Perceptron Mistake Bounds

Mehryar Mohri[1,2] and Afshin Rostamizadeh[1]

[1] Google Research
[2] Courant Institute of Mathematical Sciences

Abstract. We present a brief survey of existing mistake bounds and introduce novel bounds for the Perceptron or the kernel Perceptron algorithm. Our novel bounds generalize beyond standard margin-loss type bounds, allow for any convex and Lipschitz loss function, and admit a very simple proof.

1 Introduction

The Perceptron algorithm belongs to the broad family of on-line learning algorithms (see Cesa-Bianchi and Lugosi [2006] for a survey) and admits a large number of variants. The algorithm learns a linear separator by processing the training sample in an on-line fashion, examining a single example at each iteration [Rosenblatt, 1958]. At each round, the current hypothesis is updated if it makes a mistake, that is if it incorrectly classifies the new training point processed. The full pseudocode of the algorithm is provided in Figure 1. In what follows, we will assume that $\mathbf{w}_0 = \mathbf{0}$ and $\eta = 1$ for simplicity of presentation, however, the more general case also allows for similar guarantees which can be derived following the same methods we are presenting.

This paper briefly surveys some existing *mistake bounds* for the Perceptron algorithm and introduces new ones which can be used to derive generalization bounds in a stochastic setting. A mistake bound is an upper bound on the number of updates, or the number of mistakes, made by the Perceptron algorithm when processing a sequence of training examples. Here, the bound will be expressed in terms of the performance of any linear separator, including the best. Such mistake bounds can be directly used to derive generalization guarantees for a combined hypothesis, using existing on-line-to-batch techniques.

2 Separable case

The seminal work of Novikoff [1962] gave the first margin-based bound for the Perceptron algorithm, one of the early results in learning theory and probably one of the first based on the notion of margin. Assuming that the data is separable with some margin ρ, Novikoff showed that the number of mistakes made by the Perceptron algorithm can be bounded as a function of the normalized margin ρ/R, where R is the radius of the sphere containing the training instances. We start with a Lemma that can be used to prove Novikoff's theorem and that will be used throughout.

PERCEPTRON($\mathbf{w}_0$)

$\mathbf{w}_1 \leftarrow \mathbf{w}_0$ ▷ typically $\mathbf{w}_0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
  RECEIVE($\mathbf{x}_t$)
  $\widehat{y}_t \leftarrow \text{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$
  RECEIVE(y_t)
  if $(\widehat{y}_t \neq y_t)$ **then**
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \mathbf{x}_t$ ▷ more generally $\eta y_t \mathbf{x}_t, \eta > 0$.
  else $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
return $\mathbf{w}_{T+1}$

Fig. 1. Perceptron algorithm [Rosenblatt, 1958].

Lemma 1. *Let I denote the set of rounds at which the Perceptron algorithm makes an update when processing a sequence of training instances $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$. Then, the following inequality holds:*

$$\Big\| \sum_{t \in I} y_t \mathbf{x}_t \Big\| \leq \sqrt{\sum_{t \in I} \|\mathbf{x}_t\|^2}\,.$$

Proof. The inequality holds using the following sequence of observations,

$$\begin{aligned}
\Big\| \sum_{t \in I} y_t \mathbf{x}_t \Big\| &= \Big\| \sum_{t \in I} (\mathbf{w}_{t+1} - \mathbf{w}_t) \Big\| && \text{(definition of updates)} \\
&= \|\mathbf{w}_{T+1}\| && \text{(telescoping sum, } \mathbf{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\mathbf{w}_{t+1}\|^2 - \|\mathbf{w}_t\|^2} && \text{(telescoping sum, } \mathbf{w}_0 = 0) \\
&= \sqrt{\sum_{t \in I} \|\mathbf{w}_t + y_t \mathbf{x}_t\|^2 - \|\mathbf{w}_t\|^2} && \text{(definition of updates)} \\
&= \sqrt{\sum_{t \in I} 2 \underbrace{y_t \mathbf{w}_t \cdot \mathbf{x}_t}_{\leq 0} + \|\mathbf{x}_t\|^2} \\
&\leq \sqrt{\sum_{t \in I} \|\mathbf{x}_t\|^2}\,.
\end{aligned}$$

The final inequality uses the fact that an update is made at round t only when the current hypothesis makes a mistake, that is, $y_t(\mathbf{w}_t \cdot \mathbf{x}_t) \leq 0$. □

The lemma can be used straightforwardly to derive the following mistake bound for the separable setting.

