsymbol{W}(\boldsymbol{A})) - D(\boldsymbol{A})$ can be monitored (In the literature of optimization, the term, duality gap, is defined by the minimal gap between the primal and dual objective values, although the gap at any possible primal and dual feasible solutions is referred to as the duality gap in many machine learning literature). The duality gap vanishes at an optimum for most of the loss functions. When the duality gap is below a small positive threshold ϵ, the recovered primal variable $\boldsymbol{W}(\boldsymbol{A})$ ensures the ϵ-*accuracy*, i.e.,

$$P(\boldsymbol{W}) - \min_{\boldsymbol{W}' \in \mathbb{R}^{m\times n}} P(\boldsymbol{W}') \leq \epsilon, \tag{6}$$

which allows us to decide when to stop the iterations.

3. Unweighted Top-k Hinge

The learning algorithm for top-k SVM [10] attempts to minimize the regularized empirical risk where the empirical risk is evaluated with the average of the top-k hinge losses for training examples. The top-k hinge loss suffered for the prediction score $\boldsymbol{s} = \boldsymbol{W}^\top \boldsymbol{x}$ is defined as

$$\Phi_{\mathrm{utk}}(\boldsymbol{s}\,;\, y) := \max\left\{0, \frac{1}{k} \sum_{j=1}^k \left(\mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y \mathbf{1}_m\right)_{\pi(j)}\right\} \tag{7}$$

where $\boldsymbol{W}$ is a matrix of the model parameters. Then, how can we minimize the regularized empirical risk? Lapin et al. [10] have employed the stochastic dual coordinate ascent (SDCA) algorithm to find the minimizer in an iterative fashion. One column in $\boldsymbol{A}$ is selected at random, and updated at each iteration of SDCA. Lapin et al. [10] have developed an algorithm for updating a column and plugged it in to the framework of SDCA.

To express the convex conjugate of the top-k loss function, Lapin et al. [10] introduce the following convex polytope

$$\Delta(k, r) := \left\{\boldsymbol{\beta} \in \mathbb{R}_+^m \mid \langle \mathbf{1}, \boldsymbol{\beta}\rangle \leq r,\ \boldsymbol{\beta} \leq \frac{1}{k} \mathbf{1}\mathbf{1}^\top \boldsymbol{\beta}\right\} \tag{8}$$

and they call it the *top-k simplex*. Using the convex polytope, the top-k loss function can be re-expressed as

$$\Phi_{\mathrm{utk}}(\boldsymbol{s}\,;\,y) = \max_{\boldsymbol{\beta}\in\Delta(k,1)} \left\langle \boldsymbol{\beta}, \mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}_m \right\rangle. \tag{9}$$

From the Eq. (10), the convex conjugate can be derived as

$$\Phi^*_{\mathrm{utk}}(\boldsymbol{v}\,;\,y) = v_y \tag{10}$$

provided that the value of $\boldsymbol{v}$ satisfies

$$\langle \boldsymbol{v}, \mathbf{1}\rangle = 0, \quad \exists b_y \in \mathbb{R}, \quad \boldsymbol{v} + (b_y - v_y)\boldsymbol{e}_y \in \Delta(k,1); \tag{11}$$

otherwise, $\Phi^*_{\mathrm{utk}}(\boldsymbol{v}\,;\,y)$ goes infinity.

4. Weighted Top-k Hinge

In this section, an extension of the top-k hinge loss function is described. We use m pre-defined weights $\boldsymbol{\rho} := [\rho_1, \ldots, \rho_m]^\top$ such that $\rho_1 \geq \cdots \geq \rho_m \geq 0$. With these weights, we introduce the following loss function:

$$\Phi_{\mathrm{wtk}}(\boldsymbol{s}\,;\,y) := \max\left\{0, \sum_{j=1}^{m} \left(\mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}_m\right)_{\pi(j)} \rho_j\right\} \tag{12}$$

This function is referred to as the *weighted top-k hinge loss*. This definition is a special case of Chang et al. [3]'s extensions. They use an upper bound of the loss value, say τ. Their loss function is no more convex unless $\tau = +\infty$.

A sub-gradient of the weighted top-k hinge is given by $\forall j \in [m]$,

$$\frac{\Phi_{\mathrm{wtk}}(\boldsymbol{s}\,;\,y)}{\partial s_k} = \left(\rho_j - \langle \boldsymbol{\rho}, \mathbf{1}\rangle \delta_{k,y}\right) \mathbb{1}[\Phi_{\mathrm{wtk}}(\boldsymbol{s}\,;\,y) > 0] \tag{13}$$

for $k = \pi(j\,;\,\mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}_m)$, where $\mathbb{1}[x] = 1$ if the argument x is true; $\mathbb{1}[x] = 0$, otherwise. We have used $\delta_{\cdot,\cdot}$ to denote the Kronecker's delta. Equation (13) suggests that $O(m \log m)$ cost is needed for computing the gradient. This fact shall be used in the next section.

