Loss Functions for Top-k Error: Analysis and Insights

Maksim Lapin,[1] Matthias Hein[2] and Bernt Schiele[1]
[1]Max Planck Institute for Informatics, Saarbrücken, Germany
[2]Saarland University, Saarbrücken, Germany

Abstract

In order to push the performance on realistic computer vision tasks, the number of classes in modern benchmark datasets has significantly increased in recent years. This increase in the number of classes comes along with increased ambiguity between the class labels, raising the question if top-1 error is the right performance measure. In this paper, we provide an extensive comparison and evaluation of established multiclass methods comparing their top-k performance both from a practical as well as from a theoretical perspective. Moreover, we introduce novel top-k loss functions as modifications of the softmax and the multiclass SVM losses and provide efficient optimization schemes for them. In the experiments, we compare on various datasets all of the proposed and established methods for top-k error optimization. An interesting insight of this paper is that the softmax loss yields competitive top-k performance for all k simultaneously. For a specific top-k error, our new top-k losses lead typically to further improvements while being faster to train than the softmax.

1. Introduction

The number of classes is rapidly growing in modern computer vision benchmarks [37, 52]. Typically, this also leads to ambiguity in the labels as classes start to overlap. Even for humans, the error rates in top-1 performance are often quite high ($\approx 30\%$ on SUN 397 [50]). While previous research focuses on minimizing the top-1 error, we address top-k error optimization in this paper. We are interested in two cases: a) achieving small top-k error for *all* reasonably small k; and b) minimization of a specific top-k error.

While it is argued in [2] that the one-versus-all (OVA) SVM scheme performs on par in top-1 and top-5 accuracy with the other SVM variations based on ranking losses, we have recently shown in [23] that minimization of the *top-k hinge loss* leads to improvements in top-k performance compared to OVA SVM, multiclass SVM, and other ranking-based formulations. In this paper, we study top-k error optimization from a wider perspective. On the one hand, we compare OVA schemes and direct multiclass losses in extensive experiments, and on the other, we present theoretical discussion regarding their calibration for the top-k error. Based on these insights, we suggest 4 new families of loss functions for the top-k error. Two are smoothed versions of the top-k hinge losses [23], and the other two are top-k versions of the softmax loss. We discuss their advantages and disadvantages, and for the convex losses provide an efficient implementation based on stochastic dual coordinate ascent (SDCA) [38].

We evaluate a battery of loss functions on 11 datasets of different tasks ranging from text classification to large scale vision benchmarks, including fine-grained and scene classification. We systematically optimize and report results separately for each top-k accuracy. One interesting message that we would like to highlight is that the softmax loss is able to optimize *all top-k error measures simultaneously*. This is in contrast to multiclass SVM and is also reflected in our experiments. Finally, we show that our new top-k variants of smooth multiclass SVM and the softmax loss can further improve top-k performance for a specific k.

Related work. Top-k optimization has recently received revived attention with the advent of large scale problems [18, 23, 24, 25]. The top-k error in multiclass classification, which promotes good ranking of class *labels* for each example, is closely related to the precision@k metric in information retrieval, which counts the fraction of positive instances among the top-k ranked *examples*. In essence, both approaches enforce a desirable *ranking* of items [23].

The classic approaches optimize pairwise ranking with $\text{SVM}^{\text{struct}}$ [20, 43], RankNet [10], and LaRank [6]. An alternative direction was proposed by Usunier *et al*. [44], who described a general family of convex loss functions for ranking and classification. One of the loss functions that we consider (top-k SVM^{β} [23]) also falls into that family. Weston *et al*. [49] then introduced Wsabie, which optimizes an approximation of a ranking-based loss from [44]. A Bayesian approach was suggested by [41].

Recent works focus on the top of the ranked list [1, 8, 29, 36], scalability to large datasets [18, 23, 24], explore transductive learning [25] and prediction of tuples [35].

Method	Name	Loss function	Conjugate	SDCA update	Top-k calibrated	Convex
SVM^{OVA}	One-vs-all (OVA) SVM	$\max\{0,\, 1-a\}$	[38]	[38]	no[1] (Prop. 1)	yes
LR^{OVA}	OVA logistic regression	$\log(1+e^{-a})$			yes (Prop. 2)	
$\text{SVM}^{\text{Multi}}$	Multiclass SVM	$\max\left\{0, (a+c)_{\pi_1}\right\}$	[23, 38]	[23, 38]	no (Prop. 3)	
LR^{Multi}	Softmax (maximum entropy)	$\log\left(\sum_{j\in\mathcal{Y}}\exp(a_j)\right)$	Prop. 7	Prop. 11	yes (Prop. 4)	
top-k SVM^{α}	Top-k hinge (α)	$\max\left\{0, \frac{1}{k}\sum_{j=1}^{k}(a+c)_{\pi_j}\right\}$	[23]	[23]	open question for $k>1$	
top-k SVM^{β}	Top-k hinge (β)	$\frac{1}{k}\sum_{j=1}^{k}\max\left\{0, (a+c)_{\pi_j}\right\}$				
top-k $\text{SVM}^{\alpha}_{\gamma}$	Smooth top-k hinge (α) *	Eq. (10) w/ Δ_k^{α}	Prop. 6	Prop. 10		
top-k $\text{SVM}^{\beta}_{\gamma}$	Smooth top-k hinge (β) *	Eq. (10) w/ Δ_k^{β}				
top-k Ent	Top-k entropy *	Prop. 8	Eq. (12)	Prop. 11		
top-k Ent_{tr}	Truncated top-k entropy *	Eq. (15)	-	-	yes (Prop. 9)	no

Note that $\text{SVM}^{\text{Multi}} \equiv \text{top-1 SVM}^{\alpha} \equiv \text{top-1 SVM}^{\beta}$ and $\text{LR}^{\text{Multi}} \equiv \text{top-1 Ent} \equiv \text{top-1 Ent}_{\text{tr}}$.

We let $a \triangleq yf(x)$ (binary one-vs-all); $a \triangleq (f_j(x) - f_y(x))_{j\in\mathcal{Y}}$, $c \triangleq \mathbf{1} - e_y$ (multiclass); $\pi : a_{\pi_1} \geq \ldots \geq a_{\pi_m}$.

Table 1: Overview of the methods we consider and our contributions. *Novel loss. [1]But *smoothed* one is (Prop. 5).

Contributions. We study the problem of top-k error optimization on a diverse range of learning tasks. We consider existing methods as well as propose 4 novel loss functions for minimizing the top-k error. A brief overview of the methods is given in Table 1. For the proposed convex top-k losses, we develop an efficient optimization scheme based on SDCA[1], which can also be used for training with the softmax loss. All methods are evaluated empirically in terms of the top-k error and, whenever possible, in terms of classification calibration. We discover that the softmax loss and the proposed smooth top-1 SVM are astonishingly competitive in all top-k errors. Further small improvements can be obtained with the new top-k losses.

