Robust Top-k Multiclass SVM for Visual Category Recognition

Xiaojun Chang
Language Technologies Institute
Carnegie Mellon University
cxj273@gmail.com

Yao-Liang Yu
School of Computer Science
University of Waterloo
yaoliang.yu@uwaterloo.ca

Yi Yang*
Centre for Artificial Intelligence
University of Technology Sydney
yi.yang@uts.edu.au

ABSTRACT

Classification problems with a large number of classes inevitably involve *overlapping* or *similar* classes. In such cases it seems reasonable to allow the learning algorithm to make mistakes on similar classes, as long as the true class is still among the top-k (say) predictions. Likewise, in applications such as search engine or ad display, we are allowed to present k predictions at a time and the customer would be satisfied as long as her interested prediction is included. Inspired by the recent work of [15], we propose a very generic, robust multiclass SVM formulation that directly aims at minimizing a *weighted* and *truncated* combination of the *ordered* prediction scores. Our method includes many previous works as special cases. Computationally, using the Jordan decomposition Lemma we show how to rewrite our objective as the difference of two convex functions, based on which we develop an efficient algorithm that allows incorporating many popular regularizers (such as the l_2 and l_1 norms). We conduct extensive experiments on four real large-scale visual category recognition datasets, and obtain very promising performances.

KEYWORDS

Top-k Multiclass SVM, Visual Category Recognition

1 INTRODUCTION

The multiclass classification problem is a fundamental task in the filed of machine learning and computer vision [3, 8, 10, 20, 32]. It plays a central role in many vision applications, *e.g.*, object recognition, image segmentation, and scene classification [21], which can all be reduced to the task of discriminating multiple categories. Multiclass classification is difficult because the classifier needs to distinguish an object from a large number of categories, potentially overlapping and similar to each other [5, 17]. Indeed, even conservative estimates suggest that there are tens of thousands of object classes in the visual world [2]. The multiclass classification problem can be solved by naturally extending the binary classification technique with the 1-vs-all or 1-vs-1 strategy [22]. These include neural networks [25], decision trees [1], and Support Vector Machines [4].

When the number of visual categories becomes large, the visual recognition problem becomes extremely challenging in the presence of *overlapping* or *similar* classes [9]. We illustrate this phenomenon in Figure 1, where Fig. (a) and Fig. (b) have some overlapping classes while Fig. (c) and Fig. (d) have similar classes. One might ask, is it possible, or even expected, for a human to predict correctly on a first attempt?

Figure 1: Video examples from the FCVID dataset [11]. The objects in Figure (a) and Figure (b) have some overlapping, and the objects in Figure (c) and Figure (d) are similar.

Perhaps not. Therefore, for such challenging circumstances, it makes sense to allow the learning algorithm to present k predictions altogether to the user, as long as the true category is among the top-k predictions. This assumption also aligns with many real applications, such as ad display or search engine [12, 13, 18]. Generally, the customer will still be happy as long as her item of interest is included in the top-k candidates [14].

Recently, [15] proposed the top-k multiclass SVM as a direct method to optimize for top-k performance. It strictly generalizes the multiclass SVM based on a tight convex upper bound of the top-k error. The traditional multiclass formulation of [4] aims at separating the correct class with the top-1 confusing class, which can be too stringent for applications with severe class overlapping. In contrast, the top-k extension in [15] gives the algorithm some slack by ignoring the $k-1$ most confusing labels.

The major limitation of the existing top-k extension is its sensitivity to abnormal observations, *i.e.*, outliers, after all its loss is still convex [31]. To overcome this limitation, we propose a very generic, robust multiclass SVM formulation. Its goal is to directly minimize a *weighted* and *truncated* combination of the ordered prediction scores. Particularly, the proposed algorithm allows to "give up" focusing on any training pair that incurs an excessively

KDD'17, , August 13–17, 2017, Halifax, NS, Canada

ACM ISBN 978-1-4503-4887-4/17/08.
https://doi.org/10.1145/3097983.3097991

large loss, namely, outliers. We also show that the algorithm includes many previous multiclass SVMs as special cases. Based on the Jordan decomposition Lemma, we propose an efficient algorithm that allows incorporating many popular regularizers. Lastly, we conduct extensive experiments on several real large-scale visual category recognition datasets. The experiment results confirm the effectiveness of our algorithm.

Paper organization: We first review some related works on multiclass SVM and its recent top-k extension in Section 2. Then we introduce the proposed Robust Top-k Multiclass SVM in Section 3, followed by the detailed computational algorithm in Section 4. Experiments are conducted in Section 5. Finally, Section 6 concludes this paper.

2 PRELIMINARY

In this section we first recall the multiclass SVM of [4] and the recent top-k extension due to [15].

Given a training sample $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ and $y_i \in \mathcal{Y} := \{1, \ldots, c\}$, the multiclass SVM [4] aims at minimizing the following regularized empirical risk:

$$\min_{W=[\mathbf{w}_1,\ldots,\mathbf{w}_c]^\top \in \mathbb{R}^{c\times d}} \sum_{i=1}^{n} \bar{\ell}_i(W) + \lambda \|W\|_{\mathsf{F}}^2, \tag{1}$$

where the loss $\bar{\ell}_i$ on the i-th training example is given as

$$\bar{\ell}_i = \bar{\ell}_i(W) := \max_{j\in\mathcal{Y}} \{\mathbb{1}_{j\neq y_i} + \langle \mathbf{w}_j, \mathbf{x}_i\rangle - \langle \mathbf{w}_{y_i}, \mathbf{x}_i\rangle\}, \tag{2}$$

$\|W\|_{\mathsf{F}}$ is the Frobenius norm, and $\lambda \geq 0$ controls the tradeoff between the model fitting on the training data and the model complexity. Here we use the notation $\mathbb{1}_A$ for the indicator which is 1 iff A is true, and 0 otherwise. The inner product $\langle \mathbf{w}_j, \mathbf{x}_i\rangle$ is the score of the i-th example w.r.t. the j-th class.

To predict the label of a test point $\mathbf{x}$, we use the max-rule[1]:

$$\hat{y} = \operatorname*{argmax}_{j\in\mathcal{Y}} \langle \mathbf{w}_j, \mathbf{x}\rangle. \tag{3}$$

From this prediction rule it is clear that the prediction is correct, *i.e.*, $\hat{y} = y(\mathbf{x}) =: y$, iff the scores satisfy

$$\langle \mathbf{w}_y, \mathbf{x}\rangle > \langle \mathbf{w}_j, \mathbf{x}\rangle, \ \forall j \neq y. \tag{4}$$

Accordingly, the multiclass loss $\bar{\ell}_i$ in Equation (2) is designed to enjoy the following property:

$$\bar{\ell}_i(W) = 0 \iff \langle \mathbf{w}_{y_i}, \mathbf{x}_i\rangle \geq \langle \mathbf{w}_j, \mathbf{x}_i\rangle + \mathbb{1}_{j\neq y_i}, \tag{5}$$

which is clearly a sufficient condition to guarantee Equation (4). Due to homogeneity of the objective function (1), changing the margin parameter 1 here to any positive number $\gamma > 0$ amounts to scaling down W and scaling up λ by γ. For simplicity, we do not consider adding a bias term here.