To exploit the duality gap for a stopping criterion, the convex conjugate of the weighted top-k hinge loss is required. To derive the convex conjugate, we use the following lemma:

Lemma 1. Let $y \in [m]$ and $\boldsymbol{\delta} \in \mathbb{R}^m$ such that $\delta_y = 0$. With a non-empty convex polyhedron $\mathcal{B} \subseteq \mathbb{R}^m$, define a function $\Phi : \mathbb{R}^m \to \mathbb{R}$ as

$$\Phi(\boldsymbol{s}) := \max_{\boldsymbol{\beta}\in\mathcal{B}} \left\langle \boldsymbol{\beta}, \boldsymbol{\delta} + \boldsymbol{s} - \mathbf{1}_m s_y \right\rangle. \tag{14}$$

The convex conjugate of Φ is then expressed as

$$\Phi^*(\boldsymbol{v}) = \begin{cases} -\langle \boldsymbol{v}, \boldsymbol{\delta}\rangle & \text{if} \quad \boldsymbol{v} \in \mathrm{dom}(\Phi^*), \\ +\infty & \text{otherwise,} \end{cases} \tag{15}$$

where $\mathrm{dom}(\Phi^*)$ is the *effective domain* of Φ^* which is given by

$$\mathrm{dom}(\Phi^*) = \left\{\boldsymbol{v} \in \mathbb{R}^m \,\middle|\, \langle \boldsymbol{v}, \mathbf{1}\rangle = 0, \; \exists \beta_y \in \mathbb{R}, \quad \boldsymbol{v} + (\beta_y - v_y)\boldsymbol{e}_y \in \mathcal{B}\right\}. \tag{16}$$

In the case of the unweighted top-k hinge loss (7), the convex conjugate (10) and its effective domain (11) are indeed derived by setting $\mathcal{B} := \Delta(k, 1)$ and $\boldsymbol{\delta} = \mathbf{1} - \boldsymbol{e}_y$.

The convex conjugate of the weighted top-k hinge loss can also be derived with the use of Lemma 1 as follows. Preliminary to application of the lemma, we shall first observe that the weighted top-k hinge loss can be re-expressed in the form of Eq. (14). There exist an index set $\mathcal{K} := \{k_1, \ldots, k_{|\mathcal{K}|}\} \subseteq [m]$ and a transformed weights $\boldsymbol{\rho}' := \left[\rho'_1, \ldots, \rho'_{|\mathcal{K}|}\right]^\top$ such that

$$\Phi_{\mathrm{wtk}}(\boldsymbol{s}\,;\,y) = \max\left\{0, \sum_{\ell=1}^{|\mathcal{K}|} \rho'_\ell g_\ell(\boldsymbol{s})\right\} \tag{17}$$

where

$$\forall \ell \in |\mathcal{K}|, \quad g_\ell(\boldsymbol{s}) := \sum_{j=1}^{k_\ell} \left(\mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}_m\right)_{\pi(j)}. \tag{18}$$

If defining a convex polyhedron $\mathcal{B}_{\mathrm{wtk}}$ as

$$\mathcal{B}_{\mathrm{wtk}} := \left\{\boldsymbol{\beta} \in \mathbb{R}^m \,\middle|\, \exists \zeta \in \mathbb{R}, \; \forall \ell \in [|\mathcal{K}|], \; \exists \boldsymbol{\lambda}_\ell \in \Delta(k_\ell, \rho'_\ell k_\ell), \; \zeta = \frac{\langle \mathbf{1}, \boldsymbol{\lambda}_\ell\rangle}{k_\ell \rho'_\ell}, \quad \boldsymbol{\beta} = \boldsymbol{\lambda}_1 + \cdots + \boldsymbol{\lambda}_{|\mathcal{K}|}\right\}, \tag{19}$$

the loss function can be re-written as

$$\Phi_{\mathrm{wtk}}(\boldsymbol{s}\,;\,y) = \max_{\boldsymbol{\beta}\in\mathcal{B}_{\mathrm{wtk}}} \left\langle \boldsymbol{\beta}, \mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}_m \right\rangle. \tag{20}$$

Thusly, it has been confirmed that the weighted top-k hinge loss satisfies the assumption of Lemma 1, which leads to the following result.

Theorem 2. The convex conjugate of the weighted top-k hinge loss is expressed as

$$\Phi^*_{\mathrm{wtk}}(\boldsymbol{v}\,;\,y) = \begin{cases} v_y & \text{if } \langle \boldsymbol{v}, \mathbf{1}\rangle = 0, \\ & \quad \exists b_y \in \mathbb{R}, \; \boldsymbol{v} + (b_y - v_y)\boldsymbol{e}_y \in \mathcal{B}_{\mathrm{wtk}}, \\ +\infty & \text{o.w.} \end{cases} \tag{21}$$

Our goal is the development of optimization algorithms for ERM based on the weighted top-k hinge loss, in which no step size is required and the clear stopping criterion is provided like SDCA. Lapin et al. [10] and Chu et al. [4] have tried to develop a key ingredient of SDCA which optimizes a chosen column of the dual variable $\boldsymbol{A}$ for the unweighted top-k hinge loss. For the weighted extension of the top-k hinge, a serious obstacle against the development of such an algorithm is a much more complicated effective domain of the dual variables, $-\mathrm{dom}(\Phi^*_{\mathrm{wtk}}(\cdot\,;\,y_i))$. In the next section, we present a new optimization algorithm to avoid facing the rather complicated problem directly for updating a column of $\boldsymbol{A}$.