2. Loss Functions for Top-k Error

We consider multiclass problems with m classes where the training set $(x_i, y_i)_{i=1}^n$ consists of n examples $x_i \in \mathbb{R}^d$ along with the corresponding labels $y_i \in \mathcal{Y} \triangleq \{1, \ldots, m\}$. We use π and τ to denote a permutation of (indexes) $\mathcal{Y}$. Unless stated otherwise, a_π reorders components of a vector $a \in \mathbb{R}^m$ in descending order, *i.e.* $a_{\pi_1} \geq a_{\pi_2} \geq \ldots \geq a_{\pi_m}$. While we consider linear classifiers in our experiments, all loss functions below are formulated in the general setting where a function $f : \mathcal{X} \to \mathbb{R}^m$ is learned and prediction at test time is done via $\arg\max_{y\in\mathcal{Y}} f_y(x)$, resp. the top-$k$ predictions. For the linear case, all predictors f_y have the form $f_y(x) = \langle w_y, x\rangle$. Let $W \in \mathbb{R}^{d\times m}$ be the stacked weight matrix, $L : \mathcal{Y} \times \mathbb{R}^m \to \mathbb{R}$ be a convex loss function, and $\lambda > 0$ be a regularization parameter. We consider the following multiclass optimization problem $\min_W \frac{1}{n}\sum_{i=1}^n L(y_i, W^\top x_i) + \lambda \|W\|_F^2$.

[1] Code available at: https://github.com/mlapin/libsdca

We use the Iverson bracket notation $[\![P]\!]$, defined as $[\![P]\!] = 1$ if P is true, 0 otherwise; and introduce a shorthand $p_y(x) \triangleq \Pr(Y = y \,|\, X = x)$. We generalize the standard zero-one error and allow k guesses instead of one. Formally, the **top-k zero-one loss** (**top-k error**) is

$$\text{err}_k(y, f(x)) \triangleq [\![f_{\pi_k}(x) > f_y(x)]\!]. \tag{1}$$

Note that for $k = 1$ we recover the standard zero-one error. **Top-k accuracy** is defined as 1 minus the top-k error.

All proofs and technical details are in the supplement.

2.1. Bayes Optimality and Top-k Calibration

In this section, we establish the best achievable top-k error, determine when a classifier achieves it, and define a notion of top-k calibration.

Lemma 1. *The Bayes optimal top-k error at x is*

$$\min_{g\in\mathbb{R}^m} \mathbb{E}_{Y|X}[\text{err}_k(Y, g) \,|\, X = x] = 1 - \textstyle\sum_{j=1}^k p_{\tau_j}(x),$$

*where $p_{\tau_1}(x) \geq p_{\tau_2}(x) \geq \ldots \geq p_{\tau_m}(x)$. A classifier f is **top-k Bayes optimal at** x if and only if*

$$\{y \,|\, f_y(x) \geq f_{\pi_k}(x)\} \subset \{y \,|\, p_y(x) \geq p_{\tau_k}(x)\},$$

where $f_{\pi_1}(x) \geq f_{\pi_2}(x) \geq \ldots \geq f_{\pi_m}(x)$.

Optimization of the zero-one loss (and, by extension, the top-k error) leads to hard combinatorial problems. Instead, a standard approach is to use a convex surrogate loss which upper bounds the zero-one error. Under mild conditions on the loss function [3, 42], the optimal classifier w.r.t. the surrogate yields a Bayes optimal solution for the zero-one loss. Such loss is called *classification calibrated*, which is known

in statistical learning theory as a necessary condition for a classifier to be universally Bayes consistent [3]. We introduce now the notion of calibration for the top-k error.

Definition 1. *A loss function $L : \mathcal{Y} \times \mathbb{R}^m \to \mathbb{R}$ (or a reduction scheme) is called **top-k calibrated** if for all possible data generating measures on $\mathbb{R}^d \times \mathcal{Y}$ and all $x \in \mathbb{R}^d$*

$$\arg\min_{g \in \mathbb{R}^m} \mathbb{E}_{Y|X}[L(Y, g) \,|\, X = x]$$
$$\subseteq \arg\min_{g \in \mathbb{R}^m} \mathbb{E}_{Y|X}[\mathrm{err}_k(Y, g) \,|\, X = x].$$

If a loss is *not* top-k calibrated, it implies that even in the limit of infinite data, one does not obtain a classifier with the Bayes optimal top-k error from Lemma 1.

2.2. OVA and Direct Multiclass Approaches

The standard multiclass problem is often solved using the one-vs-all (OVA) reduction into a set of m binary classification problems. Every class is trained versus the rest which yields m classifiers $\{f_y\}_{y \in \mathcal{Y}}$.

Typically, the binary classification problems are formulated with a convex margin-based loss function $L(yf(x))$, where $L : \mathbb{R} \to \mathbb{R}$ and $y = \pm 1$. We consider in this paper:

$$L(yf(x)) = \max\{0,\, 1 - yf(x)\}, \tag{2}$$
$$L(yf(x)) = \log(1 + e^{-yf(x)}). \tag{3}$$

The **hinge** (2) and **logistic** (3) losses correspond to the SVM and logistic regression respectively. We now show when the OVA schemes are top-k calibrated, not only for $k = 1$ (standard multiclass loss) but *for all* k simultaneously.

Lemma 2. *The OVA reduction is top-k calibrated for any $1 \leq k \leq m$ if the Bayes optimal function of the convex margin-based loss $L(yf(x))$ is a strictly monotonically increasing function of $\Pr(Y = 1 \,|\, X = x)$.*

Next, we check if the one-vs-all schemes employing hinge and logistic regression losses are top-k calibrated.

Proposition 1. *OVA SVM is not top-k calibrated.*

In contrast, logistic regression is top-k calibrated.

Proposition 2. *OVA logistic regression is top-k calibrated.*

An alternative to the OVA scheme with binary losses is to use a *multiclass* loss $L : \mathcal{Y} \times \mathbb{R}^m \to \mathbb{R}$ directly. We consider two generalizations of the hinge and logistic losses below:

$$L(y, f(x)) = \max_{j \in \mathcal{Y}} \big\{[\![j \neq y]\!] + f_j(x) - f_y(x)\big\}, \tag{4}$$
$$L(y, f(x)) = \log\Big(\textstyle\sum_{j \in \mathcal{Y}} \exp(f_j(x) - f_y(x))\Big). \tag{5}$$

Both the **multiclass hinge loss** (4) of Crammer & Singer [14] and the **softmax loss** (5) are popular losses for multiclass problems. The latter is also known as the cross-entropy or multiclass logistic loss and is often used as the last layer in deep architectures [5, 21, 40]. The multiclass hinge loss has been shown to be competitive in large-scale image classification [2], however, it is known to be not calibrated [42] for the top-1 error. Next, we show that it is not top-k calibrated for any k.

Proposition 3. *Multiclass SVM is not top-k calibrated.*

Again, a contrast between the hinge and logistic losses.

Proposition 4. *The softmax loss is top-k calibrated.*

The implicit reason for top-k calibration of the OVA schemes and the softmax loss is that one can estimate the probabilities $p_y(x)$ from the Bayes optimal classifier. Loss functions which allow this are called *proper*. We refer to [31] and references therein for a detailed discussion.

