A New Perspective on an Old Perceptron Algorithm

Shai Shalev-Shwartz[1,2] and Yoram Singer[1,2]

[1] School of Computer Sci. & Eng., The Hebrew University, Jerusalem 91904, Israel
[2] Google Inc., 1600 Amphitheater Parkway, Mountain View CA 94043, USA
{shais,singer}@cs.huji.ac.il

Abstract. We present a generalization of the Perceptron algorithm. The new algorithm performs a Perceptron-style update whenever the margin of an example is smaller than a predefined value. We derive worst case mistake bounds for our algorithm. As a byproduct we obtain a new mistake bound for the Perceptron algorithm in the inseparable case. We describe a multiclass extension of the algorithm. This extension is used in an experimental evaluation in which we compare the proposed algorithm to the Perceptron algorithm.

1 Introduction

The Perceptron algorithm [1, 15, 14] is a well studied and popular classification learning algorithm. Despite its age and simplicity it has proven to be quite effective in practical problems, even when compared to the state-of-the-art large margin algorithms [9]. The Perceptron maintains a single hyperplane which separates positive instances from negative ones. Another influential learning paradigm which employs separating hyperplanes is Vapnik's Support Vector Machine (SVM) [16]. Learning algorithms for SVMs use quadratic programming for finding a separating hyperplane attaining the maximal *margin*. Interestingly, the *analysis* of the Perceptron algorithm [14] also employs the notion of margin. However, the algorithm itself does not exploit any margin information. In this paper we try to draw a connection between the two approaches by analyzing a variant of the Perceptron algorithm, called Ballseptron, which utilizes the margin. As a byproduct, we also get a new analysis for the original Perceptron algorithm.

While the Perceptron algorithm can be used as linear programming solver [4] and can be converted to a batch learning algorithm [9], it was originally studied in the *online* learning model which is also the main focus of our paper. In online learning, the learner receives instances in a sequential manner while outputting a prediction after each observed instance. For concreteness, let $\mathcal{X} = \mathbb{R}^n$ denote our instance space and let $\mathcal{Y} = \{+1, -1\}$ denote our label space. Our primary goal is to learn a classification function $f : \mathcal{X} \rightarrow \mathcal{Y}$. We confine most of our discussion to linear classification functions. That is, f takes the form $f(\mathbf{x}) = \text{sign}(\mathbf{w}\cdot\mathbf{x})$ where $\mathbf{w}$ is a weight vector in $\mathbb{R}^n$. We briefly discuss in later sections how to use Mercer kernels with the proposed algorithm. Online algorithms work in rounds. On round t an online algorithm receives an instance $\mathbf{x}_t$ and predicts a label $\hat{y}_t$ according to its current classification function $f_t : \mathcal{X} \rightarrow \mathcal{Y}$. In our case, $\hat{y}_t = f_t(\mathbf{x}_t) = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$, where $\mathbf{w}_t$ is the current weight vector used by the algorithm. The true label y_t is then revealed and the online algorithm may update its classification function. The goal of the online algorithm is to minimize its cumulative number of prediction mistakes which we denote by ε. The Perceptron initializes its weight vector to be the zero vector and employs the update rule $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$ where $\tau_t = 1$ if $\hat{y}_t \neq y_t$ and $\tau_t = 0$ otherwise.

Several authors [14, 3, 13] have shown that whenever the Perceptron is presented with a sequence of linearly separable examples, it suffers a bounded number of prediction mistakes which does not depend on the length of the sequence of examples. Formally, let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of instance-label pairs. Assume that there exists a unit vector $\mathbf{u}$ ($\|\mathbf{u}\| = 1$) and a positive scalar $\gamma > 0$ such that for all t, $y_t(\mathbf{u} \cdot \mathbf{x}_t) \geq \gamma$. In words, $\mathbf{u}$ separates the instance space into two half-spaces such that positively labeled instances reside in one half-space while the negatively labeled instances belong to the second half-space. Moreover, the distance of each instance to the separating hyperplane $\{\mathbf{x} : \mathbf{u} \cdot \mathbf{x} = 0\}$, is at least γ. We refer to γ as the margin attained by $\mathbf{u}$ on the sequence of examples. Throughout the paper we assume that the instances are of bounded norm and let $R = \max_t \|\mathbf{x}_t\|$ denote the largest norm of an instance in the input sequence. The number of prediction mistakes, ε, the Perceptron algorithm makes on the sequence of examples is at most

$$\varepsilon \;\leq\; \left(\frac{R}{\gamma}\right)^2 \quad . \tag{1}$$

Interestingly, neither the dimensionality of $\mathcal{X}$ nor the number of examples directly effect this mistake bound. Freund and Schapire [9] relaxed the separability assumption and presented an analysis for the inseparable case. Their mistake bound depends on the *hinge-loss* attained by any vector $\mathbf{u}$. Formally, let $\mathbf{u}$ be *any* unit vector ($\|\mathbf{u}\| = 1$). The hinge-loss of $\mathbf{u}$ with respect to an instance-label pair $(\mathbf{x}_t, y_t)$ is defined as $\ell_t = \max\{0, \gamma - y_t \mathbf{u} \cdot \mathbf{x}_t\}$ where γ is a fixed target margin value. This definition implies that $\ell_t = 0$ if $\mathbf{x}_t$ lies in the half-space corresponding to y_t and its distance from the separating hyperplane is at least γ. Otherwise, ℓ_t increases linearly with $-y_t(\mathbf{u} \cdot \mathbf{x}_t)$. Let D_2 denote the two-norm of the sequence of hinge-losses suffered by $\mathbf{u}$ on the sequence of examples,

$$D_2 = \left(\sum_{t=1}^{T} \ell_t^2\right)^{1/2} \quad . \tag{2}$$

Freund and Schapire [9] have shown that the number of prediction mistakes the Perceptron algorithm makes on the sequence of examples is at most,

$$\varepsilon \;\leq\; \left(\frac{R + D_2}{\gamma}\right)^2 \quad . \tag{3}$$

This mistake bound does not assume that the data is linearly separable. However, whenever the data is linearly separable with margin γ, D_2 is 0 and the bound reduces to the bound given in Eq. (1). In this paper we also provide analysis in terms of the one-norm of the hinge losses which we denote by D_1 and is defined as,

$$D_1 = \sum_{t=1}^{T} \ell_t \quad . \tag{4}$$

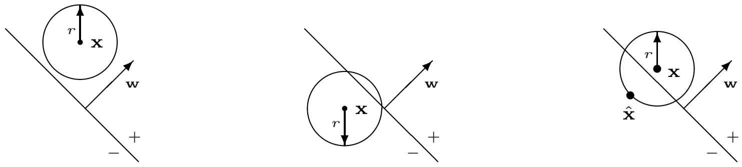

Fig. 1. An illustration of the three modes constituting the Ballseptron's update. The point $\mathbf{x}$ is labeled $+1$ and can be in one of three positions. Left: $\mathbf{x}$ is classified correctly by $\mathbf{w}$ with a margin greater than r. Middle: $\mathbf{x}$ is classified incorrectly by $\mathbf{w}$. Right: $\mathbf{x}$ is classified correctly but the ball of radius r is intersected by the separating hyper-plane. The point $\hat{\mathbf{x}}$ is used for updating $\mathbf{w}$.