5. Learning Algorithm

In this section, a new optimization algorithm for learning weighted top-k SVM is presented. The algorithm developed in this study is based on Frank-Wolfe framework [5] which iteratively maximizes a function over a convex polyhedron. In the dual problem for ERM, the polyhedron is the effective domain of the negative dual objective

Algorithm 1: General framework of Frank-Wolfe algorithm.

```
begin
    A^(0) ∈ dom(−D);
    for t := 1 to T do
        U^(t−1) ∈ argmax_{U∈dom(−D)} ⟨∇D(A^(t−1)), U⟩;
        ΔA^(t−1) := U^(t−1) − A^(t−1);
        γ^(t−1) ∈ arg max_{γ∈[0,1]} D(A^(t−1) + γΔA^(t−1));
        A^(t) := A^(t−1) + γ^(t−1) ΔA^(t−1);
    end
end
```

$$\mathrm{dom}(-D) = (-\mathrm{dom}\Phi^*(\cdot; y_1)) \times \cdots \times (-\mathrm{dom}\Phi^*(\cdot; y_n)). \quad (22)$$

As presented in Algorithm 1, each iteration of Frank-Wolfe framework consists of two steps: *direction finding step* and *line search step*.

In the direction finding step, the optimal solution that maximizes the *linearized* objective function over the polyhedron is searched, where the linearized objective function is given by

$$\left\langle \nabla D(\boldsymbol{A}^{(t-1)}), \boldsymbol{U} - \boldsymbol{A}^{(t-1)} \right\rangle + D(\boldsymbol{A}^{(t-1)}) \quad (23)$$

which is the first-order Taylor expansion of the dual objective $D(\cdot)$ around the previous solution $\boldsymbol{A}^{(t-1)}$. If denoting the solution of this linear programming (LP) problem by $\boldsymbol{U}^{(t-1)}$, the new direction is determined as $\Delta\boldsymbol{A}^{(t-1)} := \boldsymbol{U}^{(t-1)} - \boldsymbol{A}^{(t-1)}$.

In the line search step, the optimal point is searched on the line segment between $\boldsymbol{A}^{(t-1)}$ and $\boldsymbol{A}^{(t-1)} + \Delta\boldsymbol{A}^{(t-1)}$. The optimal point is expressed as $\boldsymbol{A}^{(t)} := \boldsymbol{A}^{(t-1)} + \gamma^{(t-1)}\Delta\boldsymbol{A}^{(t-1)}$ where

$$\gamma^{(t-1)} := \operatorname*{argmax}_{\gamma\in[0,1]} D\left(\boldsymbol{A}^{(t-1)} + \gamma\Delta\boldsymbol{A}^{(t-1)}\right). \quad (24)$$

The line search step can be expressed in a closed form so long as the convex conjugates of the loss functions are an affine or quadratic function. For the weighted top-k SVM, this step can be written as

$$\gamma^{(t-1)} = \max(0, \min(1, \hat{\gamma}^{(t-1)})), \quad (25)$$

where

$$\hat{\gamma}^{(t-1)} := \frac{\lambda n \left\langle \Delta\boldsymbol{A}^{(t-1)}, \boldsymbol{E}_{\boldsymbol{y}} - \boldsymbol{Z}(\boldsymbol{A}^{(t-1)}) \right\rangle}{\langle \Delta\boldsymbol{A}^{(t-1)}\boldsymbol{K}, \Delta\boldsymbol{A}^{(t-1)} \rangle}, \qquad \boldsymbol{K} := \boldsymbol{X}^\top\boldsymbol{X},$$

$$\text{and} \quad \boldsymbol{Z}(\boldsymbol{A}) := \frac{1}{\lambda n}\boldsymbol{K}\boldsymbol{A}^\top, \quad \boldsymbol{E}_{\boldsymbol{y}} := \left[\boldsymbol{e}_{y_1}, \ldots, \boldsymbol{e}_{y_n}\right]. \quad (26)$$

This step requires $O(mn\min(d, n))$ computation.

