

---

# Online Passive-Aggressive Algorithms

---

Koby Crammer Ofer Dekel Shai Shalev-Shwartz Yoram Singer  
School of Computer Science & Engineering  
The Hebrew University, Jerusalem 91904, Israel  
{kobics,oferd,shais,singer}@cs.huji.ac.il

## Abstract

We present a unified view for *online* classification, regression, and uniclass problems. This view leads to a single algorithmic framework for the three problems. We prove worst case loss bounds for various algorithms for both the realizable case and the non-realizable case. The end result is new algorithms and accompanying loss bounds for hinge-loss regression and uniclass. We also get refined loss bounds for previously studied classification algorithms.

## 1 Introduction

In this paper we describe and analyze several learning tasks through the same algorithmic prism. Specifically, we discuss online classification, online regression, and online uniclass prediction. In all three settings we receive instances in a sequential manner. For concreteness we assume that these instances are vectors in  $\mathbb{R}^n$  and denote the instance received on round  $t$  by  $\mathbf{x}_t$ . In the classification problem our goal is to find a mapping from the instance space into the set of labels,  $\{-1, +1\}$ . In the regression problem the mapping is into the reals. Our goal in the uniclass problem is to find a center-point in  $\mathbb{R}^n$  with a small Euclidean distance to all of the instances.

For clarity, we first describe the classification and regression problems. For classification and regression we restrict ourselves to mappings based on a weight vector  $\mathbf{w} \in \mathbb{R}^n$ . The mapping takes the form  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  where  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ . After receiving  $\mathbf{x}_t$  we extend a prediction  $\hat{y}_t$  using  $f$ . For regression the prediction is simply  $\hat{y}_t = f(\mathbf{x}_t)$  while for classification  $\hat{y}_t = \text{sign}(f(\mathbf{x}_t))$ . After extending the prediction  $\hat{y}_t$ , we receive the true outcome  $y_t$  where  $y_t \in \mathbb{R}$  for regression and  $y_t \in \{-1, +1\}$  for classification. We then suffer an *instantaneous* loss based on the discrepancy between  $y_t$  and  $f(\mathbf{x}_t)$ . The goal of the online learning algorithm is to make the cumulative loss that we suffer small. The losses we discuss in this paper depend on a pre-defined *insensitivity* parameter  $\epsilon$  and are denoted  $\ell_\epsilon(\mathbf{w}; (\mathbf{x}, y))$ . For regression the  $\epsilon$ -insensitive loss is,

$$\ell_\epsilon(\mathbf{w}; (\mathbf{x}, y)) = \begin{cases} 0 & |y - \mathbf{w} \cdot \mathbf{x}| \leq \epsilon \\ |y - \mathbf{w} \cdot \mathbf{x}| - \epsilon & \text{otherwise} \end{cases}, \quad (1)$$

while for classification the  $\epsilon$ -insensitive loss is defined to be,

$$\ell_\epsilon(\mathbf{w}; (\mathbf{x}, y)) = \begin{cases} 0 & y(\mathbf{w} \cdot \mathbf{x}) \geq \epsilon \\ \epsilon - y(\mathbf{w} \cdot \mathbf{x}) & \text{otherwise} \end{cases}. \quad (2)$$

As in other online algorithms we update the weight vector  $\mathbf{w}$  after receiving the feedback  $y_t$ . Therefore, we denote by  $\mathbf{w}_t$  the vector used for prediction on round  $t$ . We leave the details on the form this update takes to later sections.

Problem	Example ( $\mathbf{z}_t$ )	Discrepancy ( $\delta$ )	Update Direction ( $\mathbf{v}_t$ )
Classification	$(\mathbf{x}_t, y_t) \in \mathbb{R}^n \times \{-1, +1\}$	$-y_t(\mathbf{w}_t \cdot \mathbf{x}_t)$	$y_t \mathbf{x}_t$
Regression	$(\mathbf{x}_t, y_t) \in \mathbb{R}^n \times \mathbb{R}$	$ y_t - \mathbf{w}_t \cdot \mathbf{x}_t $	$\text{sign}(y_t - \mathbf{w}_t \cdot \mathbf{x}_t) \mathbf{x}_t$
Uniclass	$(\mathbf{x}_t, y_t) \in \mathbb{R}^n \times \{1\}$	$\ \mathbf{x}_t - \mathbf{w}_t\ $	$\frac{\mathbf{x}_t - \mathbf{w}_t}{\ \mathbf{x}_t - \mathbf{w}_t\ }$

Table 1: Summary of the settings and parameters employed by the additive PA algorithm for classification, regression, and uniclass.

