Supplementary Material: Learning Weighted Top-k Support Vector Machine

Tsuyoshi Kato KATOTSU@CS.GUNMA-U.AC.JP *Division of Electronics and Informatics, Faculty of Science and Technology, Gunma University, Kiryu, Tenjin-cho 1-5-1, 376-8515, Japan*

Yoshihiro Hirohashi UDPIP.NNCT@GMAIL.COM *DENSO CORPORATION, Tokyo Nihonbashi Tower 15F, Nihonbashi 2-7-1, Chuo-ku, Tokyo, 103-6015, Japan*

Editors: Wee Sun Lee and Taiji Suzuki

This is a longer version of the manuscript submitted to ACML2019, including the main text as well as the appendix which cannot be included in the main manuscript due to limitation of space. The other supplementary file, which is zipped, contains an HTML document using animations to illustrate how the existing method fails to converge to the optimum.

Abstract

Nowadays, the top-k accuracy is a major performance criterion when benchmarking multi-class classifier using datasets with a large number of categories. Top-k multiclass SVM has been designed with the aim to minimize the empirical risk based on the top-k accuracy. There already exist two SDCA-based algorithms to learn the top-k SVM, enjoying several preferable properties for optimization, although both the algorithms suffer from two disadvantages. A weak point is that, since the design of the algorithms are specialized only to the top-k hinge, their applicability to other variants is limited. The other disadvantage is that both the two algorithms cannot attain the optimal solution in most cases due to their theoritical imperfections. In this study, a weighted extension of top-k SVM is considered, and novel learning algorithms based on the Frank-Wolfe algorithm is devised. 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. Numerical simulations demonstrate that only the proposed Frank-Wolfe algorithms can converge to the optimum, in contrast with the failure of the two existing SDCA-based algorithms. Finally, our analytical results for these two studies are presented to shed light on the meaning of the solutions produced from their algorithms.

Keywords: Top-k SVM, Empirical risk minimization, Convex optimization, Frank-Wolfe algorithm, SDCA.

1. Introduction

Lapin et al. (2015) have devised a new loss function, named the *top-k hinge loss*, for multi-category classification. They focus on the recent multi-category classification task in which the number of categories is increasing. *Top-k error* ratio is often used as the performance measure of such classifiers for the task with a large number of categories. The performance measure is supposed to be used for the *top-k outputs* of a classifier. The top-k outputs are k category labels with the k largest prediction scores for a testing example. The top-k

hinge loss is designed to be suitable to the top-k error ratio. Nevertheless, 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 by Lapin et al. (2015). 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 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 optimization often fails without careful manual tuning of parameters for optimization. For example, the number of epochs and a step size are chosen carefully by monitoring the learning curve in training deep neural networks. Meanwhile, the framework of stochastic dual coordinate ascent (SDCA) algorithm (Shalev-Shwartz and Zhang, 2013) 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, meaning that the accuracy of the solution is 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, the rest of the blocks are fixed whereas the chosen block is optimized. Lapin et al. (2015) 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. (2018) 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, meaning that the applicability to variants of top-k hinge is limited. This is one of reasons to develop a new optimization algorithm that can also be applied to a wide class of extensions of the top-k hinge loss function.

Another motivation to devise a new algorithm for top-k SVM is due to another serious limitation of the two existing algorithms. The feasible regions derived in the two studies are narrower than the correct one. Lapin et al. (2015) have discovered a property of the convex conjugate of a particular class of convex functions which they call *compatible*, and attempted to use the property to derive the convex conjugate of the top-k hinge loss function. However, in this study, we have found that the top-k hinge loss does not belong to a class of the compatible functions. From the incorrectness, it turns out that the effective domain (Bertsekas, 1999) used by Lapin et al is just a subset of the true effective domain(See the HTML file in supplementary zip file.).Chu et al. (2018) have developed another optimization algorithm which directly utilizes the dual problem discussed by Shalev-Shwartz and Zhang (2016). In Chu et al. (2018)'s paper, it is asserted that a particular subset of the dual variables can be frozen to zero, although there is no guarantee that, at least, one of the optimal solutions satisfies the added constraints (See Figure 1). Hence, neither of the two algorithms can attain the optimum in cases where the wrongly narrower feasible regions do not intersect with the set of the optimal solutions.

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. (2017) 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.

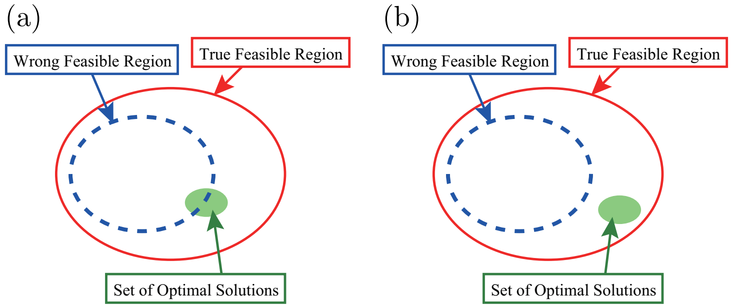


Figure 1: Sketch describing what happens if trying to find an optimal solution over a subset of the feasible region. In both panels, the true feasible region and its subset are depicted with solid and broken ellipses, respectively. (a) An optimal solution can be found if the subset has an intersection with the set of optimal solutions, whereas (b) it is impossible to attain the optimum if no intersection exists. See also the HTML file in the supplementary zip file.

The new optimization algorithm developed in this study is based on the *Frank-Wolfe algorithm* (Frank and Wolfe, 1956) 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, the algorithm can converge geometrically (Lacoste-Julien and Jaggi, 2015). The proposed algorithm can be applied even when smoothing the weighted top-k hinge, which is described in Section 6. Numerical simulations demonstrate that the proposed algorithm successfully converges to the optimum, although the two existing SDCA algorithms (Lapin et al., 2015; Chu et al., 2018) fail due to the aforementioned theoretical faults (Section 7). In Section 8, we shed light on what the theories developed in the two existing studies bring in the world, followed by the last section concluding this paper.

Notation We shall use 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 $s_{\pi(1)} \geq s_{\pi(2)} \geq \cdots \geq s_{\pi(m)}$. Let us define $\boldsymbol{\pi}(\boldsymbol{s}) := [\pi(1\,;\,\boldsymbol{s}), \dots, \pi(m\,;\,\boldsymbol{s})]^\top$ and introduce a notation for a vector with permutated components as $\boldsymbol{s}_{\boldsymbol{\pi}(\boldsymbol{s})} := \left[s_{\pi(1)}, \dots, s_{\pi(m)}\right]^\top$.

We use $\boldsymbol{e}_i$ 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 $\|\cdot\|_{\mathrm{F}}$ to denote the Frobenius norm.

2. Empirical Risk Minimization

The linear multi-class classifier discussed in this paper has a parameter $\boldsymbol{W} := [\boldsymbol{w}_1, \dots, \boldsymbol{w}_m] \in \mathbb{R}^{d\times m}$, 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 $\langle \boldsymbol{w}_1, \boldsymbol{x}\rangle, \dots, \langle \boldsymbol{w}_m, \boldsymbol{x}\rangle$. In order to determine the value of the parameter $\boldsymbol{W}$, suppose that we are given n training examples, $(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_n, y_n) \in \mathbb{R}^d \times [m]$. 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

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

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 (Shalev-Shwartz and Zhang, 2013; Hsieh et al., 2008; Lacoste-Julien et al., 2013; Lapin et al., 2015; Chu et al., 2018). The dual methods attempt to find the maximizer of the *Fenchel dual function* (Bertsekas, 1999) given by

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

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 $\boldsymbol{W}(\boldsymbol{A}) := \frac{1}{\lambda n}\boldsymbol{X}\boldsymbol{A}^\top$ where $\boldsymbol{X} := [\boldsymbol{x}_1, \dots, \boldsymbol{x}_n]$. One of 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 of machine learning literature.). The duality gap vanishes at an optimum for most of 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. $P(\boldsymbol{W}) - \min_{\boldsymbol{W}'\in\mathbb{R}^{d\times m}} P(\boldsymbol{W}') \le \epsilon$, which allows us to decide when to stop the iterations.

3. Unweighted Top-k Hinge

The learning algorithm for top-k SVM developed by Lapin et al. (2015) 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

$$\Phi_{\mathrm{utk}}(\boldsymbol{s}\,;\, y) := \max\left\{0, \frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y\boldsymbol{1}_m)_{\pi(j)}\right\} \tag{3}$$

where $\boldsymbol{W}$ is a matrix of the model parameters. Then, how can we minimize the regularized empirical risk? Lapin et al. (2015) 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. (2015) have developed an algorithm for updating a column and plugged in it to the framework of SDCA.

To express the convex conjugate of the top-k loss function, Lapin et al. (2015) introduce the following convex polytope

$$\Delta(k,r) := \left\{ \boldsymbol{\beta} \in \mathbb{R}_{+}^{m} \mid \langle \boldsymbol{1}, \boldsymbol{\beta} \rangle \leq r,\, \boldsymbol{\beta} \leq \frac{1}{k} \boldsymbol{1}\boldsymbol{1}^{\top} \boldsymbol{\beta} \right\} \tag{4}$$

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

$$\Phi_{\text{utk}}(\boldsymbol{s}\,;\,y) = \max_{\boldsymbol{\beta} \in \Delta(k,1)} \langle \boldsymbol{\beta}, \boldsymbol{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y \boldsymbol{1}_m \rangle \,. \tag{5}$$

From the equation (5), the convex conjugate can be derived as

$$\Phi_{\text{utk}}^{*}(\boldsymbol{v}\,;\,y) = v_y \tag{6}$$

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

$$\langle \boldsymbol{v}, \boldsymbol{1} \rangle = 0, \quad \exists b_y \in \mathbb{R}, \quad \boldsymbol{v} + (b_y - v_y)\boldsymbol{e}_y \in \Delta(k,1); \tag{7}$$

otherwise, $\Phi_{\text{utk}}^{*}(\boldsymbol{v}\,;\,y)$ 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 $\boldsymbol{\rho} := [\rho_1, \dots, \rho_m]^{\top}$ such that $\rho_1 \geq \cdots \geq \rho_m \geq 0$. With these weights, we introduce the following loss function:

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

This function is referred to as the *weighted top-k hinge loss.* This definition is a special case of Chang et al. (2017)'s extensions. They use an upperbound of the loss value, say τ. Their loss function is no more convex unless $\tau = +\infty$.

