On the design of robust classifiers for computer vision

Hamed Masnadi-Shirazi Vijay Mahadevan Nuno Vasconcelos
Department of Electrical and Computer Engineering
University of California, San Diego
hmasnadi@ucsd.edu, vmahadev@ucsd.edu, nuno@ece.ucsd.edu

Abstract

The design of robust classifiers, which can contend with the noisy and outlier ridden datasets typical of computer vision, is studied. It is argued that such robustness requires loss functions that penalize both large positive and negative margins. The probability elicitation view of classifier design is adopted, and a set of necessary conditions for the design of such losses is identified. These conditions are used to derive a novel robust Bayes-consistent loss, denoted Tangent loss, and an associated boosting algorithm, denoted TangentBoost. Experiments with data from the computer vision problems of scene classification, object tracking, and multiple instance learning show that TangentBoost consistently outperforms previous boosting algorithms.

1. Introduction

Over the last decade, tremendous advances have been achieved in computer vision tasks that can be formulated as classification problems. Examples include object detection [32] and recognition [30], object tracking [2], image classification and retrieval [24, 23], among others. Much of this progress is due to the widespread adoption of classification techniques, such as the support vector machine (SVM) [29], boosting [11], or logistic regression [12], which minimize the expected value of a *margin enforcing* loss. Such losses, see Figure 1 for examples, apply a large penalty to points with large *negative margin* (i.e. incorrectly classified and far from the boundary), some penalty to points of small *positive margin* (correctly classified but close to the boundary), and zero penalty to points of large positive margin (correctly classified and far from the boundary). The assignment of non-zero loss to correct classifications close to the boundary is critical to assuring a classifier of maximal margin. This, in turn, is critical to guarantee good generalization [29].

While the positive impact of large margin classifiers is undisputable, they do not overcome all challenges posed by computer vision. This is due to the prevalence, in most vision applications, of noise, outliers, ambiguity, lack of labels, small training sizes, and imbalance of positive/negative coverage by training sets. For example, patch-based image classification usually involves much more negative than positive examples per class, and is inherently outlier ridden: an image from the *buildings* class invariably contains patches from the *people*, *garden*, or *car* class [17]. Furthermore, patches are inherently ambiguous (e.g. the same circular shape could correspond to a car wheel or a boat window) [24], and "noise" is plentiful (in the form of shadows, occlusions, perspective distortions, etc.). In applications such as tracking, where a classifier is incrementally learned from data (as it is being classified), it is impossible to guarantee that there is no leakage between the sets of positives and negatives used for training [2, 31, 16, 3]. While some of these problems can be mitigated by careful human labeling, human labeled data can itself be error prone. In large-scale problems, where labeling is expensive, there is frequently a need to resort to unlabeled datasets, or labels of low-quality. In some cases, exact labels cannot even be assigned to every sample point, and there is a need to resort to a multiple instance learning (MIL) formalism, where labels only exist for bags of points [17, 7, 22, 34, 1].

Different areas of computer vision have taken varied approaches to dealing with these problems. These include resorting to MIL algorithms for scene classification [17], object detection [31], or tracking [3], modeling context to reduce ambiguity in scene analysis [28], adopting parts-based models of greater flexibility with respect to occlusions and deformation [10, 9], etc. While such improvements in representation robustness are necessary, they cannot completely eliminate the ambiguity, noise, and outlier propensity of tasks such as image classification or tracking. Hence, there is an equally important need for more robust classifiers. In this context, an issue of particular concern is a well known limitation of most current margin-enforcing losses: their *unbounded growth* with negative margins. In statistics, this type of loss growth is classically known to produce inference procedures that are too *sensitive to outliers* [13, 25], a problem that has also been extensively

studied in computer vision [21, 4, 27]. This research has shown that, for many vision applications, better results are obtained with losses of tapered growth. However, most of these results only apply to regression problems, such as surface fitting or optic flow estimation, and do not generalize to classification.

Robust classifier design has been studied in machine learning, namely in the boosting literature. Boosting algorithms, such as AdaBoost [11], have found multiple applications in vision, e.g. real-time object detection [32], tracking [2], and segmentation [33]. Yet, Adaboost is known to be particularly sensitive to noisy data [6], due to the exponential growth of its loss. Non-trivial improvements are due to [12], which introduced losses that grow *linearly* with the negative margin. The resulting boosting algorithms, e.g. LogitBoost, are known to be substantially more outlier resistant than AdaBoost [20]. Central to this contribution was the establishment, by this work, of a formal connection between the large margin approaches and classical decision theory. A number of other attempts to introduce robust classification losses, e.g. the noisy-OR [31] or sigmoidal nonlinearities [19], lack this property. The resulting classifiers are not Bayes consistent, i.e. are not guaranteed to converge to the optimal Bayes decision rule [8] as datasets increase.