The setting for uniclass is slightly different as we only observe a sequence of instances  $\mathbf{x}_t$ 's. The goal of the uniclass algorithm is to find a center-point  $\mathbf{w}$  such that all points  $\mathbf{x}_t$  fall within a radius of  $\epsilon$  from  $\mathbf{w}$ . Since we employ the framework of online learning the vector  $\mathbf{w}$  is constructed incrementally. The vector  $\mathbf{w}_t$  therefore plays the role of the *instantaneous* center and is adapted after observing each instance  $\mathbf{x}_t$ . If an example  $\mathbf{x}_t$  falls within an Euclidean distance  $\epsilon$  from  $\mathbf{w}_t$  then we suffer no loss. Otherwise, the loss is the distance to a ball of radius  $\epsilon$  centered at  $\mathbf{w}_t$ . Formally, the uniclass loss is,

$$\ell_\epsilon(\mathbf{w}_t; \mathbf{x}_t) = \begin{cases} 0 & \|\mathbf{x}_t - \mathbf{w}_t\| \leq \epsilon \\ \|\mathbf{x}_t - \mathbf{w}_t\| - \epsilon & \text{otherwise} \end{cases} . \quad (3)$$

In the next sections we give additive and multiplicative online algorithms for the above learning problems and prove respective online loss bounds. A common thread of our approach is a unified view of all three tasks which leads to a single algorithmic framework with a common analysis.

**Related work:** Our work builds on numerous techniques from online learning. Due to the lack of space we mention here only a handful of related papers. The updates we derive are based on an optimization problem directly related to the one employed by Support Vector Machines [11]. Li and Long [10] were among the first to suggest the idea of converting a batch optimization problem into an online task. Our work borrows ideas from the work of Warmuth and colleagues [8]. In particular, Gentile and Warmuth [4] generalized and adapted techniques from [8] to the hinge loss which is closely related to the losses defined in Eqs. (1)-(3). Kivinen et al. [7] discussed a general framework for gradient-based online learning where some of their bounds bare similarities to the bounds presented in this paper. Our work also generalizes and greatly improves online loss bounds for classification given in [2]. Finally, we would like to note that similar algorithms have been devised in the convex optimization community (see [1]). The main difference between these algorithms and the online algorithms presented in this paper lies in the analysis: while we derive worst case, finite horizon, loss bounds, the optimization community is mostly concerned with asymptotic convergence properties.

## 2 A Unified Loss

The three problems described in the previous section share common algebraic properties which we explore in this section. The end result is a common algorithmic framework that is applicable to all three problems and an accompanying analysis (Sec. 3). Let  $\mathbf{z}_t = (\mathbf{x}_t, y_t)$  denote the instance-target pair received on round  $t$  where in the case of uniclass we set  $y_t = 1$  as a placeholder. For a given example  $\mathbf{z}_t$ , let  $\delta(\mathbf{w}; \mathbf{z}_t)$  denote the discrepancy of  $\mathbf{w}$  on  $\mathbf{z}_t$ : for classification we set the discrepancy to be  $-y_t(\mathbf{w}_t \cdot \mathbf{x}_t)$  (the negative of the margin), for regression it is  $|y_t - \mathbf{w}_t \cdot \mathbf{x}_t|$ , and for uniclass  $\|\mathbf{x}_t - \mathbf{w}_t\|$ . Fixing  $\mathbf{z}_t$  we also view  $\delta(\mathbf{w}; \mathbf{z}_t)$  as a *convex* function of  $\mathbf{w}$ . Let  $[a]_+$  denote the step function, that is,  $[a]_+ = a$

whenever  $a > 0$  and otherwise it equals zero. Using the discrepancies defined above the three different losses given in Eqs. (1)-(3) can all be written as  $\ell_\epsilon(\mathbf{w}; \mathbf{z}) = [\delta(\mathbf{w}; \mathbf{z}) - \epsilon]_+$ , where for classification we set  $\epsilon \leftarrow -\epsilon$  since the discrepancy is defined as the negative of the margin. While this construction might seem a bit odd for classification, it is very useful in unifying the three problems. To conclude, the loss in all three problems can be derived by applying the same hinge loss to different (problem dependent) discrepancies.