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*

$$\Phi(\boldsymbol{s}) := \max_{\boldsymbol{\beta} \in \mathcal{B}} \langle \boldsymbol{\beta}, \boldsymbol{\delta} + \boldsymbol{s} - \boldsymbol{1}_m s_y \rangle \,. \tag{9}$$

The convex conjugate of Φ *is then expressed as*

$$\Phi^*(\boldsymbol{v}) = \begin{cases} -\langle \boldsymbol{v}, \boldsymbol{\delta} \rangle & \text{if} \quad \boldsymbol{v} \in \mathit{dom}(\Phi^*), \\ +\infty & \text{otherwise}, \end{cases} \tag{10}$$

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

$$\mathit{dom}(\Phi^*) = \Big\{ \boldsymbol{v} \in \mathbb{R}^m \,\Big|\, \langle \boldsymbol{v}, \mathbf{1} \rangle = 0, \quad \exists \beta_y \in \mathbb{R}, \quad \boldsymbol{v} + (\beta_y - v_y)\boldsymbol{e}_y \in \mathcal{B} \Big\}. \tag{11}$$

See Subsection A.1 for the proof of Lemma 1. In the case of the unweighted top-k hinge loss (3), the convex conjugate (6) and its effective domain (7) 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 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 (9). There exist an index set $\mathcal{K} := \left\{ k_1, \dots, k_{|\mathcal{K}|} \right\} \subseteq [m]$ and a transformed weights $\boldsymbol{\rho}' := \left[\rho'_1, \dots, \rho'_{|\mathcal{K}|} \right]^\top$ such that

$$\Phi_{\text{wtk}}(\boldsymbol{s}\,;\, y) = \max \left\{ 0, \sum_{\ell=1}^{|\mathcal{K}|} \rho'_\ell g_\ell(\boldsymbol{s}) \right\} \tag{12}$$

(See Subsection A.2 for the derivation of (12).)where

$$\forall \ell \in |\mathcal{K}|, \quad g_\ell(\boldsymbol{s}) := \sum_{j=1}^{k_\ell} \left(\mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y \mathbf{1}_m \right)_{\pi(j)}. \tag{13}$$

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

$$\mathcal{B}_{\text{wtk}} := \Big\{ \boldsymbol{\beta} \in \mathbb{R}^m \,\Big|\, \exists \zeta \in \mathbb{R}, \ \forall \ell \in [|\mathcal{K}|], \ \exists \boldsymbol{\lambda}_\ell \in \Delta(k_\ell, \rho'_\ell k_\ell), \quad \zeta = \frac{\langle \mathbf{1}, \boldsymbol{\lambda}_\ell \rangle}{k_\ell \rho'_\ell}, \quad \boldsymbol{\beta} = \boldsymbol{\lambda}_1 + \dots + \boldsymbol{\lambda}_{|\mathcal{K}|} \Big\}, \tag{14}$$

the loss function can be re-written as

$$\Phi_{\text{wtk}}(\boldsymbol{s}\,;\, y) = \max_{\boldsymbol{\beta} \in \mathcal{B}_{\text{wtk}}} \langle \boldsymbol{\beta}, \mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y \mathbf{1}_m \rangle. \tag{15}$$

See Subsection A.3 for the derivation of (15). 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*

$$\Phi^*_{wtk}(\boldsymbol{v}\,;\,y) = \begin{cases} v_y & \textit{if } \langle \boldsymbol{v}, \boldsymbol{1}\rangle = 0, \quad \exists b_y \in \mathbb{R}, \quad \boldsymbol{v} + (b_y - v_y)\boldsymbol{e}_y \in \mathcal{B}_{wtk}, \\ +\infty & \textit{o.w.} \end{cases} \tag{16}$$

Our goal is 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. (2015) and Chu et al. (2018) 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 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 (Frank and Wolfe, 1956) 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

$$\mathrm{dom}(-D) = (-\mathrm{dom}\Phi^*(\cdot; y_1)) \times \cdots \times (-\mathrm{dom}\Phi^*(\cdot; y_n)). \tag{17}$$

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

$$\left\langle \nabla D(\boldsymbol{A}^{(t-1)}), \boldsymbol{U} - \boldsymbol{A}^{(t-1)} \right\rangle + D(\boldsymbol{A}^{(t-1)}) \tag{18}$$

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

$$\gamma^{(t-1)} := \operatorname*{argmax}_{\gamma\in[0,1]} D\left(\boldsymbol{A}^{(t-1)} + \gamma\Delta\boldsymbol{A}^{(t-1)}\right). \tag{19}$$

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

$$\begin{aligned} \hat{\gamma}^{(t-1)} &:= \frac{\lambda n \left\langle \Delta \boldsymbol{A}^{(t-1)}, \boldsymbol{E}_{\boldsymbol{y}} - \boldsymbol{Z}(\boldsymbol{A}^{(t-1)}) \right\rangle}{\left\langle \Delta \boldsymbol{A}^{(t-1)} \boldsymbol{K}, \Delta \boldsymbol{A}^{(t-1)} \right\rangle}, \qquad \boldsymbol{K} := \boldsymbol{X}^\top \boldsymbol{X}, \\ \text{and} \qquad \boldsymbol{Z}(\boldsymbol{A}) &:= \frac{1}{\lambda n} \boldsymbol{K} \boldsymbol{A}^\top, \quad \boldsymbol{E}_{\boldsymbol{y}} := \left[\boldsymbol{e}_{y_1}, \ldots, \boldsymbol{e}_{y_n} \right]. \end{aligned} \tag{20}$$

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 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 $i = 1, \ldots, n$,

$$\max \quad \left\langle \frac{\partial D(\boldsymbol{A}^{(t-1)})}{\partial \boldsymbol{\alpha}_i}, \boldsymbol{u}_i \right\rangle \qquad \text{wrt} \quad \boldsymbol{u}_i \in -\text{dom}(\Phi^*_{\text{wtk}}(\cdot; y_i)). \tag{21}$$

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 $\boldsymbol{U}^{(t-1)} := \left[\boldsymbol{u}_1^{(t-1)}, \ldots, \boldsymbol{u}_n^{(t-1)} \right]$. The gradient with respect to the i-th column in $\boldsymbol{A}$ is expressed as

$$\frac{\partial D(\boldsymbol{A})}{\partial \boldsymbol{\alpha}_i} = \frac{1}{n} (\boldsymbol{z}_i(\boldsymbol{A}) - \boldsymbol{e}_{y_i}) \tag{22}$$

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 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 $\boldsymbol{\alpha} \in -\text{dom}\phi_*$, where $\boldsymbol{f} \in \mathbb{R}^m$ is a constant vector. Then, it holds that*

$$\forall \eta \in \mathbb{R}_{++}, \qquad \underset{\boldsymbol{\alpha} \in -\text{dom}(\phi_*)}{\text{argmax}} \ \langle \boldsymbol{g}, \boldsymbol{\alpha} \rangle = -\partial \phi(\boldsymbol{f} - \eta \boldsymbol{g}) \tag{23}$$

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

See Subsection A.4for the proof of Lemma 3. 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 (21) can be expressed in a closed form as

$$\boldsymbol{u}_i^{(t-1)} := -\nabla\Phi_{\text{wtk}}(\boldsymbol{z}_i(\boldsymbol{A}^{(t-1)})) \tag{24}$$

where $\nabla\Phi_{\text{wtk}}(\boldsymbol{z}_i(\boldsymbol{A}^{(t-1)}))$ is a sub-gradient of the weighted top-k hinge at $\boldsymbol{z}_i(\boldsymbol{A}^{(t-1)})$. Theories are established even if any of sub-gradient in the sub-differential is taken. On computing $\boldsymbol{Z}(\boldsymbol{A}^{(t-1)})$, it takes $O(m\log m)$ time to compute $\boldsymbol{u}_i^{(t-1)}$. These results can be summarized in the following theorem.

Theorem 4 *Consider the Frank-Wolfe algorithm for maximizing $D(\boldsymbol{A})$ with $\Phi^*(\cdot;y_i) = \Phi^*_{wtk}(\cdot;y_i)$ for $i=1,\dots,n$. 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 (2015) have proved the *global linear convergence* for the standard Frank-Wolfe algorithm and these variants. When employing the standard Frank-Wolfe algorithm, the upper bound of the objective gap $\text{dgap}(\boldsymbol{A}) := \min_{\boldsymbol{W}} P(\boldsymbol{W}) - \boldsymbol{D}(\boldsymbol{A})$ is guaranteed to geometrically decrease as $\text{dgap}(\boldsymbol{A}^{(t)}) \le \exp(-\zeta t)$ where ζ is a constant dependent on the optimization problem (Lacoste-Julien and Jaggi, 2015). Their theories are based on an assumption that the objective function must be *smooth* and *strongly convex* (Nesterov, 2014), 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* (Rockafellar, 1970) 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, (3) and (8), suffer from the discontinuity in the derivatives. Several studies (Rennie and Srebro, 2005; Shalev-Shwartz and Zhang, 2013; Lapin et al., 2016) have considered smoothing loss functions to obtain a better property for optimization. Following Lapin et al. (2016), 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

$$\Phi_{\text{stk}}(\boldsymbol{s}\,;\,y) := \min_{\boldsymbol{z}\in\mathbb{R}^m}\left(\Phi_{\text{wtk}}(\boldsymbol{z}\,;\,y) + \frac{1}{2\gamma}\|\boldsymbol{s}-\boldsymbol{z}\|^2\right) \tag{25}$$

where $\gamma > 0$ is a smoothing constant. Here we discuss how to find $\boldsymbol{W}\in\mathbb{R}^{d\times n}$ that minimizes the regularized empirical risk based on the smoothed loss, denoted by $P_{\text{stk}} : \mathbb{R}^{d\times m} \to \mathbb{R}$, which is given in (1) with $\Phi(\cdot;y_i) = \Phi_{\text{stk}}(\cdot\,;\,y_i)$ for $i=1,\dots,n$. To use dual methods for learning with this smoothed loss function, the dual objective, denoted by $D_{\text{stk}} : \mathbb{R}^{m\times n} \to -\bar{\mathbb{R}}$, 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 $-D_{\text{stk}}$ 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, 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 $i = 1, \dots, n$. Then, the optimization problem for maximizing $D_{stk}(\boldsymbol{A})$ is not only dual to the minimization problem with the primal objective $P_{stk} : \mathbb{R}^{d\times m} \to \mathbb{R}$ but also dual to the minimization problem with the objective function $\tilde{P}_{wtk} : \mathbb{R}^{(d+n)\times m} \to \mathbb{R}$ defind as*

$$\tilde{P}_{wtk}(\tilde{\boldsymbol{W}}) := \frac{\lambda}{2}\left\|\tilde{\boldsymbol{W}}\right\|_F^2 + \frac{1}{n}\sum_{i=1}^{n}\Phi_{wtk}(\tilde{\boldsymbol{W}}^\top\tilde{\boldsymbol{x}}_i\,;\,y_i). \tag{26}$$