The design of Bayes consistent robust classification losses was most recently studied in [18]. This work established a framework for the derivation of novel Bayes consistent loss functions. It also proposed a new robust loss, denoted as *Savage loss* and an associated *SavageBoost* algorithm [18]. This algorithm was shown to outperform AdaBoost and LogitBoost in outlier ridden classification problems. While our experience with the algorithms confirms this observation, the added robustness of SavagaBoost does not make a tremendous difference for all vision problems. We argue that this requires a more subtle constraint on the loss than simply bounding its growth for large negative margins: in addition to this, robustness requires *penalizing large positive margins*.

We present a simple classification problem that demonstrates this point, and show how all existing methods (including SavageBoost) fail in this case. We then derive a set of necessary conditions that any Bayes consistent loss function must satisfy, in order to guarantee a bounded penalty for *both* large negative and positive margins. These conditions are used to derive a novel robust loss, which we denote by *Tangent loss*, and an associated boosting algorithm, denoted *TangentBoost*. Experiments involving various computer vision problems, including scene classification, object tracking, recognition, and MIL show that the proposed algorithm consistently outperforms previous boosting algorithms. In fact, for some of these problems, it is shown to achieve the best results reported to date on the literature.

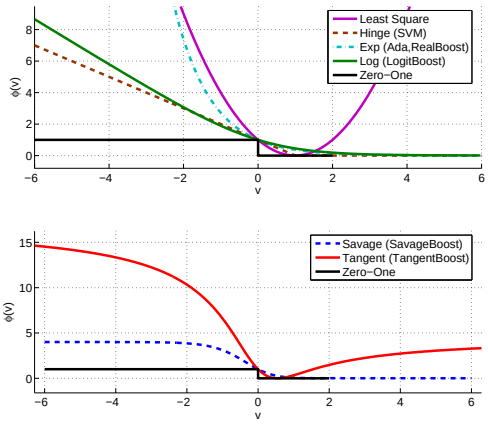

Figure 1. Loss functions used for classifier design in alternative to the non-margin enforcing $0-1$ loss. Top: classical non-robust losses. Bottom: robust losses of SavageBoost and TangentBoost.

2. Loss functions for classification

We start by briefly reviewing the theory of Bayes consistent classifier design. See [12, 5, 35, 18] for further details.

2.1. Risk minimization

A classifier h maps a feature vector $\mathbf{x} \in \mathcal{X}$ to a class label $y \in \{-1, 1\}$. This mapping can be written as $h(\mathbf{x}) = sign[f(\mathbf{x})]$ for some function $f : \mathcal{X} \rightarrow \mathbb{R}$, which is denoted as the classifier predictor. Feature vectors and class labels are drawn from probability distributions $P_{\mathbf{X}}(\mathbf{x})$ and $P_Y(y)$ respectively. Given a non-negative loss function $L(\mathbf{x}, y)$, the classifier is optimal if it minimizes the risk $R(f) = E_{\mathbf{X},Y}[L(h(\mathbf{x}), y)]$. This is equivalent to minimizing the conditional risk $E_{Y|\mathbf{X}}[L(h(\mathbf{x}), y)|\mathbf{X} = \mathbf{x}]$ for all $\mathbf{x} \in \mathcal{X}$. Classifiers are frequently designed to be optimal with respect to the zero-one loss

$$L_{0/1}(f, y) = \frac{1 - sign(yf)}{2} = \begin{cases} 0, & \text{if } y = sign(f); \\ 1, & \text{if } y \neq sign(f), \end{cases} \quad (1)$$

where we omit the dependence of f on $\mathbf{x}$ for notational simplicity. The associated conditional risk is

$$\begin{aligned} C_{0/1}(\eta, f) &= \eta \frac{1 - sign(f)}{2} + (1 - \eta) \frac{1 + sign(f)}{2} \\ &= \begin{cases} 1 - \eta, & \text{if } f \geq 0; \\ \eta, & \text{if } f < 0 \end{cases} \end{aligned}$$

with $\eta(\mathbf{x}) = P_{Y|\mathbf{X}}(1|\mathbf{x})$. Optimal predictors f^* that minimize this risk include $f^* = 2\eta - 1$, $f^* = \log \frac{\eta}{1-\eta}$, or any other function such that $f^* \geq 0$ if and only if $\eta \geq \frac{1}{2}$. The associated optimal classifier $h^* = sign[f^*]$ is the well known Bayes decision rule (BDR) and has minimum con-

Table 1. Loss ϕ, predictor $f^*_\phi(\eta)$, minimum conditional risk $C^*_\phi(\eta)$ and predictor inverse $[f^*_\phi]^{-1}(v)$ for different machine learning algorithms.