### 3 An Additive Algorithm for the Realizable Case

Equipped with the simple unified notion of loss we describe in this section a single on-line algorithm that is applicable to all three problems. The algorithm and the analysis we present in this section assume that there exist a weight vector  $\mathbf{w}^*$  and an insensitivity parameter  $\epsilon^*$  for which the data is perfectly realizable. Namely, we assume that for all  $t$   $\ell_{\epsilon^*}(\mathbf{w}^*; (\mathbf{x}_t, y_t)) = 0$  which implies that,

$$y_t(\mathbf{w}^* \cdot \mathbf{x}_t) \geq |\epsilon^*| \text{ (Class.)} \quad |y_t - \mathbf{w}^* \cdot \mathbf{x}_t| \leq \epsilon^* \text{ (Reg.)} \quad \|\mathbf{x}_t - \mathbf{w}^*\| \leq \epsilon^* \text{ (Unic.)} . \quad (4)$$

A modification of the algorithm for the unrealizable case is given in Sec. 5.

The general method we use for deriving our on-line update rule is to define the new weight vector  $\mathbf{w}_{t+1}$  as the solution to the following projection problem

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \quad \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell_\epsilon(\mathbf{w}; \mathbf{z}_t) = 0 , \quad (5)$$

where  $\epsilon$  is an insensitivity parameter. The solution to the above optimization problem is the projection of  $\mathbf{w}_t$  onto the set of all weight vectors that attain a loss of zero. We denote this set by  $S$ . For the case of classification,  $S$  is a half space,  $S = \{\mathbf{w} : -y_t \mathbf{w} \cdot \mathbf{x}_t \leq \epsilon\}$ . For regression  $S$  is an  $\epsilon$ -hyper-slab,  $S = \{\mathbf{w} : |\mathbf{w} \cdot \mathbf{x}_t - y_t| \leq \epsilon\}$  and for uniclass it is a ball of radius  $\epsilon$  centered at  $\mathbf{x}_t$ ,  $S = \{\mathbf{w} : \|\mathbf{w} - \mathbf{x}_t\| \leq \epsilon\}$ . In Fig. 2 we illustrate the projection for the three cases. This optimization problem attempts to keep  $\mathbf{w}_{t+1}$  as close to  $\mathbf{w}_t$  as possible, while forcing  $\mathbf{w}_{t+1}$  to achieve a zero loss on the most recent example. The resulting algorithm is *passive* whenever the loss is zero, that is,  $\mathbf{w}_{t+1} = \mathbf{w}_t$  whenever  $\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) = 0$ . In contrast, on rounds for which  $\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) > 0$  we *aggressively* force  $\mathbf{w}_{t+1}$  to satisfy the constraint  $\ell_\epsilon(\mathbf{w}_{t+1}; \mathbf{z}_t) = 0$ . Therefore we name the algorithm passive-aggressive or PA for short. In the following we show that for the three problems described above the solution to the optimization problem in Eq. (5) yields the following update rule,

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{v}_t , \quad (6)$$

where  $\mathbf{v}_t$  is minus the gradient of the discrepancy and  $\tau_t = \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) / \|\mathbf{v}_t\|^2$ . (Note that although the discrepancy might not be differentiable everywhere, its gradient exists whenever the loss is greater than zero). To see that the update from Eq. (6) is the solution to the problem defined by Eq. (5), first note that the equality constraint  $\ell_\epsilon(\mathbf{w}; \mathbf{z}_t) = 0$  is equivalent to the inequality constraint  $\delta(\mathbf{w}; \mathbf{z}_t) \leq \epsilon$ . The Lagrangian of the optimization problem is

$$\mathcal{L}(\mathbf{w}, \tau) = \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 + \tau(\delta(\mathbf{w}; \mathbf{z}_t) - \epsilon) , \quad (7)$$

<p><b>Parameter:</b> Insensitivity: <math>\epsilon</math>  <b>Initialize:</b> Set <math>\mathbf{w}_1 = 0</math> (R&amp;C); <math>\mathbf{w}_1 = \mathbf{x}_0</math> (U)  For <math>t = 1, 2, \dots</math></p> <ul style="list-style-type: none"> <li>• Get a new instance: <math>\mathbf{z}_t \in \mathbb{R}^n</math></li> <li>• Suffer loss: <math>\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)</math></li> <li>• If <math>\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) &gt; 0</math> : <ol style="list-style-type: none"> <li>1. Set <math>\mathbf{v}_t</math> (see Table 1)</li> <li>2. Set <math>\tau_t = \frac{\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)}{\ \mathbf{v}_t\ ^2}</math></li> <li>3. Update: <math>\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{v}_t</math></li> </ol> </li> </ul>
<p>Figure 1: The additive PA algorithm.</p>