See Subsection A.5 for the proof of Proposition 5. 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:

$$\begin{aligned}\text{Gap}_{\text{stk}}(\boldsymbol{A}) &:= \tilde{P}_{\text{wtk}}(\tilde{\boldsymbol{W}}(\boldsymbol{A})) - D_{\text{stk}}(\boldsymbol{A}) \\ &= \frac{\gamma}{n}\|\boldsymbol{A}\|_{\text{F}}^2 + \frac{1}{\lambda n^2}\left\langle \boldsymbol{A}\boldsymbol{K} - \lambda n\boldsymbol{E}_{\boldsymbol{y}}, \boldsymbol{A}\right\rangle + \frac{1}{n}\sum_{i=1}^{n}\Phi_{\text{wtk}}(\boldsymbol{z}_i(\boldsymbol{A}) + \gamma\boldsymbol{\alpha}_i).\end{aligned} \tag{27}$$

The above observations suggest that neither the $(d+n)$-dimensional vectors $\tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_n$ nor the model parameters $\tilde{\boldsymbol{W}} \in \mathbb{R}^{(d+n)\times m}$ do not have to be unfolded in the computational memory to implement the Frank-Wolfe algorithm for minimizing $\tilde{P}_{\text{utk}}(\tilde{\boldsymbol{W}})$ 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 will be available at https://github.com/hirohashi/wtopk

7.1. Convergence Behavior for Non-Smooth Unweighted Top-k SVM

The proposed Frank-Wolfe algorithms were compared with three existing SDCA-based algorithms for learning the unweighted top-k SVM. Two of the three existing algorithms, denoted by `Chu I` and `Chu II`, were proposed by Chu et al. (2018). `Chu I` always uses a Newton method for SDCA update, whereas `Chu II` switches the SDCA update method from the Newton method to the variable fixing method (Kiwiel, 2007) under some condition. The remaining one, denoted by `Lapin`, was Lapin et al. (2015)'s SDCA algorithm. Implementations published in the authors' GitHub repositories [1] were utilized to run the three existing SDCA algorithms. In their codes, different loss functions, which shall be shown in (31), were implemented. In our experiments, the corresponding code was replaced to the correct one for comparison.

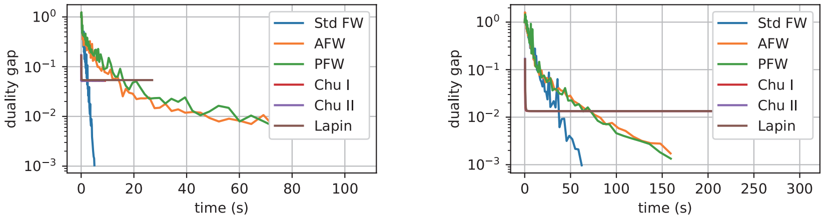
Panels (a) and (b) in Figure 2 show the duality gap $P(\boldsymbol{W}(\boldsymbol{A})) - D(\boldsymbol{A})$ against the CPU times on two datasets, FMD and News20. Each algorithm was terminated at 1,000th epoch. FMD contains $n = 1,000$ training examples divided into $m = 10$ categories and each feature vector $\boldsymbol{x}_i$ is $d = 4,096$ dimensional; for News20, $n = 15,935$, $d = 1,024$, and $m = 20$. In this experiment, $k = 3$ and $\lambda = 1/n$. On the two datasets, the standard Frank-Wolfe algorithm, denoted by *Std FW* attained the duality gap of 10^{-3} for the shortest times compared to the two variants, AFW and PFW. The running times were 5.23 and 62.17 seconds, respectively, to make the duality gap below 10^{-3} on FMD and News20, and those were 53.42 and 1102.35 seconds to get 10^{-5}-accurate solutions. PFW took 172.56 seconds for News20 to obtain 10^{-3}-accurate solutions, but AFW could not attain 10^{-3}-accurate solutions within 1,000 iterations.

The three algorithms, `Chu I`, `Chu II`, and `Lapin,` converge to the almost same value. Therefore, the curves of `Chu I` and `Chu II` look overlapped with `Lapin`'s. The three existing SDCA methods could reduce the duality gap to $5.12 \cdot 10^{-2}$ and $1.34 \cdot 10^{-2}$ quickly on FMD and News20 (0.44 and 156.99 seconds, at minimum), respectively. However, the duality gaps could not be decreased further, and eventually remain over 10^{-2} at 1,000th epoch. If comparing the values of the primal objective, the regularized empirical risk, at 1,000th epoch, the differences from that of Std FW, $P(\boldsymbol{W}(\boldsymbol{A}^{(1000)})) - P(\boldsymbol{W}(\boldsymbol{A}_{\text{FW}}^{(1000)}))$, — where $\boldsymbol{A}_{\text{FW}}^{(1000)}$ was the solution generated with Std FW at 1,000th epoch — were seriously large (Minimums among three SDCA were $4.51 \cdot 10^{-2}$ and $4.17 \cdot 10^{-3}$ for the two datasets, respectively), indicating that any of existing SDCA algorithms could not reach accurate solutions for the two datasets. In the next section, what prevents the existing methods from converging to the optimum shall be analyzed.

7.2. How Does Smoothing Affect Convergence?

We next 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 3 plots the duality gaps against the number of iterations. The duality gaps produced with Std FW, AFW, and PFW, respectively, are shown in Figure 3(a),(b),(c). The dataset used here is News20. When using

[1] `https://github.com/djchu/topkmsvm` with `8d17418` and `https://github.com/mlapin/libsdca` with `fd5c1f1`



(a) FMD

(b) News20

Figure 2: Convergence behavior for the unweighted top-k hinge loss. In (a) and (b), the convergences of the proposed Frank-Wolfe algorithms, Std FW, AFW, and PFW, are compared to those of the three SDCA-based algorithms, Chu I, Chu II, and Lapin for learning the unweighted top-k SVM with two datasets, FMD and News20. The horizontal and vertical axes, respectively, indicate the CPU time and the duality gap which vanishes at the optimum. The classical Frank-Wolfe algorithm quickly converged to the optimum on both the datasets, in contradistinction to the failures of the existing SDCA-based algorithms.

$\gamma = 10^{-1}$, 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 phenomena discussed in Lacoste-Julien and Jaggi (2015). Meanwhile, such 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 was did not differ largely from the ones of non-smooth loss.

7.3. 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 (8). 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 $\rho_j^{\text{flat}} = \frac{1}{k}$, $\rho_j^{\text{linear}} = 2(k+1-j)/((k+1)k)$, $\rho_j^{\text{exp}} = \exp(-j/k)/\left(\sum_{j'=1}^{k} \exp(-j'/k)\right)$ for $j \leq k$, and the remaining weights were zero. The `flat` recover 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 Figure 4. No significant differences amongst the three weight types were observed despite the effective domains complicated by weighting.

7.4. Pattern Recognition Performance

Finally, the pattern recognition performances of the proposed learning methods were investigated. We used the top-k accuracy for the performance measure for multi-class 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, denoted by `UTk (ours)`, `WTk (linear)`, and `WTk (exp)`, where `UTk (ours)` was equivalent to the unweighted top-k SVM. These three multi-class SVMs were trained with the standard Frank-Wolfe algorithms presented in Section 5. Each algorithm was terminated when the difference between the primal and dual objective values reached 10^{-3}. The regularization parameter was chosen by $\lambda = 1/nC$ where $C = 10^{-3}, 10^{-2}, \dots, 10^{+3}$. 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. (2015)'s and Chu et al. (2018)'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. Our methods achieved the highest accuracies except Letter. The differences in top-k accuracies might cause due to the success of convergence to the optimum. As demonstrated in Subsection 7.1, 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 the next section. These results empirically suggest that solutions more accurate in optimality are of benefit to better pattern recognition performance.

8. Discussions

In this section, we discuss why the existing methods fail to converge to the optimum. The reason is due to their defective theories. The dual objective functions derived by Lapin et al. (2015) and Chu et al. (2018) are correct, although their feasible regions are smaller than the true ones. The results reported in Subsection 7.1, in which large duality gaps have remained, suggest that the set of optimal solutions are out of the wrongly derived feasible regions. It may not be simple to modify their algorithms with the correct feasible regions because their theories are founded on the wrongly smaller feasible regions. In what follows, we shall elucidate what the solutions derived from their theories mean.

Lapin et al. (2015) have introduced a new concept named *y-compatible* in Definition 2 of their paper for $y \in [m]$. For simplicity, we here assume that $y = m$. Lapin et al. (2015) have defined a convex function $\phi(\cdot\,; y) : \mathbb{R}^m \to \mathbb{R}$ to be y-compatible if

$$\forall \boldsymbol{v}^{\setminus y} \in \mathbb{R}^{m-1}, \quad \sup_{\boldsymbol{s}^{\setminus y} \in \mathbb{R}^{m-1}} \left(\left\langle \boldsymbol{v}^{\setminus y}, \boldsymbol{s}^{\setminus y} \right\rangle - \phi\left(\left[(\boldsymbol{s}^{\setminus y})^\top, 0 \right]^\top ; y \right) \right) = \phi^*\left(\left[(\boldsymbol{v}^{\setminus y})^\top, 0 \right]^\top ; y \right). \tag{28}$$

where we have used the notation $\boldsymbol{x}^{\setminus y} \in \mathbb{R}^{m-1}$ to denote the $(m-1)$-dimensional vector generated by excluding the y-th entry from a vector $\boldsymbol{x} \in \mathbb{R}^m$. In Proposition 3 in Lapin et al. (2015)'s paper, it is stated that the function

$$\phi_{\text{utk}}(\boldsymbol{s}\,; y) := \max \left\{ 0, \frac{1}{k} \sum_{j=1}^{k} (\mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s})_{\pi(j)} \right\} \tag{29}$$

is y-compatible, and the convex conjugate of the unweighted top-k hinge loss has been derived with dependence on their Proposition 3. However, in this study, we have found a result that contradicts with Lapin et al. (2015)'s Proposition 3.