We have established that the OVA logistic regression and the softmax loss are top-k calibrated for any k, so why should we be interested in defining new loss functions for the top-k error? The reason is that calibration is an asymptotic property as the Bayes optimal functions are obtained pointwise. The picture changes if we use linear classifiers, since they obviously cannot be minimized independently at each point. Indeed, most of the Bayes optimal classifiers cannot be realized by linear functions.

In particular, convexity of the softmax and multiclass hinge losses leads to phenomena where $\mathrm{err}_k(y, f(x)) = 0$, but $L(y, f(x)) \gg 0$. This happens if $f_{\pi_1}(x) \gg f_y(x) \geq f_{\pi_k}(x)$ and adds a bias when working with "rigid" function classes such as linear ones. The loss functions which we introduce in the following are modifications of the above losses with the goal of alleviating that phenomenon.

2.3. Smooth Top-k Hinge Loss

Recently, we introduced two top-k versions of the multiclass hinge loss (4) in [23], where the second version is based on the family of ranking losses introduced earlier by [44]. We use our notation from [23] for direct comparison and refer to the first version as α and the second one as β. Let $c = \mathbf{1} - e_y$, where $\mathbf{1}$ is the all ones vector, e_y is the y-th basis vector, and let $a \in \mathbb{R}^m$ be defined componentwise as $a_j \triangleq \langle w_j, x\rangle - \langle w_y, x\rangle$. The two **top-$k$ hinge losses** are

$$L(a) = \max\big\{0, \tfrac{1}{k}\textstyle\sum_{j=1}^{k}(a + c)_{\pi_j}\big\}\ \big(\text{top-}k\ \text{SVM}^{\alpha}\big), \tag{6}$$
$$L(a) = \tfrac{1}{k}\textstyle\sum_{j=1}^{k}\max\big\{0, (a + c)_{\pi_j}\big\}\ \big(\text{top-}k\ \text{SVM}^{\beta}\big), \tag{7}$$

where $(a)_{\pi_j}$ is the j-th largest component of a. It was shown in [23] that (6) is a tighter upper bound on the top-k error than (7), however, both losses performed similarly in our experiments. In the following, we simply refer to them as the top-k hinge or the top-k SVM loss.

Both losses reduce to the multiclass hinge loss (4) for $k = 1$. Therefore, they are unlikely to be top-k calibrated, even though we can currently neither prove nor disprove

this for $k > 1$. The multiclass hinge loss is not calibrated as it is non-smooth and does not allow to estimate the class conditional probabilities $p_y(x)$. Our new family of *smooth* top-k hinge losses is based on the Moreau-Yosida regularization [4, 26]. This technique has been used in [38] to smooth the binary hinge loss (2). Interestingly, smooth binary hinge loss fulfills the conditions of Lemma 2 and leads to a top-k calibrated OVA scheme. The hope is that the smooth top-k hinge loss becomes top-k calibrated as well.

Smoothing works by adding a quadratic term to the conjugate function[2], which then becomes strongly convex. Smoothness of the loss, among other things, typically leads to much faster optimization as we discuss in Section 3.

Proposition 5. *OVA smooth hinge is top-k calibrated.*

Next, we introduce the *multiclass* smooth top-k hinge losses, which extend the top-k hinge losses (6) and (7). We define the **top-k simplex** (α and β) of radius r as

$$\Delta_k^\alpha(r) \triangleq \left\{x \,|\, \langle \mathbf{1}, x\rangle \leq r,\ 0 \leq x_i \leq \tfrac{1}{k}\langle \mathbf{1}, x\rangle,\ \forall i\right\}, \quad (8)$$

$$\Delta_k^\beta(r) \triangleq \left\{x \,|\, \langle \mathbf{1}, x\rangle \leq r,\ 0 \leq x_i \leq \tfrac{1}{k} r,\ \forall i\right\}. \quad (9)$$

We also let $\Delta_k^\alpha \triangleq \Delta_k^\alpha(1)$ and $\Delta_k^\beta \triangleq \Delta_k^\beta(1)$.

Smoothing applied to the top-k hinge loss (6) yields the following **smooth top-k hinge loss** (α). Smoothing of (7) is done similarly, but the set $\Delta_k^\alpha(r)$ is replaced with $\Delta_k^\beta(r)$.

Proposition 6. *Let $\gamma > 0$ be the smoothing parameter. The smooth top-k hinge loss (α) and its conjugate are*

$$L_\gamma(a) = \tfrac{1}{\gamma}\left(\langle a + c, p\rangle - \tfrac{1}{2}\langle p, p\rangle\right), \quad (10)$$

$$L_\gamma^*(b) = \tfrac{\gamma}{2}\langle b, b\rangle - \langle c, b\rangle, \text{ if } b \in \Delta_k^\alpha,\ +\infty \text{ o/w}, \quad (11)$$

where $p = \operatorname{proj}_{\Delta_k^\alpha(\gamma)}(a + c)$ is the Euclidean projection of $(a + c)$ on $\Delta_k^\alpha(\gamma)$. Moreover, $L_\gamma(a)$ is $1/\gamma$-smooth.

There is no analytic expression for (10) and evaluation requires computing a projection onto the top-k simplex $\Delta_k^\alpha(\gamma)$, which can be done in $O(m \log m)$ time as shown in [23]. The non-analytic nature of smooth top-k hinge losses currently prevents us from proving their top-k calibration.

2.4. Top-k Entropy Loss

As shown in § 4 on synthetic data, top-1 and top-2 error optimization, when limited to linear classifiers, lead to completely different solutions. The softmax loss, primarily aiming at top-1 performance, produces a solution that is reasonably good in top-1 error, but is far from what can be achieved in top-2 error. That reasoning motivated us to adapt the softmax loss to top-k error optimization. Inspired by the conjugate of the top-k hinge loss, we introduce in this section the top-k entropy loss.

[2] The **convex conjugate** of f is $f^*(x^*) = \sup_x\{\langle x^*, x\rangle - f(x)\}$.

Recall that the conjugate functions of multiclass SVM [14] and the top-k SVM [23] differ only in their effective domain[3] while the conjugate function is the same. Instead of the standard simplex, the conjugate of the top-k hinge loss is defined on a subset, the top-k simplex.

This suggests a way to *construct novel losses* with specific properties by taking the conjugate of an existing loss function, and modifying its essential domain in a way that enforces the desired properties. The motivation for doing so comes from the interpretation of the dual variables as forces with which every training example pushes the decision surface in the direction given by the ground truth label. The absolute value of the dual variables determines the magnitude of these forces and the optimal values are often attained at the boundary of the feasible set (which coincides with the essential domain of the loss). Therefore, by reducing the feasible set we can limit the maximal contribution of a given training example.

We begin with the **conjugate** of the **softmax loss**. Let $a^{\setminus y}$ be obtained by removing the y-th coordinate from a.