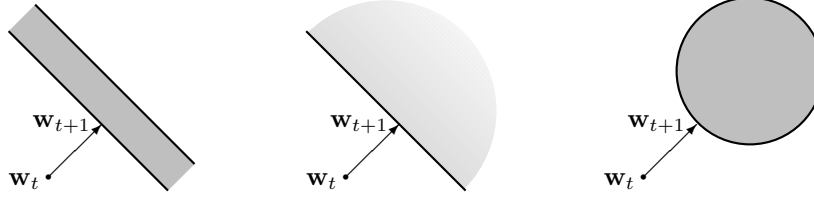


Figure 2: An illustration of the update:  $\mathbf{w}_{t+1}$  is found by projecting the current vector  $\mathbf{w}_t$  onto the set of vectors attaining a zero loss on  $\mathbf{z}_t$ . This set is a stripe in the case of regression, half-space for classification, and a sphere for uniclass.

where  $\tau \geq 0$  is a Lagrange multiplier. To find a saddle point of  $\mathcal{L}$  we first differentiate  $\mathcal{L}$  with respect to  $\mathbf{w}$  and use the fact that  $\mathbf{v}_t$  is minus the gradient of the discrepancy to get that,

$$\nabla_{\mathbf{w}}(\mathcal{L}) = \mathbf{w} - \mathbf{w}_t + \tau \nabla_{\mathbf{w}} \delta = 0 \Rightarrow \mathbf{w} = \mathbf{w}_t + \tau \mathbf{v}_t .$$

To find the value of  $\tau$  we use the KKT conditions. Hence, whenever  $\tau$  is positive (as in the case of non-zero loss), the inequality constraint,  $\delta(\mathbf{w}; \mathbf{z}_t) \leq \epsilon$ , becomes an equality. Simple algebraic manipulations yield that the value  $\tau$  for which  $\delta(\mathbf{w}; \mathbf{z}_t) = \epsilon$  for all three problems is equal to,  $\tau_t = \ell_\epsilon(\mathbf{w}; \mathbf{z}_t) / \|\mathbf{v}_t\|^2$ .

A summary of the discrepancy functions and their respective updates is given in Table 1. The pseudo-code of the additive algorithm for all three settings is given in Fig. 1. Note that for uniclass the update given by Eq. (6) sets  $\mathbf{w}_{t+1}$  to be a convex combination of  $\mathbf{w}_t$  and  $\mathbf{x}_t$ . To see this we expand  $\mathbf{w}_{t+1}$ ,

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{v}_t = \mathbf{w}_t + \frac{(\|\mathbf{w}_t - \mathbf{x}_t\| - \epsilon)}{\|\mathbf{w}_t - \mathbf{x}_t\|} (\mathbf{x}_t - \mathbf{w}_t) = (1 - \nu_t) \mathbf{w}_t + \nu_t \mathbf{x}_t ,$$

where  $\nu_t = \frac{(\|\mathbf{w}_t - \mathbf{x}_t\| - \epsilon)}{\|\mathbf{w}_t - \mathbf{x}_t\|} \in [0, 1]$ .

To conclude the description of the additive PA algorithm, we would like to discuss the initialization of  $\mathbf{w}_1$ . For classification and regression a reasonable choice for  $\mathbf{w}_1$  is the zero vector. However, in the case of uniclass initializing  $\mathbf{w}_1$  to be the zero vector might incur large losses if, for instance, all the instances are located far away from the origin. A more sensible choice for uniclass is to initialize  $\mathbf{w}_1$  to be one of the examples. For simplicity of the description we assume that we are provided with an example  $\mathbf{x}_0$  prior to the run of the algorithm. This initialization enables us to expand  $\mathbf{w}_{t+1}$  and write it as a convex combination of the examples observed thus far,

$$\mathbf{w}_{t+1} = \sum_{s=0}^t \alpha_s \mathbf{x}_s \quad \text{where} \quad \alpha_s = \nu_s \prod_{i=s+1}^t (1 - \nu_i) .$$

The above representation for the weight vector for uniclass enables to employ kernels [11]. It is also easy to verify that in classification and regression the weight vector lies in the linear subspace spanned by the instances, therefore we can use kernels for these problems as well.

## 4 Analysis

The following theorem provides a unified loss bound for all three settings. After proving the theorem we discuss a few of its implications.