The multiclass SVM formulation above has been successfully applied to many real applications, however, it is less appropriate when we are allowed to predict *a set of labels* $\hat{Y}$ for a test point $\mathbf{x}$ and the prediction is deemed "correct" iff $y = y(\mathbf{x}) \in \hat{Y}$. Recently, [15] considered the following set prediction rule:

$$\hat{Y} = \operatorname*{argmax}_{Y\subseteq\mathcal{Y}, |Y|=k} \sum_{j\in Y} \langle \mathbf{w}_j, \mathbf{x}\rangle, \tag{6}$$

where the cardinality of $\hat{Y}$ is constrained to be k. Namely, we present the k labels whose scores are among the top-k, and the prediction is considered "correct" iff the scores satisfy[2]:

$$\langle \mathbf{w}_y, \mathbf{x}\rangle > (W_{\setminus y}\mathbf{x})_{[k]}, \tag{7}$$

where we use $a_{[k]}$ to denote the k-th largest entry of the vector $\mathbf{a}$, and the matrix $W_{\setminus y} \in \mathbb{R}^{(c-1)\times d}$ removes the y-th row of W. Accordingly, [15] modified the multiclass SVM loss Equation (2) as follows:

$$\tilde{\ell}_i(W) := \max\{0, (W_{\setminus y_i}\mathbf{x}_i)_{[k]} + 1 - \langle \mathbf{w}_{y_i}, \mathbf{x}_i\rangle\} \tag{8}$$

$$= \max\left\{0, \left((W - \mathbf{1}\mathbf{w}_{y_i}^\top)\mathbf{x}_i + \mathbf{1} - \mathbf{e}_{y_i}\right)_{[k]}\right\}, \tag{9}$$

where $\mathbf{e}_j$ is the j-th canonical basis vector, and $\mathbf{1}$ is the all ones vector. Indeed,

$$\tilde{\ell}_i(W) = 0 \iff \langle \mathbf{w}_{y_i}, \mathbf{x}_i\rangle \geq (W_{\setminus y_i}\mathbf{x}_i)_{[k]} + 1, \tag{10}$$

which is a sufficient condition to guarantee Equation (7). Again, the margin parameter 1 here can be replaced with any positive number.

The above top-k extension due to [15] is indeed useful for the following reasons:

- It strictly generalizes the multiclass SVM: Setting $k = 1$ we recover the original multiclass SVM of [4], which is clear by comparing Equation (2) with Equation (8), Equation (3) with Equation (6), Equation (4) with Equation (7), and Equation (5) with Equation (10).
- In retrieval tasks such as search engine or ad display, we may be allowed to present k predictions altogether to the user, who will remain happy as long as she can easily identify the desired outcome from the k candidates.
- The multiclass formulation of [4] aims at separating the correct class with the top-1 confusing class, which is wasteful in applications (*e.g.* image classification with extremely many labels) where the labels inevitably overlap a lot. In contrast, the top-k extension gives the algorithm some slack by ignoring the $k-1$ most confusing labels (which could well resemble the correct class).

The top-k loss in Equation (8) is unfortunately nonconvex. Instead, [15] proposed the following convex upper bound (up to a rescaling of factor k):

$$\tilde{\ell}_i(W) = \max\left\{0, \sum_{j=1}^{k} \left((W - \mathbf{1}\mathbf{w}_{y_i}^\top)\mathbf{x}_i + \mathbf{1} - \mathbf{e}_{y_i}\right)_{[j]}\right\}. \tag{11}$$

However, this convex loss may still be dominated by the top-1 confusing class. Moreover, it grows unboundedly as $\|W\|_{\mathsf{F}} \to \infty$ in the nonseparable case — an indication of non-robustness w.r.t. outliers [31].

3 ROBUST TOP-K MULTICLASS SVM

In this section we further extend the top-k multiclass SVM in two aspects: We introduce weights on the ordered scores, and we truncate the loss to induce robustness. Due to its generality, our formulation includes the aforementioned multiclass SVM works as special cases.

[1] Ties can be broken in any consistent way.

[2] Note that for the set prediction rule Equation (6), we can assume w.l.o.g. that $k \leq c-1$, for otherwise the problem is trivial.

The major limitation of the convex surrogate Equation (11) and also the original nonconvex top-k loss Equation (8), is their sensitivity to abnormal observations, *i.e.* outliers. This is mostly due to the unboundedness of the corresponding losses (as $\|W\|_{\mathsf{F}} \to \infty$). To overcome this limitation, we propose the following truncated loss that extends the multicategory ψ-loss in [19]:

$$\ell_i(W) := \min\Big\{\max\Big\{0, \sum_{j=1}^{c} \alpha_j s^i_{[j]}\Big\}, \tau\Big\}, \tag{12}$$

where we use the following abbreviation for the scores

$$\mathbf{s}^i = (W - \mathbf{1}\mathbf{w}_{y_i}^\top)\mathbf{x}_i + \mathbf{1} - \mathbf{e}_{y_i}, \tag{13}$$

and $\boldsymbol{\alpha} \in \mathbb{R}^k$ is an arbitrary weight that we choose to combine the *ordered* scores. Here $\tau > 0$ is a hyperparameter that we use to *cap* the loss for any training pair. In particular, it allows the algorithm to "give up" focusing on any training pair that incurs an excessively large loss, namely, outliers. In real large application where the training data are inevitably noisy, it is beneficial to use a small τ to exclude outliers.

It is clear that our formulation includes many previous multiclass SVMs as special cases:

- If $\tau = \infty$, $\alpha_1 = 1, \alpha_2 = \cdots = \alpha_c = 0$, then we recover the multiclass SVM of [4].
- If $\tau = \infty$, $\alpha_k = 1, \alpha_1 = \cdots = \alpha_{k-1} = \alpha_{k+1} = \cdots = \alpha_c = 0$, then we recover the nonconvex top-k multiclass SVM of [15].
- If $\tau = \infty$, $\alpha_1 = \cdots = \alpha_k = 1, \alpha_{k+1} = \cdots = \alpha_c = 0$, then we recover the convex surrogate of [15].
- If $\tau = 2$, $\alpha_1 = 1, \alpha_2 = \cdots = \alpha_c = 0$, then we recover a similar formulation as [19].

We can combine the robust multiclass loss Equation (12) with a regularization function to promote structure, resulting in the composite minimization problem:

$$\min_W \sum_{i=1}^{n} \ell_i(W) + \lambda f(W), \tag{14}$$

where we can choose for instance the $\mathfrak{l}_2^2$-norm $f(W) = \|W\|_{\mathsf{F}}^2$ for generalization or the $\mathfrak{l}_1$-norm $f(W) = \|W\|_1$ for sparsity. In general, our rosbut loss $\ell_i(W)$ in Equation (12) is not convex, due to the truncation by τ and the arbitrariness of the weight α. However, if $\tau = \infty$ and $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_c$ is monotonically decreasing, such as the multiclass SVM of [4] and the convex surrogate of [15], then the formulation Equation (14) becomes a convex minimization problem. In the next section we develop an efficient algorithm for the general nonconvex setting.