Proposition 7. *The convex conjugate of (5) is*

$$L^*(v) = \begin{cases} \sum_{j \neq y} v_j \log v_j + (1 + v_y)\log(1 + v_y), \\ \qquad \text{if } \langle \mathbf{1}, v\rangle = 0 \text{ and } v^{\setminus y} \in \Delta, \\ +\infty \quad \text{otherwise}, \end{cases} \quad (12)$$

where $\Delta \triangleq \left\{x \,|\, \langle \mathbf{1}, x\rangle \leq 1,\ 0 \leq x_j \leq 1,\ \forall j\right\}$.

The **conjugate** of the **top-k entropy loss** is obtained by replacing Δ in (12) with Δ_k^α. A β version could be obtained using the Δ_k^β instead, which defer to future work. There is no closed-form solution for the primal top-k entropy loss for $k > 1$, but we can evaluate it as follows.

Proposition 8. *Let $u_j \triangleq f_j(x) - f_y(x)$ for all $j \in \mathcal{Y}$. The **top-k entropy loss** is defined as*

$$L(u) = \max\Big\{\langle u^{\setminus y}, x\rangle - (1 - s)\log(1 - s) - \langle x, \log x\rangle \,|\, x \in \Delta_k^\alpha,\ \langle \mathbf{1}, x\rangle = s\Big\}. \quad (13)$$

Moreover, we recover the softmax loss (5) if $k = 1$.

We show in the supplement how this problem can be solved efficiently. The non-analytic nature of the loss for $k > 1$ does not allow us to check if it is top-k calibrated.

2.5. Truncated Top-k Entropy Loss

A major limitation of the softmax loss for top-k error optimization is that it cannot ignore the highest scoring predictions, which yields a high loss even if the top-k error is zero. This can be seen by rewriting (5) as

$$L(y, f(x)) = \log\left(1 + \textstyle\sum_{j \neq y} \exp(f_j(x) - f_y(x))\right). \quad (14)$$

[3] The **effective domain** of f is $\operatorname{dom} f = \{x \in X \,|\, f(x) < +\infty\}$.

If there is only a *single* j such that $f_j(x) - f_y(x) \gg 0$, then $L(y, f(x)) \gg 0$ even though $\mathrm{err}_2(y, f(x)) = 0$.

This problem is is also present in all top-k hinge losses considered above and is an inherent limitation due to their convexity. The origin of the problem is the fact that ranking based losses [44] are based on functions such as

$$\phi(f(x)) = \tfrac{1}{m} \textstyle\sum_{j \in \mathcal{Y}} \alpha_j f_{\pi_j}(x) - f_y(x).$$

The function ϕ is convex if the sequence (α_j) is monotonically non-increasing [9]. This implies that convex ranking based losses have to put *more* weight on the highest scoring classifiers, while we would like to put *less* weight on them. To that end, we drop the first $(k-1)$ highest scoring predictions from the sum in (14), sacrificing convexity of the loss, and define the **truncated top-k entropy loss** as follows

$$L(y, f(x)) = \log\left(1 + \textstyle\sum_{j \in \mathcal{J}_y} \exp(f_j(x) - f_y(x))\right), \quad (15)$$

where $\mathcal{J}_y$ are the indexes corresponding to the $(m-k)$ *smallest* components of $(f_j(x))_{j \neq y}$. This loss can be seen as a smooth version of the top-k error (1), as it is small whenever the top-k error is zero. Below, we show that this loss is top-k calibrated.

Proposition 9. *The truncated top-k entropy loss is top-s calibrated for any $k \leq s \leq m$.*

As the loss (15) is nonconvex, we use solutions obtained with the softmax loss (5) as initial points and optimize them further via gradient descent. However, the resulting optimization problem seems to be "mildly nonconvex" as the same-quality solutions are obtained from different initializations. In Section 4, we show a synthetic experiment, where the advantage of discarding the highest scoring classifier in the loss becomes apparent.

3. Optimization Method

In this section, we briefly discuss how the proposed smooth top-k hinge losses and the top-k entropy loss can be optimized efficiently within the SDCA framework of [38]. Further implementation details are given in the supplement.

The primal and dual problems. Let $X \in \mathbb{R}^{d \times n}$ be the matrix of training examples $x_i \in \mathbb{R}^d$, $K = X^\top X$ the corresponding Gram matrix, $W \in \mathbb{R}^{d \times m}$ the matrix of primal variables, $A \in \mathbb{R}^{m \times n}$ the matrix of dual variables, and $\lambda > 0$ the regularization parameter. The primal and Fenchel dual [7] objective functions are given as

$$\begin{aligned} P(W) &= +\frac{1}{n}\sum_{i=1}^{n} L\left(y_i, W^\top x_i\right) + \frac{\lambda}{2}\operatorname{tr}\left(W^\top W\right), \\ D(A) &= -\frac{1}{n}\sum_{i=1}^{n} L^*\left(y_i, -\lambda n a_i\right) - \frac{\lambda}{2}\operatorname{tr}\left(AKA^\top\right), \end{aligned} \quad (16)$$

where L^* is the convex conjugate of L. SDCA proceeds by randomly picking a variable a_i (which in our case is a vector of dual variables over all m classes for a sample x_i) and modifying it to achieve maximal increase in the dual objective $D(A)$. It turns out that this update step is equivalent to a proximal problem, which can be seen as a regularized projection onto the essential domain of L^*.

The update step for top-k SVM$^\alpha_\gamma$**.** Let $a^{\setminus y}$ be obtained by removing the y-th coordinate from vector a. We show that performing an update step for the smooth top-k hinge loss is equivalent to projecting a certain vector b, computed from the prediction scores $W^\top x_i$, onto the essential domain of L^*, the top-k simplex, with an added regularization $\rho \langle \mathbf{1}, x \rangle^2$, which biases the solution to be orthogonal to $\mathbf{1}$.

Proposition 10. *Let L and L^* in (16) be respectively the* top-k SVM$^\alpha_\gamma$ *loss and its conjugate as in Proposition 6. The update $\max_{a_i}\{D(A) \mid \langle \mathbf{1}, a_i \rangle = 0\}$ is equivalent with the change of variables $x \leftrightarrow -a_i^{\setminus y_i}$ to solving*

$$\min_x \{\|x - b\|^2 + \rho \langle \mathbf{1}, x \rangle^2 \mid x \in \Delta^\alpha_k(\tfrac{1}{\lambda n})\}, \quad (17)$$

where $b = \frac{1}{\langle x_i, x_i \rangle + \gamma n \lambda}\left(q^{\setminus y_i} + (1 - q_{y_i})\mathbf{1}\right)$, $q = W^\top x_i - \langle x_i, x_i \rangle a_i$, *and* $\rho = \frac{\langle x_i, x_i \rangle}{\langle x_i, x_i \rangle + \gamma n \lambda}$.

Note that setting $\gamma = 0$, we recover the update step for the non-smooth top-k hinge loss [23]. It turns out that we can employ their projection procedure for solving (17) with only a minor modification of b and ρ.

The update step for the top-k SVM$^\beta_\gamma$ loss is derived similarly using the set Δ^β_k in (17) instead of Δ^α_k. The resulting projection problem is a biased continuous quadratic knapsack problem, which is discussed in the supplement of [23].

Smooth top-k hinge losses converge significantly faster than their nonsmooth variants as we show in the scaling experiments below. This can be explained by the theoretical results of [38] on the convergence rate of SDCA. They also had similar observations for the smoothed binary hinge loss.