**Theorem 1** Let  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t, \dots$  be a sequence of examples for one of the problems described in Table 1. Assume that there exist  $\mathbf{w}^*$  and  $\epsilon^*$  such that  $\ell_{\epsilon^*}(\mathbf{w}^*; \mathbf{z}_t) = 0$  for all  $t$ . Then if the additive PA algorithm is run with  $\epsilon > \epsilon^*$ , the following bound holds for any  $T \geq 1$

$$\sum_{t=1}^T (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t))^2 + 2(\epsilon - \epsilon^*) \sum_{t=1}^T \ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) \leq B \|\mathbf{w}^* - \mathbf{w}_1\|^2, \quad (10)$$

where for classification and regression  $B$  is a bound on the squared norm of the instances ( $\forall t : B \geq \|\mathbf{x}_t\|_2^2$ ) and  $B = 1$  for uniclass.

**Proof:** Define  $\Delta_t = \|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2$ . We prove the theorem by bounding  $\sum_{t=1}^T \Delta_t$  from above and below. First note that  $\sum_{t=1}^T \Delta_t$  is a telescopic sum and therefore

$$\sum_{t=1}^T \Delta_t = \|\mathbf{w}_1 - \mathbf{w}^*\|^2 - \|\mathbf{w}_{T+1} - \mathbf{w}^*\|^2 \leq \|\mathbf{w}_1 - \mathbf{w}^*\|^2. \quad (11)$$

This provides an upper bound on  $\sum_t \Delta_t$ . In the following we prove the lower bound

$$\Delta_t \geq \frac{\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t)}{B} (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) + 2(\epsilon - \epsilon^*)) .$$

To prove the above bound, note that we do not modify  $\mathbf{w}_t$  if  $\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) = 0$ . Therefore, the above inequality trivially holds when  $\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) = 0$  and thus we can restrict ourselves to the rounds on which the discrepancy was larger than  $\epsilon$  which implies that  $\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) = \delta(\mathbf{w}_t; \mathbf{z}_t) - \epsilon$ . Let  $t$  be such a round and expand  $\mathbf{w}_{t+1}$ ,

$$\begin{aligned} \Delta_t &= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2 \\ &= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_t + \tau_t \mathbf{v}_t - \mathbf{w}^*\|^2 \\ &= \|\mathbf{w}_t - \mathbf{w}^*\|^2 - (\tau_t^2 \|\mathbf{v}_t\|^2 + 2\tau_t (\mathbf{v}_t \cdot (\mathbf{w}_t - \mathbf{w}^*))) + \|\mathbf{w}_t - \mathbf{w}^*\|^2 \\ &= -\tau_t^2 \|\mathbf{v}_t\|^2 + 2\tau_t \mathbf{v}_t \cdot (\mathbf{w}^* - \mathbf{w}_t) . \end{aligned} \quad (12)$$

We now use the fact that  $-\mathbf{v}_t$  is a gradient of the convex function  $\delta(\mathbf{w}; \mathbf{z}_t)$  at  $\mathbf{w}_t$ . Therefore, for any  $\mathbf{w} \in \mathbb{R}^n$  the following inequality holds:

$$\delta(\mathbf{w}; \mathbf{z}_t) - \delta(\mathbf{w}_t; \mathbf{z}_t) \geq (-\mathbf{v}_t) \cdot (\mathbf{w} - \mathbf{w}_t) .$$

Specifically, for  $\mathbf{w} = \mathbf{w}^*$  we get

$$\mathbf{v}_t \cdot (\mathbf{w}^* - \mathbf{w}_t) \geq \delta(\mathbf{w}_t; \mathbf{z}_t) - \delta(\mathbf{w}^*; \mathbf{z}_t) = \delta(\mathbf{w}_t; \mathbf{z}_t) - \epsilon + \epsilon - \delta(\mathbf{w}^*; \mathbf{z}_t) . \quad (13)$$

Recall that  $\delta(\mathbf{w}_t; \mathbf{z}_t) - \epsilon = \ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t)$  and that  $\epsilon^* \geq \delta(\mathbf{w}^*; \mathbf{z}_t)$ . Therefore,

$$(\delta(\mathbf{w}_t; \mathbf{z}_t) - \epsilon) + (\epsilon - \delta(\mathbf{w}^*; \mathbf{z}_t)) \geq \ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) + (\epsilon - \epsilon^*) . \quad (14)$$

Combining Eqs. (12-14) we get

$$\begin{aligned} \Delta_t &\geq -\tau_t^2 \|\mathbf{v}_t\|^2 + 2\tau_t (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) + (\epsilon - \epsilon^*)) \\ &= \tau_t (-\tau_t \|\mathbf{v}_t\|^2 + 2\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) + 2(\epsilon - \epsilon^*)) . \end{aligned} \quad (15)$$