Proposition 6 *The function $\phi_{utk}(\cdot\,;\,y) : \mathbb{R}^m \to \mathbb{R}$ defined in (29) is not y-compatible for $2 \leq k < m$.*

See Subsection A.6for the proof of Proposition 6. Their imperfection leads to an incorrect convex conjugate function $\Phi^*_{\text{ptk}}(\boldsymbol{v}\,;\,y) = v_y$ with the effective domain

$$\text{dom}\Phi^*_{\text{ptk}}(\cdot \mid y) = \{\boldsymbol{v} \in \mathbb{R}^m \mid \langle \boldsymbol{v}, \boldsymbol{1} \rangle = 0, \quad \boldsymbol{v} - v_y \boldsymbol{e}_y \in \Delta(k,1)\} \tag{30}$$

which is a subset of the correct one, $\text{dom}\Phi^*_{\text{utk}}(\cdot\,;\,y)$, when $2 \leq k < m$. In fact, the above function $\Phi^*_{\text{ptk}}(\cdot\,;\,y)$ is the convex conjugate of the following function (See Subsection A.7 for the derivation):

$$\Phi_{\text{ptk}}\left(\boldsymbol{s}\,;\,y\right) = \max\left\{0, \frac{1}{k}\sum_{j=1}^{k}\left((1-s_y)\boldsymbol{1}_{m-1} + \boldsymbol{s}^{\setminus y}\right)_{\pi(j)}\right\} \tag{31}$$

which is no more the unweighted top-k hinge loss function. We refer to $\Phi_{\text{ptk}}\left(\cdot\,;\,y\right)$ as the *pseudo top-k hinge loss* below. Let us denote by P_{ptk} and P_{utk}, the regularized empirical risk (1) with $\Phi(\cdot; y_i) = \Phi_{\text{ptk}}(\cdot; y_i)$ and $\Phi(\cdot; y_i) = \Phi_{\text{utk}}(\cdot; y_i)$ for $i = 1, \ldots, n$, respectively. It can be observed that, $\forall y \in [m]$, $\forall \boldsymbol{s} \in \mathbb{R}^m$, $\Phi_{\text{ptk}}\left(\boldsymbol{s}\,;\,y\right) \leq \Phi_{\text{utk}}\left(\boldsymbol{s}\,;\,y\right)$, implying that $\forall \boldsymbol{W} \in \mathbb{R}^{d\times m}$, $P_{\text{ptk}}(\boldsymbol{W}) \leq P_{\text{utk}}(\boldsymbol{W})$. Therefore, the duality gap derived from the true top-k hinge loss can remain positive even when the duality gap from the pseudo top-k hinge loss vanishes.

Chu et al. (2018) have employed another formulation for the Fenchel dual function of the regularized empirical risk, which is given by

$$d_{\text{utk}}(\boldsymbol{A}) := -\frac{1}{2\lambda n^2}\left\|\sum_{i=1}^{n}\left(\boldsymbol{H}_{y_i}^\top \otimes \boldsymbol{x}_i\right)\boldsymbol{\alpha}_i\right\|^2 - \frac{1}{n}\sum_{i=1}^{n}\phi^*_{\text{utk}}(-\boldsymbol{\alpha}_i\,;\,y_i), \tag{32}$$

where $\boldsymbol{H}_y := \boldsymbol{I} - \boldsymbol{1}\boldsymbol{e}_y^\top$ and the operator $\otimes$ denotes the Kronecker product. This formulation is similar to the one presented in Shalev-Shwartz and Zhang (2016). In maximizing $d_{\text{utk}}(\boldsymbol{A})$, no feasibility condition but $\boldsymbol{A} \in \text{dom}(-d_{\text{utk}})$ must be given to the dual variable $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. Nonetheless, Chu et al. (2018) insist that the (y_i, i)-th entries in $\boldsymbol{A}$, say $\alpha_{y_i,i}$, for $i \in [n]$ can be fixed to zero, and their algorithm has been developed on the basis of this fixation. Their constraints inevitably make the feasible region narrower than the true one. Fixing these n entries to zero would not be harmful only when the set of the optimal solutions contained a matrix with $\alpha_{y_i,i} = 0$ for $\forall i \in [n]$, although unfortunately the empirical results in this study suggest that such a case is very rare. The observation for their failure of the convergence has prompted us to analyze their theories, which has brought the following proposition.

Proposition 7 *The maximization problem with objective $d_{utk}(\boldsymbol{A})$ subject to a constraint $\boldsymbol{E_y} \odot \boldsymbol{A} = \boldsymbol{O}$, where $\odot$ is the operator of the entrywise product, is dual to the ERM problem for minimizing $P_{ptk}(\boldsymbol{W})$.*

See Subsection A.9 for the proof of Proposition 7. From the above discussions, this study has unraveled a new fact that the two existing theories developed by Lapin et al. (2015) and Chu et al. (2018) are not for learning the unweighted top-k SVM, but for ERM with the pseudo top-k hinge (31).

9. Conclusions

In this paper, a novel approach to answering the question of how to solve the dual problem for learning the top-k multiclass SVM was presented. Due to the theoretical incompleteness in the previous studies (Lapin et al., 2015; Chu et al., 2018) tackling the same question, this study turned out the first to provide a correct answer. The experimental results demonstrated that the proposed algorithms work well even if the loss functions are weighted and smoothed.

Besides the proposed Frank-Wolfe algorithm, there remain substantial choices for learning the weighted top-k SVM. One might employ the stochastic sub-gradient method, in which a sub-linear convergence is guaranteed (e.g. Shalev-Shwartz et al. (2011)). Several variants such as SAG (Roux et al., 2012) and SVRG (Johnson and Zhang, 2013) converge geometrically when smoothed loss is employed. Disadvantage of these approaches is lack of clear stopping criterion. Another choice for learning the weighted top-k SVM may be the block coordinate Frank-Wolfe (BCFW) algorithm (Lacoste-Julien et al., 2013; Osokin et al., 2016). Interestingly, it can be shown that BCFW applied to the dual problem for the weighted top-k SVM is exactly same as ProxSDCA Option II (Shalev-Shwartz and Zhang, 2016) which updates the solution in the same direction, and performs the exact line search in a closed form. Numerical comparison with these methods is left to future work.

References

D.P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 1999.

Xiaojun Chang, Yao-Liang Yu, and Yi Yang. Robust top-k multiclass SVM for visual category recognition. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 17*. ACM Press, 2017.

Dejun Chu, Rui Lu, Jin Li, Xintong Yu, Changshui Zhang, and Qing Tao. Optimizing top-k multiclass SVM via semismooth newton algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):6264–6275, December 2018.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2): 95–110, March 1956. doi:10.1002/nav.3800030109.

Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 408–415, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4. doi:10.1145/1390156.1390208.

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.

K. C. Kiwiel. Variable fixing algorithms for the continuous quadratic knapsack problem. *Journal of Optimization Theory and Applications*, 136(3):445–458, November 2007. doi:10.1007/s10957-007-9317-7.

Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 496–504, 2015.

Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, and Patrick Pletscher. Block-coordinate Frank-Wolfe optimization for structural SVMs. 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 53–61, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Top-k multiclass svm. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 325–333, Cambridge, MA, USA, 2015. MIT Press.

Maksim Lapin, Matthias Hein, and Bernt Schiele. Loss functions for top-k error: Analysis and insights. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. doi: 10.1109/cvpr.2016.163.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014. ISBN 9781461346913.

Anton Osokin, Jean-Baptiste Alayrac, Isabella Lukasewitz, Puneet Dokania, and Simon Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 593–602, New York, New York, USA, 20–22 Jun 2016. PMLR.

Jason Rennie and Nathan Srebro. Loss functions for preference levels: Regression with discrete ordered labels. *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling*, 01 2005.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

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.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *J. Mach. Learn. Res.*, 14(1):567–599, February 2013. ISSN 1532-4435.

Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1):105–145, 2016. [pdf].

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: primal estimated sub-gradient solver for SVM. *Math. Program.*, 127(1):3–30, 2011.

Appendix A. Proofs and Derivations

A.1. Proof for Lemma 1

Without loss of generality, we assume $y = m$ in this proof. We can then write $\boldsymbol{x} = \left[(\boldsymbol{x}^{\setminus y})^\top, x_y\right]^\top$. Fixing $\boldsymbol{v} \in \mathbb{R}^m$ in Lemma 1, let us define a function $J_1 : \mathbb{R}^m \to \bar{\mathbb{R}}$ as

$$J_1(\boldsymbol{\beta}) := \max_{\boldsymbol{s}^{\setminus y} \in \mathbb{R}^{m-1}} \left\langle \boldsymbol{s}^{\setminus y}, \boldsymbol{v}^{\setminus y} - \boldsymbol{\beta}^{\setminus y} \right\rangle + \max_{s_y \in \mathbb{R}} \left(v_y + \left\langle \boldsymbol{\beta}^{\setminus y}, \mathbf{1} \right\rangle \right) s_y, \tag{33}$$

to express the convex conjugate of Φ as

$$\begin{aligned}
\Phi^\star(\boldsymbol{v}) &= \max_{\boldsymbol{s} \in \mathbb{R}^m} \left(\langle \boldsymbol{s}, \boldsymbol{v} \rangle - \Phi(\boldsymbol{s}) \right) \\
&= \max_{\boldsymbol{s} \in \mathbb{R}^m} \left(\langle \boldsymbol{s}, \boldsymbol{v} \rangle - \max_{\boldsymbol{\beta} \in \mathcal{B}} \langle \boldsymbol{\beta}, \boldsymbol{\delta} + \boldsymbol{s} - s_y \mathbf{1} \rangle \right) \\
&= \min_{\boldsymbol{\beta} \in \mathcal{B}} \left(- \langle \boldsymbol{\beta}, \boldsymbol{\delta} \rangle + \max_{\boldsymbol{s} \in \mathbb{R}^m} \left(\langle \boldsymbol{s}, \boldsymbol{v} - \boldsymbol{\beta} \rangle + s_y \langle \boldsymbol{\beta}, \mathbf{1} \rangle \right) \right) \qquad (34) \\
&= \min_{\boldsymbol{\beta} \in \mathcal{B}} \left(- \langle \boldsymbol{\beta}, \boldsymbol{\delta} \rangle + \max_{\boldsymbol{s}^{\setminus m} \in \mathbb{R}^m} \left\langle \boldsymbol{s}^{\setminus y}, \boldsymbol{v}^{\setminus y} - \boldsymbol{\beta}^{\setminus y} \right\rangle + \max_{s_y \in \mathbb{R}} \left(v_y + \left\langle \boldsymbol{\beta}^{\setminus y}, \mathbf{1} \right\rangle \right) s_y \right) \\
&= \min_{\boldsymbol{\beta} \in \mathcal{B}} \left(J_1(\boldsymbol{\beta}) - \langle \boldsymbol{\beta}, \boldsymbol{\delta} \rangle \right).
\end{aligned}$$