While the analysis of the Perceptron employs the notion of separation with margin, the Perceptron algorithm itself is oblivious to the absolute value of the margin attained by any of the examples. Specifically, the Perceptron does not modify the hyperplane used for classification even for instances whose margin is very small so long as the predicted label is correct. While this property of the Perceptron has numerous advantages (see for example [8]) it also introduces some deficiencies which spurred work on algorithms that incorporate the notion of margin (see the references below). For instance, if we know that the data is linearly separable with a margin value γ we can deduce that our current hyperplane is not optimal and make use of this fact in updating the current hyperplane. In the next section we present an algorithm that updates its weight vector whenever it either makes a *prediction mistake* or suffers a *margin error*. Formally, let r be a positive scalar. We say that the algorithm suffers a margin error with respect to r if the current instance $\mathbf{x}_t$ is correctly classified but it lies too close to the separating hyper-plane, that is,

$$0 \; < \; y_t \left(\frac{\mathbf{w}_t}{\|\mathbf{w}_t\|} \cdot \mathbf{x}_t \right) \; \leq \; r \; . \tag{5}$$

Analogously to the definition of ε, we denote by $\tilde{\varepsilon}$ the number of margin errors our algorithm suffers on the sequence of examples.

Numerous online margin-based learning algorithms share similarities with the work presented in this paper. See for instance [12, 10, 11, 2, 5]. Many of the algorithms can be viewed as variants and enhancements of the Perceptron algorithm. However, the mistake bounds derived for these algorithms are not directly comparable to that of the Perceptron, especially when the examples are not linearly separable. In contrast, under certain conditions discussed in the sequel, the mistake bound for the algorithm described in this paper is superior to that of the Perceptron. Moreover, our analysis carries over to the original Perceptron algorithm.

The paper is organized as follows. We start in Sec. 2 with a description of our new online algorithm, the Ballseptron. In Sec. 3 we analyze the algorithm using the mistake bound model and discuss the implications on the original Perceptron algorithm. Next, in Sec. 4, we describe a multiclass extension of the Ballseptron algorithm. This extension is used in Sec. 5 in which we present few experimental results that underscore some of the algorithmic properties of the Ballseptron algorithm in the light of its formal analysis. Finally, we discuss possible future directions in Sec. 6.

2 The Ballseptron algorithm

```
PARAMETER: radius $r$
INITIALIZE: $\mathbf{w}_1 = \mathbf{0}$
For $t = 1, 2, \ldots$
    Receive an instance $\mathbf{x}_t$
    Predict: $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$
    If $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \leq 0$
        Update: $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\mathbf{x}_t$
    Else If $y_t(\mathbf{w}_t \cdot \mathbf{x}_t)/\|\mathbf{w}_t\| \leq r$
        Set: $\hat{\mathbf{x}}_t = \mathbf{x}_t - y_t r \mathbf{w}_t/\|\mathbf{w}_t\|$
        Update: $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\hat{\mathbf{x}}_t$
    Else // No margin mistake
        Update: $\mathbf{w}_{t+1} = \mathbf{w}_t$
    End
Endfor
```

Fig. 2. The Ballseptron algorithm.

In this section we present the Ballseptron algorithm which is a simple generalization of the classical Perceptron algorithm. As in the Perceptron algorithm, we maintain a single vector which is initially set to be the zero vector. On round t, we first receive an instance $\mathbf{x}_t$ and output a prediction according to the current vector, $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$. We then receive the correct label y_t. In case of a prediction mistake, i.e. $\hat{y}_t \neq y_t$, we suffer a unit loss and update $\mathbf{w}_t$ by adding to it the vector $y_t\mathbf{x}_t$. The updated vector constitutes the classifier to be used on the next round, thus $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\mathbf{x}_t$. In contrast to the Perceptron algorithm, we also update the classifier whenever the margin attained on $\mathbf{x}_t$ is smaller than a pre-specified parameter r. Formally, denote by $B(\mathbf{x}_t, r)$ the ball of radius r centered at $\mathbf{x}_t$. We impose the assumption that all the points in $B(\mathbf{x}_t, r)$ must have the same label as the center $\mathbf{x}_t$ (see also [6]). We now check if there is a point in $B(\mathbf{x}_t, r)$ which is misclassified by $\mathbf{w}_t$. If such a point exists then $\mathbf{w}_t$ intersects $B(\mathbf{x}_t, r)$ into two parts. We now generate a pseudo-instance, denoted $\hat{\mathbf{x}}_t$ which corresponds to the point in $B(\mathbf{x}_t, r)$ attaining the worst (negative) margin with respect to $\mathbf{w}_t$. (See Fig. 1 for an illustration.) This is obtained by moving r units away from $\mathbf{x}_t$ in the direction of $-y_t\mathbf{w}_t$, that is $\hat{\mathbf{x}}_t = \mathbf{x}_t - \frac{y_t r}{\|\mathbf{w}_t\|}\mathbf{w}_t$. To show this formally, we solve the following constrained minimization problem,

$$\hat{\mathbf{x}}_t = \operatorname*{argmin}_{\mathbf{x} \in B(\mathbf{x}_t, r)} y_t(\mathbf{w}_t \cdot \mathbf{x}) \quad . \tag{6}$$

To find $\hat{\mathbf{x}}_t$ we recast the constraint $\mathbf{x} \in B(\mathbf{x}_t, r)$ as $\|\mathbf{x} - \mathbf{x}_t\|^2 \leq r^2$. Note that both the objective function $y_t(\mathbf{w}_t \cdot \mathbf{x})$ and the constraint $\|\mathbf{x} - \mathbf{x}_t\|^2 \leq r^2$ are convex in $\mathbf{x}$. In addition, the relative interior of the $B(\mathbf{x}_t, r)$ is not empty. Thus, Slater's optimality conditions hold and we can find $\hat{\mathbf{x}}_t$ by examining the saddle point of the problem's Lagrangian which is, $L(\mathbf{x}, \alpha) = y_t(\mathbf{w}_t \cdot \mathbf{x}) + \alpha\left(\|\mathbf{x} - \mathbf{x}_t\|^2 - r^2\right)$. Taking the derivative of the Lagrangian w.r.t. each of the components of $\mathbf{x}$ and setting the resulting vector to zero gives,

$$y_t\mathbf{w}_t + 2\alpha(\mathbf{x} - \mathbf{x}_t) = 0 \quad . \tag{7}$$

Since $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) > 0$ (otherwise, we simply undergo a simple Perceptron update) we have that $\mathbf{w}_t \neq \mathbf{0}$ and $\alpha > 0$. Hence we get that the solution of Eq. (7) is $\hat{\mathbf{x}}_t = \mathbf{x}_t - (y_t/2\alpha)\mathbf{w}_t$. To find α we use the complementary slackness condition. That is, since $\alpha > 0$ we must have that $\|\mathbf{x} - \mathbf{x}_t\| = r$. Replacing $\mathbf{x} - \mathbf{x}_t$ with $-y_t\mathbf{w}_t/(2\alpha)$, the slackness condition yields that, $\frac{\|\mathbf{w}_t\|}{2\alpha} = r$ which let us express $\frac{1}{2\alpha}$ as $\frac{r}{\|\mathbf{w}_t\|}$. We thus get that $\hat{\mathbf{x}}_t = \mathbf{x}_t - \frac{y_t r}{\|\mathbf{w}_t\|}\mathbf{w}_t$. By construction, if $y_t(\mathbf{w}_t \cdot \hat{\mathbf{x}}_t) > 0$ we know that all the points in the ball of radius r centered at $\mathbf{x}_t$ are correctly classified and we set $\mathbf{w}_{t+1} = \mathbf{w}_t$.