Theorem 1 ([Novikoff, 1962]). *Let $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$ be a sequence of T points with $\|\mathbf{x}_t\| \leq r$ for all $t \in [1, T]$, for some $r > 0$. Assume that*

there exist $\rho > 0$ *and* $\mathbf{v} \in \mathbb{R}^N$, $\mathbf{v} \neq 0$, *such that for all* $t \in [1, T]$, $\rho \leq \frac{y_t(\mathbf{v} \cdot \mathbf{x}_t)}{\|\mathbf{v}\|}$. *Then, the number of updates made by the Perceptron algorithm when processing* $\mathbf{x}_1, \ldots, \mathbf{x}_T$ *is bounded by* r^2/ρ^2.

Proof. Let I denote the subset of the T rounds at which there is an update, and let M be the total number of updates, i.e., $|I| = M$. Summing up the inequalities yields:

$$M\rho \leq \frac{\mathbf{v} \cdot \sum_{t \in I} y_t \mathbf{x}_t}{\|\mathbf{v}\|} \leq \Big\| \sum_{t \in I} y_t \mathbf{x}_t \Big\| \leq \sqrt{\sum_{t \in I} \|\mathbf{x}_t\|^2} \leq \sqrt{M r^2},$$

where the second inequality holds by the Cauchy-Schwarz inequality, the third by Lemma 1 and the final one by assumption. Comparing the left- and right-hand sides gives $\sqrt{M} \leq r/\rho$, that is, $M \leq r^2/\rho^2$. □

3 Non-separable case

In real-world problems, the training sample processed by the Perceptron algorithm is typically not linearly separable. Nevertheless, it is possible to give a margin-based mistake bound in that general case in terms of the radius of the sphere containing the sample and the margin-based loss of an arbitrary weight vector. We present two different types of bounds: first, a bound that depends on the L_1-norm of the vector of ρ-margin hinge losses, or the vector of more general losses that we will describe, next a bound that depends on the L_2-norm of the vector of margin losses, which extends the original results presented by Freund and Schapire [1999].

3.1 L_1-norm mistake bounds

We first present a simple proof of a mistake bound for the Perceptron algorithm that depends on the L_1-norm of the losses incurred by an arbitrary weight vector, for a general definition of the loss function that covers the ρ-margin hinge loss. The family of *admissible* loss functions is quite general and defined as follows.

Definition 1 (γ-admissible loss function). *A* γ-admissible *loss function* $\phi_\gamma \colon \mathbb{R} \to \mathbb{R}_+$ *satisfies the following conditions:*

1. *The function* ϕ_γ *is convex.*
2. ϕ_γ *is non-negative:* $\forall x \in \mathbb{R}, \phi_\gamma(x) \geq 0$.
3. *At zero, the* ϕ_γ *is strictly positive:* $\phi_\gamma(0) > 0$.
4. ϕ_γ *is* γ*-Lipschitz:* $|\phi_\gamma(x) - \phi_\gamma(y)| \leq \gamma |x - y|$, *for some* $\gamma > 0$.

These are mild conditions satisfied by many loss functions including the hinge-loss, the squared hinge-loss, the Huber loss and general p-norm losses over bounded domains.