In case that $\boldsymbol{\beta}^{\setminus y} \neq \boldsymbol{v}^{\setminus y}$, then $J_1(\boldsymbol{\beta}) = +\infty$; otherwise, i.e. in case of $\boldsymbol{\beta}^{\setminus y} = \boldsymbol{v}^{\setminus y}$,

$$v_y + \left\langle \boldsymbol{\beta}^{\setminus y}, \mathbf{1} \right\rangle = v_y + \left\langle \boldsymbol{v}^{\setminus y}, \mathbf{1} \right\rangle = \langle \boldsymbol{v}, \mathbf{1} \rangle \tag{35}$$

implying that $J_1(\boldsymbol{\beta})$ goes to infinity if $\langle \boldsymbol{v}, \mathbf{1} \rangle \neq 0$. Combining the two cases yields

$$J_1(\boldsymbol{\beta}) = \begin{cases} 0 & \text{if } \langle \boldsymbol{v}, \mathbf{1} \rangle = 0 \text{ and } \boldsymbol{\beta}^{\setminus m} = \boldsymbol{v}^{\setminus m}, \\ +\infty & \text{o.w.} \end{cases} \tag{36}$$

We now look back at $\Phi^*(\boldsymbol{v})$. Provided that $\boldsymbol{v}$ satisfies $\langle \boldsymbol{v}, \mathbf{1} \rangle = 0$ and there exists $\boldsymbol{\beta} \in \mathcal{B}$ such that $\boldsymbol{\beta}^{\setminus y} = \boldsymbol{v}^{\setminus y}$, we have

$$\Phi^*(\boldsymbol{v}) = \min_{\boldsymbol{\beta} \in \mathcal{B}} \left(J_1(\boldsymbol{\beta}) - \langle \boldsymbol{\beta}, \boldsymbol{\delta} \rangle \right) = - \langle \boldsymbol{v}, \boldsymbol{\delta} \rangle ; \tag{37}$$

otherwise, $\Phi^*(\boldsymbol{v}) = +\infty$. Therein, the last equality in (37) follows from the assumption that $\delta_y = 0$. Hence, we have

$$\Phi^*(\boldsymbol{v}) := \begin{cases} - \langle \boldsymbol{v}, \boldsymbol{\delta} \rangle & \text{if } \langle \boldsymbol{v}, \mathbf{1} \rangle = 0 \text{ and } \exists \beta_y \text{ s.t. } \begin{bmatrix} \boldsymbol{v}^{\setminus y} \\ \beta_y \end{bmatrix} \in \mathcal{B} \\ +\infty & \text{o.w.} \end{cases} \tag{38}$$

concluding the proof.

A.2. Derivation of (12)

Recall that $\{\rho_k\}_k$ is non-negative and (non-strictly) monotonically descreasing. Letting $\rho_{m+1} = 0$, the set $\mathcal{K}$ can be found by

$$\mathcal{K} := \{k \mid \rho_k > \rho_{k+1}\} \tag{39}$$

and we can write the entries in $\mathcal{K}$ as $\mathcal{K} = \{k_1, \ldots, k_{|\mathcal{K}|}\}$ with $k_1 \geq \cdots \geq k_{|\mathcal{K}|}$. Let

$$\rho'_\ell := \rho_{k_\ell} - \rho_{k_\ell+1} \tag{40}$$

for $\ell \in [|\mathcal{K}|]$ to have

$$\forall \boldsymbol{x} \in \mathbb{R}^m, \qquad \sum_{j=1}^{m} \rho_j x_{\pi(j)} = \sum_{\ell=1}^{|\mathcal{K}|} \rho'_\ell \sum_{j=1}^{k_\ell} x_{\pi(j)}. \tag{41}$$

Setting $\boldsymbol{x} := \mathbf{1} - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}$, we get

$$\sum_{j=1}^{m} \left(\mathbf{1}_m - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}_m\right)_{\pi(j)} \rho_j = \sum_{\ell=1}^{|\mathcal{K}|} \rho'_\ell g_\ell(\boldsymbol{s}) \tag{42}$$

which immediately yields the equality (12).

A.3. Derivation of (15)

Following Lapin et al. (2015)'s paper, we use the following lemma:

Lemma 8 (Lemma 1 in Lapin et al. (2015))

$$\left\langle \mathbf{1}, \boldsymbol{x}_{\boldsymbol{\pi}(1:k)} \right\rangle = \min_{t \in \mathbb{R}} \left(kt + \langle \mathbf{1}, \max(\mathbf{0}, \boldsymbol{x} - t\mathbf{1})\rangle\right). \tag{43}$$

Let $\boldsymbol{x} := \mathbf{1} - \boldsymbol{e}_y + \boldsymbol{s} - s_y\mathbf{1}$ and denote its subvector containing the largest k entries by

$$\boldsymbol{x}_{\pi(1:k)} := \left[x_{\pi(1)}, \ldots, x_{\pi(k)}\right]^\top. \tag{44}$$

It then suffices to show that

$$\max\left\{0, \sum_{\ell=1}^{|\mathcal{K}|} \rho'_\ell \left\langle \mathbf{1}, \boldsymbol{x}_{\pi(1:k_\ell)} \right\rangle\right\} = \max_{\boldsymbol{\beta} \in \mathcal{B}_{\text{wtk}}} \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle. \tag{45}$$

We define $\boldsymbol{k} := \left[k_1, \dots, k_{|\mathcal{K}|}\right]^\top$ and $\boldsymbol{\rho}' := \left[\rho'_1, \dots, \rho'_{|\mathcal{K}|}\right]^\top$, to rearrange the left hand side of (45) as

$$\begin{aligned}\text{LHS of (45)} &= \min_{s \in \mathbb{R}_+} \left\{ s \,\middle|\, s \geq \sum_{\ell=1}^{|\mathcal{K}|} \rho'_\ell \left\langle \mathbf{1}, \boldsymbol{x}_{\pi(1:k_\ell)} \right\rangle \right\} \\ &= \min_{s \in \mathbb{R}_+, \boldsymbol{t} \in \mathbb{R}^{|\mathcal{K}|}} \left\{ s \,\middle|\, s \geq \sum_{\ell=1}^{|\mathcal{K}|} \left(k_\ell t_\ell + \langle \mathbf{1}, \max(\mathbf{0}, \boldsymbol{x} - t_\ell \mathbf{1}) \rangle \right) \rho'_\ell \right\} \\ &= \min_{s_+ \in \mathbb{R}, \boldsymbol{t} \in \mathbb{R}^{|\mathcal{K}|}, \boldsymbol{\Xi} \in \mathbb{R}_+^{m \times |\mathcal{K}|}} \left\{ s \,\middle|\, s \geq \left\langle \boldsymbol{k} \odot \boldsymbol{t} + \boldsymbol{\Xi}^\top \mathbf{1}, \boldsymbol{\rho}' \right\rangle,\ \boldsymbol{\Xi} \geq \boldsymbol{x} \mathbf{1}^\top - \mathbf{1} \boldsymbol{t}^\top \right\} \\ &= \max_{\zeta, \eta \in \mathbb{R}_+,\, \boldsymbol{\Lambda}, \boldsymbol{M} \in \mathbb{R}_+^{m \times |\mathcal{K}|}} \left(\min_{s \in \mathbb{R}_+, \boldsymbol{t} \in \mathbb{R}^{|\mathcal{K}|}, \boldsymbol{\Xi} \in \mathbb{R}_+^{m \times |\mathcal{K}|}} L(s, \boldsymbol{t}, \boldsymbol{\Xi}, \zeta, \eta, \boldsymbol{\Lambda}, \boldsymbol{M}) \right)\end{aligned} \tag{46}$$

where $L(\cdot, \cdot, \cdot, \cdot, \cdot, \cdot, \cdot)$ is the Lagrangian function for the linear program with the primal variables, $s \in \mathbb{R}_+$, $\boldsymbol{t} \in \mathbb{R}^{|\mathcal{K}|}$, and $\boldsymbol{\Xi} = \left[\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{|\mathcal{K}|}\right] \in \mathbb{R}_+^{m \times |\mathcal{K}|}$, and the Lagrangian multipliers, $\zeta, \eta \in \mathbb{R}_+$, $\boldsymbol{\Lambda} := \left[\boldsymbol{\lambda}_1, \dots, \boldsymbol{\lambda}_{|\mathcal{K}|}\right]$, $\boldsymbol{M} := \left[\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_{|\mathcal{K}|}\right] \in \mathbb{R}_+^{m \times |\mathcal{K}|}$, defined as

$$\begin{aligned}& L(s, \boldsymbol{t}, \boldsymbol{\Xi}, \zeta, \eta, \boldsymbol{\Lambda}, \boldsymbol{M}) \\ & := s + \left(-s + \left\langle \boldsymbol{k} \odot \boldsymbol{t} + \boldsymbol{\Xi}^\top \mathbf{1}, \boldsymbol{\rho}' \right\rangle \right) \zeta \quad + \left\langle \boldsymbol{x} \mathbf{1}^\top - \mathbf{1} \boldsymbol{t}^\top - \boldsymbol{\Xi}, \boldsymbol{\Lambda} \right\rangle - s\eta - \langle \boldsymbol{\Xi}, \boldsymbol{M} \rangle \\ & = (1 - \zeta - \eta) s + \left\langle \zeta \cdot \boldsymbol{k} \odot \boldsymbol{\rho}' - \boldsymbol{\Lambda}^\top \mathbf{1}, \boldsymbol{t} \right\rangle + \left\langle \zeta \cdot \mathbf{1} \left(\boldsymbol{\rho}'\right)^\top - \boldsymbol{\Lambda} - \boldsymbol{M}, \boldsymbol{\Xi} \right\rangle + \langle \boldsymbol{\Lambda} \mathbf{1}, \boldsymbol{x} \rangle .\end{aligned} \tag{47}$$

At the saddle point, it holds that

$$1 - \zeta - \eta = 0, \quad \zeta \cdot \boldsymbol{k} \odot \boldsymbol{\rho}' - \boldsymbol{\Lambda}^\top \mathbf{1} = \mathbf{0}, \tag{48}$$

$$\zeta \cdot \mathbf{1} \left(\boldsymbol{\rho}'\right)^\top - \boldsymbol{\Lambda} - \boldsymbol{M} = \boldsymbol{O}. \tag{49}$$

From these equations and the non-negativity of dual variables, we get, $\forall \ell \in [|\mathcal{K}|]$,

$$1 \geq \zeta = \frac{\langle \mathbf{1}, \boldsymbol{\lambda}_\ell \rangle}{k_\ell \rho'_\ell}, \quad \boldsymbol{\lambda}_\ell \leq \zeta \rho'_\ell \mathbf{1} = \frac{1}{k_\ell} \mathbf{1}\mathbf{1}^\top \boldsymbol{\lambda}_\ell, \tag{50}$$

equivalently,

$$\forall \ell \in [|\mathcal{K}|], \quad \boldsymbol{\lambda}_\ell \in \Delta(k_\ell, k_\ell \rho'_\ell),\ \zeta = \frac{\langle \mathbf{1}, \boldsymbol{\lambda}_\ell \rangle}{k_\ell \rho'_\ell}. \tag{51}$$

If introducing a vector

$$\boldsymbol{\beta} := \boldsymbol{\lambda}_1 + \dots + \boldsymbol{\lambda}_{|\mathcal{K}|} = \boldsymbol{\Lambda} \mathbf{1}, \tag{52}$$

the saddle point condition is that, for any $(\zeta, \boldsymbol{\Lambda}) \in \mathbb{R}_+ \times \mathbb{R}_+^{m \times |\mathcal{K}|}$, satisfying (51), the vector $\boldsymbol{\beta}$ is in the set $\mathcal{B}_{\text{wtk}}$. Hence, we obtain

$$\max_{\zeta, \eta \in \mathbb{R}_+,\, \boldsymbol{\Lambda}, \boldsymbol{M} \in \mathbb{R}_+^{m \times |\mathcal{K}|}} \left(\min_{s \in \mathbb{R}_+, \boldsymbol{t} \in \mathbb{R}^{|\mathcal{K}|}, \boldsymbol{\Xi} \in \mathbb{R}_+^{m \times |\mathcal{K}|}} L(s, \boldsymbol{t}, \boldsymbol{\Xi}, \zeta, \eta, \boldsymbol{\Lambda}, \boldsymbol{M}) \right) = \max_{\boldsymbol{\beta} \in \mathcal{B}_{\text{wtk}}} \langle \boldsymbol{x}, \boldsymbol{\beta} \rangle \tag{53}$$

which proves the equation (45).