Then, how to compute the LP solution required in the direction finding step? Does the LP problem for this step entail the use of a general-purpose solver in every iteration? The answer is no. This study has discovered that the direction finding step can be given in a closed form and takes only $O(nm\log m)$ computation. Below we shall derive the algorithm. From the expressions of the linearization approximation and the effective domain of the dual objective, it is seen that the linear programming problem can be divided into n independent and smaller LP problems: for $i = 1, \ldots, n$,

$$\max \quad \left\langle \frac{\partial D(\boldsymbol{A}^{(t-1)})}{\partial \boldsymbol{\alpha}_i}, \boldsymbol{u}_i \right\rangle \qquad \text{wrt} \quad \boldsymbol{u}_i \in -\mathrm{dom}(\Phi^*_{\mathrm{wtk}}(\cdot; y_i)). \quad (27)$$

The LP solution for the direction finding step $\boldsymbol{U}^{(t-1)}$ is obtained by solving each of n smaller LP problems and concatenating these n optimal solutions $\boldsymbol{u}_i^{(t-1)}$ as $\boldsymbol{U}^{(t-1)} := [\boldsymbol{u}_1^{(t-1)}, \ldots, \boldsymbol{u}_n^{(t-1)}]$. The gradient with respect to the i-th column in $\boldsymbol{A}$ is expressed as

$$\frac{\partial D(\boldsymbol{A})}{\partial \boldsymbol{\alpha}_i} = \frac{1}{n}(\boldsymbol{z}_i(\boldsymbol{A}) - \boldsymbol{e}_{y_i}) \quad (28)$$

where $\boldsymbol{z}_i(\boldsymbol{A})$ is the i-th column of $\boldsymbol{Z}(\boldsymbol{A})$.

A naïve way to finding the optimal solution $\boldsymbol{u}_i^{(t-1)}$ to each of n LPs is the use of a general-purpose LP solvers. The variables to be determined in each LP problem are $\boldsymbol{u}_i \in \mathbb{R}^m$ as well as $\beta_y, \zeta \in \mathbb{R}$ and $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{|\mathcal{K}|} \in \mathbb{R}^m$ in its LP form. The computational time for solving each LP with a general-purpose solver is prohibitive if the number of classes is large. In this study, the following lemma has been found, which brings an $O(m\log m)$ algorithm for solving each LP.

Lemma 3. Let $\phi : \mathbb{R}^m \to \mathbb{R}$ be a convex function whose convex conjugate ϕ_* is given by $-\phi_*(-\boldsymbol{\alpha}) = \langle \boldsymbol{f}, \boldsymbol{\alpha} \rangle$ for $\boldsymbol{\alpha} \in -\mathrm{dom}\phi_*$, where $\boldsymbol{f} \in \mathbb{R}^m$ is a constant vector. Then, it holds that

$$\forall \eta \in \mathbb{R}_{++}, \qquad \operatorname*{argmax}_{\boldsymbol{\alpha}\in-\mathrm{dom}(\phi_*)} \langle \boldsymbol{g}, \boldsymbol{\alpha} \rangle = -\partial\phi(\boldsymbol{f} - \eta\boldsymbol{g}) \quad (29)$$

where $\partial\phi(\boldsymbol{x})$ is the sub-differential of ϕ at $\boldsymbol{x} \in \mathbb{R}^m$.

By substituting $\boldsymbol{f} := \boldsymbol{e}_{y_i}$, $\boldsymbol{g} := (\boldsymbol{e}_{y_i} - \boldsymbol{z}_i(\boldsymbol{A}^{(t-1)}))/n$ and $\eta := n$ into the result of Lemma 3, a solution optimal to the LP (27) can be expressed in a closed form as

$$\boldsymbol{u}_i^{(t-1)} := -\nabla\Phi_{\mathrm{wtk}}(\boldsymbol{z}_i(\boldsymbol{A}^{(t-1)}); y_i) \quad (30)$$

where $\nabla\Phi_{\mathrm{wtk}}(\boldsymbol{z}_i(\boldsymbol{A}^{(t-1)}))$ is a sub-gradient of the weighted top-k hinge at $\boldsymbol{z}_i(\boldsymbol{A}^{(t-1)})$. On computing $\boldsymbol{Z}(\boldsymbol{A}^{(t-1)})$, it takes $O(m\log m)$ time to compute $\boldsymbol{u}_i^{(t-1)}$, as shown in the previous section. These results can be summarized in the following theorem.

Theorem 4. Consider the Frank-Wolfe algorithm for maximizing $D(\boldsymbol{A})$ with $\Phi^*(\cdot; y_i) = \Phi^*_{\mathrm{wtk}}(\cdot; y_i)$ for $i = 1, \ldots, n$. Every iteration consisting of the direction finding step and the line search step can be done in $O(nm(\min(d, n) + \log m))$ computational time.

The techniques presented in this section make efficient every iteration not only of the classical Frank-Wolfe but also of its variants such as *away-step Frank-Wolfe* (AFW) and *pairwise Frank-Wolfe* (PFW) algorithms. Recently Lacoste-Julien and Jaggi [8] have proved the *linear convergence* for some variants of the Frank-Wolfe algorithm. When employing these variants of the Frank-Wolfe algorithm, the upper bound of the objective gap $\mathrm{dgap}(\boldsymbol{A}) := \min_{\boldsymbol{W}} P(\boldsymbol{W}) - \boldsymbol{D}(\boldsymbol{A})$ is guaranteed to geometrically decrease as $\mathrm{dgap}(\boldsymbol{A}^{(t)}) \leq \exp(-\zeta t)$ where ζ is a constant dependent on the optimization problem [8]. Their theories are based on an assumption that the objective function must be *smooth* and *strongly convex* [12], although $-D(\cdot)$ does not possess the strongly convex property in the setting discussed so far. In the next section, we introduce the technique of *Moreau envelope* [14] to the weighted top-k hinge, which endows the objective with the strong convexity.