Theorem 2. *Let* I *denote the set of rounds at which the Perceptron algorithm makes an update when processing a sequence of training instances* $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$. *For any vector* $\mathbf{u} \in \mathbb{R}^N$ *with* $\|\mathbf{u}\| \leq 1$ *and any* γ*-admissible loss function* ϕ_γ, *consider the vector of losses incurred by*

$\mathbf{u}$: $\mathbf{L}_{\phi_\gamma}(\mathbf{u}) = \big[\phi_\gamma(y_t(\mathbf{u}\cdot\mathbf{x}_t))\big]_{t\in I}$. *Then, the number of updates* $M_T = |I|$ *made by the Perceptron algorithm can be bounded as follows:*

$$M_T \leq \inf_{\gamma>0,\|\mathbf{u}\|\leq 1} \frac{1}{\phi_\gamma(0)}\|\mathbf{L}_{\phi_\gamma}(\mathbf{u})\|_1 + \frac{\gamma}{\phi_\gamma(0)}\sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}\,. \tag{1}$$

If we further assume that $\|\mathbf{x}_t\| \leq r$ *for all* $t \in [1,T]$*, for some* $r > 0$*, this implies*

$$M_T \leq \inf_{\gamma>0,\|\mathbf{u}\|\leq 1}\left(\frac{\gamma r}{\phi_\gamma(0)} + \sqrt{\frac{\|\mathbf{L}_{\phi_\gamma}(\mathbf{u})\|_1}{\phi_\gamma(0)}}\right)^2. \tag{2}$$

Proof. For all $\gamma > 0$ and $\mathbf{u}$ with $\|\mathbf{u}\| \leq 1$, the following statements hold. By convexity of ϕ_γ we have $\frac{1}{M_T}\sum_{t\in I}\phi_\gamma(y_t\mathbf{u}\cdot\mathbf{x}_t) \geq \phi_\gamma(\mathbf{u}\cdot\mathbf{z})$, where $\mathbf{z} = \frac{1}{M_T}\sum_{t\in I} y_t\mathbf{x}_t$. Then, by using the Lipschitz property of ϕ_γ we have,

$$\begin{aligned}\phi_\gamma(\mathbf{u}\cdot\mathbf{z}) &= \phi_\gamma(\mathbf{u}\cdot\mathbf{z}) - \phi_\gamma(0) + \phi_\gamma(0)\\ &= -\big|\phi_\gamma(0) - \phi_\gamma(\mathbf{u}\cdot\mathbf{z})\big| + \phi_\gamma(0)\\ &\geq -\gamma|\mathbf{u}\cdot\mathbf{z}| + \phi_\gamma(0)\,.\end{aligned}$$

Combining the two inequalities above and multiplying both sides by M_T implies

$$M\phi_\gamma(0) \leq \sum_{t\in I}\phi_\gamma(y_t\mathbf{u}\cdot\mathbf{x}_t) + \gamma\Big|\sum_{t\in I} y_t\mathbf{u}\cdot\mathbf{x}_t\Big|\,.$$

Finally, using the Cauchy-Schwartz inequality and Lemma 1 yields

$$\Big|\sum_{t\in I} y_t\mathbf{u}\cdot\mathbf{x}_t\Big| = \Big|\mathbf{u}\cdot\Big(\sum_{t\in I} y_t\mathbf{x}_t\Big)\Big| \leq \|\mathbf{u}\|\Big\|\sum_{t\in I} y_t\mathbf{x}_t\Big\| \leq \sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}\,,$$

which completes the proof of the first statement after re-arranging terms. If it is further assumed that $\|\mathbf{x}_t\| \leq r$ for all $t \in I$, then this implies $M\phi_\gamma(0) - r\sqrt{M} - \sum_{t\in I}\phi_\gamma(y_t\mathbf{u}\cdot\mathbf{x}_t) \leq 0$. Solving this quadratic expression in terms of $\sqrt{M}$ proves the second statement. □

It is straightforward to see that the ρ-margin hinge loss $\phi_\rho(x) = (1 - x/\rho)_+$ is $(1/\rho)$-admissible with $\phi_\rho(0) = 1$ for all ρ, which gives the following corollary.