Algorithm	$\phi(v)$	$f^*_\phi(\eta)$	$C^*_\phi(\eta)$	$[f^*_\phi]^{-1}(v)$
Least squares	$(1-v)^2$	$2\eta-1$	$4\eta(1-\eta)$	$\frac{1}{2}(v+1)$
SVM	$\max(1-v,0)$	$sign(2\eta-1)$	$1-\|2\eta-1\|$	NA
Boosting	$\exp(-v)$	$\frac{1}{2}\log\frac{\eta}{1-\eta}$	$2\sqrt{\eta(1-\eta)}$	$\frac{e^{2v}}{1+e^{2v}}$
Logistic Regression	$\log(1+e^{-v})$	$\log\frac{\eta}{1-\eta}$	$-\eta\log\eta-(1-\eta)\log(1-\eta)$	$\frac{e^v}{1+e^v}$

ditional risk

$$C^*_{0/1}(\eta) = \eta\left(\frac{1}{2}-\frac{1}{2}sign(2\eta-1)\right)+ (1-\eta)\left(\frac{1}{2}+\frac{1}{2}sign(2\eta-1)\right). \quad (2)$$

A loss which is minimized by the BDR is denoted as Bayes consistent. A number of Bayes consistent alternatives to the 0-1 loss are commonly used in machine learning. These include the exponential loss of boosting, the log loss of logistic regression, and the hinge loss of SVMs, which are shown in the top of Figure 1. They have the form $L_\phi(f,y)=\phi(yf)$, for different functions ϕ of the margin yf. The non-zero penalty assigned to small positive margins encourages the creation of a margin, a property not shared by the 0-1 loss. The resulting *large-margin* classifiers have better generalization than those produced by the latter [29]. The associated conditional risk

$$C_\phi(\eta,f)=\eta\phi(f)+(1-\eta)\phi(-f) \quad (3)$$

is minimized by the predictor

$$f^*_\phi(\eta)=\arg\min_f C_\phi(\eta,f) \quad (4)$$

and has minimum $C^*_\phi(\eta)=C_\phi(\eta,f^*_\phi)$. The $\phi(v)$, $f^*_\phi(\eta)$, and $C^*_\phi(\eta)$ associated with popular algorithms for classifier design are shown in Table 1. See [35] for their derivations.

2.2. Probability elicitation

Conditional risk minimization is closely related to classical probability elicitation in statistics [26]. Here, the goal is to find the probability estimator $\hat\eta$ that maximizes the expected reward

$$I(\eta,\hat\eta)=\eta I_1(\hat\eta)+(1-\eta)I_{-1}(\hat\eta), \quad (5)$$

where $I_1(\hat\eta)$ is the reward for prediction $\hat\eta$ when event $y=1$ holds and $I_{-1}(\hat\eta)$ the corresponding reward when $y=-1$. The functions $I_1(\cdot), I_{-1}(\cdot)$ must be such that the expected reward is maximal when $\hat\eta=\eta$, i.e.

$$I(\eta,\hat\eta)\le I(\eta,\eta)=J(\eta),\ \ \forall\eta \quad (6)$$

with equality if and only if $\hat\eta=\eta$. It can be shown [26] that (6) holds if and only if 1) the maximal reward function $J(\eta)$ is strictly convex and 2)

$$I_1(\eta) = J(\eta)+(1-\eta)J'(\eta) \quad (7)$$
$$I_{-1}(\eta) = J(\eta)-\eta J'(\eta). \quad (8)$$

The connection between risk minimization and probability elicitation has been studied in [18]. This work has shown that if 1) $J(\eta)=J(1-\eta)$, and 2) the predictor f is invertible and has symmetry $f^{-1}(-v)=1-f^{-1}(v)$, the functions $I_1(\cdot)$ and $I_{-1}(\cdot)$ of (7) and (8) satisfy the following equalities

$$I_1(\eta) = -\phi(f(\eta)) \quad (9)$$
$$I_{-1}(\eta) = -\phi(-f(\eta)), \quad (10)$$

for the loss

$$\phi(v)=-J[f^{-1}(v)]-(1-f^{-1}(v))J'[f^{-1}(v)]. \quad (11)$$

In this case, probability elicitation by maximization of (5) is equivalent to risk minimization with (3), and the minimum conditional risk is related to the maximal expected reward through $C^*_\phi(\eta)=-J(\eta)$. This establishes a new path for the design of learning algorithms. Rather than specifying a loss ϕ and minimizing $C_\phi(\eta,f)$, so as to obtain whatever optimal predictor f^*_ϕ and minimum expected risk $C^*_\phi(\eta)$ results, it is possible to specify f^*_ϕ and $C^*_\phi(\eta)$ and derive, from (11) with $J(\eta)=-C^*_\phi(\eta)$, the underlying loss ϕ. The only conditions are that $C^*_\phi(\eta)$ is strictly concave, f^*_ϕ is invertible, and

$$C^*_\phi(\eta) = C^*_\phi(1-\eta) \quad (12)$$
$$[f^*_\phi]^{-1}(-v) = 1-[f^*_\phi](v). \quad (13)$$