4 COMPUTATIONAL ALGORITHM

The main challenge to numerically solve our formulation Equation (14) is that the robust loss $\ell_i(W)$ defined in Equation (12) is both nonconvex and nonsmooth — a setting where not many algorithms are applicable. Here we propose an efficient implementation based on the difference of convex algorithm (DCA) [28]. The main idea is to decompose a nonconvex function, say ℓ, as the difference of two convex functions, *i.e.* $\ell = g - h$, where both g and h are convex. Note that *not* all nonconvex functions can be written this way, but it is the case for our robust loss (as we shall demonstrate soon). The decomposition is not unique either and we will simply pick a convenient one. The next step is to linearize h at the current iterate, say W_t, so that we have the upper bound

$$\ell(W) \leq g(W) - \langle W - W_t, \nabla h(W_t)\rangle - h(W_t), \tag{15}$$

where $\nabla h(W_t)$ is any subgradient of h at W_t and the inequality follows from the convexity of h. Since the upper bound is now a convex function of W, we can apply our favorite convex optimization algorithm to find its minimizer, which will be W_{t+1}, our next iterate. It is easy to prove that the objective value will decrease monotonically, and with more efforts it can be shown that the algorithm converges to a critical point, see [28]. We note that another possibility is to approximate the robust loss ℓ_i with some (nonconvex) differentiable function, and then apply the ordinary gradient descent.

To implement the above DCA procedure, let us first decompose $\ell_i(W)$ into the difference of two convex functions. For this we need the following well-known (Jordan) decomposition:

Lemma 4.1. *For any vector $\alpha \in \mathbb{R}^c$, we have*

$$\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-, \ \textit{where } \forall j = 1, \ldots, c, \tag{16}$$

$$\alpha_j^+ = \sum_{p=1}^{j} \max\{0, \alpha_p - \alpha_{p-1}\}, \tag{17}$$

$$\alpha_j^- = \sum_{p=1}^{j} \min\{0, \alpha_p - \alpha_{p-1}\}, \tag{18}$$

and by convention $\alpha_0 := 0$.

Clearly, $\boldsymbol{\alpha}^-$ is monotonically decreasing and $\boldsymbol{\alpha}^+$ is monotonically increasing. For instance, for the top-k loss:

$$\boldsymbol{\alpha} = [0, \ldots, 0, \underbrace{1}_{k^{\text{th}}}, 0, \ldots, 0] \tag{19}$$

$$\boldsymbol{\alpha}^+ = [0, \ldots, 0, \underbrace{1}_{k^{\text{th}}}, 1, \ldots, 1] \tag{20}$$

$$\boldsymbol{\alpha}^- = [0, \ldots, 0, \underbrace{-1}_{(k+1)^{\text{th}}}, -1, \ldots, -1], \tag{21}$$

while for the convex upper bound of [15], *i.e.* Equation (11), we have

$$\boldsymbol{\alpha} = [1, \ldots, 1, \underbrace{1}_{k^{\text{th}}}, 0, \ldots, 0] \tag{22}$$

$$\boldsymbol{\alpha}^+ = [1, \ldots, 1, \underbrace{1}_{k^{\text{th}}}, 1, \ldots, 1] \tag{23}$$

$$\boldsymbol{\alpha}^- = [0, \ldots, 0, \underbrace{-1}_{(k+1)^{\text{th}}}, -1, \ldots, -1]. \tag{24}$$

Of course, the decomposition is not unique. For instance, we could also choose $\boldsymbol{\alpha}^- = \boldsymbol{\alpha}, \boldsymbol{\alpha}^+ = \mathbf{0}$ for the latter case above.

Now we can proceed to the robust top-k loss in Equation (12). Since

$$\begin{aligned}\min\{\max\{a, 0\}, \tau\} &= \max\{a, 0\} - \max\{a - \tau, 0\} \\ &= \max\{a^+ + a^-, 0\} - \max\{a^+ + a^- - \tau, 0\} \\ &= \max\{a^-, -a^+\} - \max\{a^- - \tau, -a^+\},\end{aligned}$$

where in the last equality we subtract a^+ from both terms. Therefore, we have the following decomposition for the i-th robust loss:

$$\ell_i(W) = \max\left\{\sum_{j=1}^{c}\alpha_j^- s_{[j]}^i,\ \sum_{j=1}^{c}(-\alpha_j^+)s_{[j]}^i\right\} - \max\left\{\sum_{j=1}^{c}\alpha_j^- s_{[j]}^i - \tau,\ \sum_{j=1}^{c}(-\alpha_j^+)s_{[j]}^i\right\} \tag{25}$$

Since both $\boldsymbol{\alpha}^-$ and $-\boldsymbol{\alpha}^+$ are monotonically decreasing, all four sums in Equation (25) are convex functions of W. Since convexity is preserved under taking the maximum, we have successfully decomposed the robust loss $\ell_i(W)$ into the difference of two convex functions.

The next step is to linearize the subtrahend convex function, for which we will need a formula for the subgradient of the convex function $W \mapsto \sum_{j=1}^{c}\beta_j s_{[j]}$, where recall from Equation (13) that $\mathbf{s}$ is a linear function of W and $\boldsymbol{\beta}$ is any monotonically decreasing vector. For this purpose we need the following reformulation:

$$\sum_{j=1}^{c}\beta_j s_{[j]} = \max_P \boldsymbol{\beta}^\top P[(W - \mathbf{1}\mathbf{w}_y^\top)\mathbf{x} + \mathbf{1} - \mathbf{e}_y], \tag{26}$$

where the maximization is over all permutation matrices P. Given W_t, we can find a maximizer P_t easily in (almost) linear time (essentially sorting $\mathbf{s}$), then we can choose the subgradient to be

$$P_t^\top \boldsymbol{\beta}\mathbf{x}^\top - (\boldsymbol{\beta}^\top \mathbf{1})\mathbf{e}_y\mathbf{x}^\top. \tag{27}$$

Note that the second term does not depend on W_t hence can be precomputed once in the beginning. Lastly, let us recall the well-known subdifferential rule for the max-function $\max\{g, h\}$: the subgradient can be simply chosen as the subgradient of g (resp. h) if g (resp. h) is bigger at the evaluated point. To summarize, a subgradient of the subtrahend in Equation (25) at the current iterate W_t is given in Equation (27) where $\boldsymbol{\beta} = \boldsymbol{\alpha}^-$ if the first term inside the max operator is bigger and $\boldsymbol{\beta} = -\boldsymbol{\alpha}^+$ otherwise.