(See also the left-most plot in Fig. 1.) If on the other hand $y_t(\mathbf{w}_t \cdot \hat{\mathbf{x}}_t) \leq 0$ (right-most plot in Fig. 1) we use $\hat{\mathbf{x}}_t$ as a pseudo-example and set $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\hat{\mathbf{x}}_t$.

Note that we can rewrite the condition $y_t(\mathbf{w}_t \cdot \hat{\mathbf{x}}_t) \leq 0$ as $y_t(\mathbf{w}_t \cdot \mathbf{x}_t)/\|\mathbf{w}_t\| \leq r$. The pseudocode of the Ballseptron algorithm is given in Fig. 2. and an illustration of the different cases encountered by the algorithm is given in Fig. 1. Last, we would like to note in passing that $\mathbf{w}_t$ can be written as a linear combination of the instances, $\mathbf{w}_t = \sum_{i=1}^{t-1} \alpha_t \mathbf{x}_t$, and therefore, $\mathbf{w}_t \cdot \mathbf{x}_t = \sum_{i=1}^{t-1} \alpha_i(\mathbf{x}_i \cdot \mathbf{x}_t)$. The inner products $\mathbf{x}_i \cdot \mathbf{x}_t$ can be replaced with an inner products defined via a Mercer kernel, $K(\mathbf{x}_i, \mathbf{x}_t)$, without any further changes to our derivation. Since the analysis in the next section does not depend on the dimensionality of the instances, all of the formal results still hold when the algorithm is used in conjunction with kernel functions.

3 Analysis

In this section we analyze the Ballseptron algorithm. Analogous to the Perceptron bounds, the bounds that we obtain do not depend on the dimension of the instances but rather on the geometry of the problem expressed via the margin of the instances and the radius of the sphere enclosing the instances. As mentioned above, most of our analysis carries over to the original Perceptron algorithm and we therefore dedicate the last part of this section to a discussion of the implications for the original Perceptron algorithm. A desirable property of the Ballseptron would have been that it does not make more prediction mistakes than the Perceptron algorithm. Unfortunately, without any restrictions on the radius r that the Ballseptron algorithm employs, such a property cannot be guaranteed. For example, suppose that the instances are drawn from $\mathbb{R}$ and all the input-label pairs in the sequence $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ are the same and equal to $(\mathbf{x}, y) = (1, 1)$. The Perceptron algorithm makes a single mistake on this sequence. However, if the radius r that is relayed to the Ballseptron algorithm is 2 then the algorithm would make $T/2$ prediction mistakes on the sequence. The crux of this failure to achieve a small number of mistakes is due to the fact that the radius r was set to an excessively large value. To achieve a good mistake bound we need to ensure that r is set to be less than the target margin γ employed by the competing hypothesis $\mathbf{u}$. Indeed, our first theorem implies that the Ballseptron attains the same mistake bound as the Perceptron algorithm provided that r is small enough.

Theorem 1. *Let $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_T, y_T)$ be a sequence of instance-label pairs where $\mathbf{x}_t \in \mathbb{R}^n$, $y_t \in \{-1, +1\}$, and $\|\mathbf{x}_t\| \leq R$ for all t. Let $\mathbf{u} \in \mathbb{R}^n$ be a vector whose norm is 1, $0 < \gamma \leq R$ an arbitrary scalar, and denote $\ell_t = \max\{0, \gamma - y_t\mathbf{u} \cdot \mathbf{x}_t\}$. Let D_2 be as defined by Eq. (2). Assume that the Ballseptron algorithm is run with a parameter r which satisfies $0 \leq r < (\sqrt{2} - 1)\,\gamma$. Then, the number of prediction mistakes the Ballseptron makes on the sequence is at most,*

$$\left(\frac{R + D_2}{\gamma} \right)^2 .$$

Proof. We prove the theorem by bounding $\mathbf{w}_{T+1} \cdot \mathbf{u}$ from below and above while comparing the two bounds. Starting with the upper bound, we need to examine three different cases for every t. If $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \leq 0$ then $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\mathbf{x}_t$ and therefore,

$$\|\mathbf{w}_{t+1}\|^2 \;=\; \|\mathbf{w}_t\|^2 + \|\mathbf{x}_t\|^2 + 2y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \;\leq\; \|\mathbf{w}_t\|^2 + \|\mathbf{x}_t\|^2 \;\leq\; \|\mathbf{w}_t\|^2 + R^2 \;\;.$$

In the second case where $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) > 0$ yet the Ballseptron suffers a margin mistake, we know that $y_t(\mathbf{w}_t \cdot \hat{\mathbf{x}}_t) \leq 0$ and thus get

$$\|\mathbf{w}_{t+1}\|^2 \;=\; \|\mathbf{w}_t + y_t\hat{\mathbf{x}}_t\|^2 \;=\; \|\mathbf{w}_t\|^2 + \|\hat{\mathbf{x}}_t\|^2 + 2y_t(\mathbf{w}_t \cdot \hat{\mathbf{x}}_t) \;\leq\; \|\mathbf{w}_t\|^2 + \|\hat{\mathbf{x}}_t\|^2 \;\;.$$

Recall that $\hat{\mathbf{x}}_t = \mathbf{x}_t - y_t r \mathbf{w}_t / \|\mathbf{w}_t\|$ and therefore,

$$\|\hat{\mathbf{x}}_t\|^2 = \|\mathbf{x}_t\|^2 + r^2 - 2y_t r(\mathbf{x}_t \cdot \mathbf{w}_t)/\|\mathbf{w}_t\| \;<\; \|\mathbf{x}_t\|^2 + r^2 \;\leq\; R^2 + r^2 \;\;.$$

Finally in the third case where $y_t(\mathbf{w}_t \cdot \hat{\mathbf{x}}_t) > 0$ we have $\|\mathbf{w}_{t+1}\|^2 = \|\mathbf{w}_t\|^2$. We can summarize the three different scenarios by defining two variables: $\tau_t \in \{0,1\}$ which is 1 iff $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \leq 0$ and similarly $\tilde{\tau}_t \in \{0,1\}$ which is 1 iff $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) > 0$ and $y_t(\mathbf{w}_t \cdot \hat{\mathbf{x}}_t) \leq 0$. Unraveling the bound on the norm of $\mathbf{w}_{T+1}$ while using the definitions of τ_t and $\tilde{\tau}_t$ gives,

$$\|\mathbf{w}_{T+1}\|^2 \;\leq\; R^2 \sum_{t=1}^{T} \tau_t + (R^2 + r^2) \sum_{t=1}^{T} \tilde{\tau}_t \;\;.$$