3. Robust loss functions for computer vision

Computer vision problems frequently deviate from the canonical classification problem, due to the prevalence of noise, outliers, ambiguity, and imbalance of positive/negative training set sizes, in many vision applications. In this context, the losses shown at the top of Figure 1 are problematic in two ways. The first is their unbounded growth with negative values of the margin yf. This type of growth is well known to produce inference procedures that are too sensitive to outliers [13, 25]. For vision applications, better results are invariably obtained with loss functions of tapered growth [21, 4]. The second is the null

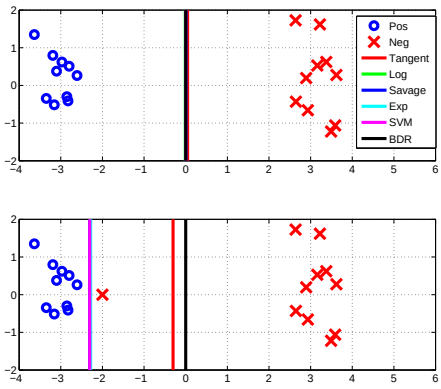

Figure 2. Minimum risk decision boundary for different loss functions. Top: outlier free problem. Bottom: impact of a single outlier.

penalty assigned to very large positive margins. This creates an incentive for the classifier to push, as far as possible from the boundary, the maximum possible number of points. Although less studied than the first problem, we contend that this can have an equally nefarious effect in terms of sensitivity to outliers.

We illustrate this point in Figure 2. The figure depicts the linearly separable problem that motivates the design of large-margin classifiers. The data come from two distributions that are uniform in the vertical direction and Gaussian, with equal variance and means $\mu = \pm 3$, in the horizontal direction. Given these distributions, the BDR is the vertical line $x = 0$. Figure-2 (top) shows ten data points sampled from each class and the decision boundary resulting from the minimization of the (empirical) risk associated with each loss. All losses of Figure 1 produce approximately the same boundary, close to the BDR.

Figure-2 (bottom) shows the impact of adding a single negative at location $(-2, 0)$. Both the classical losses and the robust Savage loss move the boundary substantially, to the vicinity of $x = -2.3$. This is due to the fact that this boundary classifies all points correctly, and the existing losses assign small penalty to correctly classified points. The result in as unwarranted leverage on the boundary by the outlier at $(-2, 0)$, compromising the generalization ability of the classifier. Also shown in the figure is the boundary produced by the loss (the tangent loss) proposed in this work. This loss, which is derived in the following sections, penalizes *both* large positive and large negative margins. The penalty assigned to large positive margins discourages solutions where large numbers of points are classified "too correctly". The force to classify the outlier correctly is countered by the force to avoid large numbers of points far away from the boundary. In result, the boundary remains close to the BDR ($x = -0.303$).

3.1. Robust losses

The discussion above suggests that a robust loss for classifier design should have the following properties:

1. saturate for large margins: $\phi'(\infty) = \phi'(-\infty) = 0$;
2. bounded penalty for large negative margins: $\phi(-\infty) = k_1 < \infty$;
3. smaller positive penalty for large positive margins: $0 < \phi(\infty) = k_2 < k_1$;
4. margin enforcing: $\phi(0) > 0$

where we use the simplified notation $\phi(\infty) = \lim_{v\to\infty} \phi(v)$. As usual, the loss should be non-negative.

It can be shown, from (11), that

$$\phi'(v) = -[1 - f^{-1}(v)] \times J''[f^{-1}(v)] \times [f^{-1}]'(v) \quad (14)$$

From the strict convexity of $J(\eta)$, and (13), it follows that property 1 holds if

$$[f^{-1}]'(\infty) = [f^{-1}]'(-\infty) = 0. \quad (15)$$

This implies that the optimal predictor saturates as $v \to \pm\infty$. Furthermore, using the fact that $J(\eta) = J(1-\eta)$, $J'(\eta) = -J'(1-\eta)$, and (13),

$$\begin{aligned} \phi(v) - \phi(-v) &= -J'[f^{-1}(v)] \\ (\phi(v) - \phi(-v))' &= -J''[f^{-1}(v)] \times [f^{-1}(v)]'. \end{aligned}$$

It follows from (15) that $|\phi(v) - \phi(-v)|$ is maximum as $|v| \to \infty$. The condition $k_2 < k_1$ requires that $J'\left[f^{-1}(\infty)\right] > 0$. From the convexity and symmetry of $J(\eta)$ ($J'(1/2) = 0$) this holds whenever

$$f^{-1}(\infty) > \frac{1}{2}.$$

Defining $\gamma(v) = f^{-1}(-v) \times J'[f^{-1}(-v)]$, $k_2 > 0$ requires that $-J[f^{-1}(\infty)] > -\gamma(\infty)$, or $0 < C_\phi^*[f^{-1}(\infty)] + \gamma(\infty)$. Similarly, $k_1 < \infty$ requires that $C_\phi^*[f^{-1}(\infty)] + \gamma(-\infty) < \infty$. Finally, from (13), $f^{-1}(0) = \frac{1}{2}$ and, from (11) and $J'(1/2) = 0$, it follows that $\phi(0) = -J(1/2) = C_\phi^*(1/2) > 0$. In summary, the four properties are satisfied if

$$\begin{aligned} [f^{-1}]'(\infty) &= [f^{-1}]'(-\infty) = 0 && (16) \\ f^{-1}(\infty) &> \frac{1}{2} = f^{-1}(0) && (17) \\ C_\phi^*(1/2) &> 0 && (18) \\ C_\phi^*[f^{-1}(\infty)] &+ \gamma(\infty) > 0 && (19) \\ C_\phi^*[f^{-1}(\infty)] &+ \gamma(-\infty) < \infty && (20) \end{aligned}$$