The update step for top-k Ent**.** We now discuss the optimization of the proposed top-k entropy loss in the SDCA framework. Note that the top-k entropy loss reduces to the softmax loss for $k = 1$. Thus, our SDCA approach can be used for *gradient-free* optimization of the softmax loss without having to tune step sizes or learning rates.

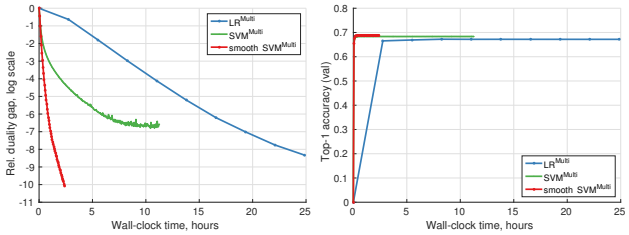
Proposition 11. *Let L in (16) be the* top-k Ent *loss (13) and L^* be its convex conjugate as in (12) with Δ replaced by Δ^α_k. The update $\max_{a_i}\{D(A) \mid \langle \mathbf{1}, a_i \rangle = 0\}$ is equivalent with the change of variables $x \leftrightarrow -\lambda n a_i^{\setminus y_i}$ to solving*

$$\begin{aligned} \min_{x \in \Delta^\alpha_k} \; & \tfrac{\alpha}{2}(\langle x, x \rangle + \langle \mathbf{1}, x \rangle^2) - \langle b, x \rangle + \\ & \langle x, \log x \rangle + (1 - \langle \mathbf{1}, x \rangle)\log(1 - \langle \mathbf{1}, x \rangle) \end{aligned} \quad (18)$$

where $\alpha = \frac{\langle x_i, x_i \rangle}{\lambda n}$, $b = q^{\setminus y_i} - q_{y_i}\mathbf{1}$, $q = W^\top x_i - \langle x_i, x_i \rangle a_i$.

Note that this optimization problem is similar to (17), but is more difficult to solve due to the presence of logarithms in the objective. We propose to tackle this problem using the Lambert W function introduced below.

Lambert W function. The Lambert W function is defined to be the inverse of the function $w \mapsto we^w$ and is widely used in many fields [13, 17, 45]. Taking logarithms on both sides of the defining equation $z = We^W$, we obtain $\log z = W(z) + \log W(z)$. Therefore, if we are given an equation of the form $x + \log x = t$ for some $t \in \mathbb{R}$, we can directly "solve" it in closed-form as $x = W(e^t)$. The crux of the problem is that the function $V(t) \triangleq W(e^t)$ is transcendental [17] just like the logarithm and the exponent. There exist highly optimized implementations for the latter and we argue that the same can be done for the Lambert W function. In fact, there is already some work on this topic [17, 45], which we also employ in our implementation.



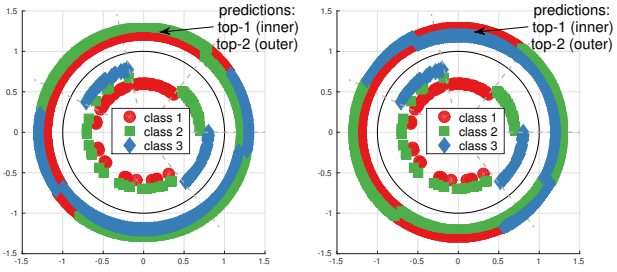
(a) Relative duality gap vs. time (b) Top-1 accuracy vs. time

Figure 1: SDCA convergence with $\mathrm{LR}^{\mathrm{Multi}}$, $\mathrm{SVM}^{\mathrm{Multi}}$, and top-1 SVM^{α}_1 objectives on ILSVRC 2012.

Runtime. We compare the wall-clock runtime of the top-1 multiclass SVM [23] ($\mathrm{SVM}^{\mathrm{Multi}}$) with our smooth multiclass SVM (smooth $\mathrm{SVM}^{\mathrm{Multi}}$) and the softmax loss ($\mathrm{LR}^{\mathrm{Multi}}$) objectives in Figure 1. We plot the relative duality gap $(P(W) - D(A))/P(W)$ and the validation accuracy versus time for the best performing models on ILSVRC 2012. We obtain substantial improvement of the convergence rate for smooth top-1 SVM compared to the non-smooth baseline. Moreover, top-1 accuracy saturates after a few passes over the training data, which justifies the use of a fairly loose stopping criterion (we used 10^{-3}). For $\mathrm{LR}^{\mathrm{Multi}}$, the cost of each epoch is significantly higher compared to the top-1 SVMs, which is due to the difficulty of solving (18). This suggests that one can use the smooth top-1 SVM^{α}_1 and obtain competitive performance (see § 5) at a lower training cost.

Gradient-based optimization. Finally, we note that the proposed smooth top-k hinge and the truncated top-k entropy losses are easily amenable to gradient-based optimization, in particular, for training deep architectures. The computation of the gradient of (15) is straightforward, while for the smooth top-k hinge loss (10) we have [7, § 3, Ex. 12.d]

$$\nabla L_\gamma(a) = \tfrac{1}{\gamma} \operatorname{proj}_{\Delta^\alpha_k(\gamma)}(a + c).$$



(a) top-1 SVM_1 test accuracy (top-1 / top-2): 65.7% / 81.3%

(b) top-2 $\mathrm{Ent}_{\mathrm{tr}}$ test accuracy (top-1 / top-2): 29.4%, 96.1%

Figure 2: Synthetic data on the unit circle in $\mathbb{R}^2$ (inside black circle) and visualization of top-1 and top-2 predictions (outside black circle). **(a)** Smooth top-1 SVM_1 optimizes top-1 error which impedes its top-2 error. **(b)** Trunc. top-2 entropy loss ignores top-1 scores and optimizes directly top-2 errors leading to a much better top-2 result.

4. Synthetic Example

In this section, we demonstrate in a synthetic experiment that our proposed top-2 losses outperform the top-1 losses when one aims at optimal top-2 performance. The dataset with three classes is shown in the inner circle of Figure 2. The description of the distribution from which we sample can be found in the supplement. Samples of different classes are plotted next to each other for better visibility as there is significant class overlap. We visualize top-1/2 predictions with two colored circles (outside the black circle). We sample 200/200/200K points for training/validation/testing and tune the $C = \frac{1}{\lambda n}$ parameter in the range 2^{-18} to 2^{18}. Results are in Table 2.

Circle (synthetic)

Method	Top-1	Top-2	Method	Top-1	Top-2
$\mathrm{SVM}^{\mathrm{OVA}}$	54.3	85.8	top-1 SVM_1	**65.7**	83.9
$\mathrm{LR}^{\mathrm{OVA}}$	54.7	81.7	top-2 $\mathrm{SVM}_{0/1}$	54.4 / 54.5	87.1 / 87.0
$\mathrm{SVM}^{\mathrm{Multi}}$	58.9	89.3	top-2 Ent	54.6	87.6
$\mathrm{LR}^{\mathrm{Multi}}$	54.7	81.7	top-2 $\mathrm{Ent}_{\mathrm{tr}}$	58.4	**96.1**

Table 2: Top-k accuracy (%) on synthetic data. **Left:** Baselines methods. **Right:** Top-k SVM (nonsmooth / smooth) and top-k softmax losses (convex and nonconvex).