Let us now denote by $\varepsilon = \sum_{t=1}^{T} \tau_t$ the number of mistakes the Ballseptron makes and analogously by $\tilde{\varepsilon} = \sum_{t=1}^{T} \tilde{\tau}_t$ the number of margin errors of the Ballseptron. Using the two definitions along with the Cauchy-Schwartz inequality yields that,

$$\mathbf{w}_{T+1} \cdot \mathbf{u} \;\leq\; \|\mathbf{w}_{T+1}\| \, \|\mathbf{u}\| \;=\; \|\mathbf{w}_{T+1}\| \;\leq\; \sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2 + r^2)} \;\;. \tag{8}$$

This provides us with an upper bound on $\mathbf{w}_{T+1} \cdot \mathbf{u}$. We now turn to derive a lower bound on $\mathbf{w}_{T+1} \cdot \mathbf{u}$. As in the derivation of the upper bound, we need to consider three cases. The definition of ℓ_t immediately implies that $\ell_t \geq \gamma - y_t \mathbf{x}_t \cdot \mathbf{u}$. Hence, in the first case (a prediction mistake), we can bound $\mathbf{w}_{t+1} \cdot \mathbf{u}$ as follows,

$$\mathbf{w}_{t+1} \cdot \mathbf{u} \;=\; (\mathbf{w}_t + y_t\mathbf{x}_t) \cdot \mathbf{u} \;\geq\; \mathbf{w}_t \cdot \mathbf{u} + \gamma - \ell_t \;\;,$$

In the second case (a margin error) the Ballseptron's update is $\mathbf{w}_{t+1} = \mathbf{w}_t + y_t\hat{\mathbf{x}}_t$ which results in the following bound,

$$\begin{aligned}\mathbf{w}_{t+1} \cdot \mathbf{u} = (\mathbf{w}_t + y_t\hat{\mathbf{x}}_t) \cdot \mathbf{u} \;=\; & \left(\mathbf{w}_t + y_t\mathbf{x}_t - r\frac{\mathbf{w}_t}{\|\mathbf{w}_t\|}\right) \cdot \mathbf{u} \\ \geq \mathbf{w}_t \cdot \mathbf{u} + \gamma - \ell_t - r & \left(\frac{\mathbf{w}_t}{\|\mathbf{w}_t\|} \cdot \mathbf{u}\right) \;\;.\end{aligned}$$

Since the norm of $\mathbf{u}$ is assumed to be 1, by using Cauchy-Schwartz inequality we can bound $\frac{\mathbf{w}_t}{\|\mathbf{w}_t\|} \cdot \mathbf{u}$ by 1. We thus get that, $\mathbf{w}_{t+1} \cdot \mathbf{u} \geq \mathbf{w}_t \cdot \mathbf{u} + \gamma - \ell_t - r$. Finally, on rounds for which there was neither a prediction mistake nor a margin error we immediately get that, $\mathbf{w}_{t+1} \cdot \mathbf{u} = \mathbf{w}_t \cdot \mathbf{u}$. Combining the three cases while using the definitions of $\tau_t, \tilde{\tau}_t, \varepsilon$ and $\tilde{\varepsilon}$ we get that,

$$\mathbf{w}_{T+1} \cdot \mathbf{u} \ \geq \ \varepsilon\gamma + \tilde{\varepsilon}(\gamma - r) - \sum_{t=1}^{T} (\tau_t + \tilde{\tau}_t)\ell_t \ . \tag{9}$$

We now apply Cauchy-Schwartz inequality once more to obtain that,

$$\sum_{t=1}^{T} (\tau_t + \tilde{\tau}_t)\ell_t \ \leq \ \left(\sum_{t=1}^{T} (\tau_t + \tilde{\tau}_t)^2 \right)^{\frac{1}{2}} \left(\sum_{t=1}^{T} (\ell_t)^2 \right)^{\frac{1}{2}} \ = \ D_2 \sqrt{\varepsilon + \tilde{\varepsilon}} \ .$$

Combining the above inequality with Eq. (9) we get the following lower bound on $\mathbf{w}_{T+1} \cdot \mathbf{u}$,

$$\mathbf{w}_{T+1} \cdot \mathbf{u} \ \geq \ \varepsilon\gamma + \tilde{\varepsilon}(\gamma - r) - D_2 \sqrt{\varepsilon + \tilde{\varepsilon}} \ . \tag{10}$$

We now tie the lower bound on $\mathbf{w}_{T+1} \cdot \mathbf{u}$ from Eq. (10) with the upper bound from Eq. (8) to obtain that,

$$\sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2 + r^2)} \geq \varepsilon\gamma + \tilde{\varepsilon}(\gamma - r) - D_2 \sqrt{\varepsilon + \tilde{\varepsilon}} \ . \tag{11}$$

Let us now denote by $g(\varepsilon, \tilde{\varepsilon})$ the difference between the two sides of the above equation, that is,

$$g(\varepsilon, \tilde{\varepsilon}) = \varepsilon\gamma + \tilde{\varepsilon}(\gamma - r) - \sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2 + r^2)} - D_2 \sqrt{\varepsilon + \tilde{\varepsilon}} \ . \tag{12}$$

Eq. (11) implies that $g(\varepsilon, \tilde{\varepsilon}) \leq 0$ for the particular values of ε and $\tilde{\varepsilon}$ obtained by the Ballseptron algorithm. We now use the this fact to show that ε cannot exceed $((R + D_2)/\gamma)^2$. First note that if $\tilde{\varepsilon} = 0$ then g is a quadratic function in $\sqrt{\varepsilon}$ and therefore $\sqrt{\varepsilon}$ is at most the positive root of the equation $g(\varepsilon, 0) = 0$ which is $(R + D_2)/\gamma$. We thus get,

$$g(\varepsilon, 0) \leq 0 \quad \Rightarrow \quad \varepsilon \ \leq \ \left(\frac{R + D_2}{\gamma} \right)^2 \ .$$

If $\tilde{\varepsilon} \geq 1$ and $\varepsilon + \tilde{\varepsilon} \leq ((R + D_2)/\gamma)^2$ then the bound stated in the theorem immediately holds. Therefore, we only need to analyze the case in which $\tilde{\varepsilon} \geq 1$ and $\varepsilon + \tilde{\varepsilon} > ((R + D_2)/\gamma)^2$. In this case we derive the mistake bound by showing first that the function $g(\varepsilon, \tilde{\varepsilon})$ is monotonically increasing in $\tilde{\varepsilon}$ and therefore $g(\varepsilon, 0) \leq g(\varepsilon, \tilde{\varepsilon}) \leq 0$. To prove the monotonicity of g we need the following simple inequality which holds for $a > 0$, $b \geq 0$ and $c > 0$,

$$\sqrt{a + b + c} - \sqrt{a + b} \ = \ \frac{c}{\sqrt{a + b + c} + \sqrt{a + b}} \ < \ \frac{c}{2\sqrt{a}} \ . \tag{13}$$