Plugging  $\tau_t = \ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) / \|\mathbf{v}_t\|^2$  into Eq. (15) we get

$$\Delta_t \geq \frac{\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t)}{\|\mathbf{v}_t\|^2} (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) + 2(\epsilon - \epsilon^*)) .$$

For uniclass  $\|\mathbf{v}_t\|^2$  is always equal to 1 by construction and for classification and regression we have  $\|\mathbf{v}_t\|^2 = \|\mathbf{x}_t\|^2 \leq B$  which gives,

$$\Delta_t \geq \frac{\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t)}{B} (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) + 2(\epsilon - \epsilon^*)) .$$

Comparing the above lower bound with the upper bound in Eq. (11) we get

$$\sum_{t=1}^T (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t))^2 + \sum_{t=1}^T 2(\epsilon - \epsilon^*)\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) \leq B\|\mathbf{w}^* - \mathbf{w}_1\|^2 .$$

This concludes the proof. ■

Let us now discuss the implications of Thm. 1. First, recall that for classification and regression  $\mathbf{w}_1$  is the zero vector. Thus, the loss bound reduces to  $B\|\mathbf{w}^*\|^2$ . This form of bound is common to online algorithms for classification such as ROMMA [10]. However, here we are bounding the hinge-loss simultaneously for both regression and classification. Due to our initialization, we get that for uniclass  $\|\mathbf{w}^* - \mathbf{w}_1\|^2 \leq (\epsilon^*)^2$  and the bound is translation invariant – clearly a desired property for the uniclass problem. It is straightforward to derive a mistake bound from the loss bound of Thm. 1 in the case of classification. First note that the theorem implies that,  $\sum_t (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t))^2 \leq B\|\mathbf{w}^*\|^2$ . Second, whenever there is a classification error we have that  $(\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t))^2 \geq (\epsilon)^2$ . Therefore, the number of classification errors is bound above by  $B\|\mathbf{w}^*\|^2/\epsilon^2$ . For classification however the bound has one degree of freedom as we can scale  $\mathbf{w}^*$  and thus  $\epsilon$  accordingly. If we get rid of this degree of freedom by choosing the optimal value for  $\epsilon$  and setting the norm of  $\mathbf{w}^*$  to 1 we obtain the mistake bound of the Perceptron algorithm. We also like to note that the PA algorithm is the same as the MIRA algorithm [2] for binary problems. However, the loss bound from Thm. 1 is much more refined than the bound in [2].

We would like to note that the online loss bound can be used to derive a simple progress bound for a batch setting. The bound is achieved by cycling through the training set multiple times and calling the online algorithm consecutively. This cycling is guaranteed to decrease the loss of the entire training set like  $O(1/\sqrt{k})$  where  $k$  is the number of epochs. Moreover, it is possible to show that in the case of classification  $\|\mathbf{w}_{t+1}\| \leq 2\|\mathbf{w}^*\|$  for  $t > 0$ . This implies that the PA algorithm is guaranteed to find a separating hyperplane whose margin is at least a half of the best margin achievable by a batch algorithm.

## 5 A Modification for the Unrealizable Case

We now describe an algorithm for the case where the data is unrealizable. The algorithm employs two parameters. The first is the insensitivity parameter  $\epsilon$  which defines the loss function as in the realizable case. However, in this case we do not assume that there exists  $\mathbf{w}^*$  that achieves zero  $\epsilon$ -loss over the sequence. Rather, we measure the loss of the online algorithm *relative* to the loss of any vector  $\mathbf{w}^*$ . The second parameter,  $\gamma > 0$ , is a clipping parameter. Before describing the effect of this parameter we define the update step for the unrealizable case. As in the realizable case, the algorithm is conservative, that is if the  $\epsilon$ -loss over example  $\mathbf{z}_t$  is zero then  $\mathbf{w}_{t+1} = \mathbf{w}_t$ . In case the loss is positive the update rule is  $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \mathbf{v}_t$ . The update direction  $\mathbf{v}_t$  is the same as in the realizable case. However, the scale factor  $\tau_t$  is clipped by  $\gamma$ . That is,  $\tau_t = \min\{\gamma, \ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t)\}/\|\mathbf{v}_t\|^2$ . The following theorem provides a bound on the cumulative loss of our online algorithm *relative* to the loss of any fixed weight vector  $\mathbf{w}^*$ .