In each column we provide the results for the model that optimizes the corresponding top-k accuracy, which is in general different for top-1 and top-2. First, we note that all top-1 baselines perform similar in top-1 performance, except for $\mathrm{SVM}^{\mathrm{Multi}}$ and top-1 SVM_1 which show better results. Next, we see that our top-2 losses improve the top-2 accuracy and the improvement is most significant for the nonconvex top-2 $\mathrm{Ent}_{\mathrm{tr}}$ loss, which is close to the optimal solution for this dataset. This is because top-2 $\mathrm{Ent}_{\mathrm{tr}}$ is a tight bound on the top-2 error and ignores top-1 errors in the loss. Unfortunately, similar significant improvements were not observed on the real-world data sets that we tried.

	ALOI				**Letter**				**News 20**				**Caltech 101 Silhouettes**			
State-of-the-art	93 ± 1.2 [34]				97.98 [19] (RBF kernel)				86.9 [32]				62.1	79.6	83.4	[41]
Method	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10
SVM^{OVA}	82.4	89.5	91.5	93.7	63.0	82.0	88.1	94.6	84.3	95.4	97.9	**99.5**	61.8	76.5	80.8	86.6
LR^{OVA}	86.1	93.0	94.8	96.6	68.1	86.1	90.6	96.2	84.9	96.3	97.8	99.3	63.2	80.4	84.4	89.4
$\text{SVM}^{\text{Multi}}$	90.0	95.1	96.7	98.1	76.5	89.2	93.1	97.7	85.4	94.9	97.2	99.1	62.8	77.8	82.0	86.9
LR^{Multi}	89.8	95.7	97.1	98.4	75.3	90.3	94.3	98.0	84.5	96.4	98.1	**99.5**	63.2	**81.2**	85.1	89.7
top-3 SVM	89.2	95.5	97.2	98.4	74.0	91.0	94.4	97.8	85.1	96.6	98.2	99.3	63.4	79.7	83.6	88.3
top-5 SVM	87.3	95.6	97.4	98.6	70.8	**91.5**	95.1	98.4	84.3	96.7	98.4	99.3	63.3	80.0	84.3	88.7
top-10 SVM	85.0	95.5	97.3	**98.7**	61.6	88.9	96.0	99.6	82.7	96.5	98.4	99.3	63.0	80.5	84.6	89.1
top-1 SVM_1	**90.6**	95.5	96.7	98.2	**76.8**	89.9	93.6	97.6	**85.6**	96.3	98.0	99.3	**63.9**	80.3	84.0	89.0
top-3 SVM_1	89.6	95.7	97.3	98.4	74.1	90.9	94.5	97.9	85.1	96.6	98.4	99.4	63.3	80.1	84.0	89.2
top-5 SVM_1	87.6	95.7	**97.5**	98.6	70.8	**91.5**	95.2	98.6	84.5	96.7	98.4	99.4	63.3	80.5	84.5	89.1
top-10 SVM_1	85.2	95.6	97.4	**98.7**	61.7	89.1	95.9	**99.7**	82.9	96.5	98.4	**99.5**	63.1	80.5	84.8	89.1
top-3 Ent	89.0	95.8	97.2	98.4	73.0	90.8	94.9	98.5	84.7	96.6	98.3	99.4	63.3	81.1	85.0	89.9
top-5 Ent	87.9	95.8	97.2	98.4	69.7	90.9	95.1	98.8	84.3	**96.8**	**98.6**	99.4	63.2	80.9	85.2	89.9
top-10 Ent	86.0	95.6	97.3	98.5	65.0	89.7	**96.2**	99.6	82.7	96.4	98.5	99.4	62.5	80.8	**85.4**	90.1
top-3 Ent_{tr}	89.3	**95.9**	97.3	98.5	63.6	91.1	95.6	98.8	83.4	96.4	98.3	99.4	60.7	81.1	85.2	**90.2**
top-5 Ent_{tr}	87.9	95.7	97.3	98.6	50.3	87.7	96.1	99.4	83.2	96.0	98.2	99.4	58.3	79.8	85.2	**90.2**
top-10 Ent_{tr}	85.2	94.8	97.1	98.5	46.5	80.9	93.7	99.6	82.9	95.7	97.9	99.4	51.9	78.4	84.6	**90.2**

	Indoor 67				**CUB**				**Flowers**				**FMD**		
State-of-the-art	82.0 [48]				62.8 [12] / 76.37 [51]				86.8 [30]				77.4 [12] / 82.4 [12]		
Method	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5
SVM^{OVA}	81.9	94.3	96.5	98.0	60.6	77.1	83.4	89.9	82.0	91.7	94.3	96.8	77.4	92.4	96.4
LR^{OVA}	82.0	94.9	97.2	98.7	62.3	80.5	87.4	93.5	82.6	92.2	94.8	97.6	79.6	94.2	**98.2**
$\text{SVM}^{\text{Multi}}$	82.5	**95.4**	97.3	99.1	61.0	79.2	85.7	92.3	82.5	92.2	94.8	96.4	77.6	93.8	97.2
LR^{Multi}	82.4	95.2	**98.0**	99.1	62.3	81.7	87.9	**93.9**	82.9	92.4	95.1	97.8	79.0	94.6	97.8
top-3 SVM	81.6	95.1	97.7	99.0	61.3	80.4	86.3	92.5	81.9	92.2	95.0	96.1	78.8	94.6	97.8
top-5 SVM	79.9	95.0	97.7	99.0	60.9	81.2	87.2	92.9	81.7	92.4	95.1	97.8	78.4	94.4	97.6
top-10 SVM	78.4	95.1	97.4	99.0	59.6	81.3	87.7	93.4	80.5	91.9	95.1	97.7			
top-1 SVM_1	**82.6**	95.2	97.6	99.0	61.9	80.2	86.9	93.1	**83.0**	92.4	95.1	97.6	78.6	93.8	98.0
top-3 SVM_1	81.6	95.1	97.8	99.0	61.9	81.1	86.6	93.2	82.5	92.3	95.2	97.7	79.0	94.4	98.0
top-5 SVM_1	80.4	95.1	97.8	99.1	61.3	81.3	87.4	92.9	82.0	**92.5**	95.1	97.8	79.4	94.4	97.6
top-10 SVM_1	78.3	95.1	97.5	99.0	59.8	81.4	87.8	93.4	80.6	91.9	95.1	97.7			
top-3 Ent	81.4	**95.4**	97.6	**99.2**	**62.5**	81.8	87.9	**93.9**	82.5	92.0	**95.3**	97.8	**79.8**	94.8	98.0
top-5 Ent	80.3	95.0	97.7	99.0	62.0	**81.9**	88.1	93.8	82.1	92.2	95.1	**97.9**	79.4	94.4	98.0
top-10 Ent	79.2	95.1	97.6	99.0	61.2	81.6	**88.2**	93.8	80.9	92.1	95.0	97.7			
top-3 Ent_{tr}	79.8	95.0	97.5	99.1	62.0	81.4	87.6	93.4	82.1	92.2	95.2	97.6	78.4	**95.4**	**98.2**
top-5 Ent_{tr}	76.4	94.3	97.3	99.0	61.4	81.2	87.7	93.7	81.4	92.0	95.0	97.7	77.2	94.0	97.8
top-10 Ent_{tr}	72.6	92.8	97.1	98.9	59.7	80.7	87.2	93.4	77.9	91.1	94.3	97.3			