A.4. Proof for Lemma 3

Observe that, $\forall \boldsymbol{\alpha} \in -\mathrm{dom}\phi_*$,

$$\langle \boldsymbol{\alpha}, \boldsymbol{g} \rangle = \langle -\boldsymbol{\alpha}, \boldsymbol{f} - \boldsymbol{g} \rangle + \langle \boldsymbol{\alpha}, \boldsymbol{f} \rangle = \langle -\boldsymbol{\alpha}, \boldsymbol{f} - \boldsymbol{g} \rangle - \phi_*(-\boldsymbol{\alpha}) \tag{54}$$

which leads to

$$\begin{aligned} \mathrm{argmax}_{\boldsymbol{\alpha} \in -\mathrm{dom}\phi_*} \langle \boldsymbol{\alpha}, \boldsymbol{g} \rangle &= \mathrm{argmax}_{\boldsymbol{\alpha} \in -\mathrm{dom}\phi_*} \langle -\boldsymbol{\alpha}, \boldsymbol{f} - \boldsymbol{g} \rangle - \phi_*(-\boldsymbol{\alpha}) \\ &= -\mathrm{argmax}_{\boldsymbol{v} \in \mathrm{dom}\phi_*} \langle \boldsymbol{v}, \boldsymbol{f} - \boldsymbol{g} \rangle - \phi_*(\boldsymbol{v}) \\ &= -\partial\phi(\boldsymbol{f} - \boldsymbol{g}). \end{aligned} \tag{55}$$

Hence, we have

$$\begin{aligned} -\partial\phi(\boldsymbol{f} - \eta\boldsymbol{g}) &= \mathrm{argmax}_{\boldsymbol{\alpha} \in -\mathrm{dom}\phi_*} \langle \boldsymbol{\alpha}, \eta\boldsymbol{g} \rangle \\ &= \mathrm{argmax}_{\boldsymbol{\alpha} \in -\mathrm{dom}\phi_*} \eta \langle \boldsymbol{\alpha}, \boldsymbol{g} \rangle = \mathrm{argmax}_{\boldsymbol{\alpha} \in -\mathrm{dom}\phi_*} \langle \boldsymbol{\alpha}, \boldsymbol{g} \rangle \end{aligned} \tag{56}$$

where the last equality follows since the set of the optimal solutions is unchanged even if the objective function is divided by a positive value η.

A.5. Proof for Proposition 5

We shall show that the problem

$$\max \quad D_{\mathrm{stk}}(\boldsymbol{A}) \quad \mathrm{wrt} \quad \boldsymbol{A} \in \mathbb{R}^{m \times n} \tag{57}$$

is dual to the problem

$$\min \quad \tilde{P}_{\mathrm{wtk}}(\tilde{\boldsymbol{W}}) \qquad \mathrm{wrt} \quad \tilde{\boldsymbol{W}} \in \mathbb{R}^{(d+n) \times m}. \tag{58}$$

Let

$$\tilde{\boldsymbol{X}} := [\tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_n] = \begin{bmatrix} \boldsymbol{X} \\ \sqrt{\gamma\lambda n}\boldsymbol{I}_n \end{bmatrix}. \tag{59}$$

It can be seen immediately that the following problem is dual to (58):

$$\begin{aligned} &\max \quad \tilde{D}_{\mathrm{wtk}}(\boldsymbol{A}) \qquad \mathrm{wrt} \quad \boldsymbol{A} \in \mathbb{R}^{m \times n} \\ &\text{where} \quad \tilde{D}_{\mathrm{wtk}}(\boldsymbol{A}) := -\frac{\lambda}{2} \left\| \frac{\tilde{\boldsymbol{X}}\boldsymbol{A}^\top}{\lambda n} \right\|_{\mathrm{F}}^2 - \frac{1}{n}\sum_{i=1}^{n} \Phi^*_{\mathrm{wtk}}(-\boldsymbol{\alpha}_i\,;\, y_i), \end{aligned} \tag{60}$$

We decompose the first term of $\tilde{D}_{\mathrm{wtk}}(\boldsymbol{A})$ as

$$\left\| \frac{\tilde{\boldsymbol{X}}\boldsymbol{A}^\top}{\lambda n} \right\|_{\mathrm{F}}^2 = \left\| \frac{\boldsymbol{X}\boldsymbol{A}^\top}{\lambda n} \right\|_{\mathrm{F}}^2 + \frac{\gamma\lambda n}{2\lambda^2 n^2}\|\boldsymbol{A}\|_{\mathrm{F}}^2 = \|\boldsymbol{W}(\boldsymbol{A})\|_{\mathrm{F}}^2 + \frac{\gamma}{2\lambda n}\sum_{i=1}^{n}\|\boldsymbol{\alpha}_i\|^2 \tag{61}$$

to obtain

$$\begin{aligned} \tilde{D}_{\mathrm{wtk}}(\boldsymbol{A}) &= -\frac{\lambda}{2}\|\boldsymbol{W}(\boldsymbol{A})\|_{\mathrm{F}}^2 - \frac{1}{n}\sum_{i=1}^{n}\left(\Phi^*_{\mathrm{wtk}}(-\boldsymbol{\alpha}_i\,;\, y_i) + \frac{\gamma}{2}\|\boldsymbol{\alpha}_i\|^2\right) \\ &= -\frac{\lambda}{2}\|\boldsymbol{W}(\boldsymbol{A})\|_{\mathrm{F}}^2 - \frac{1}{n}\sum_{i=1}^{n}\Phi^*_{\mathrm{stk}}(-\boldsymbol{\alpha}_i\,;\, y_i) = D_{\mathrm{stk}}(\boldsymbol{A}). \end{aligned} \tag{62}$$

Therefore, the problem (57) is dual to the problem (58).

A.6. Proof for Proposition 6

We can set $y = m$ without loss of generality. From Proposition 2 of Lapin et al. (2015)'s paper, the convex conjugate of $\phi_{\text{utk}}(\cdot\,;\,y)$ is given by:

$$\phi^*_{\text{utk}}(\boldsymbol{v}\,;\,y) = \begin{cases} -\langle \boldsymbol{v}, \boldsymbol{1} - \boldsymbol{e}_y \rangle & \text{if } \boldsymbol{v} \in \Delta(k,1), \\ +\infty & \text{o.w.} \end{cases} \tag{63}$$

which allows us to rewrite RHS of (28) as

$$\text{RHS of (28)} = \phi^*_{\text{utk}}\left(\begin{bmatrix} \boldsymbol{v}^{\setminus y} \\ 0 \end{bmatrix} ; y \right) = \begin{cases} -\left\langle \boldsymbol{v}^{\setminus y}, \boldsymbol{1} \right\rangle & \text{if } \boldsymbol{v}^{\setminus y} \in \mathcal{B}_{\text{rhs}}, \\ +\infty & \text{o.w.} \end{cases} \tag{64}$$

where we have defined

$$\mathcal{B}_{\text{rhs}} := \left\{ \boldsymbol{v}^{\setminus y} \in \mathbb{R}^{m-1} \,\middle|\, \begin{bmatrix} \boldsymbol{v}^{\setminus y} \\ 0 \end{bmatrix} \in \Delta(k,1) \right\}. \tag{65}$$

By the way, the unweighted top-k hinge loss has another equivalent expression:

$$\phi_{\text{utk}}\left(\begin{bmatrix} \boldsymbol{s}^{\setminus y} \\ s_y \end{bmatrix} ; y \right) = \max_{\boldsymbol{\lambda} \in \Delta(k,1)} \left\langle \boldsymbol{\lambda}, \begin{bmatrix} \boldsymbol{s}^{\setminus y} + \boldsymbol{1} \\ s_y \end{bmatrix} \right\rangle. \tag{66}$$

The derivation for the above equality can be found in the proof for Proposition 2 of Lapin et al. (2015)'s paper. We exploit the above equality to rearrange LHS of (28) as

$$\begin{aligned} \text{LHS of (28)} &= \max_{\boldsymbol{s}^{\setminus y} \in \mathbb{R}^{m-1}} \left\{ \left\langle \boldsymbol{s}^{\setminus y}, \boldsymbol{v}^{\setminus y} \right\rangle - \phi_{\text{utk}}\left(\begin{bmatrix} \boldsymbol{s}^{\setminus y} \\ 0 \end{bmatrix} \right) \right\} \\ &= \max_{\boldsymbol{s}^{\setminus y} \in \mathbb{R}^{m-1}} \left\{ \left\langle \boldsymbol{s}^{\setminus y}, \boldsymbol{v}^{\setminus y} \right\rangle - \max_{\boldsymbol{\lambda} \in \Delta(k,1)} \left\langle \boldsymbol{\lambda}^{\setminus y}, \boldsymbol{s}^{\setminus y} + \boldsymbol{1} \right\rangle \right\} \\ &= \min_{\boldsymbol{\lambda} \in \Delta(k,1)} \left\{ -\left\langle \boldsymbol{\lambda}^{\setminus y}, \boldsymbol{1} \right\rangle + \max_{\boldsymbol{s}^{\setminus y} \in \mathbb{R}^{m-1}} \left\langle \boldsymbol{s}^{\setminus y}, \boldsymbol{v}^{\setminus y} - \boldsymbol{\lambda}^{\setminus y} \right\rangle \right\} \\ &= \begin{cases} -\left\langle \boldsymbol{v}^{\setminus y}, \boldsymbol{1} \right\rangle & \text{if } \boldsymbol{v}^{\setminus y} \in \mathcal{B}_{\text{lhs}}, \\ +\infty & \text{o.w.} \end{cases} \end{aligned} \tag{67}$$

where we have defined

$$\mathcal{B}_{\text{lhs}} := \left\{ \boldsymbol{v}^{\setminus y} \in \mathbb{R}^{m-1} \,\middle|\, \exists \beta \in \mathbb{R}, \quad \begin{bmatrix} \boldsymbol{v}^{\setminus y} \\ \beta \end{bmatrix} \in \Delta(k,1) \right\}. \tag{68}$$

Setting the entries in the vector $\boldsymbol{v}^{\setminus y}$ to

$$v_{m-1} := \frac{1}{k}, \qquad \forall h \in [m-2], \quad v_h := \frac{(m-1)k - m}{(m-2)mk}, \tag{69}$$

it can be readily seen that $\boldsymbol{v}^{\setminus y} \in \mathcal{B}_{\text{lhs}} \setminus \mathcal{B}_{\text{rhs}}$ implying $\mathcal{B}_{\text{lhs}} \neq \mathcal{B}_{\text{rhs}}$, which concludes that the unweighted top-k hinge is not y-compatible if $2 \leq k < m$.