Corollary 1. *Let* I *denote the set of rounds at which the Perceptron algorithm makes an update when processing a sequence of training instances* $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$*. For any* $\rho > 0$ *and any* $\mathbf{u} \in \mathbb{R}^N$ *with* $\|\mathbf{u}\| \leq 1$*, consider the vector of* ρ*-hinge losses incurred by* $\mathbf{u}$: $\mathbf{L}_\rho(\mathbf{u}) = \big[(1 - \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho})_+\big]_{t\in I}$*. Then, the number of updates* $M_T = |I|$ *made by the Perceptron algorithm can be bounded as follows:*

$$M_T \leq \inf_{\rho>0\|\mathbf{u}\|\leq 1} \|\mathbf{L}_\rho(\mathbf{u})\|_1 + \frac{\sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}}{\rho}\,. \tag{3}$$

If we further assume that $\|\mathbf{x}_t\| \leq r$ *for all* $t \in [1,T]$*, for some* $r > 0$*, this implies*

$$M_T \leq \inf_{\rho>0,\|\mathbf{u}\|\leq 1}\left(\frac{r}{\rho} + \sqrt{\|\mathbf{L}_\rho(\mathbf{u})\|_1}\right)^2. \tag{4}$$

The mistake bound (3) appears already in Cesa-Bianchi et al. [2004] but we could not find its proof either in that paper or in those it references for this bound.
Another application of Theorem 2 is to the squared-hinge loss $\phi_\rho(x) = (1 - x/\rho)_+^2$. Assume that $\|\mathbf{x}\| \leq r$, then the inequality $\|y(\mathbf{u} \cdot \mathbf{x})\| \leq \|\mathbf{u}\|\|\mathbf{x}\| \leq r$ implies that the derivative of the hinge-loss is also bounded, achieving a maximum absolute value $|\phi'_\rho(r)| = |\frac{2}{\rho}(\frac{r}{\rho} - 1)| \leq \frac{2r}{\rho^2}$. Thus, the ρ-margin squared hinge loss is $(2r/\rho^2)$-admissible with $\phi_\rho(0) = 1$ for all ρ. This leads to the following corollary.

Corollary 2. *Let I denote the set of rounds at which the Perceptron algorithm makes an update when processing a sequence of training instances $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$ with $\|\mathbf{x}_t\| \leq r$ for all $t \in [1, T]$. For any $\rho > 0$ and any $\mathbf{u} \in \mathbb{R}^N$ with $\|\mathbf{u}\| \leq 1$, consider the vector of ρ-margin squared hinge losses incurred by $\mathbf{u}$: $\mathbf{L}_\rho(\mathbf{u}) = \left[(1 - \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho})_+^2\right]_{t\in I}$. Then, the number of updates $M_T = |I|$ made by the Perceptron algorithm can be bounded as follows:*

$$M_T \leq \inf_{\rho>0\|\mathbf{u}\|\leq 1} \|\mathbf{L}_\rho(\mathbf{u})\|_1 + \frac{2r\sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}}{\rho^2}\,. \tag{5}$$

This also implies

$$M_T \leq \inf_{\rho>0,\|\mathbf{u}\|\leq 1} \left(\frac{2r^2}{\rho^2} + \sqrt{\|\mathbf{L}_\rho(\mathbf{u})\|_1}\right)^2. \tag{6}$$

Theorem 2 can be similarly used to derive mistake bounds in terms of other admissible losses.

3.2 L_2-norm mistake bounds

The original results of this section are due to Freund and Schapire [1999]. Here, we extend their proof to derive finer mistake bounds for the Perceptron algorithm in terms of the L_2-norm of the vector of hinge losses of an arbitrary weight vector at points where an update is made.

Theorem 3. *Let I denote the set of rounds at which the Perceptron algorithm makes an update when processing a sequence of training instances $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$. For any $\rho > 0$ and any $\mathbf{u} \in \mathbb{R}^N$ with $\|\mathbf{u}\| \leq 1$, consider the vector of ρ-hinge losses incurred by $\mathbf{u}$: $\mathbf{L}_\rho(\mathbf{u}) = \left[(1 - \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho})_+\right]_{t\in I}$. Then, the number of updates $M_T = |I|$ made by the Perceptron algorithm can be bounded as follows:*

$$M_T \leq \inf_{\rho>0,\|\mathbf{u}\|\leq 1} \left(\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2}{2} + \sqrt{\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2^2}{4} + \frac{\sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}}{\rho}}\right)^2. \tag{7}$$