Let us now examine $g(\varepsilon, \tilde{\varepsilon}+1) - g(\varepsilon, \tilde{\varepsilon})$. Expanding the definition of g from Eq. (12) and using Eq. (13) we get that,

$$\begin{aligned} g(\varepsilon, \tilde{\varepsilon}+1) - g(\varepsilon, \tilde{\varepsilon}) &= \gamma - r - \sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2+r^2) + R^2 + r^2} \\ &\quad + \sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2+r^2)} - D_2\sqrt{\varepsilon + \tilde{\varepsilon} + 1} + D_2\sqrt{\varepsilon + \tilde{\varepsilon}} \\ &\geq \gamma - r - \frac{R^2+r^2}{2R\sqrt{\varepsilon+\tilde{\varepsilon}}} - \frac{D_2}{2\sqrt{\varepsilon+\tilde{\varepsilon}}} \\ &= \gamma - r - \frac{R + D_2 + r^2/R}{2\sqrt{\varepsilon+\tilde{\varepsilon}}} \ . \end{aligned}$$

We now use the assumption that $\varepsilon + \tilde{\varepsilon} > ((R+D_2)/\gamma)^2$ and that $\gamma \leq R$ to get that,

$$\begin{aligned} g(\varepsilon, \tilde{\varepsilon}+1) - g(\varepsilon, \tilde{\varepsilon}) &\geq \gamma \left(1 - \frac{r}{\gamma} - \frac{R+D_2}{2\gamma\sqrt{\varepsilon+\tilde{\varepsilon}}} - \frac{r^2}{2R(R+D_2)}\right) \\ &> \gamma \left(1 - \frac{r}{\gamma} - \frac{1}{2} - \frac{1}{2}\left(\frac{r}{\gamma}\right)^2\right) \ . \end{aligned} \qquad (14)$$

The condition that $r \leq (\sqrt{2}-1)\,\gamma$ implies that the term $0.5 - r/\gamma - 0.5(r/\gamma)^2$ is strictly positive. We have thus shown that $g(\varepsilon, \tilde{\varepsilon}+1) - g(\varepsilon, \tilde{\varepsilon}) > 0$ hence g is monotonically increasing in $\tilde{\varepsilon}$. Therefore, from Eq. (11) we get that $0 \geq g(\varepsilon, \tilde{\varepsilon}) > g(\varepsilon, 0)$. Finally, as already argued above, the condition $0 \geq g(\varepsilon, 0)$ ensures that $\varepsilon \leq ((R+D_2)/\gamma)^2$. This concludes our proof. □

The above bound ensures that whenever r is less than $(\sqrt{2}-1)\,\gamma$, the Ballseptron mistake bound is as good as Freund and Schapire's [9] mistake bound for the Perceptron. The natural question that arises is whether the Ballseptron entertains any advantage over the less complex Perceptron algorithm. As we now argue, the answer is yes so long as the number of margin errors, $\tilde{\varepsilon}$, is strictly positive. First note that if $\varepsilon + \tilde{\varepsilon} \leq ((R+D_2)/\gamma)^2$ and $\tilde{\varepsilon} > 0$ then $\varepsilon \leq ((R+D_2)/\gamma)^2 - \tilde{\varepsilon}$ which is strictly smaller than the mistake bound from [9]. The case when $\varepsilon + \tilde{\varepsilon} > ((R+D_2)/\gamma)^2$ needs some deliberation. To simplify the derivation let $\beta = 0.5 - r/\gamma - 0.5\,(r/\gamma)^2$. The proof of Thm. 1 implies that $g(\varepsilon, \tilde{\varepsilon}+1) - g(\varepsilon, \tilde{\varepsilon}) \geq \beta\gamma$. From the same proof we also know that $g(\varepsilon, \tilde{\varepsilon}) \leq 0$. We thus get that $g(\varepsilon, 0) + \tilde{\varepsilon}\beta\gamma \ \leq \ g(\varepsilon, \tilde{\varepsilon}) \ \leq \ 0$. Expanding the term $g(\varepsilon, 0) + \tilde{\varepsilon}\beta\gamma$ we get the following inequality,

$$\varepsilon\gamma - \sqrt{\varepsilon R^2} - D_2\sqrt{\varepsilon} + \tilde{\varepsilon}\beta\gamma \ = \ \varepsilon\gamma - \sqrt{\varepsilon}(R+D_2) + \tilde{\varepsilon}\beta\gamma \ \leq \ 0 \ . \qquad (15)$$

The left-hand side of Eq. (15) is a quadratic function in $\sqrt{\varepsilon}$. Thus, $\sqrt{\varepsilon}$ cannot exceed the positive root of this function. Therefore, the number of prediction mistakes, ε, can be bounded above as follows,

$$\begin{aligned}\varepsilon &\leq \left(\frac{R + D_2 + \sqrt{(R+D_2)^2 - 4\beta\gamma^2\tilde{\varepsilon}}}{2\gamma}\right)^2 \\ &\leq \frac{(R+D_2)^2 + 2\,(R+D_2)\,\sqrt{(R+D_2)^2 - 4\beta\gamma^2\tilde{\varepsilon}} + (R+D_2)^2 - 4\beta\gamma^2\tilde{\varepsilon}}{4\gamma^2} \\ &\leq \left(\frac{R+D_2}{\gamma}\right)^2 \;-\; \beta\tilde{\varepsilon}\ \ .\end{aligned}$$

We have thus shown that whenever the number of margin errors $\tilde{\varepsilon}$ is strictly positive, the number of prediction mistakes is smaller than $((R+D_2)/\gamma)^2$, the bound obtained by Freund and Schapire for the Perceptron algorithm. In other words, the mistake bound we obtained puts a cap on a function which depends both on ε and on $\tilde{\varepsilon}$. Margin errors naturally impose more updates to the classifier, yet they come at the expense of sheer prediction mistakes. Thus, the Ballseptron algorithm is most likely to suffer a smaller number of prediction mistakes than the standard Perceptron algorithm. We summarize these facts in the following corollary.

Corollary 1. *Under the same assumptions of Thm. 1, the number of prediction mistakes the Ballseptron algorithm makes is at most,*

$$\left(\frac{R+D_2}{\gamma}\right)^2 \;-\; \tilde{\varepsilon}\left(\frac{1}{2} - \frac{r}{\gamma} - \frac{1}{2}\left(\frac{r}{\gamma}\right)^2\right)\quad ,$$

where $\tilde{\varepsilon}$ is the number of margin errors of the Ballseptron algorithm.

Thus far, we derived mistake bounds that depend on R,γ, and D_2 which is the square-root of the sum of the squares of hinge-losses. We now turn to an analogous mistake bound which employs D_1 instead of D_2. Our proof technique is similar to the proof of Thm. 1 and we thus confine the next proof solely to the modifications that are required.