**Theorem 2** *Let  $\mathbf{z}_1 = (\mathbf{x}_1, y_1), \mathbf{z}_2 = (\mathbf{x}_2, y_2), \dots, \mathbf{z}_t = (\mathbf{x}_t, y_t), \dots$  be a sequence of examples. Let  $\mathbf{w}^* \in \mathbb{R}^n$  be any vector. Then if the PA algorithm for the unrealizable case is run with  $\epsilon$ , and with  $\gamma > 0$ , the following bound holds for any  $T \geq 1$*

$$\sum_{t=1}^T \ell_{\epsilon+\gamma}(\mathbf{w}_t; \mathbf{z}_t) \leq 2 \sum_{t=1}^T \ell_{\epsilon}(\mathbf{w}^*; \mathbf{z}_t) + \frac{B\|\mathbf{w}^* - \mathbf{w}_1\|^2}{\gamma} ,$$

where  $B$  is as defined in Thm. 1.

**Proof:** The proof technique is similar to the proof of Thm. 1. We define  $\Delta_t = \|\mathbf{w}_t - \mathbf{w}^*\|^2 - \|\mathbf{w}_{t+1} - \mathbf{w}^*\|^2$ . We upper bound  $\sum_t \Delta_t$  by  $\|\mathbf{w}_1 - \mathbf{w}^*\|^2$ . We expand  $\mathbf{w}_{t+1}$  and get the lower bound,

$$\Delta_t \geq \tau_t (-\tau_t \|\mathbf{v}_t\|^2 + 2\mathbf{v}_t \cdot (\mathbf{w}^* - \mathbf{w}_t)) \quad . \quad (17)$$

We use the fact that  $-\mathbf{v}_t$  is a gradient to get

$$\mathbf{v}_t \cdot (\mathbf{w}^* - \mathbf{w}_t) \geq \delta(\mathbf{w}_t; \mathbf{z}_t) - \delta(\mathbf{w}^*; \mathbf{z}_t) = \delta(\mathbf{w}_t; \mathbf{z}_t) - \epsilon + \epsilon - \delta(\mathbf{w}^*; \mathbf{z}_t) \quad .$$

Since we make progress only if  $\delta(\mathbf{w}_t; \mathbf{z}_t) \geq \epsilon$ , we get that  $\delta(\mathbf{w}_t; \mathbf{z}_t) - \epsilon = \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)$ . In addition, we use the fact that  $\ell_\epsilon(a) \geq a - \epsilon$  for any  $a$  and therefore  $-(\delta(\mathbf{w}^*; \mathbf{z}_t) - \epsilon) \geq -\ell_\epsilon(\mathbf{w}^*; \mathbf{z}_t)$ . Therefore we get,

$$\Delta_t \geq \tau_t (-\tau_t \|\mathbf{v}_t\|^2 + 2(\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) - \ell_\epsilon(\mathbf{w}^*; \mathbf{z}_t))) \quad . \quad (18)$$

We assign the value  $\tau_t = \min\{\gamma, \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)\} / \|\mathbf{v}_t\|^2$  in Eq. (18) and use the simple fact that  $\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) \geq \min\{\gamma, \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)\}$  to get the lower bound

$$\Delta_t \geq \frac{\min\{\gamma, \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)\}}{\|\mathbf{v}_t\|^2} (\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) - 2\ell_\epsilon(\mathbf{w}^*; \mathbf{z}_t)) \quad .$$

Combining the above bound with the upper bound for  $\sum_t \Delta_t$ , and with the bound  $\|\mathbf{v}_t\|^2 \leq B$  yields

$$B\|\mathbf{w}_1 - \mathbf{w}^*\|^2 \geq \sum_{t=1}^T \min\{\gamma, \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)\} (\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) - 2\ell_\epsilon(\mathbf{w}^*; \mathbf{z}_t)) \quad . \quad (19)$$

Now note that since  $\min\{\gamma, \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)\} \leq \gamma$  we have

$$-\min\{\gamma, \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)\} \ell_\epsilon(\mathbf{w}^*; \mathbf{z}_t) \geq -\gamma \ell_\epsilon(\mathbf{w}^*; \mathbf{z}_t) \quad . \quad (20)$$

In addition,

$$\min\{\gamma, \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t)\} \ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) \geq \gamma \ell_{\epsilon+\gamma}(\mathbf{w}_t; \mathbf{z}_t) \quad , \quad (21)$$

for all  $t$ , since if  $\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) \leq \gamma$  then the right-hand side of Eq. (21) is zero and if  $\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) \geq \gamma$  we use the fact the loss is monotonically decreasing with respect to the insensitivity parameter. Combining Eq. (20) and Eq. (21) with Eq. (19) gives,

$$\sum_{t=1}^T \ell_{\epsilon+\gamma}(\mathbf{w}_t; \mathbf{z}_t) \leq 2 \sum_{t=1}^T \ell_\epsilon(\mathbf{w}^*; \mathbf{z}_t) + \frac{B\|\mathbf{w}_1 - \mathbf{w}^*\|^2}{\gamma} \quad .$$