3.2. The Tangent loss

In this section we seek to design a loss with the four properties discussed above, through the selection of a predictor $f^*_\phi(\eta)$ and minimum risk $C^*_\phi(\eta)$ that comply with conditions (16)-(20). We start by noting that some of these conditions hold for any sensible choice of these functions. For example, (17) and (18) are met by all methods of Table 1. On the other hand, (16) disqualifies the predictor of least squares, but leaves the sigmoidal predictors of boosting and logistic regression as potential solutions. This suggests that conditions (19) and (20) are the most stringent. In fact, they fail to hold for all methods of Table 1.

Consider any of the sigmoidal predictors. Since $f^{-1}(\infty) = 1$, for any of the C^*_ϕ in the table, $C^*_\phi[f^{-1}(\infty)] = 0$. This simplifies (19) and (20) into

$$\gamma(\infty) = -f^{-1}(-\infty) \times [C^*_\phi]'[f^{-1}(-\infty)] > 0 \quad (21)$$
$$\gamma(-\infty) = -f^{-1}(\infty) \times [C^*_\phi]'[f^{-1}(\infty)] < \infty. \quad (22)$$

Since $f^{-1}(-\infty) = 0$, (21) requires $[C^*_\phi]'(0) = -\infty$. In fact, because the sigmoid converges to 0 *exponentially* fast, (21) requires the derivative of $[C^*_\phi](\eta)$ to decay to $-\infty$ (as $\eta \to 0$) at a (faster) exponential rate. This is not easy to guarantee, and does certainly not hold for any of the risks of Table 1. In summary, it appears that none of the predictors in the table is suitable for robust loss design. What is needed is a predictor such that $f^{-1}(v)$ saturates at $\pm\infty$, so as to satisfy (16), but at a *slower than exponential* rate.

One possibility is the tangent

$$f(\eta) = \tan(\eta - 0.5) \quad (23)$$
$$f^{-1}(v) = .5 + \arctan(v). \quad (24)$$

It has the symmetry of (13), a *quadratic* decay rate ($[f^{-1}]'(v) = (1 + x^2)^{-1}$) and is compatible for combination with the minimal conditional risk of least squares, $C^*_\phi(\eta) = 4\eta(1 - \eta)$, resulting in

$$C^*_\phi[f^{-1}(\infty)] + \gamma(\infty) = (1 - \pi)^2 > 0$$
$$C^*_\phi[f^{-1}(\infty)] + \gamma(-\infty) = (1 + \pi)^2 < \infty.$$

It can be easily verified that conditions (16)-(18) also hold. Using (11) it is possible to derive the ϕ function, which we denote by *Tangent loss*,

$$\phi(v) = (2\arctan(v) - 1)^2. \quad (25)$$

Figure-1 (bottom) shows that the Tangent loss is similar to the Savage loss in the sense that it is non convex, and bounded for large negative margins. It, however, also penalizes points of large *positive* margin. This penalty is, once again, bounded and of smaller value than that assigned to large negative margins. Overall, the tangent loss is margin enforcing, and encourages all points to be classified correctly. However, it discourages situations where a large number of points are classified "too correctly". We will see, in Section 5, that this leads to superior performance for a number of vision problems.

4. The TangentBoost algorithm

In this section we derive a boosting algorithm based on the Tangent loss. This consists of minimizing the empirical risk

$$R = \sum_i \phi(yf(x)) \quad (26)$$

by gradient descent on the space of linear combinations of weak learners. The fact that this is a sum of squared values, suggests performing the minimization with the Gauss algorithm. For a general sum of squares problem

$$S(x) = \sum_{i=1}^{N} r_i^2(x) \quad (27)$$

this has update step

$$x^{n+1} = x^n + \frac{-r(x)}{\frac{\partial r}{\partial x}} \quad (28)$$

As in the case of LogitBoost [12], it is more convenient to work with the intermediate probability estimates $\eta(x_i)$ than the points x_i. For the Tangent loss

$$r(\eta) = 2\arctan(yf(\eta)) - 1 \quad (29)$$

the optimal solution is

$$f^* = \arg\min_f \sum_{i=1}^{N} (2\arctan(yf(\eta(x_i))) - 1)^2. \quad (30)$$

The Gauss update is

$$f(\eta)^{n+1} = f(\eta)^n + \Delta f(\eta) = f(\eta)^n - \frac{r(\eta)}{\frac{\partial r}{\partial \eta}} \quad (31)$$
$$= f(\eta)^n - \frac{2\arctan(yf(\eta)) - 1}{\frac{2y}{1+f(\eta)^2}}$$
$$= f(\eta)^n - \frac{(2\arctan yf(\eta) - 1)(1 + f(\eta)^2)}{2y}.$$