Theorem 2. *Under the same assumptions of Thm. 1, the number of prediction mistakes the Ballseptron algorithm makes is at most,*

$$\left(\frac{R + \sqrt{\gamma\, D_1}}{\gamma}\right)^2 \quad .$$

Proof. Following the proof outline of Thm. 1, we start by modifying the lower bound on $\mathbf{w}_{T+1}\cdot\mathbf{u}$. First, note that the lower bound given by Eq. (9) still holds. In addition, $\tau_t + \tilde{\tau}_t \leq 1$ for all t since on each round there exists a mutual exclusion between a prediction mistake and a margin error. We can therefore simplify Eq. (9) and rewrite it as, $\mathbf{w}_{T+1}\cdot\mathbf{u} \;\geq\; \varepsilon\gamma - \sum_{t=1}^{T}\ell_t + \tilde{\varepsilon}(\gamma - r)$. Combining this lower bound on $\mathbf{w}_{T+1}\cdot\mathbf{u}$ with the upper bound on $\mathbf{w}_{T+1}\cdot\mathbf{u}$ given in Eq. (8) we get that,

$$\varepsilon\gamma + \tilde{\varepsilon}(\gamma - r) - \sum_{t=1}^{T}\ell_t \;\leq\; \sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2 + r^2)}\ \ . \tag{16}$$

Similar to the definition of g from Thm. 1, we define the following auxiliary function,

$$q(\varepsilon,\tilde{\varepsilon}) = \varepsilon\gamma + \tilde{\varepsilon}(\gamma - r) - \sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2 + r^2)} - D_1 \ \ .$$

Thus, Eq. (16) yields that $q(\varepsilon,\tilde{\varepsilon}) \leq 0$. We now show that $q(\varepsilon,\tilde{\varepsilon}) \leq 0$ implies that ε cannot exceed $((R + \sqrt{\gamma D_1})/\gamma)^2$. First, note that if $\tilde{\varepsilon} = 0$ then q becomes a quadratic function in $\sqrt{\varepsilon}$. Therefore, $\sqrt{\varepsilon}$ cannot be larger than the positive root of the equation $q(\varepsilon, 0) = 0$ which is,

$$\frac{R + \sqrt{R^2 + 4\gamma D_1}}{2\gamma} \ \leq \ \frac{R + \sqrt{\gamma D_1}}{\gamma} \ \ .$$

We have therefore shown that,

$$q(\varepsilon, 0) \leq 0 \quad \Rightarrow \quad \varepsilon \ \leq \ \left(\frac{R + \sqrt{\gamma D_1}}{\gamma}\right)^2 \ \ .$$

We thus assume that $\tilde{\varepsilon} \geq 1$. Again, if $\varepsilon + \tilde{\varepsilon} \leq (R/\gamma)^2$ then the bound stated in the theorem immediately holds. We are therefore left with the case $\varepsilon + \tilde{\varepsilon} > (R/\gamma)^2$ and $\tilde{\varepsilon} > 0$. To prove the theorem we show that $q(\varepsilon,\tilde{\varepsilon})$ is monotonically increasing in $\tilde{\varepsilon}$. Expanding the function q and using as before the bound given in Eq. (13) we get that,

$$\begin{aligned} q(\varepsilon,\tilde{\varepsilon}+1) - q(\varepsilon,\tilde{\varepsilon}) &= \gamma - r - \sqrt{\varepsilon R^2 + (\tilde{\varepsilon}+1)(R^2 + r^2)} + \sqrt{\varepsilon R^2 + \tilde{\varepsilon}(R^2 + r^2)} \\ &> \gamma - r - \frac{R^2 + r^2}{2\sqrt{(\varepsilon + \tilde{\varepsilon})R^2}} \ = \ \gamma - r - \frac{R + r^2/R}{2\sqrt{\varepsilon + \tilde{\varepsilon}}} \ \ . \end{aligned}$$

Using the assumption that $\varepsilon + \tilde{\varepsilon} > (R/\gamma)^2$ and that $\gamma \leq R$ let us further bound the above as follows,

$$q(\varepsilon,\tilde{\varepsilon}+1) - q(\varepsilon,\tilde{\varepsilon}) > \gamma - r - \frac{\gamma}{2} - \frac{\gamma r^2}{2R^2} \ \geq \ \gamma\left(\frac{1}{2} - \frac{r}{\gamma} - \frac{1}{2}\left(\frac{r}{\gamma}\right)^2\right) \ \ .$$

The assumption that $r \leq (\sqrt{2} - 1)\gamma$ yields that $q(\varepsilon,\tilde{\varepsilon}+1) - q(\varepsilon,\tilde{\varepsilon}) \geq 0$ and therefore $q(\varepsilon,\tilde{\varepsilon})$ is indeed monotonically increasing in $\tilde{\varepsilon}$ for $\varepsilon + \tilde{\varepsilon} > R^2/\gamma^2$. Combining the inequality $q(\varepsilon,\tilde{\varepsilon}) \leq 0$ with the monotonicity property we get that $q(\varepsilon, 0) \leq q(\varepsilon,\tilde{\varepsilon}) \leq 0$ which in turn yields the bound of the theorem. This concludes our proof. □

The bound of Thm. 2 is similar to the bound of Thm. 1. The natural question that arises is whether we can obtain a tighter mistake bound whenever we know the number of margin errors $\tilde{\varepsilon}$. As for the bound based on D_2, the answer for the D_1-based bound is affirmative. Recall that we define the value of $1/2 - r/\gamma - 1/2(r/\gamma)^2$ by β. We now show that the number of prediction mistakes is bounded above by,

$$\varepsilon \ \leq \ \left(\frac{R + \sqrt{\gamma D_1}}{\gamma}\right)^2 \ - \ \tilde{\varepsilon}\beta \ \ . \tag{17}$$

First, if $\varepsilon + \tilde{\varepsilon} \leq (R/\gamma)^2$ then the bound above immediately holds. In the proof of Thm. 2 we have shown that if $\varepsilon + \tilde{\varepsilon} > (R/\gamma)^2$ then $q(\varepsilon,\tilde{\varepsilon}+1) - q(\varepsilon,\tilde{\varepsilon}) \geq \beta\gamma$.

Therefore, $q(\varepsilon, \tilde{\varepsilon}) \geq q(\varepsilon, 0) + \tilde{\varepsilon}\beta\gamma$. Recall that Eq. (16) implies that $q(\varepsilon, \tilde{\varepsilon}) \leq 0$ and thus we get that $q(\varepsilon, 0) + \tilde{\varepsilon}\beta\gamma \leq 0$ yielding the following,

$$\varepsilon\gamma - R\sqrt{\varepsilon} - D_1 + \tilde{\varepsilon}\beta\gamma \ \leq \ 0 \ .$$

The left-hand side of the above inequality is yet again a quadratic function in $\sqrt{\varepsilon}$. Therefore, once more $\sqrt{\varepsilon}$ is no bigger than the positive root of the equation and we get that,

$$\sqrt{\varepsilon} \leq \frac{R + \sqrt{R^2 + 4\gamma D_1 - 4\gamma^2\beta\tilde{\varepsilon}}}{2\gamma} \ ,$$

and thus,

$$\begin{aligned} \varepsilon &\leq \frac{R^2 + 2\,R\,\sqrt{R^2 + 4\gamma D_1 - 4\gamma^2\beta\tilde{\varepsilon}} + R^2 + 4\gamma D_1 - 4\gamma^2\beta\tilde{\varepsilon}}{4\gamma^2} \\ &\leq \frac{R^2 + 2R\sqrt{\gamma D_1} + \gamma D_1}{\gamma^2} - \beta\tilde{\varepsilon} \ , \end{aligned}$$

which can be translated to the bound on ε from Eq. (17).