A.7. Convex Conjugate of Pseudo Top-k Hinge

Define $\phi_{\mathrm{ptk}}\left(\cdot\,;\,y\right):\mathbb{R}^m\to\mathbb{R}$ as

$$\phi_{\mathrm{ptk}}\left(\boldsymbol{s}\,;\,y\right):=\max\left\{0,1+\frac{1}{k}\sum_{j=1}^{k}\left(\boldsymbol{s}^{\setminus y}\right)_{\pi(j;\boldsymbol{s}^{\setminus y})}\right\}. \tag{70}$$

Its convex conjugate is given by:

$$\phi^*_{\mathrm{ptk}}\left(\boldsymbol{v}\,;\,y\right)=\begin{cases}-\left\langle\boldsymbol{v}^{\setminus y},\boldsymbol{1}\right\rangle & \text{if } v_m=0\quad\text{and}\quad\boldsymbol{v}^{\setminus y}\in\Delta_{k,m-1},\\ +\infty & \text{o.w.}\end{cases} \tag{71}$$

where $\Delta_{k,m-1}$ is the top-k simplex in $(m-1)$-dimensional space:

$$\Delta_{k,m-1}:=\left\{\boldsymbol{\beta}\in\mathbb{R}^{m-1}_{+}\mid\langle\boldsymbol{1},\boldsymbol{\beta}\rangle\le 1,\,\boldsymbol{\beta}\le\frac{1}{k}\boldsymbol{1}\boldsymbol{1}^\top\boldsymbol{\beta}\right\}. \tag{72}$$

Equation (71) implies that $\phi_{\mathrm{ptk}}(\cdot;y)$ is y-compatible. Combining the fact of $\Phi_{\mathrm{ptk}}\left(\boldsymbol{s}\,;\,y\right)=\phi_{\mathrm{ptk}}\left(\boldsymbol{H}_y\boldsymbol{s}\,;\,y\right)$ with Lemma 2 in Lapin et al. (2015)'s paper, we obtain the convex conjugate of $\Phi_{\mathrm{ptk}}\left(\cdot\,;\,y\right)$ as

$$\begin{aligned}\Phi^*_{\mathrm{ptk}}\left(\boldsymbol{v}\,;\,y\right)&=\begin{cases}\phi^*_{\mathrm{ptk}}\left(\boldsymbol{v}-v_y\boldsymbol{e}_y\,;\,y\right) & \text{if }\langle\boldsymbol{v},\boldsymbol{1}\rangle=0,\\ +\infty & \text{o.w.}\end{cases}\\ &=\begin{cases}v_m & \text{if }\langle\boldsymbol{v},\boldsymbol{1}\rangle=0\text{ and }\begin{bmatrix}\boldsymbol{v}_{\setminus y}\\ 0\end{bmatrix}\in\Delta(k,1),\\ \\ +\infty & \text{o.w.}\end{cases}\end{aligned} \tag{73}$$

A.8. Derivation of (71): Convex Conjugate of $\phi_{\mathrm{ptk}}\left(\cdot\,;\,y\right)$

Here we describe how the convex conjugate of $\phi_{\mathrm{ptk}}\left(\cdot\,;\,y\right):\mathbb{R}^m\to\mathbb{R}$ defined in (70) is derived. With help of Lemma 8, the function $\phi_{\mathrm{ptk}}\left(\cdot\,;\,y\right)$ can be rewritten as

$$\phi_{\mathrm{ptk}}\left(\boldsymbol{s}\,;\,y\right)=\max_{\boldsymbol{\lambda}^{\setminus y}\in\Delta_{k,m-1}}\left\langle\boldsymbol{\lambda}^{\setminus y},\boldsymbol{s}^{\setminus y}+\boldsymbol{1}\right\rangle. \tag{74}$$

Using this, the convex conjugate of this function can be obtained as

$$\begin{aligned}\phi^*_{\mathrm{ptk}}\left(\boldsymbol{v}\,;\,y\right)&=\max_{\boldsymbol{s}\in\mathbb{R}^m}\left(\langle\boldsymbol{s},\boldsymbol{v}\rangle-\max_{\boldsymbol{\lambda}^{\setminus y}\in\Delta_{k,m-1}}\left\langle\boldsymbol{\lambda}^{\setminus y},\boldsymbol{1}+\boldsymbol{s}^{\setminus y}\right\rangle\right)\\ &=\min_{\boldsymbol{\lambda}^{\setminus y}\in\Delta_{k,m-1}}\left(-\left\langle\boldsymbol{\lambda}^{\setminus y},\boldsymbol{1}\right\rangle+\max_{\boldsymbol{s}^{\setminus y}\in\mathbb{R}^m}\left\langle\boldsymbol{s}^{\setminus y},\boldsymbol{v}^{\setminus y}-\boldsymbol{\lambda}^{\setminus y}\right\rangle+\max_{s\in\mathbb{R}}v_m s_m\right)\\ &=\begin{cases}-\left\langle\boldsymbol{v}^{\setminus y},\boldsymbol{1}\right\rangle & \text{if } v_m=0\quad\text{and}\quad\boldsymbol{v}^{\setminus y}\in\Delta_{k,m-1},\\ +\infty & \text{o.w.}\end{cases}\end{aligned} \tag{75}$$

Thus, (71) is derived.

A.9. Proof for Proposition 7

Let us introduce $d_{\text{ptk}} : \mathbb{R}^{m \times n} \to \mathbb{R}$ as

$$d_{\text{ptk}}(\boldsymbol{A}) := -\frac{1}{2\lambda n^2} \left\| \sum_{i=1}^{n} \left(\boldsymbol{H}_{y_i}^{\top} \otimes \boldsymbol{x}_i \right) \boldsymbol{\alpha}_i \right\|_{\text{F}}^2 - \frac{1}{n} \sum_{i=1}^{n} \phi_{\text{ptk}}^{*}(-\boldsymbol{\alpha}_i \,;\, y_i). \tag{76}$$

This function d_{ptk} is the Fenchel dual of

$$p_{\text{ptk}}(\boldsymbol{w}) := \frac{\lambda}{2} \left\| \boldsymbol{w} \right\|^2 + \frac{1}{n} \sum_{i=1}^{n} \phi_{\text{ptk}} \left((\boldsymbol{x}_i^{\top} \otimes \boldsymbol{H}_{y_i}) \boldsymbol{w} \,;\, y_i \right). \tag{77}$$

It can be seen that $\forall \boldsymbol{W} \in \mathbb{R}^{d \times m}$, $P_{\text{ptk}}(\boldsymbol{W}) = p_{\text{ptk}}(\text{vec}(\boldsymbol{W}))$. The Fenchel dual of $p_{\text{ptk}} : \mathbb{R}^{md} \to \mathbb{R}$ has the following property:

$$\forall \boldsymbol{A} \in \mathbb{R}^{m \times n}, \quad d_{\text{ptk}}(\boldsymbol{A}) = d_{\text{ptk}}(\boldsymbol{A} - (\boldsymbol{A} \odot \boldsymbol{E}_{\boldsymbol{y}})) = d_{\text{utk}}(\boldsymbol{A} - (\boldsymbol{A} \odot \boldsymbol{E}_{\boldsymbol{y}})) \tag{78}$$

which implies that

$$\max_{\boldsymbol{A} \in \mathbb{R}^{m \times n}} \left\{ d_{\text{ptk}}(\boldsymbol{A}) \right\} = \max_{\boldsymbol{A} \in \mathbb{R}^{m \times n}} \left\{ d_{\text{utk}}(\boldsymbol{A}) \,|\, \boldsymbol{A} \odot \boldsymbol{E}_{\boldsymbol{y}} = \boldsymbol{O} \right\}. \tag{79}$$

Therefore, it is concluded that the problem of maximizing $d_{\text{utk}}(\boldsymbol{A})$ subject to $\boldsymbol{A} \odot \boldsymbol{E}_{\boldsymbol{y}} = \boldsymbol{O}$ is equivalent to the unconstrained problem of maximizing $d_{\text{ptk}}(\boldsymbol{A})$, which is dual to the problem of minimizing the regularized empirical risk based on the pseudo top-k hinge.

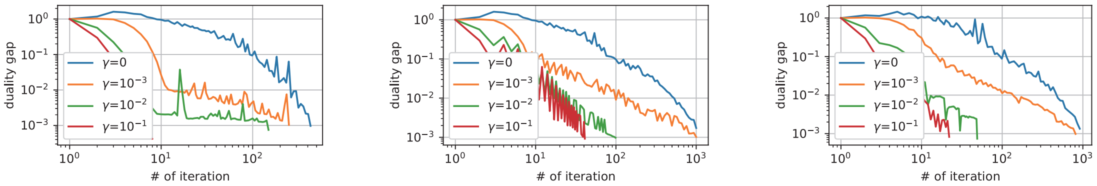
Appendix B. Additional Experimental Results

B.1. Convergence Speeds to the Optimum

Here, additional plots of the duality gaps against the number of iterations are shown in Figures 3 and 4.