6. Optimization for Smoothed Top-k Hinge

The two aforementioned top-k hinge losses, Eqs. (7) and (12), suffer from the discontinuity in the derivatives. Several Refs. [11], [13], [15] have considered smoothing loss functions to obtain a better property for optimization. Following Ref. [11], the Moreau envelope, which is a typical approach to smoothing, is introduced for the weighted top-k hinge loss in this study. The *smoothed weighted top-k hinge loss* is given by

$$\Phi_{\mathrm{stk}}(\boldsymbol{s}\,;\,y) := \min_{\boldsymbol{z}\in\mathbb{R}^m}\left(\Phi_{\mathrm{wtk}}(\boldsymbol{z}\,;\,y) + \frac{1}{2\gamma}\|\boldsymbol{s}-\boldsymbol{z}\|^2\right) \tag{31}$$

where $\gamma > 0$ is a smoothing constant. Here we discuss how to find $\boldsymbol{W} \in \mathbb{R}^{d\times m}$ that minimizes the regularized empirical risk based on the smoothed loss, denoted by $P_{\mathrm{stk}} : \mathbb{R}^{d\times m} \to \mathbb{R}$, which is given in (3) with $\Phi(\cdot\,;y_i) = \Phi_{\mathrm{stk}}(\cdot\,;\,y_i)$ for $i = 1,\ldots,n$. To use dual methods for learning with this smoothed loss function, the dual objective, denoted by $D_{\mathrm{stk}} : \mathbb{R}^{m\times n} \to -\bar{\mathbb{R}}$, must be maximized with respect to the dual variables $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. It can be seen that the dual objective $-D_{\mathrm{stk}}$ is strongly convex with coefficient γ/n which is proportional to the constant ζ. It is not straightforward to develop an efficient Frank-Wolfe iteration again to solve this dual problem, because the convex conjugate of the smoothed loss is no longer a linear function which violates the assumption of Lemma 3. Nonetheless, the Frank-Wolfe framework is re-used in this study, with the help of the following proposition.

Proposition 5. Let $\tilde{\boldsymbol{x}}_i := \left[\boldsymbol{x}_i^\top, \sqrt{\gamma\lambda n}\boldsymbol{e}_i^\top\right]^\top \in \mathbb{R}^{d+n}$ for $i = 1,\ldots,n$. Then, the optimization problem for maximizing $D_{\mathrm{stk}}(\boldsymbol{A})$ is not only dual to the minimization problem with the primal objective $P_{\mathrm{stk}} : \mathbb{R}^{d\times m} \to \mathbb{R}$ but also dual to the minimization problem with the objective function $\tilde{P}_{\mathrm{wtk}} : \mathbb{R}^{(d+n)\times m} \to \mathbb{R}$ defind as

$$\tilde{P}_{\mathrm{wtk}}(\tilde{\boldsymbol{W}}) := \frac{\lambda}{2}\left\|\tilde{\boldsymbol{W}}\right\|_{\mathrm{F}}^2 + \frac{1}{n}\sum_{i=1}^{n}\Phi_{\mathrm{wtk}}(\tilde{\boldsymbol{W}}^\top\tilde{\boldsymbol{x}}_i\,;\,y_i). \tag{32}$$

This proposition suggests that the learning problem for the smoothed loss can be transformed back to that for the non-smoothed loss. This enables us to re-use the algorithm presented in Section 5 — the trick for direction finding step, in particular — with the kernel matrix $\boldsymbol{K}$ replaced to $\tilde{\boldsymbol{K}} := \boldsymbol{X}^\top\boldsymbol{X} + \gamma\lambda n\boldsymbol{I}$. The iterations can be stopped when the following duality gap is small enough:

$$\begin{aligned}\mathrm{Gap}_{\mathrm{stk}}(\boldsymbol{A}) &:= \tilde{P}_{\mathrm{wtk}}(\tilde{\boldsymbol{W}}(\boldsymbol{A})) - D_{\mathrm{stk}}(\boldsymbol{A}) \\ &= \frac{\gamma}{n}\|\boldsymbol{A}\|_{\mathrm{F}}^2 + \frac{1}{\lambda n^2}\left\langle \boldsymbol{A}\boldsymbol{K} - \lambda n\boldsymbol{E}_{\boldsymbol{y}}, \boldsymbol{A}\right\rangle \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}\Phi_{\mathrm{wtk}}(\boldsymbol{z}_i(\boldsymbol{A}) + \gamma\boldsymbol{\alpha}_i).\end{aligned} \tag{33}$$

The above observations suggest that neither the $(d+n)$-dimensional vectors $\tilde{\boldsymbol{x}}_1,\ldots,\tilde{\boldsymbol{x}}_n$ nor the model parameters $\tilde{\boldsymbol{W}} \in \mathbb{R}^{(d+n)\times m}$ have to be unfolded in the computational memory to implement the Frank-Wolfe algorithm for minimizing $\tilde{P}_{\mathrm{utk}}(\tilde{\boldsymbol{W}})$ and to monitor the duality gap.