Using the known form of the optimal predictor $f(\eta) = \tan(\eta - 0.5)$ and its inverse $\eta = \arctan(f(\eta)) + 0.5$ we redefine the above updates as follows. For $y = 1$,

$$z(\eta)_1 = -\frac{(2\arctan(f(\eta)) - 1)(1 + f(\eta)^2)}{2}$$
$$= -(\eta - 1)(1 + \tan^2(\eta - 0.5)) \quad (32)$$

Algorithm 1 TangentBoost

Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where $y \in \{1, -1\}$ is the class label of example $\mathbf{x}$, and number M of weak learners in the final decision rule.

Initialization: Set uniformly distributed probabilities $\eta^{(1)}(\mathbf{x}_i) = \frac{1}{2} \;\; \forall \mathbf{x}_i$ and $\hat{f}^{(1)}(\mathbf{x}) = 0$.

for $m = \{1, \ldots, M\}$ **do**

compute the working responses $z_i^{(m)}$ as in (32) and (33) and weights $w_i^{(m)} = \eta^{(m)}(x_i)(1 - \eta^{(m)}(x_i))$.

for $k = \{1, \ldots, K\}$ **do**

compute the solution to the least squares problem,

$$a_{\phi_k} = \frac{\langle 1 \rangle_w \cdot \langle \phi_k(\mathbf{x}_i) z_i \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w \cdot \langle z_i \rangle_w}{\langle 1 \rangle_w \cdot \langle \phi_k^2(\mathbf{x}_i) \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w^2}$$

$$b_{\phi_k} = \frac{\langle \phi_k(\mathbf{x}_i)^2 \rangle_w \cdot \langle z_i \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w \cdot \langle \phi_k(\mathbf{x}_i) z_i \rangle_w}{\langle 1 \rangle_w \cdot \langle \phi_k^2(\mathbf{x}_i) \rangle_w - \langle \phi_k(\mathbf{x}_i) \rangle_w^2}$$

where we have defined

$$\langle q(\mathbf{x}_i) \rangle_w \doteq \sum_i w_i^{(m)} q(\mathbf{x}_i).$$

end for

select the direction of minimal regression error according to $k^* = \arg\min_k \sum_i w_i^{(m)} (z_i - a_{\phi_k} \phi_k(\mathbf{x}_i) - b_{\phi_k})^2$.

set $\hat{f}^{(m+1)}(\mathbf{x}_i) = \hat{f}^{(m)}(\mathbf{x}_i) + (a_{\phi_k} \phi_k(\mathbf{x}_i) + b_{\phi_k})$.

update $\eta^{(m+1)}(\mathbf{x}_i) = \arctan(\hat{f}^{(m+1)}(\mathbf{x}_i)) + 0.5$.

end for

Output: decision rule $h(\mathbf{x}) = \text{sgn}[\hat{f}^{(M)}(\mathbf{x})]$.

and for $y = -1$ as

$$z(\eta)_{-1} = -\frac{(-2 \arctan(f(\eta)) - 1)(1 + f(\eta)^2)}{-2} = -\eta(1 + \tan^2(\eta - 0.5)) \quad (33)$$

The linear regression model can now be used to approximate $z(\eta)$, as is done in logistic regression. This leads to the TangentBoost algorithm described in Algorithm 1.

5. Experiments

In this section we describe several experiments designed to test the performance of TangentBoost in classification problems involving outliers and noisy data.

We start with a simple classification problem, which provides some insight on the benefits of the Tangent loss. This problem involves the Letter-1 dataset, from the UCI database. It addresses the classification of the highly confusable letter "O" from the other letters of the alphabet, resulting in an unbalanced problem with many outliers. Figure 3 shows the histogram of the positive margins on the test set (a very similar histogram exists on the train set), for classifiers learned with TangentBoost and Adaboost. Note that the TangentBoost margins are below 0.7 and much smaller

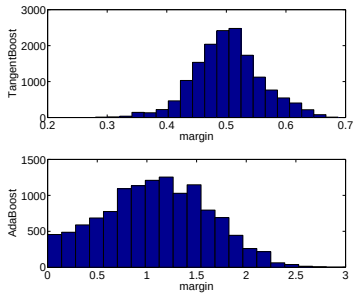

Figure 3. Histogram of positive test margins for the TangentBoost (top) and AdaBoost (bottom) algorithms on the Letter-1 dataset.