Summing up, the Ballseptron algorithm entertains two mistake bounds: the first is based on the root of the cumulative square of losses (D_2) while the second is based directly on the cumulative sum of hinge losses (D_1). Both bounds imply that the Ballseptron would make fewer prediction mistakes than the original Perceptron algorithm so long as the Ballseptron suffers margin errors along its run. Since margin errors are likely to occur for reasonable choices of r, the Ballseptron is likely to attain a smaller number of prediction mistakes than the Perceptron algorithm. Indeed, preliminary experiments reported in Sec. 5 indicate that for a wide range of choices for r the number of online prediction mistakes of the Ballseptron is significantly lower than that of the Perceptron.

The bounds of Thm. 1 and Thm. 2 hold for any $r \leq (\sqrt{2} - 1)\gamma$, in particular for $r = 0$. When $r = 0$, the Ballseptron algorithm reduces to the Perceptron algorithm. In the case of Thm. 1 the resulting mistake bound for $r = 0$ is identical to the bound of Freund and Schapire [9]. Our proof technique though is substantially different than the one in [9] which embeds each instance in a high dimensional space rendering the problem separable. Setting r to zero in Thm. 2 yields a new mistake bound for the Perceptron with $\sqrt{\gamma D_1}$ replacing D_2 in the bound. The latter bound is likely to be tighter in the presence of noise which may cause large margin errors. Specifically, the bound of Thm. 2 is better than that of Thm. 1 when

$$\gamma \sum_{t=1}^{T} \ell_t \ \leq \ \sum_{t=1}^{T} \ell_t^2 \ .$$

We therefore expect the bound in Thm. 1 to be better when ℓ_t is small and otherwise the new bound is likely to be better. We further investigate the difference between the two bounds in Sec. 5.

4 An Extension to Multiclass Problems

In this section we describe a generalization of the Ballseptron to the task of multiclass classification. For concreteness we assume that there are k different possible labels and denote the set of all possible labels by $\mathcal{Y} = \{1, \ldots, k\}$. There are several adaptations of the Perceptron algorithm to multiclass settings (see for example [5, 7, 16, 17]), many of which are also applicable to the Ballseptron. We now outline one possible multiclass extension in which we associate a weight vector with each class. Due to the lack of space proofs of the mistake bound obtained by our construction are omitted. Let $\mathbf{w}^r$ denote the weight vector associated with a label $r \in \mathcal{Y}$. We also refer to $\mathbf{w}^r$ as the r'th prototype. As in the binary case we initialize each of the prototypes to be the zero vector. The predicted label of an instance $\mathbf{x}_t$ is defined as,

$$\hat{y}_t \;=\; \operatorname*{argmax}_{r \in \mathcal{Y}} \mathbf{w}_t^r \cdot \mathbf{x}_t \quad .$$

Upon receiving the correct label y_t, if $\hat{y}_t \neq y_t$ we perform the following update which is a multiclass generalization of the Perceptron rule,

$$\mathbf{w}_{t+1}^{y_t} = \mathbf{w}_t^{y_t} + \mathbf{x}_t \;;\; \mathbf{w}_{t+1}^{\hat{y}_t} = \mathbf{w}_t^{\hat{y}_t} - \mathbf{x}_t \;;\; \mathbf{w}_{t+1}^{r} = \mathbf{w}_t^{r} \;(\forall r \in \mathcal{Y} \setminus \{y_t, \hat{y}_t\}) \quad . \tag{18}$$

In words, we add the instance $\mathbf{x}_t$ to the prototype of the correct label and subtract $\mathbf{x}_t$ from the prototype of $\hat{y}_t$. The rest of the prototypes are left intact. If $\hat{y}_t = y_t$, we check whether we still encounter a margin error. Let $\tilde{y}_t$ denote the index of the prototype whose inner-product with $\mathbf{x}_t$ is the second largest, that is,

$$\tilde{y}_t \;=\; \operatorname*{argmax}_{y \neq y_t} \left(\mathbf{w}_t^{y} \cdot \mathbf{x}_t\right) \quad .$$

Analogous to the definition of $\hat{\mathbf{x}}_t$ in the binary classification problem, we define $\hat{\mathbf{x}}_t$ as the solution to the following optimization problem,

$$\hat{\mathbf{x}}_t \;=\; \operatorname*{argmin}_{\mathbf{x} \in B(\mathbf{x}_t, r)} \left(\mathbf{w}_t^{y_t} \cdot \mathbf{x} - \mathbf{w}_t^{\hat{y}_t} \cdot \mathbf{x}\right) \quad . \tag{19}$$

Note that if $\mathbf{w}_t^{y_t} \cdot \hat{\mathbf{x}}_t > \mathbf{w}_t^{\tilde{y}_t} \cdot \hat{\mathbf{x}}_t$ then all the points in $B(\mathbf{x}_t, r)$ are labeled correctly and there is no margin error. If this is the case we leave all the prototypes intact. If however $\mathbf{w}_t^{y_t} \cdot \hat{\mathbf{x}}_t \leq \mathbf{w}_t^{\tilde{y}_t} \cdot \hat{\mathbf{x}}_t$ we perform the update given by Eq. (18) using $\hat{\mathbf{x}}_t$ instead of $\mathbf{x}_t$ and $\tilde{y}_t$ instead of $\hat{y}_t$. The same derivation described in Sec. 2, yields that $\hat{\mathbf{x}}_t = \mathbf{x}_t + r(\mathbf{w}_t^{\tilde{y}_t} - \mathbf{w}_t^{y_t}) / \|\mathbf{w}_t^{\tilde{y}_t} - \mathbf{w}_t^{y_t}\|$. The analysis of the Ballseptron from Sec. 3 can be adapted to the multiclass version of the algorithm as we now briefly describe. Let $\{\mathbf{u}^1, \ldots, \mathbf{u}^k\}$ be a set of k prototype vectors such that $\sum_{i=1}^{k} \|\mathbf{u}^i\|^2 = 1$. For each multiclass example $(\mathbf{x}_t, y_t)$ define the hinge-loss of the above prototypes on this example as,

$$\ell_t \;=\; \max\left\{0 \,,\; \max_{y \neq y_t} \left(\gamma - (\mathbf{u}^{y_t} - \mathbf{u}^{y}) \cdot \mathbf{x}_t\right)\right\} \quad .$$

We now redefine D_2 and D_1 using the above definition of the hinge-loss. In addition, we need to redefine R to be $R = \sqrt{2} \max_t \|\mathbf{x}_t\|$. Using these definitions, it can be shown that slightly weaker versions of the bounds from Sec. 3 can be obtained.