7. Experiments

We shall demonstrate the convergence behaviors of the proposed Frank-Wolfe algorithms for the top-k SVM learning, followed by reporting the pattern recognition performances with top-k accuracies on several datasets for benchmarking multi-class classifiers. The proposed Frank-Wolfe algorithms were implemented in Python. The Python code is available at https://github.com/hirohashi/wtopk.

7.1 How Does Smoothing Affect Convergence?

We investigated how the smoothing technique affected the convergence. In Section 6, the smoothed weighted top-k SVM can be trained again with the Frank-Wolfe algorithm for non-smooth weighted top-k SVM presented in Section 5. Theoretically, a faster convergence rate can be achieved if the coefficient of strong convexity is larger, and the larger coefficient can be generated with a larger smoothing coefficient γ. In the experiments presented here, the smoothing coefficient γ is varied with 0, 10^{-3}, 10^{-2}, and 10^{-1}, where the value $\gamma = 0$ does not change the non-smooth loss function. **Figure 1** plots the duality gaps against the number of iterations. The duality gaps produced with Std FW, AFW, and PFW, respectively, are shown in Fig. 1 (a), (b), (c). The dataset used here is News20. When using $\gamma = 10^{-1}$, the duality gap felt below 10^{-3} within only eight iterations. For $\gamma = 10^{-3}$ and $\gamma = 10^{-2}$, the dual gaps are decreased quickly for the first several iterations, although the convergence speeds slowed down suddenly. This might be the zigzag phenomenon discussed in Ref. [8]. Meanwhile, such a slowdown was not observed when using AFW and PFW with $\gamma = 10^{-2}$. The duality gaps for $\gamma = 10^{-3}$ were decreased almost linearly on the log-log plots, although, due to the mild slopes, the number of iterations to attain 10^{-3} of the duality gap did not differ largely from the ones of non-smooth loss.

7.2 Convergence Behavior for Weighted Top-k SVM

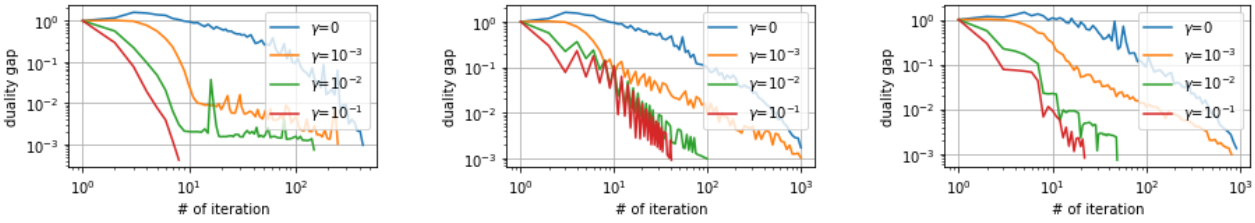
One advantage of the proposed algorithms is that dual variables can be optimized within the feasible region that has a much more complicated shape by having weights for differences in the prediction scores, as defined in Eq. (12). In the experiments reported here, three types of the weights, called `flat`, `linear` and `exp`, were examined. The `flat`, `linear` and `exp` weights, respectively, were designed as

$$\rho_j^{\mathrm{flat}} := \frac{1}{k},\quad \rho_j^{\mathrm{linear}} := \frac{2(k+1-j)}{(k+1)k},\quad \rho_j^{\mathrm{exp}} := \frac{\exp(-j/k)}{\sum_{j'=1}^{k}\exp(-j'/k)}$$

for $j \le k$, and the remaining weights are zero. The `flat` recovers the unweighted top-k hinge, whereas the `linear` and `exp` weights, respectively, decrease the coefficient ρ_j linearly and exponentially as j goes larger. The convergence behaviors for the three weight types on FMD and News20 were plotted in **Fig. 2**. No significant differences amongst the three weight types were observed despite the effective domains complicated by weighting.

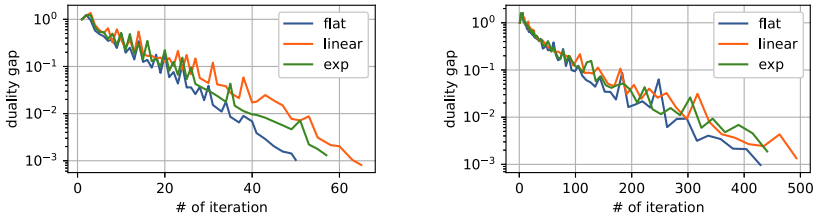
7.3 Pattern Recognition Performance

Finally, the pattern recognition performances of the proposed



(a) Std FW (b) AFW (c) PFW

Fig. 1 Convergence behavior for smooth unweighted top-k hinge. The duality gaps against the number of iterations with three Frank-Wolfe algorithms, Std FW, AFW, and PFW, are plotted in (a), (b), and (c), respectively. The smoothing parameter γ was varied with 0, 10^{-3}, 10^{-2}, and 10^{-1}. Larger γ yields smoother loss. The unweighted top-k hinge smoothed with $\gamma = 0$ is still non-smooth. Convergence was faster with larger γ.