Denote the subgradient above in Equation (27) as H_t. The next step is to minimize the convex upper bound:

$$\min_W \sum_{i=1}^{n}\ell_i(W; W_t) + \lambda f(W), \quad \text{where} \tag{28}$$

$$\ell_i(W; W_t) = \max\left\{\sum_{j=1}^{c}\alpha_j^- s_{[j]}^i,\ \sum_{j=1}^{c}(-\alpha_j^+)s_{[j]}^i\right\} - \langle H_t^i, W\rangle$$

Conveniently, we can evaluate the subgradient of the convex upper bound $\ell_i(W; W_t)$ similarly as before. Note that H_t^i is fixed here. To solve the convex upper bound Equation (28), we use the stochastic forward-backward splitting algorithm of [6]. The algorithm is again iterative, and consists mostly of two steps:

$$W \leftarrow W - \eta\nabla\ell_I(W; W_t) \tag{29}$$

$$W \leftarrow \mathsf{P}_f^{\eta\lambda}(W) := \operatorname*{argmin}_Z \tfrac{1}{2\eta}\|W - Z\|_{\mathsf{F}}^2 + \lambda f(Z), \tag{30}$$

where $\nabla\ell_I(W; W_t)$ is the subgradient evaluated at a (uniformly) randomly chosen sample index I, $\eta > 0$ is a small step size, and $\mathsf{P}_f^{\eta\lambda}(W)$ is the proximity operator of the regularizer f. For the

Algorithm 1: DCA for robust top-k multiclass SVM

```
Initialize W, η, α.
for j = 1, ..., c do
    α_j^+ = α_{j-1}^+ + max{0, α_j − α_{j-1}}
    α_j^- = α_{j-1}^- + min{0, α_j − α_{j-1}}
for t = 1, 2, ... do
    H ← 0
    for i = 1, ..., n do
        s ← (W − 1 w_{y_i}^⊤) x_i + 1 − e_{y_i}
        [s, P] ← sort(s)
        if s^⊤ α^- − τ > s^⊤ (−α^+) then
            β ← α^-
        else
            β ← −α^+
        H ← H + P^⊤ β x_i^⊤ − (β^⊤ 1) e_{y_i} x_i^⊤
    for m = 1, 2, ... do
        randomly draw I from {1, ..., n}
        s ← (W − 1 w_{y_I}^⊤) x_I + 1 − e_{y_I}
        [s, P] ← sort(s)
        if s^⊤ α^- > s^⊤ (−α^+) then
            β ← α^-
        else
            β ← −α^+
        G ← P^⊤ β x_I^⊤ − (β^⊤ 1) e_{y_I} x_I^⊤
        W ← W − η(G − H)
        W ← P_f^{ηλ}(W)
```

l_2^2-norm regularizer we have

$$\mathsf{P}^{\eta\lambda}_{\|\cdot\|_{\mathsf{F}}^2}(W) = \tfrac{1}{1+2\eta\lambda}W \tag{31}$$

while for the l_1-norm regularizer we have

$$\mathsf{P}^{\eta\lambda}_{\|\cdot\|_1}(W) = \mathrm{sign}(W) * \max\{|W| - \eta\lambda, 0\}, \tag{32}$$

where the algebraic operations are component-wise. For the step size η we can either choose a small value by try-and-error, or use the diminishing rule $\eta_m = O(1/\sqrt{m})$. As shown by [6], this iteration will converge to the minimum objective value in expectation (and in high probability).

We summarize the entire procedure in Algorithm 1, where we make one final modification: Instead of computing the entire subgradient of $\ell_I(W; W_t)$, we rearrange the terms as follows:

$$\min_W \sum_i \max\left\{\sum_{j=1}^{c}\alpha_j^- s_{[j]}^i,\ \sum_{j=1}^{c}(-\alpha_j^+)s_{[j]}^i\right\} \tag{33}$$

$$-\langle\sum_i H_t^i, W\rangle + \lambda f(W), \tag{34}$$

where we treat the terms in Equation (34) as the "regularizer". Therefore, in each iteration we need only sample and compute the subgradient of the first max-term in Equation (33), followed by the proximity operator of the sum in Equation (34), for which we can apply Theorem 3 of [30]. This amounts to subtracting the term $\sum_i H_t^i$ instead of the sampled one H_t^I, hence potentially making

more progress in each iteration. It is clear that in each iteration the time and space complexity is (almost) linear in terms of the problem size.

We note that [15] adopted an entirely different algorithm (SDCA of [23]) for their convex upper bound loss Equation (11). However, SDCA optimizes the dual problem and relies on a strongly convex regularizer (such as the l_2^2-norm). Instead, our algorithm here, building on the work of [28] and [6], works for a variety of robust losses (by choosing different weight vector $\boldsymbol{\alpha}$) and any regularizer (as long as its proximity operator is available cheaply, such as the l_1-norm).

5 EXPERIMENTS

In this section, we carry out extensive experiments to validate the performance of the proposed robust top-k multiclass SVM, abbreviated as **rtop-k SVM**.

5.1 Experimental Setup

Datasets: We test on four real visual category recognition datasets.

- Caltech 101 Silhouettes [27]: This dataset was created based on the CalTech 101 image annotations. Each image in the CalTech 101 data set includes a high-quality polygon outline of the primary object in the scene. To create the CalTech 101 Silhouettes data set, each outline is centered and scaled, and then rendered on a *DtimesD* pixel image-plane. It contains 101 classes, each of which has at most 100 training instances. We use features provided by [27].
- MIT Indoor 67 [21]: This is a dataset of 15,620 images over 67 indoor scenes assembled by [21]. The indoor scenes range from specific categories (*e.g.*, dental office) to generic concepts (*e.g.*, mall). We follow their experimental setting in [21] by using 80 images from each class for training and 20 for testing. We extract CNN features of a pre-trained CNN (fc7 layer after ReLU) [24], resulting in 4096-dimensional feature representation.
- UCF 101 [26]: This dataset consists of 13,320 videos with an average length of 6.2 seconds belonging to 101 different action categories. The dataset has 3 standard train/test splits with the training set containing around 9,500 vidoes in each split (the rest are used as testing data). Following the hybrid deep learning framework proposed in [29], we first extract spatial and motion features with two CNNs trained static frames and stacked optical flows respectively. The two types of features are used separately as inputs of the LSTM network for long-term temporal modeling. We apply a regularized fusion network to combine the two features on video level.
- FCVID 239 [11]: The FCVID239 dataset contains 91,223 web videos annotated into 239 categories, covering a wide range of topics like social events (*e.g.*, tailgate part), procedural events (*e.g.*, making cak), objects (*e.g.*, panda), scenes (*e.g.*, beach), *etc.* The "data" category has the largest number of positive videos while "making egg tarts" is the most infrequent category containing only 108 samples. The total duration of FCVID is 4,232 hours with an average video duration of 167 seconds. We use the 4096-dimensional CNN features provided by [11].

Compared Algorithms: To evaluate the performance of the proposed algorithm, we compare with the following alternatives.

- Probabilistic Model for Top-k Classification [27]: The proabilistic model explicitly includes a prior distribution over the number of variables that take on each label. The authors illustrate the utility of the model by exploring applications to top-K classification.
- SVM$^{\text{OVA}}$: The most common technique for multiclass SVMs has been to build the classifier in a one-versus-rest fashion, and to choose the class which classifiers the test datum with greatest margin.
- TopPush [16]: TopPush aims to optimize accuracy at the top that has computational complexity linear in the number of training instances.
- Top-k SVM [15]: The top-k multiclass SVM is a direct method to optimize for top-k performance. The generalization of the well-known multiclass SVM is based on a tight convex upper bound of the top-k error.