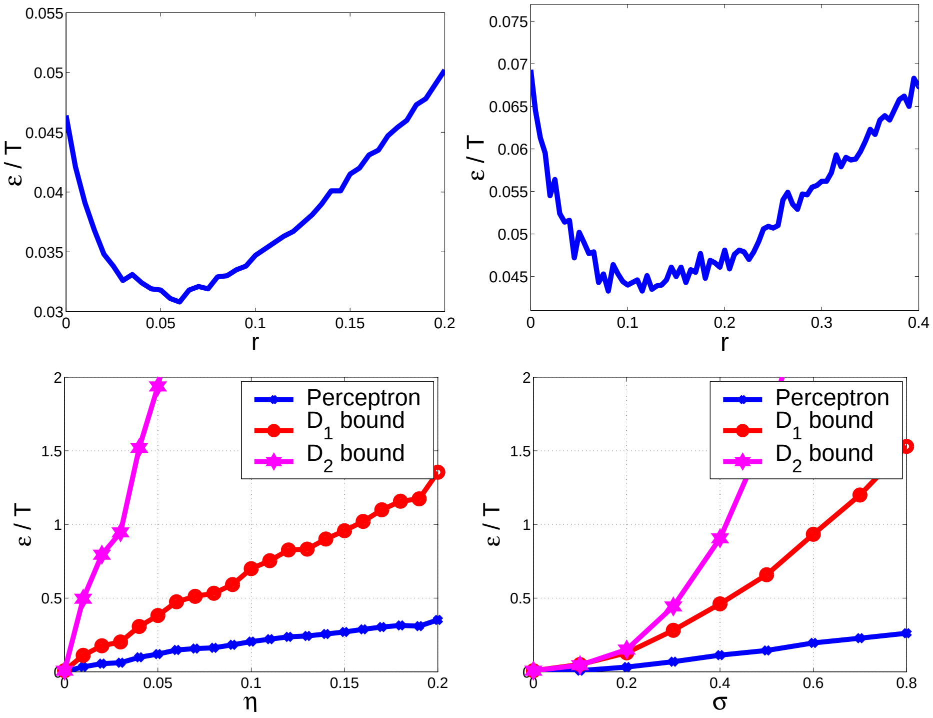

Fig. 3. Top plots: The fraction of prediction mistakes (ε/T) as a function of the radius parameter r for the MNIST (left) and USPS (right) datasets. Bottom plots: The behavior of the mistake bounds as a function of a label noise rate (left) and an instance noise rate (right).

5 Experimental Results

In this section we present experimental results that demonstrate different aspects of the Ballseptron algorithm and its accompanying analysis. In the first experiment we examine the effect of the radius r employed by the Ballseptron on the number of prediction mistakes it makes. We used two standard datasets: the MNIST dataset which consists of $60,000$ training examples and the USPS dataset which has 7291 training examples. The examples in both datasets are images of handwritten digits where each image belongs to one of the 10 digit classes. We thus used the multiclass extension of the Ballseptron described in the previous section. In both experiments we used a fifth degree polynomial kernel with a bias term of $1/2$ as our inner-product operator. We shifted and scaled the instances so that the average instance becomes the zero vector and the average norm over all instances becomes 1. For both datasets, we run the online Ballseptron algorithm with different values for the radius r. In the two plots on the top of Fig. 3 we depict ε/T, the number of prediction mistakes ε divided by the number of online rounds T as a function of r. Note that $r = 0$ corresponds to the original Perceptron algorithm. As can be seen from the figure, many choices of r result in a significant reduction in the number of online prediction mistakes. However, as anticipated, setting r to be excessively large deteriorates the performance of the algorithm.

The second experiment compares the mistake bound of Thm. 1 with that of Thm. 2. To facilitate a clear comparison, we set the parameter r to be zero hence we simply confined the experiment to the Perceptron algorithm. We compared the mistake bound of the Perceptron from Eq. (3) derived by Freund and Schapire [9] to the new mistake bound given in Thm. 2. For brevity we refer to the bound of Freund and Schapire as the D_2-bound and to the new mistake bound as the D_1-bound. We used two synthetic datasets each consisting of $10{,}000$ examples. The instances in the two datasets, were picked from the unit circle in $\mathbb{R}^2$. The labels of the instances were set so that the examples are linearly separable with a margin of 0.15. Then, we contaminated the instances with two different types of noise, resulting in two different datasets. For the first dataset we flipped the label of each example with probability η. In the second dataset we kept the labels intact but added to each instance a random vector sampled from a 2-dimensional Gaussian distribution with a zero mean vector and a covariance matrix $\sigma^2 I$. We then run the Perceptron algorithm on each of the datasets for different values of η and σ. We calculated the mistake bounds given in Eq. (3) and in Thm. 2 for each of the datasets and for each value of η and σ. The results are depicted on the two bottom plots of Fig. 3. As can be seen from the figure, the D_1-bound is clearly tighter than the D_2-bound in the presence of label noise. Specifically, whenever the label noise level is greater than 0.03, the D_2-bound is greater than 1 and therefore meaningless. Interestingly, the D_1-bound is also slightly better than the D_2-bound in the presence of instance noise. We leave further comparisons of the two bounds to future work.

6 Discussion and future work

We presented a new algorithm that uses the Perceptron as its infrastructure. Our algorithm naturally employs the notion of margin. Previous online margin-based algorithms yielded essentially the same mistake bound obtained by the Perceptron. In contrast, under mild conditions, our analysis implies that the mistake bound of the Ballseptron is superior to the Perceptron's bound. We derived two mistake bounds, both are also applicable to the original Perceptron algorithm. The first bound reduces to the original bound of Freund and Schpire [9] while the second bound is new and is likely to be tighter than the first in many settings. Our work can be extended in several directions. A few variations on the proposed approach, which replaces the original example with a pseudo-example, can be derived. Most notably, we can update $\mathbf{w}_t$ based on $\hat{\mathbf{x}}_t$ even for cases where there is a prediction mistake. Our proof technique is still applicable, yielding a different mistake bound. More complex prediction problems such as hierarchical classification may also be tackled in a similar way to the proposed multiclass extension. Last, we would like to note that the Ballseptron can be used as a building block for finding an arbitrarily close approximation to the max-margin solution in a separable batch setting.

Acknowledgments

We would like to thank the COLT committee members for their constructive comments. This research was funded by EU Project PASCAL and by NSF ITR award 0205594.

References

1. S. Agmon. The relaxation method for linear inequalities. *Canadian Journal of Mathematics*, 6(3):382–392, 1954.
2. J. Bi and T. Zhang. Support vector classification with input data uncertainty. In *Advances in Neural Information Processing Systems 17*, 2004.
3. H. D. Block. The perceptron: A model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962. Reprinted in ”Neurocomputing” by Anderson and Rosenfeld.
4. A. Blum and J.D. Dunagan. Smoothed analysis of the perceptron algorithm for linear programming. In *SODA*, 2002.
5. K. Crammer, O. Dekel, S. Shalev-Shwartz, and Y. Singer. Online passive aggressive algorithms. In *Advances in Neural Information Processing Systems 16*, 2003.
6. K. Crammer, R. Gilad-Bachrach, A. Navot, and N. Tishby. Margin analysis of the LVQ algorithm. In *Advances in Neural Information Processing Systems 15*, 2002.
7. K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Jornal of Machine Learning Research*, 3:951–991, 2003.
8. S. Floyd and M. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.
9. Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.
10. C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.
11. J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.
12. Y. Li and P. M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1–3):361–387, 2002.
13. M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, 1969.
14. A. B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.
15. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. (Reprinted in *Neurocomputing* (MIT Press, 1988).).
16. V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
17. J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *Proceedings of the Seventh European Symposium on Artificial Neural Networks*, April 1999.