(a) FMD (b) News20

Fig. 2 Comparisons of the weighted and unweighted top-k hinges. The `flat` indicates the unweighted top-k hinge, whereas the `linear` and `exp` represent two types of weighted top-k hinges with the weights decreasing linearly and exponentially, respectively. In spite of the complicated effective domains, the convergence behaviors of the weighted top-k hinges resemble those of the unweighted top-k hinges.

Table 1 Top-k accuracies of the proposed weighted top-k SVMs and the unweighted top-k SVMs trained with different optimization algorithms. `WTk (linear)` and `WTk (exp)`, respectively, indicate the weighted top-k SVMs with weight coefficients decreasing linearly and exponentially. `UTk (ours)`, `UTk (lapin)`, and `UTk (chu)` are the unweighted top-k SVMs obtained with the proposed Frank-Wolfe algorithm, Lapin et al. [10]'s algorithm, and Chu et al. [4]'s algorithm. In most cases, the weighted top-k SVMs and the unweighted one learnt by our algorithm achieved better pattern recognition performances than the existing learning methods.

(a) ALOI

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours)	0.841	0.929	0.948	0.973
WTk (linear)	**0.842**	0.929	0.949	0.973
WTk (exp)	0.841	**0.930**	**0.949**	**0.974**
UTk (Lapin)	0.834	0.929	0.949	0.965
UTk (Chu)	0.825	0.920	0.949	0.972

(b) Caltech101

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours)	0.548	0.719	**0.777**	0.844
WTk (linear)	**0.550**	**0.723**	0.774	0.843
WTk (exp)	0.547	0.722	0.775	**0.844**
UTk (Lapin)	0.544	0.723	0.767	0.827
UTk (Chu)	0.535	0.718	0.773	0.829

(c) CUB

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours)	**0.592**	0.777	**0.847**	**0.908**
WTk (linear)	**0.592**	**0.780**	0.844	0.908
WTk (exp)	**0.592**	0.778	0.843	0.908
UTk (Lapin)	0.580	0.770	0.834	0.901
UTk (Chu)	0.579	0.768	0.842	0.903

(d) Indoor67

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours)	0.697	0.878	0.925	**0.969**
WTk (linear)	**0.697**	**0.881**	0.930	0.968
WTk (exp)	0.697	0.879	**0.931**	0.968
UTk (Lapin)	0.688	0.877	0.927	0.966
UTk (Chu)	0.683	0.875	0.924	0.968

(e) Letter

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours)	0.759	0.910	**0.961**	0.994
WTk (linear)	**0.766**	0.909	0.955	0.991
WTk (exp)	0.765	0.908	0.957	0.991
UTk (Lapin)	0.761	0.907	0.951	0.988
UTk (Chu)	0.761	**0.910**	0.960	**0.995**

(f) News20

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours)	0.666	**0.876**	**0.929**	0.975
WTk (linear)	0.666	0.872	0.929	0.976
WTk (exp)	**0.666**	0.875	0.929	**0.976**
UTk (Lapin)	0.657	0.875	0.926	0.972
UTk (Chu)	0.662	0.865	0.922	0.975

learning methods were investigated. We used the top-k accuracy for the performance measure for multi-category classifiers, where the top-k accuracy is the ratio of testing examples each of which the prediction score of the correct category is in the top-k outputs. We chose $k = 1, 3, 5, 10$. For weighted top-k SVM, three types of weights, $\boldsymbol{\rho}^{\text{flat}}$, $\boldsymbol{\rho}^{\text{linear}}$, and $\boldsymbol{\rho}^{\text{exp}}$, were examined. These are denoted by `UTk (ours)`, `WTk (linear)`, and `WTk (exp)`, respectively. These three multi-category SVMs were trained with the standard Frank-Wolfe algorithms presented in Section 5. The regularization parameter was chosen by $\lambda = 1/(nC)$ where $C = 10^{-3}, 10^{-2}, \ldots, 10^{+3}$. The smoothing parameter was chosen from $\gamma = 0, 10^{-3}, 10^{-2}, 10^{-1}$. Three-fold cross-validation within training dataset was performed to determine the values of these hyper-parameters. These proposed methods were compared with

Lapin et al. [10]'s and Chu et al. [4]'s methods for learning the unweighted top-k SVM.