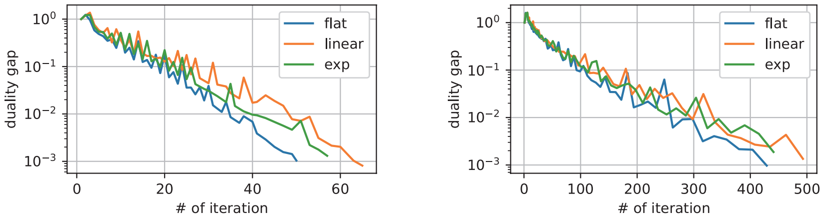
B.2. Top-k Accuracies

The unweighted top-k hinge is a convex surrogate of the top-k 0/1 loss, and has a hyper-parameter k. The three weighting schemes for the weighted top-k hinge, used in our experiments, also has a hyper-parameter k. Table 4 reports the best one amongst for top-k accuracies with $k = 1, 3, 5, 10$ on six datasets due to space limitation. The individual top-k accuracies on the six datasets are shown in Tables 2, 3, 4, 5, 6, and 7.



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

Figure 3: 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

Figure 4: 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. (2015)'s algorithm, and Chu et al. (2018)'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

Table 2: Top-k accuracies on ALOI.

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours, $k = 1$)	0.841	0.928	0.948	0.971
UTk (ours, $k = 3$)	0.837	0.927	0.946	0.971
UTk (ours, $k = 5$)	0.837	0.929	0.948	0.970
UTk (ours, $k = 10$)	0.826	0.924	0.946	0.973
WTk (linear, $k = 1$)	0.841	0.928	0.948	0.971
WTk (linear, $k = 3$)	**0.842**	0.928	0.949	0.971
WTk (linear, $k = 5$)	0.839	0.929	0.947	0.970
WTk (linear, $k = 10$)	0.836	0.927	0.947	0.973
WTk (exp, $k = 1$)	0.841	0.928	0.948	0.971
WTk (exp, $k = 3$)	0.839	0.929	0.948	0.971
WTk (exp, $k = 5$)	0.838	**0.930**	0.947	0.972
WTk (exp, $k = 10$)	0.833	0.925	**0.949**	**0.974**
UTk (Lapin, $k = 1$)	0.826	0.913	0.935	0.962
UTk (Lapin, $k = 3$)	0.819	0.915	0.940	0.965
UTk (Lapin, $k = 5$)	0.831	0.915	0.941	0.964
UTk (Lapin, $k = 10$)	0.834	0.929	0.949	0.962
UTk (Chu, $k = 1$)	0.825	0.909	0.937	0.964
UTk (Chu, $k = 3$)	0.817	0.913	0.941	0.964
UTk (Chu, $k = 5$)	0.808	0.920	0.949	0.970
UTk (Chu, $k = 10$)	0.783	0.904	0.936	0.972

Table 3: Top-k accuracies on Caltech101.

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours, $k = 1$)	0.547	0.710	0.763	0.827
UTk (ours, $k = 3$)	0.548	0.711	0.765	0.827
UTk (ours, $k = 5$)	0.542	0.718	0.773	0.842
UTk (ours, $k = 10$)	0.539	0.719	**0.777**	0.844
WTk (linear, $k = 1$)	0.547	0.710	0.763	0.827
WTk (linear, $k = 3$)	**0.550**	0.712	0.761	0.827
WTk (linear, $k = 5$)	0.549	0.714	0.767	0.840
WTk (linear, $k = 10$)	0.545	**0.723**	0.774	0.843
WTk (exp, $k = 1$)	0.547	0.710	0.763	0.827
WTk (exp, $k = 3$)	0.546	0.709	0.763	0.826
WTk (exp, $k = 5$)	0.536	0.717	0.769	0.828
WTk (exp, $k = 10$)	0.542	0.722	0.775	**0.844**
UTk (Lapin, $k = 1$)	0.527	0.675	0.725	0.803
UTk (Lapin, $k = 3$)	0.537	0.699	0.748	0.817
UTk (Lapin, $k = 5$)	0.541	0.703	0.757	0.821
UTk (Lapin, $k = 10$)	0.544	0.723	0.767	0.827
UTk (Chu, $k = 1$)	0.528	0.675	0.725	0.804
UTk (Chu, $k = 3$)	0.532	0.699	0.753	0.823
UTk (Chu, $k = 5$)	0.535	0.709	0.768	0.825
UTk (Chu, $k = 10$)	0.508	0.718	0.773	0.829

Table 4: Top-k accuracies on CUB.

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours, $k = 1$)	**0.592**	0.771	0.831	0.902
UTk (ours, $k = 3$)	0.591	0.775	0.841	**0.908**
UTk (ours, $k = 5$)	0.584	0.777	**0.847**	0.904
UTk (ours, $k = 10$)	0.576	0.767	0.845	0.903
WTk (linear, $k = 1$)	**0.592**	0.771	0.831	0.902
WTk (linear, $k = 3$)	0.588	0.775	0.831	0.908
WTk (linear, $k = 5$)	0.586	**0.780**	0.843	0.907
WTk (linear, $k = 10$)	0.580	0.765	0.844	0.908
WTk (exp, $k = 1$)	**0.592**	0.771	0.831	0.902
WTk (exp, $k = 3$)	0.586	0.778	0.838	0.903
WTk (exp, $k = 5$)	0.588	0.773	0.843	0.908
WTk (exp, $k = 10$)	0.578	0.773	0.843	0.907
UTk (Lapin, $k = 1$)	0.580	0.762	0.824	0.890
UTk (Lapin, $k = 3$)	0.578	0.762	0.824	0.891
UTk (Lapin, $k = 5$)	0.579	0.766	0.823	0.899
UTk (Lapin, $k = 10$)	0.579	0.770	0.834	0.901
UTk (Chu, $k = 1$)	0.579	0.762	0.824	0.888
UTk (Chu, $k = 3$)	0.578	0.761	0.826	0.896
UTk (Chu, $k = 5$)	0.575	0.766	0.836	0.899
UTk (Chu, $k = 10$)	0.555	0.768	0.842	0.903

Table 5: Top-k accuracies on Indoor67.

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours, $k = 1$)	0.697	0.873	0.924	0.963
UTk (ours, $k = 3$)	0.693	0.873	0.923	0.963
UTk (ours, $k = 5$)	0.673	0.878	0.925	0.966
UTk (ours, $k = 10$)	0.645	0.878	0.924	**0.969**
WTk (linear, $k = 1$)	0.697	0.873	0.924	0.963
WTk (linear, $k = 3$)	**0.697**	0.873	0.923	0.963
WTk (linear, $k = 5$)	0.688	0.876	0.924	0.964
WTk (linear, $k = 10$)	0.669	**0.881**	0.930	0.968
WTk (exp, $k = 1$)	0.697	0.873	0.924	0.963
WTk (exp, $k = 3$)	0.697	0.874	0.924	0.963
WTk (exp, $k = 5$)	0.685	0.877	0.925	0.966
WTk (exp, $k = 10$)	0.662	0.879	**0.931**	0.968
UTk (Lapin, $k = 1$)	0.683	0.857	0.905	0.953
UTk (Lapin, $k = 3$)	0.686	0.868	0.917	0.958
UTk (Lapin, $k = 5$)	0.688	0.873	0.923	0.961
UTk (Lapin, $k = 10$)	0.683	0.877	0.927	0.966
UTk (Chu, $k = 1$)	0.683	0.857	0.905	0.953
UTk (Chu, $k = 3$)	0.672	0.874	0.920	0.963
UTk (Chu, $k = 5$)	0.666	0.875	0.924	0.965
UTk (Chu, $k = 10$)	0.632	0.870	0.924	0.968

Table 6: Top-k accuracies on Letter.

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours, $k = 1$)	0.756	0.892	0.937	0.984
UTk (ours, $k = 3$)	0.759	0.900	0.944	0.984
UTk (ours, $k = 5$)	0.725	0.910	0.951	0.987
UTk (ours, $k = 10$)	0.650	0.898	**0.961**	0.994
WTk (linear, $k = 1$)	0.756	0.892	0.937	0.984
WTk (linear, $k = 3$)	**0.766**	0.891	0.941	0.983
WTk (linear, $k = 5$)	0.756	0.857	0.934	0.985
WTk (linear, $k = 10$)	0.713	0.909	0.955	0.991
WTk (exp, $k = 1$)	0.756	0.892	0.937	0.984
WTk (exp, $k = 3$)	0.765	0.884	0.943	0.984
WTk (exp, $k = 5$)	0.744	0.905	0.947	0.986
WTk (exp, $k = 10$)	0.688	0.908	0.957	0.991
UTk (Lapin, $k = 1$)	0.760	0.887	0.925	0.973
UTk (Lapin, $k = 3$)	0.761	0.901	0.940	0.978
UTk (Lapin, $k = 5$)	0.747	0.906	0.945	0.981
UTk (Lapin, $k = 10$)	0.735	0.907	0.951	0.988
UTk (Chu, $k = 1$)	0.761	0.881	0.923	0.973
UTk (Chu, $k = 3$)	0.748	0.905	0.940	0.976
UTk (Chu, $k = 5$)	0.712	**0.910**	0.949	0.986
UTk (Chu, $k = 10$)	0.605	0.890	0.960	**0.995**

Table 7: Top-k accuracies on News20.

Method	Top-1	Top-3	Top-5	Top-10
UTk (ours, $k = 1$)	0.666	0.870	0.921	0.973
UTk (ours, $k = 3$)	0.664	0.874	0.926	0.972
UTk (ours, $k = 5$)	0.656	**0.876**	0.927	0.975
UTk (ours, $k = 10$)	0.621	0.867	**0.929**	0.975
WTk (linear, $k = 1$)	0.666	0.870	0.921	0.973
WTk (linear, $k = 3$)	0.665	0.871	0.925	0.972
WTk (linear, $k = 5$)	0.665	0.872	0.925	0.973
WTk (linear, $k = 10$)	0.650	0.872	0.929	0.976
WTk (exp, $k = 1$)	0.666	0.870	0.921	0.973
WTk (exp, $k = 3$)	**0.666**	0.872	0.924	0.973
WTk (exp, $k = 5$)	0.662	0.875	0.926	0.974
WTk (exp, $k = 10$)	0.641	0.873	0.929	**0.976**
UTk (Lapin, $k = 1$)	0.656	0.837	0.900	0.964
UTk (Lapin, $k = 3$)	0.653	0.860	0.911	0.969
UTk (Lapin, $k = 5$)	0.657	0.867	0.920	0.970
UTk (Lapin, $k = 10$)	0.655	0.875	0.926	0.972
UTk (Chu, $k = 1$)	0.656	0.838	0.900	0.964
UTk (Chu, $k = 3$)	0.662	0.863	0.917	0.969
UTk (Chu, $k = 5$)	0.644	0.865	0.922	0.972
UTk (Chu, $k = 10$)	0.612	0.863	0.920	0.975