If we further assume that $\|\mathbf{x}_t\| \leq r$ for all $t \in [1, T]$, for some $r > 0$, this implies

$$M_T \leq \inf_{\rho>0,\|\mathbf{u}\|\leq 1} \left(\frac{r}{\rho} + \|\mathbf{L}_\rho(\mathbf{u})\|_2\right)^2. \tag{8}$$

Proof. We first reduce the problem to the separable case by mapping each input vector $\mathbf{x}_t \in \mathbb{R}^N$ to a vector in $\mathbf{x}'_t \in \mathbb{R}^{N+T}$ as follows:

$$\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \end{bmatrix} \mapsto \mathbf{x}'_t = \begin{bmatrix} x_{t,1} \ \dots \ x_{t,N} \ 0 \dots 0 & \underbrace{\Delta}_{\substack{(N+t)\text{th} \\ \text{component}}} & 0 \dots 0 \end{bmatrix}^\top,$$

where the first N components of $\mathbf{x}'_t$ coincide with those of $\mathbf{x}$ and the only other non-zero component is the $(N+t)$th component which is set to Δ, a parameter Δ whose value will be determined later. Define l_t by $l_t = (1 - \frac{y_t \mathbf{u} \cdot \mathbf{x}_t}{\rho}) \mathbf{1}_{t \in I}$. Then, the vector $\mathbf{u}$ is replaced by the vector $\mathbf{u}'$ defined by

$$\mathbf{u}' = \begin{bmatrix} \frac{u_1}{Z} & \dots & \frac{u_N}{Z} & \frac{y_1 l_1 \rho}{\Delta Z} & \dots & \frac{y_T l_T \rho}{\Delta Z} \end{bmatrix}^\top.$$

The first N components of $\mathbf{u}'$ are equal to the components of $\mathbf{u}/Z$ and the remaining T components are functions of the labels and hinge losses. The normalization factor Z is chosen to guarantee that $\|\mathbf{u}'\| = 1$: $Z = \sqrt{1 + \frac{\rho^2 \|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2}}$. Since the additional coordinates of the instances are non-zero exactly once, the predictions made by the Perceptron algorithm for $\mathbf{x}'_t$, $t \in [1, T]$ coincide with those made in the original space for $\mathbf{x}_t$, $t \in [1, T]$. In particular, a change made to the additional coordinates of $\mathbf{w}'$ does no affect any subsequent prediction. Furthermore, by definition of $\mathbf{u}'$ and $\mathbf{x}'_t$, we can write for any $t \in I$:

$$\begin{aligned} y_t(\mathbf{u}' \cdot \mathbf{x}'_t) &= y_t \Big(\frac{\mathbf{u} \cdot \mathbf{x}_t}{Z} + \Delta \frac{y_t l_t \rho}{Z \Delta} \Big) \\ &= \frac{y_t \mathbf{u} \cdot \mathbf{x}_t}{Z} + \frac{l_t \rho}{Z} \\ &\geq \frac{y_t \mathbf{u} \cdot \mathbf{x}_t}{Z} + \frac{\rho - y_t(\mathbf{u} \cdot \mathbf{x}_t)}{Z} = \frac{\rho}{Z}, \end{aligned}$$

where the inequality results from the definition of l_t. Summing up the inequalities for all $t \in I$ and using Lemma 1 yields $M_T \frac{\rho}{Z} \leq \sum_{t \in I} y_t(\mathbf{u}' \cdot \mathbf{x}'_t) \leq \sqrt{\sum_{t \in I} \|\mathbf{x}'_t\|^2}$. Substituting the value of Z and re-writing in terms of $\mathbf{x}$ implies:

$$\begin{aligned} M_T^2 &\leq \Big(\frac{1}{\rho^2} + \frac{\|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2} \Big) \Big(R^2 + M_T \Delta^2 \Big) \\ &= \frac{R^2}{\rho^2} + \frac{R^2 \|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2} + \frac{M_T \Delta^2}{\rho^2} + M^T \|\mathbf{L}_\rho(\mathbf{u})\|^2, \end{aligned}$$

where $R = \sqrt{\sum_{t \in I} \|\mathbf{x}_t\|^2}$. Now, solving for Δ to minimize this bound gives $\Delta^2 = \frac{\rho \|\mathbf{L}_\rho(\mathbf{u})\| R}{\sqrt{M_T}}$ and further simplifies the bound

$$\begin{aligned} M_T^2 &\leq \frac{R^2}{\rho^2} + 2 \frac{\sqrt{M_T} \|\mathbf{L}_\rho(\mathbf{u})\| R}{\rho} + M_T \|\mathbf{L}_\rho(\mathbf{u})\|^2 \\ &= \Big(\frac{R}{\rho} + \sqrt{M_T} \|\mathbf{L}_\rho(\mathbf{u})\|_2 \Big)^2. \end{aligned}$$