In **Table 1**, the top-k accuracies are reported on six benchmarking datasets, ALOI (n = 10,800, d = 128, m = 1,000), Caltech101 (n = 6,339, d = 256, m = 101), CUB (n = 6,033, d = 4,096, m = 200), Indoor67 (n = 15,607, d = 4,096, m = 67), Letter (n = 15,000, d = 16, m = 26), and News20 (n = 15,935, d = 1,024, m = 20). For CUB and Indoor67, feature vectors were extracted by the `fc7` layer in the deep structure VGG16 trained on ImageNet. For each of five methods, the highest among four top-k accuracies corresponding $k = 1, 3, 5, 10$ were reported in Table 1. Differences in the top-k accuracies among three weighting schemes are small, although these weighted top-k SVM achieved higher accuracies in most cases compared to the unweighted top-k SVM trained with the algorithms developed by Refs. [4], [10]. The superiority of top-k accuracies over the previous studies might be caused due to the success of convergence to the optimum. As demonstrated in Ref. [7], the two existing methods always fail to minimize the regularized empirical risk for top-k SVM. This is due to wrongly smaller feasible regions, which shall be analyzed in Ref. [7]. These results empirically suggest that solutions more accurate in optimality are of benefit to better pattern recognition performance.

8. Conclusions

In this study, a weighted extension of top-k SVM and a novel learning algorithm based on the Frank-Wolfe algorithm were developed. The new learning algorithms possess all the favorable properties of SDCA as well as the applicability not only to the original top-k SVM but also to the weighted extension. Geometrical convergence is achieved by smoothing the loss functions. Experimental results suggested that the weighted top-k SVM is a promising option for multi-category classification.

References

[1] Berrada, L., Zisserman, A. and Kumar, M.P.: Smooth Loss Functions for Deep Top-k Classification, *6th International Conference on Learning Representations, ICLR 2018, Conference Track Proceedings* (2018).

[2] Bertsekas, D.: *Nonlinear Programming*, Athena Scientific (1999).

[3] Chang, X., Yu, Y.-L. and Yang, Y.: Robust Top-k Multiclass SVM for Visual Category Recognition, *Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 17*, ACM Press (2017).

[4] Chu, D., Lu, R., Li, J., Yu, X., Zhang, C. and Tao, Q.: Optimizing Top-k Multiclass SVM via Semismooth Newton Algorithm, *IEEE Trans. Neural Networks and Learning Systems*, Vol.29, No.12, pp.6264–6275 (2018).

[5] Frank, M. and Wolfe, P.: An algorithm for quadratic programming, *Naval Research Logistics Quarterly*, Vol.3, No.1-2, pp.95–110 (1956).

[6] Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S.S. and Sundararajan, S.: A dual coordinate descent method for large-scale linear SVM, *Proc. 25th International Conference on Machine Learning*, *ICML '08*, pp.408–415, ACM (2008).

[7] Kato, T. and Hirohashi, Y.: Learning Weighted Top-k Support Vector Machine, *Proc. 10th Asian Conference on Machine Learning*, Lee, W.S. and Suzuki, T. (Eds.), *Proc. Machine Learning Research*, Vol.101, PMLR (2019).

[8] Lacoste-Julien, S. and Jaggi, M.: On the Global Linear Convergence of Frank-Wolfe Optimization Variants, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*, pp.496–504 (2015).

[9] Lacoste-Julien, S., Jaggi, M., Schmidt, M. and Pletscher, P.: Block-Coordinate Frank-Wolfe Optimization for Structural SVMs, *Proc. 30th International Conference on Machine Learning*, Dasgupta, S. and McAllester, D. (Eds.), *Proc. Machine Learning Research*, Vol.28, No.1, pp.53–61, PMLR (2013).

[10] Lapin, M., Hein, M. and Schiele, B.: Top-k Multiclass SVM, *Proc. 28th International Conference on Neural Information Processing Systems - Volume 1*, *NIPS'15*, pp.325–333, MIT Press (2015).

[11] Lapin, M., Hein, M. and Schiele, B.: Loss Functions for Top-k Error: Analysis and Insights, *2016 IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), IEEE (2016).

[12] Nesterov, Y.: *Introductory Lectures on Convex Optimization: A Basic Course*, Springer Publishing Company, Incorporated, 1 edition (2014).

[13] Rennie, J. and Srebro, N.: Loss functions for preference levels: Regression with discrete ordered labels, *Proc. IJCAI Multidisciplinary Workshop on Advances in Preference Handling* (2005).

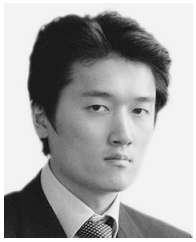
[14] Rockafellar, R.T.: *Convex Analysis*, Princeton University Press (1970).

[15] Shalev-Shwartz, S. and Zhang, T.: Stochastic Dual Coordinate Ascent Methods for Regularized Loss, *J. Mach. Learn. Res.*, Vol.14, No.1, pp.567–599 (2013).

[16] Tan, H.: An Exact Penalty Method for Top-k Multiclass Classification Based on L0 Norm Minimization, *Proc. 2019 11th International Conference on Machine Learning and Computing*, *ICMLC '19*, pp.338–343, ACM (2019).

[17] Yang, F. and Koyejo, S.: On the Consistency of Top-k Surrogate Losses, *CoRR*, Vol.abs/1901.11141 (2019).

Yoshihiro Hirohashi received his B.E. and M.E. degrees from Gunma University, Gunma, Japan in 2013, and 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.



Tsuyoshi Kato received his B.E., M.E., and Ph.D. 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.