5.2 Visual Recognition Experiment

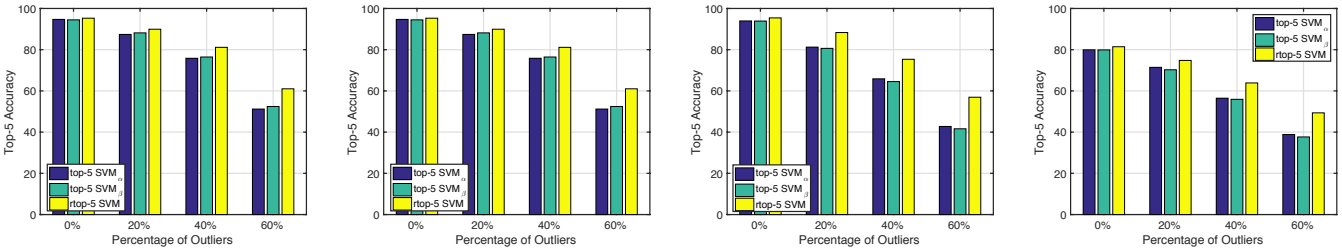
For all the compared algorithms, we cross-validate the regularization parameter λ in the range of 10^{-5} to 10^3, extending it when the optimal value is at the boundary. The oficial splits are used for all the compared algorithms. For the proposed algorithm, we also cross-validate the hyperparameter τ in the range of 10^{-1} to 10^3. The multiclass SVM of [4] is compared as a baseline. We use LibLinear [7] for SVM$^{\text{OVA}}$ and the code provided by [16] for TopPush. For the latter, we use its one-vs-all version for scalability. Two versions of top-k multiclass SVM [15], namely top-k SVM$_\alpha$ and top-k SVM$_\beta$, are also compared. Performance is measured in terms of the top-k accuracy, *i.e.*, the percentage of labels with at least one matching item retrieved within the top-k predictions.

To begin with, we compare all the algorithms with l_2^2-norm regularization. The experimental results on all the used datasets are illustrated in Table 1 and Table 2, from which we make the following observations: First, all alternatives of top-k SVM generally outperform the multiclass SVM of [4], hence demonstrating the benefit of allowing k simultaneous predictions. Second, it is interesting to notice that the alternatives of top-k SVM achieve a decreased top-1 accuracy on MIT Indoor 67 dataset, while achieving a significant improvement in the top-1 accuracy on the other datasets, like UCF 101 and FCVID 239 datasets. This phenomenon is consistent with the intuition that optimizing the top-k accuracy is more appropriate for datasets with a large number of categories, especially when overlapping or similar categories commonly exist. Third, the proposed robust top-k SVM performs better than the other alternatives of top-k SVM. For example, in top-5 accuracy with the proposed rtop-10 SVM, compared with top-10 SVM$_\alpha$: +2.06% on Caltech 101, +1.91% on MIT Indoor 67, +2.39% on UCF 101 and +2.39% on FCVID 239 dataset. This confirms that enhanced robustness does lead to improved performance. Overall, we get systematic increase in top-k accuracy over all the dataset that we used.

To step further, we also compare the performance of top-k SVM$_\alpha$, top-k SVM$_\beta$ and the proposed algorithm, when l_1-norm regularization is used for all the compared alternatives. For space limitation, we only use the Caltech 101 Sihouettes and MIT Indoor 67 datasets in this experiment. The experimental results are reported in Table 3, from which we observe that it is clear the proposed algorithm rtop-k SVM performs the best on both datasets. We attribute this

Table 1: Performance comparison between different multi-class classification algorithms on Caltech 101 Sihouettes and MIT Indoor 67 datasets. Top-k accuracy is used as an evaluation metric. A larger value indicates a better performance.

	Caltech 101 Sihouettes						MIT Indoor 67					
	Top-1 Acc	Top-2 Acc	Top-3 Acc	Top-4 Acc	Top-5 Acc	Top-10 Acc	Top-1 Acc	Top-2 Acc	Top-3 Acc	Top-4 Acc	Top-5 Acc	Top-10 Acc
Top-1 [27]	62.13	73.68	79.59	80.05	83.08	88.65	69.76	78.64	81.64	83.17	84.62	90.83
Top-2 [27]	61.42	72.11	79.23	79.94	83.42	88.93	68.84	77.33	80.87	82.69	83.96	90.14
Top-5 [27]	60.24	70.65	78.71	79.16	83.42	88.93	69.43	78.52	81.29	82.95	84.25	90.47
SVM^{OVA}	61.81	73.13	76.25	77.76	78.89	83.57	71.72	81.49	84.93	86.49	87.39	90.45
TopPush	63.11	75.16	78.46	80.19	81.97	86.95	70.52	83.13	86.94	90.00	91.64	95.90
top-1 SVM_α	62.81	74.60	77.76	80.02	81.97	86.91	73.96	85.22	89.25	91.94	93.43	96.94
top-2 SVM_α	63.11	76.16	79.02	81.01	82.75	87.65	73.06	85.67	90.37	92.24	94.48	97.31
top-3 SVM_α	63.37	76.72	79.67	81.49	83.57	88.25	71.57	86.27	91.12	93.21	94.70	97.24
top-4 SVM_α	63.20	76.64	79.76	82.36	84.05	88.64	71.42	85.67	90.75	93.28	94.78	97.84
top-5 SVM_α	63.29	76.81	80.02	82.75	84.31	88.69	70.67	85.75	90.37	93.21	94.70	97.91
top-10 SVM_α	62.98	77.33	80.49	82.66	84.57	89.55	70.00	85.45	90.00	93.13	94.63	97.76
top-20 SVM_α	59.21	75.64	80.88	83.49	85.39	90.33	65.90	84.10	89.93	92.69	94.25	97.54
top-1 SVM_β	62.81	74.60	77.76	80.02	81.97	86.91	73.96	85.22	89.25	91.94	93.43	96.94
top-2 SVM_β	63.55	76.25	79.28	81.14	82.62	87.91	74.03	85.90	89.78	92.24	94.10	97.31
top-3 SVM_β	63.94	76.64	79.71	81.36	83.44	87.99	72.99	86.34	90.60	92.76	94.40	97.24
top-4 SVM_β	63.94	76.85	80.15	82.01	83.53	88.73	73.06	86.19	90.82	92.69	94.48	97.69
top-5 SVM_β	63.59	77.03	80.36	82.57	84.18	89.03	72.61	85.60	90.75	92.99	94.48	97.61
top-10 SVM_β	64.02	77.11	80.49	83.01	84.87	89.42	71.87	85.30	90.45	93.36	94.40	97.76
top-20 SVM_β	63.37	77.24	81.06	83.31	85.18	90.03	71.94	85.30	90.07	92.46	94.33	97.39
rtop-1 SVM	64.89	77.03	81.18	83.17	84.49	88.49	74.02	85.13	89.96	92.36	94.03	97.69
rtop-2 SVM	64.97	77.65	82.53	83.42	84.73	89.03	73.75	85.58	90.11	92.49	94.06	97.46
rtop-3 SVM	65.23	77.84	82.64	83.29	84.98	89.44	74.12	85.03	89.59	92.28	93.85	96.58
rtop-4 SVM	65.49	78.35	83.12	83.57	84.77	89.97	74.69	85.98	90.03	92.79	94.13	97.39
rtop-5 SVM	65.83	78.89	83.04	83.29	84.53	90.48	73.84	85.86	91.05	92.86	95.16	97.84
rtop-10 SVM	66.21	78.64	83.28	84.02	85.38	91.73	73.28	86.87	91.68	94.74	96.42	98.22
rtop-20 SVM	66.46	77.63	83.21	84.14	84.35	90.21	72.69	85.68	90.28	94.12	94.89	97.85