Table 2. Classification error of each boosting method on Letter-1.

Dataset	Ada	Real	Savage	Logit	Tangent
LETT.1	3621	2681	647	616	**602**

than those of AdaBoost (largest margin greater than 2.5). On the other hand, the number of classification errors on the test set is 602 for TangentBoost and 3621 for AdaBoost. This shows that larger margins do not necessarily lead to better classification when there are outliers. In its effort to push points away from the boundary, AdaBoost sacrifices classification performance. On the other hand, TangentBoost has a much cleaner margin distribution, with no points of positive margin smaller than $.25$.

It should be noted that, while this problem is serious for AdaBoost, it affects most boosting algorithms in current use. Table-2 presents the error rates achieved by some of these, after 1000 iterations of training. Adaboost and Realboost, which employ the exponential loss, have disproportionately high error. The bounded Savage loss and the linearly increasing loss of logistic regression produce a dramatic improvement. Finally, TangentBoost has the best performance. The benefits of employing a bounded loss function (Savage and Tangent) or a gradually sloping loss (logistic) are evident.

5.1. The MUSK dataset

Various authors have formulated outlier ridden vision problems, such as image classification [17], object detection [31], and tracking [3], as MIL problems. Unfortunately, these formulations are not directly comparable, and some of the datasets used are not in the public domain. An alternative is the MUSK [7] dataset, which is the standard benchmark for the broader MIL research community [7, 22, 34, 17, 7, 1]. It is a good dataset to evaluate outlier robustness, since it is naturally contaminated with misclassified data points. We learned classifiers with AdaBoost, RealBoost, LogitBoost, SavageBoost, and TangentBoost on the MUSK2 dataset, using the training and testing protocol of [7]: 10-fold cross validation, with the 10 dataset partitions defined by [7]. The test error achieved by each classi-

Table 3. MIL accuracy on the MUSK2 dataset.

Boosting Alg.	Real	Ada	Logit	Savage	Tangent	
MUSK2	67.25	82.69	84.07	85.19	85.39	
MIL Alg.	MI-NN[22]	mi-SVM[1]	DD [17]	MI-SVM [1]	EMDD [34]	IAPR [7]
MUSK2	82.5	83.6	84	84.3	84.9	89.2

fier is reported in Table-3, which also includes results from various MIL algorithms not based on boosting. Note that although SavageBoost and TangentBoost do not fit the traditional MIL definition (don't operate on bags of points), they outperform this broad selection of state-of-the-art MIL procedures. The only exception is IAPR [7] which is an algorithm specifically designed for the MUSK dataset.

5.2. Results on scene classification

We next considered the vision problem of scene classification on the 15-class dataset of [14]. Here, label noise occurs naturally, as each picture can be attributed to multiple scene categories (e.g. an image containing patches of both highway and buildings). State-of-the-art results on this dataset were recently reported in [23, 24]. These methods represent images as points on a semantic space, where each feature is the probability of the image belonging to one of the 15 classes. The two methods differ in the computation of these probabilities, one using Gaussian mixtures [23] and the other mixtures of Dirichlet distributions [24]. The probability vectors are fed to an SVM classifier, which we replaced by one learned with TangentBoost.

Table-4 compares results to different methods reported in the literature. TangentBoost(A), learned from Gaussian mixture probabilities, achieved the *highest accuracy reported for this dataset in the literature*, with 76.28%. Note that this is 2% better than the accuracy achieved with Adaboost under the same setting. This gain can only be attributed to the increased robustness of TangentBoost to outliers and noise. Also reported are the results of TangentBoost(B), where we have combined the Gaussian and Dirichelet mixture probabilities, by simply concatenating the 15 class features of both into a 30 dimensional vector. This further increased performance to 76.74% accuracy. It is also interesting to note that the greatest improvements in accuracy are achieved for the classes where [23] performs worst. These are classes that 1) are easily confusable with other classes in the dataset, and 2) contain many outliers. For example, the classification of scenes of "street", "highway", and "tall building" improves in accuracy by 21%, 12%, and 10%. Similarly, the easily confused classes of "mountain", "open country", "forest", and "coast" have relative increase in accuracy of 14%, 7%, 6%, and 6% . Finally, "bedroom" displays a 20% increase in accuracy.

Table 4. Classification accuracy for 15 scene categories.

Method	Dimensions	Accuracy%
TangentBoost(B)	30	**76.74**
TangentBoost(A)	15	**76.28**
AdaBoost	15	74.79
Rasiwasia et al. [24]	15	72.5
Rasiwasia et al. [23]	15	72.2
Liu et al. [15]	20	63.32
Liu et al. [15]	200	75.16
Lazebnik et al. [14]	200	72.2

5.3. Results on object tracking

Discriminant tracking has recently been shown to be a very effective solution to the object tracking problem [2]. It is also a prime domain for testing the effectiveness of classifiers in the presence of noise and outliers. This arises from the fact that the positive and negative training sets are collected from windows centered at the location of the current detection. In challenging scenes, object boundaries are not necessarily well defined, and the target object can be subject to occlusion, shadows, and others sources of "noise". These cause drift, since a poor localization of the target will contaminate the training data with outliers, i.e. background features labeled as target and vice-versa.