	SUN 397 (10 splits)				**Places 205** (val)				**ILSVRC 2012** (val)			
State-of-the-art	66.9 [48]				60.6		88.5	[48]	76.3		93.2	[40]
Method	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10	Top-1	Top-3	Top-5	Top-10
$\text{SVM}^{\text{Multi}}$	65.8 ± 0.1	85.1 ± 0.2	90.8 ± 0.1	95.3 ± 0.1	58.4	78.7	84.7	89.9	68.3	82.9	87.0	91.1
LR^{Multi}	**67.5 ± 0.1**	**87.7 ± 0.2**	**92.9 ± 0.1**	**96.8 ± 0.1**	59.0	**80.6**	**87.6**	**94.3**	67.2	83.2	87.7	92.2
top-3 SVM	66.5 ± 0.2	86.5 ± 0.1	91.8 ± 0.1	95.9 ± 0.1	58.6	80.3	87.3	93.3	68.2	84.0	88.1	92.1
top-5 SVM	66.3 ± 0.2	87.0 ± 0.2	92.2 ± 0.2	96.3 ± 0.1	58.4	80.5	87.4	94.0	67.8	**84.1**	88.2	92.4
top-10 SVM	64.8 ± 0.3	87.2 ± 0.2	92.6 ± 0.1	96.6 ± 0.1	58.0	80.4	87.4	**94.3**	67.0	83.8	88.3	**92.6**
top-1 SVM_1	67.4 ± 0.2	86.8 ± 0.1	92.0 ± 0.1	96.1 ± 0.1	**59.2**	80.5	87.3	93.8	**68.7**	83.9	88.0	92.1
top-3 SVM_1	67.0 ± 0.2	87.0 ± 0.1	92.2 ± 0.1	96.2 ± 0.0	58.9	80.5	**87.6**	93.9	68.2	**84.1**	88.2	92.3
top-5 SVM_1	66.5 ± 0.2	87.2 ± 0.1	92.4 ± 0.2	96.3 ± 0.0	58.5	80.5	87.5	94.1	67.9	**84.1**	**88.4**	92.5
top-10 SVM_1	64.9 ± 0.3	87.3 ± 0.2	92.6 ± 0.2	96.6 ± 0.1	58.0	80.4	87.5	**94.3**	67.1	83.8	88.3	**92.6**
top-3 Ent	67.2 ± 0.2	**87.7 ± 0.2**	**92.9 ± 0.1**	**96.8 ± 0.1**	58.7	**80.6**	**87.6**	94.2	66.8	83.1	87.8	92.2
top-5 Ent	66.6 ± 0.3	**87.7 ± 0.2**	**92.9 ± 0.1**	**96.8 ± 0.1**	58.1	80.4	87.4	94.2	66.5	83.0	87.7	92.2
top-10 Ent	65.2 ± 0.3	87.4 ± 0.1	92.8 ± 0.1	**96.8 ± 0.1**	57.0	80.0	87.2	94.1	65.8	82.8	87.6	92.1

Table 3: Top-k accuracy (%) on various datasets. The first line is a reference to the state-of-the-art on each dataset and reports top-1 accuracy except when the numbers are aligned with Top-k. We compare the one-vs-all and multiclass baselines with the top-k SVM^{α} [23] as well as the proposed smooth top-k $\text{SVM}^{\alpha}_{\gamma}$, top-$k$ Ent, and the nonconvex top-k Ent_{tr}.

5. Experimental Results

The goal of this section is to provide an extensive empirical evaluation of the top-k performance of different losses in multiclass classification. To this end, we evaluate the loss functions introduced in § 2 on 11 datasets (500 to 2.4M training examples, 10 to 1000 classes), from various problem domains (vision and non-vision; fine-grained, scene and general object classification). The detailed statistics of the datasets is given in Table 4.

Dataset	m	n	d	Dataset	m	n	d
ALOI [34]	1K	54K	128	Indoor 67 [28]	67	5354	4K
Caltech 101 Sil [41]	101	4100	784	Letter [19]	26	10.5K	16
CUB [47]	202	5994	4K	News 20 [22]	20	15.9K	16K
Flowers [27]	102	2040	4K	Places 205 [52]	205	2.4M	4K
FMD [39]	10	500	4K	SUN 397 [50]	397	19.9K	4K
ILSVRC 2012 [37]	1K	1.3M	4K				

Table 4: Statistics of the datasets used in the experiments (m – # classes, n – # training examples, d – # features).

Please refer to Table 1 for an overview of the methods and our naming convention. Due to space constraints, we only report a limited selection of all the results we obtained. Please refer to the supplement for a complete report. As other ranking based losses did not perform well in [23], we do no further comparison here.

Solvers. We use `LibLinear` [16] for the one-vs-all baselines SVM^{OVA} and LR^{OVA}; and our code from [23] for top-k SVM. We extended the latter to support the smooth top-k SVM_γ and top-k Ent. The multiclass loss baselines $\text{SVM}^{\text{Multi}}$ and LR^{Multi} correspond respectively to top-1 SVM and top-1 Ent. For the nonconvex top-k Ent_{tr}, we use the LR^{Multi} solution as an initial point and perform gradient descent with line search. We cross-validate hyperparameters in the range 10^{-5} to 10^3, extending it when the optimal value is at the boundary.

Features. For ALOI, Letter, and News20 datasets, we use the features provided by the `LibSVM` [11] datasets. For ALOI, we randomly split the data into equally sized training and test sets preserving class distributions. The Letter dataset comes with a separate validation set, which we used for model selection only. For News20, we use PCA to reduce dimensionality of sparse features from 62060 to 15478 preserving all non-singular PCA components[4].

For Caltech101 Silhouettes, we use the features and the train/val/test splits provided by [41].

For CUB, Flowers, FMD, and ILSVRC 2012, we use `MatConvNet` [46] to extract the outputs of the last fully connected layer of the `imagenet-vgg-verydeep-16` model which is pre-trained on ImageNet [15] and achieves state-of-the-art results in image classification [40].

For Indoor 67, SUN 397, and Places 205, we use the `Places205-VGGNet-16` model by [48] which is pre-trained on Places 205 [52] and outperforms the ImageNet pre-trained model on scene classification tasks [48]. Further results can be found in the supplement. In all cases we obtain a similar behavior in terms of the ranking of the considered losses as discussed below.