(a) Caltech 101 Sihouettes **(b) MIT Indoor 67** **(c) UCF101** **(d) FCVID**

Figure 2: Robust evaluation of the proposed algorithm on Caltech 101 Sihouettes, MIT Indoor, UCF 101 and FCVID datasets.

superiority of the proposed algorithm to the novel truncated loss function in Equation (12).

5.3 Robustness Evaluation

To conduct additional experiments to illustrate robustness of the proposed algorithm, we corrupt a varying percentage (0%, 20%, 40%, 60%) of training samples with outliers and compare the performance of top-k SVM and robust top-k SVM (rtop-k SVM). In this section, we use Top-5 accuracy as an evaluation metric and choose $k = 5$ as an example. We report the experimental results in Figure 2. From the experimental results we have the following observations: (1) With the increase of outlier percentages, the performance of all the

Table 2: Performance comparison between different multi-class classification algorithms on UCF 101 and FCVID datasets. Top-k accuracy is used as an evaluation metric. A larger value indicates a better performance.

	UCF101						FCVID					
	Top-1 Acc	Top-2 Acc	Top-3 Acc	Top-4 Acc	Top-5 Acc	Top-10 Acc	Top-1 Acc	Top-2 Acc	Top-3 Acc	Top-4 Acc	Top-5 Acc	Top-10 Acc
Top-1 [27]	70.85	81.57	86.46	87.28	88.35	90.28	56.85	67.57	72.46	73.29	74.35	76.28
Top-2 [27]	70.24	80.83	86.18	86.93	87.85	89.88	56.24	66.84	72.18	72.94	73.85	75.88
Top-5 [27]	70.66	81.16	86.33	87.15	88.04	90.16	56.67	67.17	72.33	73.15	74.05	76.16
SVM^{OVA}	70.94	83.39	87.29	88.42	89.17	91.68	56.95	69.39	73.30	74.43	75.17	77.68
TopPush	72.49	85.16	88.14	88.98	89.85	92.56	58.50	71.16	74.14	74.98	75.86	78.56
top-1 SVM_α	78.83	88.21	91.20	92.89	93.97	96.80	64.84	74.21	77.21	78.89	79.98	82.80
top-2 SVM_α	78.56	88.18	91.28	93.10	93.87	96.78	64.56	74.19	77.28	79.10	79.88	82.78
top-3 SVM_α	77.98	87.87	91.22	93.13	93.95	96.80	63.99	73.88	77.23	79.13	79.95	82.81
top-4 SVM_α	77.58	87.81	91.12	93.02	93.87	96.75	63.59	73.82	77.12	79.03	79.87	82.76
top-5 SVM_α	77.06	87.68	91.09	92.86	93.95	96.85	63.07	73.68	77.09	78.86	79.95	82.85
top-10 SVM_α	75.61	87.18	90.80	92.84	93.58	96.78	61.62	73.19	76.80	78.85	79.59	82.78
top-20 SVM_α	73.83	85.78	89.92	92.33	93.37	96.71	59.84	71.79	75.93	78.33	79.37	82.71
top-1 SVM_β	78.83	88.21	91.20	92.89	93.97	96.80	64.83	74.21	77.20	78.90	79.98	82.81
top-2 SVM_β	79.14	88.78	91.63	93.02	94.18	97.32	65.15	74.78	77.63	79.02	80.18	83.32
top-3 SVM_β	79.37	88.95	91.84	92.96	93.96	97.11	65.37	74.95	77.85	78.97	79.96	83.11
top-4 SVM_β	79.05	88.36	91.38	92.23	93.96	96.86	65.06	74.37	77.39	78.24	79.96	82.87
top-5 SVM_β	77.85	87.95	91.54	92.51	93.89	95.92	63.85	73.95	77.55	78.52	79.89	81.92
top-10 SVM_β	76.53	87.36	91.33	92.87	93.84	95.87	62.53	73.37	77.34	78.87	79.85	81.87
top-20 SVM_β	74.17	86.04	91.28	92.69	93.12	95.46	60.17	72.04	77.28	78.69	79.13	81.46
rtop-1 SVM	78.97	89.59	92.12	94.03	95.16	97.88	65.01	75.58	78.21	80.04	81.12	83.89
rtop-2 SVM	79.23	90.18	92.58	94.16	95.64	98.02	65.21	76.05	78.51	80.12	81.58	84.04
rtop-3 SVM	79.85	90.35	92.29	94.11	95.23	97.61	65.85	76.19	78.27	80.13	81.06	83.52
rtop-4 SVM	79.28	89.59	91.95	93.64	95.02	97.76	65.38	75.69	77.95	79.64	81.02	83.75
rtop-5 SVM	80.17	90.37	92.58	94.25	95.25	97.62	66.21	76.42	78.68	80.29	81.30	83.62
rtop-10 SVM	81.27	91.54	92.87	94.56	95.76	98.32	67.01	77.39	78.89	80.45	81.65	84.42
rtop-20 SVM	80.15	90.27	92.49	94.01	95.51	98.03	65.97	76.12	78.25	89.86	81.25	84.02

compared algorithms dropped. For example, the performance of the propsoed algorithm decreased from 95.28% to 61.02% when the outlier percentage increases from 0% to 60% on FCVID. (2) As the outlier percentage increases, we can see the performance gain of the proposed robust algorithm becomes more significant, which confirms the robustness of the proposed algorithm.

5.4 Sensitivity Analysis

We conduct some sensitivity ayalysis in this section, to draw further insights of the proposed learning algorithm.

Effect of τ and λ: We conduct experiments to assess the sensitivity of the proposed algorithm w.r.t. the hyperparameter τ. To be more specific, we fix $\lambda = 1$ and record the top-k accuracy by varying τ. We use Top-5 accuracy as an evaluation metric and choose $k = 5$ as an example. The experimental results are reported in Figure 3, from which we observe that the performance is relatively robust against the paremter τ. Generally speaking, the best performance is obtained when τ is in the range of $\{10^0, 10^1\}$. Then we fix τ at 1 and test the sensitivity against the regularization parameter λ. The top-5 accuracy with varying λ is shown in Figure 4, from which we see that the performance degrades when λ is overly large. The best performance is obtained when λ is in the range of $\{10^{-3}, 10^{-2}, 10^{-1}\}$.