The original ensemble tracker of [2] was based on AdaBoost. It has however been noted that, in the tracking context, AdaBoost is quite susceptible to the outlier problem, and various approaches have recently been shown to outperform it [16, 3]. We consider here the discriminant saliency tracker (DST) of [16], which maps the video frames into a feature space where the target is *salient* compared to the background. Tracking is implemented with a weak classifier, which simply sums the saliency maps produced by each feature. Here, we investigate the use of boosting to combine these saliency maps in a discriminant manner. We implemented both AdaBoost and TangentBoost to achieve this combination. The results of the boosted tracker, for 2 noisy clips used in [16], are shown in Table-5. The error rates are measured as defined in [16]. It can be seen that the tracker based on AdaBoost has substantially larger error, in fact losing the target at some point in these sequences. On the other hand, TangentBoost produces a tracker that does not loose the target, and has an overall low error rate. Two representative frames of the process are shown in Figure 4.

Table 5. Tracking error rates on two noisy sequences.

Clip	AdaBoost	TangentBoost
athlete	0.89	**0.29**
gravel	0.70	**0.04**

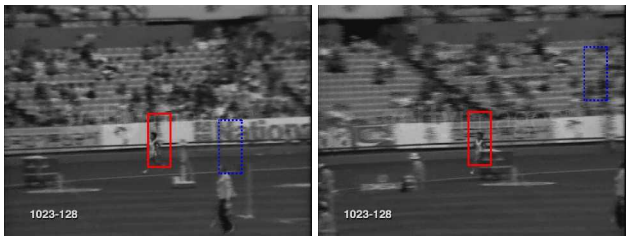

Figure 4. Frames comparing the performance of TangentBoost with AdaBoost in conjunction with a discriminant saliency tracker. Red box: TangentBoost, blue box: AdaBoost.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568, 2003.

[2] S. Avidan. Ensemble tracking. *IEEE PAMI*, 29(2):261–271, 2007.

[3] B. Babenko, M.-H. Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009.

[4] M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vision*, 1996.

[5] A. Buja, W. Stuetzle, and Y. Shen. Loss functions for binary class probability estimation and classification: Structure and applications. *(Technical Report) University of Pennsylvania*, 2005.

[6] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 2000.

[7] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley Sons Inc, New York, 2001.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, page 264, 2003.

[11] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997.

[12] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 2000.

[13] P. J. Huber and E. M. Ronchetti. *Robust Statistics*. John Wiley Sons Inc, New York, 2009.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[15] J. Liu and M. Shah. Scene modeling using co-clustering. In *ICCV*, 2007.

[16] V. Mahadevan and N. Vasconcelos. Saliency-based discriminant tracking. In *CVPR*, 2009.

[17] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *ICML*, pages 341–349, 1998.

[18] H. Masnadi-Shirazi and N. Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *NIPS*, 2009.

[19] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting Algorithms as Gradient Descent. In *NIPS*, 2000.

[20] R. McDonald, D. Hand, and I. Eckley. An empirical comparison of three boosting algorithms on real data sets with artificial class noise. In *International Workshop on Multiple Classifier Systems*, 2003.

[21] P. Meer, C. V. Stewart, and D. E. Tyler. Robust computer vision: an interdisciplinary challenge. *Comput. Vis. Image Underst.*, 2000.

[22] J. Ramon and L. De Raedt. Multi instance neural networks. In *ICML*, 2000.

[23] N. Rasiwasia and N. Vasconcelos. Scene classification with low-dimensional semantic spaces and weak supervision. In *CVPR*, 2008.

[24] N. Rasiwasia and N. Vasconcelos. Holistic context modeling using semantic co-occurrences. In *CVPR*, 2009.

[25] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John Wiley Sons Inc, New York, 2003.

[26] L. J. Savage. The elicitation of personal probabilities and expectations. *JASA*, 66:783–801, 1971.

[27] H. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *(IEEE) PAMI*, 1996.

[28] A. Torralba. Contextual priming for object detection. *Int. J. Comput. Vision*, 2003.

[29] V. N. Vapnik. *Statistical Learning Theory*. John Wiley Sons Inc, 1998.

[30] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *ICCV*, 2007.

[31] P. Viola, J. C. Platt, and C. Zhang. Multiple instance boosting for object detection. In *NIPS*, 2006.

[32] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 2004.

[33] B. Wu and R. Nevatia. Simultaneous object detection and segmentation by boosting local shape feature based classifier. In *CVPR*, 2007.

[34] Q. Zhang and S. A. Goldman. Em-dd: An improved multiple-instance learning technique. In *NIPS*, pages 1073–1080, 2001.

[35] T. Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, 2004.