■

## 6 Extensions

There are numerous potential extensions to our approach. For instance, if all the components of the instances are non-negative we can derive a multiplicative version of the PA algorithm. The multiplicative PA algorithm maintains a weight vector  $\mathbf{w}_t \in \mathbb{P}^n$  where  $\mathbb{P}^n = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}_+^n, \sum_{j=1}^n \mathbf{x}_j = 1\}$ . The multiplicative update of  $\mathbf{w}_t$  is,

$$w_{t+1,j} = (1/Z_t) w_{t,j} e^{\tau_t v_{t,j}} \quad ,$$

where  $\mathbf{v}_t$  is the same as the one used in the additive algorithm (Table 1),  $\tau_t$  now becomes  $4\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) / \|\mathbf{v}_t\|_\infty^2$  for regression and classification and  $\ell_\epsilon(\mathbf{w}_t; \mathbf{z}_t) / (8\|\mathbf{v}_t\|_\infty^2)$  for uniclass and  $Z_t = \sum_{j=1}^n w_{t,j} e^{\tau_t v_{t,j}}$  is a normalization factor. For the multiplicative PA we can prove the following loss bound.

**Theorem 3** Let  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_t = (\mathbf{x}_t, y_t), \dots$  be a sequence of examples such that  $x_{t,j} \geq 0$  for all  $t$ . Let  $D_{RE}(\mathbf{w} \parallel \mathbf{w}') = \sum_j w_j \log(w_j/w'_j)$  denote the relative entropy between  $\mathbf{w}$  and  $\mathbf{w}'$ . Assume that there exist  $\mathbf{w}^*$  and  $\epsilon^*$  such that  $\ell_{\epsilon^*}(\mathbf{w}^*; \mathbf{z}_t) = 0$  for all  $t$ . Then when the multiplicative version of the PA algorithm is run with  $\epsilon > \epsilon^*$ , the following bound holds for any  $T \geq 1$ ,

$$\sum_{t=1}^T (\ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t))^2 + 2(\epsilon - \epsilon^*) \sum_{t=1}^T \ell_{\epsilon}(\mathbf{w}_t; \mathbf{z}_t) \leq \frac{1}{2} B D_{RE}(\mathbf{w}^* \parallel \mathbf{w}_1) ,$$

where for classification and regression  $B$  is a bound on the square of the infinity norm of the instances ( $\forall t : B \geq \|\mathbf{x}_t\|_{\infty}^2$ ) and  $B = 16$  for uniclass.

The proof of the theorem is rather technical and uses the proof technique of Thm. 1 in conjunction with inequalities on the logarithm of  $Z_t$  (see for instance [5, 8, 6]).

An interesting question is whether the unified view of classification, regression, and uniclass can be exported and used with other algorithms for classification such as ROMMA [10] and ALMA [3]. Another, rather general direction for possible extension surfaces when replacing the Euclidean distance between  $\mathbf{w}_{t+1}$  and  $\mathbf{w}_t$  with other distances and divergences such as the Bregman divergence. The resulting optimization problem may be solved via Bregman projections. In this case it might be possible to derive general loss bounds, see for example [9]. Last, we would like to note we are currently exploring generalizations of our framework to other decision tasks such as distance-learning [12] and online convex programming [13].

## References

- [1] Y. Censor and S.A. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, New York, NY, USA, 1997.
- [2] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991, 2003.
- [3] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.
- [4] C. Gentile and M. Warmuth. Linear hinge loss and average margin. In *Advances in Neural Information Processing Systems 10*, 1998.
- [5] D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. A comparison of new and old algorithms for a mixture estimation problem. In *Proceedings of the Eighth Annual Conference on Computational Learning Theory*, pages 69–78, 1995.
- [6] J. Kivinen, D.P Helmbold, and M. Warmuth. Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, 10(6):1291–1304, 1999.
- [7] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.
- [8] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–64, January 1997.
- [9] J. Kivinen and M. K. Warmuth. Relative loss bounds for multidimensional regression problems. *Journal of Machine Learning*, 45(3):301–329, July 2001.
- [10] Y. Li and P. M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1–3):361–387, 2002.
- [11] V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.
- [12] E. Xing, A.Y. Ng, M. Jordan, and S. Russel. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15*, 2003.
- [13] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.