Discussion. The experimental results are given in Table 3. There are several interesting observations that one can make. While the OVA schemes perform quite similar to the multiclass approaches (logistic OVA vs. softmax, hinge OVA vs. multiclass SVM), which confirms earlier observations in [2, 33], the OVA schemes performed worse on ALOI and Letter. Therefore it seems safe to recommend to use multiclass losses instead of the OVA schemes.

Comparing the softmax vs. multiclass SVM losses, we see that there is no clear winner in top-1 performance, but softmax consistently outperforms multiclass SVM in top-k performance for $k > 1$. This might be due to the strong property of softmax being top-k calibrated for all k. Please note that this trend is uniform across all datasets, in particular, also for the ones where the features are not coming from a convnet. Both the smooth top-k hinge and the top-k entropy losses perform slightly better than softmax if one compares the corresponding top-k errors. However, the good performance of the truncated top-k loss on synthetic data does not transfer to the real world datasets. This might be due to a relatively high dimension of the feature spaces, but requires further investigation. We also report a number of fine-tuning experiments[5] in the supplementary material.

We conclude that a safe choice for multiclass problems seems to be the softmax loss as it yields competitive results in all top-k errors. An interesting alternative is the smooth top-k hinge loss which is faster to train (see Section 3) and achieves competitive performance. If one wants to optimize directly for a top-k error (at the cost of a higher top-1 error), then further improvements are possible using either the smooth top-k SVM or the top-k entropy losses.

6. Conclusion

We have done an extensive experimental study of top-k performance optimization. We observed that the softmax loss and the smooth top-1 hinge loss are competitive across all top-k errors and should be considered the primary candidates in practice. Our new top-k loss functions can further improve these results slightly, especially if one is targeting a particular top-k error as the performance measure. Finally, we would like to highlight our new optimization scheme based on SDCA for the top-k entropy loss which also includes the softmax loss and is of an independent interest.

[4] The top-k SVM solvers that we used were designed for dense inputs.

[5] Code: https://github.com/mlapin/caffe/tree/topk

References

[1] S. Agarwal. The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *SDM*, pages 839–850, 2011. 1

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Good practice in large-scale learning for image classification. *PAMI*, 36(3):507–520, 2014. 1, 3, 8

[3] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe. Convexity, classification and risk bounds. *Journal of the American Statistical Association*, 101:138–156, 2006. 2, 3

[4] A. Beck and M. Teboulle. Smoothing and first order methods, a unified framework. *SIAM Journal on Optimization*, 22:557–580, 2012. 4

[5] Y. Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009. 3

[6] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with LaRank. In *ICML*, pages 89–96, 2007. 1

[7] J. M. Borwein and A. S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Cms Books in Mathematics Series. Springer Verlag, 2000. 5, 6

[8] S. Boyd, C. Cortes, M. Mohri, and A. Radovanovic. Accuracy at the top. In *NIPS*, pages 953–961, 2012. 1

[9] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 5

[10] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, pages 89–96, 2005. 1

[11] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:1–27, 2011. 8

[12] M. Cimpoi, S. Maji, and A. Vedaldi. Deep filter banks for texture recognition and segmentation. In *CVPR*, 2015. 7

[13] R. M. Corless, G. H. Gonnet, D. E. Hare, D. J. Jeffrey, and D. E. Knuth. On the lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, 1996. 6

[14] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *JMLR*, 2:265–292, 2001. 3, 4

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 8

[16] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008. 8

[17] T. Fukushima. Precise and fast computation of Lambert W-functions without transcendental function evaluations. *Journal of Computational and Applied Mathematics*, 244:77 – 89, 2013. 6

[18] M. R. Gupta, S. Bengio, and J. Weston. Training highly multiclass classifiers. *JMLR*, 15:1461–1492, 2014. 1

[19] C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *Neural Networks*, 13(2):415–425, 2002. 7, 8

[20] T. Joachims. A support vector method for multivariate performance measures. In *ICML*, pages 377–384, 2005. 1

[21] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 3

[22] K. Lang. Newsweeder: Learning to filter netnews. In *ICML*, pages 331–339, 1995. 8

[23] M. Lapin, M. Hein, and B. Schiele. Top-k multiclass SVM. In *NIPS*, 2015. 1, 2, 3, 4, 5, 6, 7, 8

[24] N. Li, R. Jin, and Z.-H. Zhou. Top rank optimization in linear time. In *NIPS*, pages 1502–1510, 2014. 1

[25] L. Liu, T. G. Dietterich, N. Li, and Z. Zhou. Transductive optimization of top k precision. *CoRR*, abs/1510.05976, 2015. 1

[26] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005. 4

[27] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, pages 722–729, 2008. 8

[28] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 8

[29] A. Rakotomamonjy. Sparse support vector infinite push. In *ICML*, pages 1335–1342. ACM, 2012. 1

[30] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *CVPRW, DeepVision workshop*, 2014. 7

[31] M. Reid and B. Williamson. Composite binary losses. *JMLR*, 11:2387–2422, 2010. 3

[32] J. D. Rennie. Improving multi-class text classification with naive bayes. Technical report, Massachusetts Institute of Technology, 2001. 7

[33] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *JMLR*, 5:101–141, 2004. 8

[34] A. Rocha and S. Klein Goldenstein. Multiclass from binary: Expanding one-versus-all, one-versus-one and ecoc-based approaches. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(2):289–302, 2014. 7, 8

[35] S. Ross, J. Zhou, Y. Yue, D. Dey, and D. Bagnell. Learning policies for contextual submodular prediction. In *ICML*, pages 1364–1372, 2013. 1

[36] C. Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *JMLR*, 10:2233–2271, 2009. 1

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge, 2014. 1, 8

[38] S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, pages 1–41, 2014. 1, 2, 4, 5

[39] L. Sharan, R. Rosenholtz, and E. Adelson. Material perception: What can you see in a brief glance? *Journal of Vision*, 9(8):784–784, 2009. 8

[40] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3, 7, 8

[41] K. Swersky, B. J. Frey, D. Tarlow, R. S. Zemel, and R. P. Adams. Probabilistic n-choose-k models for classification and ranking. In *NIPS*, pages 3050–3058, 2012. 1, 7, 8

[42] A. Tewari and P. Bartlett. On the consistency of multiclass classification methods. *JMLR*, 8:1007–1025, 2007. 2, 3

[43] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, pages 1453–1484, 2005. 1

[44] N. Usunier, D. Buffoni, and P. Gallinari. Ranking with ordered weighted pairwise classification. In *ICML*, pages 1057–1064, 2009. 1, 3, 5

[45] D. Veberič. Lambert W function for applications in physics. *Computer Physics Communications*, 183(12):2622–2628, 2012. 6

[46] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015. 8

[47] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. Technical report, California Institute of Technology, 2011. 8

[48] L. Wang, S. Guo, W. Huang, and Y. Qiao. Places205-vggnet models for scene recognition. *CoRR*, abs/1508.01667, 2015. 7, 8

[49] J. Weston, S. Bengio, and N. Usunier. Wsabie: scaling up to large vocabulary image annotation. *IJCAI*, pages 2764–2770, 2011. 1

[50] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 8

[51] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based rcnn for fine-grained detection. In *ECCV*, 2014. 7

[52] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 1, 8