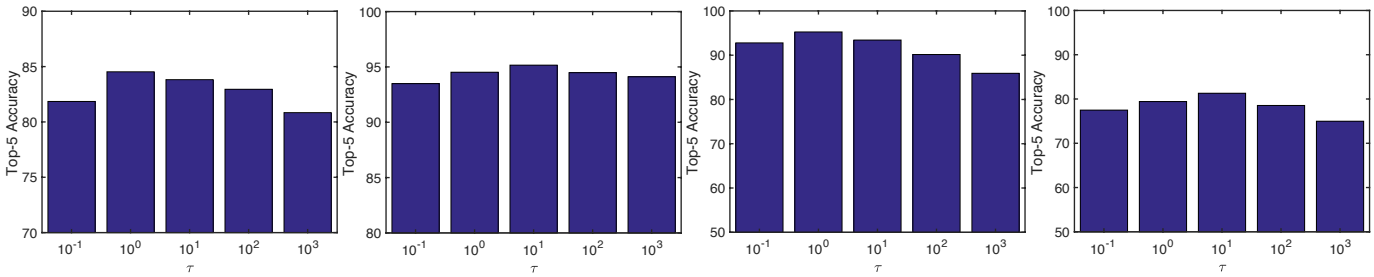
Effect of Initialization: The general setting of Equation (14) is nonconvex, hence in theory it could have multiple local optima. In practisce we observed that the computational algorithm always converged to a reasonable solution. To test this point, we repeatedly run Algorithm 1 20 times, each with a different initialization. The results in terms of top-5 accuracy on the four datasets are reported in Figure 5, which confirmed that the proposed algorithm can always get a promising result on all the datasets.

6 CONCLUSION

We have presented a generic, robust multiclass SVM formulation that directly aims at minimizing a weighted and truncated combination of the ordered prediction scores. Computationally, we have proposed an efficient implementation based on the difference of convex algorithm (DCA). Extensive experiments are conducted on four

Table 3: Performance comparison between top-k SVM$_\alpha$, top-k SVM$_\beta$ and the proposed rtop-k SVM while ℓ_1 regularizer is used. Caltech 101 Sihouettes and MIT Indoor 67 datasets are utilized. Top-k accuracy is used as an evaluation metric. A larger value indicates a better performance.

	Caltech 101 Sihouettes						MIT Indoor 67					
	Top-1 Acc	Top-2 Acc	Top-3 Acc	Top-4 Acc	Top-5 Acc	Top-10 Acc	Top-1 Acc	Top-2 Acc	Top-3 Acc	Top-4 Acc	Top-5 Acc	Top-10 Acc
top-1 SVM$_\alpha$	64.21	76.16	78.85	81.47	83.16	88.22	74.03	85.13	89.42	92.29	93.85	97.27
top-2 SVM$_\alpha$	64.98	77.58	79.89	82.32	83.95	89.14	73.15	85.72	90.08	92.55	94.02	97.17
top-3 SVM$_\alpha$	65.23	78.95	80.33	83.19	84.62	89.83	72.18	85.47	89.96	92.89	94.25	97.13
top-4 SVM$_\alpha$	64.98	77.28	80.79	83.58	85.29	90.32	72.23	85.98	91.05	93.19	94.26	97.17
top-5 SVM$_\alpha$	65.03	78.01	81.57	84.19	86.74	91.59	72.26	85.68	90.84	93.08	94.95	97.66
top-10 SVM$_\alpha$	65.32	78.57	81.92	85.23	87.98	91.23	71.93	86.17	91.14	93.98	95.26	98.05
top-20 SVM$_\alpha$	62.48	76.92	82.59	85.31	86.84	92.42	69.87	84.85	90.19	93.22	94.59	97.67
top-1 SVM$_\beta$	64.28	76.29	79.14	82.57	83.17	88.19	74.08	85.18	89.54	92.17	93.94	97.19
top-2 SVM$_\beta$	64.97	78.15	80.48	84.12	84.09	89.98	73.23	85.81	90.04	92.42	94.01	97.25
top-3 SVM$_\beta$	65.38	78.19	81.59	83.27	85.29	89.48	72.14	86.42	89.95	92.75	94.18	97.08
top-4 SVM$_\beta$	65.84	78.47	82.48	84.22	85.29	90.58	72.09	85.83	90.87	93.11	94.18	97.05
top-5 SVM$_\beta$	65.28	79.42	81.98	84.27	86.28	91.24	72.16	85.59	90.81	93.02	94.86	97.34
top-10 SVM$_\beta$	66.42	79.48	82.86	85.21	86.48	91.16	71.83	85.98	90.79	93.58	94.97	97.65
top-20 SVM$_\beta$	65.84	79.11	82.98	85.19	87.04	91.98	71.89	85.19	90.12	92.25	94.19	97.48
rtop-1 SVM	66.52	78.56	82.64	84.67	85.95	89.97	74.13	85.27	90.01	92.88	94.17	97.76
rtop-2 SVM	66.74	79.14	84.15	84.93	86.27	90.52	73.86	85.74	90.26	92.94	94.17	97.58
rtop-3 SVM	66.81	79.37	84.16	84.76	86.18	90.95	74.28	85.19	89.75	92.53	93.99	96.72
rtop-4 SVM	67.03	79.86	84.63	85.02	86.27	91.48	74.84	86.12	90.19	93.04	94.32	97.61
rtop-5 SVM	67.58	80.14	84.59	84.78	86.08	91.92	74.01	85.97	91.14	93.19	95.28	98.04
rtop-10 SVM	67.96	80.39	84.88	85.49	86.63	93.26	73.47	86.98	91.79	95.01	96.54	98.43
rtop-20 SVM	67.28	79.15	84.62	85.36	86.11	91.84	72.85	85.75	90.43	94.26	95.08	97.92



(a) Caltech 101 Sihouettes **(b) MIT Indoor 67** **(c) UCF101** **(d) FCVID**

Figure 3: Sensitivity analysis on τ on Caltech 101 Sihouettes, MIT Indoor, UCF 101 and FCVID datasets.

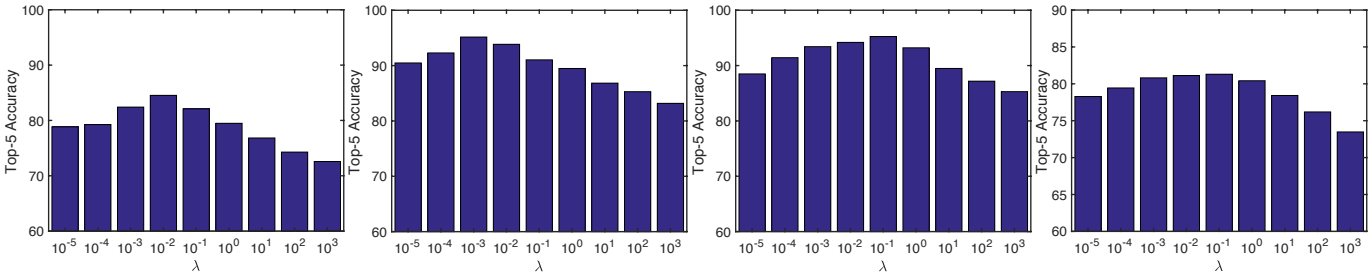
real visual recognition datasets. The experimental results confirm the superiority of the proposed algorithm. In the future, we plan to deploy the proposed robust rtop-k SVM to other applications, *i.e.*, person re-identification problems.

ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments. This work was partially supported by the Data to Decisions Cooperative Research Centre (www.d2dcrc.com.au) and partially supported by NSERC. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Data to Decisions Cooperative Research Centre (www.d2dcrc.com.au) and NSERC.

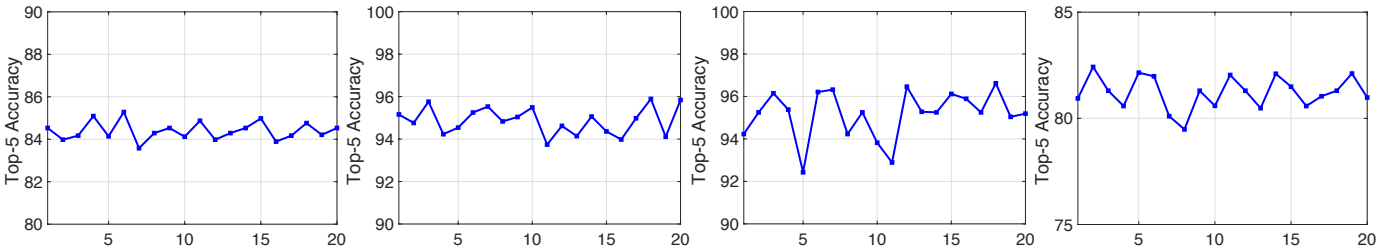
REFERENCES

[1] Samy Bengio, Jason Weston, and David Grangier. 2010. Label Embedding Trees for Large Multi-Class Tasks. In *NIPS*.
[2] Irving Biederman. 1987. Recognition-by-components: a theory of human image understanding. *Psychol Rev* (1987).
[3] Antoine Bordes, Léon Bottou, Patrick Gallinari, and Jason Weston. 2007. Solving multiclass support vector machines with LaRank. In *ICML*.



(a) Caltech 101 Sihouettes (b) MIT Indoor 67 (c) UCF101 (d) FCVID

Figure 4: Sensitivity analysis on λ on Caltech 101 Sihouettes, MIT Indoor, UCF 101 and FCVID datasets.



(a) Caltech 101 Sihouettes (b) MIT Indoor 67 (c) UCF101 (d) FCVID

Figure 5: Sensitivity analysis w.r.t. different initializations on Caltech 101 Sihouettes, MIT Indoor, UCF 101 and FCVID datasets.

[4] Koby Crammer and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research* 2 (2001), 265–292.

[5] Jia Deng, Jonathan Krause, Alexander C. Berg, and Fei-Fei Li. 2012. Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In *CVPR*.

[6] John C. Duchi and Yoram Singer. 2009. Efficient Online and Batch Learning Using Forward Backward Splitting. *Journal of Machine Learning Research* 10 (2009), 2899–2934.

[7] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9 (2008), 1871–1874.

[8] Tianshi Gao and Daphne Koller. 2011. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *ICCV*.

[9] Maya R. Gupta, Samy Bengio, and Jason Weston. 2014. Training highly multiclass classifiers. *Journal of Machine Learning Research* 15, 1 (2014), 1461–1492.

[10] Chih-Wei Hsu and Chih-Jen Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* 13, 2 (2002), 415–425.

[11] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. 2015. Exploiting Feature and Class Relationships in Video Categorization with Regularized Deep Neural Networks. (2015). arXiv:1502.07209.

[12] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *ACM SIGKDD*.

[13] M. Hadi Kiapour, Xufeng Han, Svetlana Lazebnik, Alexander C. Berg, and Tamara L. Berg. 2015. In *ICCV*.

[14] Maksim Lapin, Matthias Hein, and Bernt Schiele. 2015. Loss Functions for Top-k Error: Analysis and Insights. *CoRR* abs/1512.00486 (2015).

[15] Maksim Lapin, Bernt Schiele, and Matthias Hein. 2015. Top-k Multiclass SVM. In *NIPS*.

[16] Nan Li, Rong Jin, and Zhi-Hua Zhou. 2014. Top Rank Optimization in Linear Time. In *NIPS*.

[17] Baoyuan Liu, Fereshteh Sadeghi, Marshall F. Tappen, Ohad Shamir, and Ce Liu. 2013. Probabilistic Label Trees for Efficient Large Scale Image Classification. In *CVPR*.

[18] Li-Ping Liu, Thomas G. Dietterich, Nan Li, and Zhi-Hua Zhou. 2015. Transductive Optimization of Top k Precision. *CoRR* abs/1510.05976 (2015).

[19] Yufeng Liu and Xiaotong Shen. 2006. Multicategory ψ-Learning. *J. Amer. Statist. Assoc.* 101, 474 (2006), 500–509.

[20] John C. Platt, Nello Cristianini, and John Shawe-Taylor. 1999. Large Margin DAGs for Multiclass Classification. In *NIPS*.

[21] Ariadna Quattoni and Antonio Torralba. 2009. Recognizing indoor scenes. In *CVPR*.

[22] Ryan M. Rifkin and Aldebaro Klautau. 2004. In Defense of One-Vs-All Classification. *Journal of Machine Learning Research* 5 (2004), 101–141.

[23] Shai Shalev-Shwartz and Tong Zhang. 2016. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization. *Mathematical Programming* 155, 1 (2016), 105–145.

[24] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

[25] Hyun Oh Song, Stefan Zickler, Tim Althoff, Ross B. Girshick, Mario Fritz, Christopher Geyer, Pedro F. Felzenszwalb, and Trevor Darrell. 2012. Sparselet Models for Efficient Multiclass Object Detection. In *ECCV*.

[26] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *CoRR* abs/1212.0402 (2012).

[27] Kevin Swersky, Daniel Tarlow, Ryan P. Adams, Richard S. Zemel, and Brendan J. Frey. 2012. Probabilistic n-Choose-k Models for Classification and Ranking. In *NIPS*.

[28] Pham Dinh Tao and Le Thi Hoai An. 1998. A D.C. Optimization Algorithm for Solving the Trust-Region Subproblem. *SIAM Journal on Optimization* 8, 2 (1998), 476–505.

[29] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. 2015. Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. In *ACM MM*.

[30] Yaoliang Yu. 2013. On Decomposing the Proximal Map. In *NIPS*.

[31] Yaoliang Yu, Özlem Aslan, and Dale Schuurmans. 2012. A Polynomial-time Form of Robust Regression. In *NIPS*.

[32] Tong Zhang. 2004. Statistical Analysis of Some Multi-Category Large Margin Classification Methods. *Journal of Machine Learning Research* 5 (2004), 1225–1251.