Solving the second-degree inequality $M_T - \sqrt{M_T}\|\mathbf{L}_\rho(\mathbf{u})\|_2 - \frac{R}{\rho} \leq 0$ proves the first statement of the theorem. The second theorem is obtained by first bounding R with $r\sqrt{M_T}$ and then solving the second-degree inequality. □

3.3 Discussion

One natural question this survey raises is the respective quality of the L_1- and L_2-norm bounds. The comparison of (4) and (8) for the ρ-margin hinge loss shows that, for a fixed ρ, the bounds differ only by the following two quantities:

$$\min_{\|\mathbf{u}\| \leq 1} \|\mathbf{L}_\rho(\mathbf{u})\|_1 = \min_{\|\mathbf{u}\| \leq 1} \sum_{t \in I} (1 - y_t(\mathbf{u} \cdot \mathbf{x}_t)/\rho)_+$$
$$\min_{\|\mathbf{u}\| \leq 1} \|\mathbf{L}_\rho(\mathbf{u})\|_2^2 = \min_{\|\mathbf{u}\| \leq 1} \sum_{t \in I} (1 - y_t(\mathbf{u} \cdot \mathbf{x}_t)/\rho)_+^2.$$

These two quantities are data-dependent and in general not comparable. For a vector $\mathbf{u}$ for which the individual losses $(1 - y_t(\mathbf{u} \cdot \mathbf{x}_t))$ are all less than one, we have $\|\mathbf{L}_\rho(\mathbf{u})\|_2^2 \leq \|\mathbf{L}_\rho(\mathbf{u})\|_1$, while the contrary holds if the individual losses are larger than one.

4 Generalization Bounds

In this section, we consider the case where the training sample processed is drawn according to some distribution D. Under some mild conditions on the loss function, the hypotheses returned by an on-line learning algorithm can then be combined to define a hypothesis whose generalization error can be bounded in terms of its regret. Such a hypothesis can be determined via cross-validation Littlestone [1989] or using the online-to-batch theorem of Cesa-Bianchi et al. [2004]. The latter can be combined with any of the mistake bounds presented in the previous section to derive generalization bounds for the Perceptron predictor.

Given $\delta > 0$, a sequence of labeled examples $(x_1, y_1), \ldots, (y_T, x_T)$, a sequence of hypotheses $h_1, \ldots, h_T$, and a loss function L, define the *penalized risk minimizing hypothesis* as $\widehat{h} = h_{i^*}$ with

$$i^* = \operatorname*{argmin}_{i \in [1,T]} \frac{1}{T - i + 1} \sum_{t=i}^{T} L(y_t h_i(x_t)) + \sqrt{\frac{\log \frac{T(T+1)}{\delta}}{2(T - i + 1)}}.$$

The following theorem gives a bound on the expected loss of $\widehat{h}$ on future examples.

Theorem 4 (Cesa-Bianchi et al. [2004]). *Let S be a labeled sample $((x_1, y_1), \ldots, (x_T, y_T))$ drawn i.i.d. according to D, L a loss function bounded by one, and $h_1, \ldots, h_T$ the sequence of hypotheses generated by an on-line algorithm $\mathcal{A}$ sequentially processing S. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following holds:*

$$\mathop{\mathrm{E}}_{(x,y) \sim D}[L(y\widehat{h}(x))] \leq \frac{1}{T} \sum_{i=1}^{T} L(y_i h_i(x_i)) + 6\sqrt{\frac{1}{T} \log \frac{2(T+1)}{\delta}}. \quad (9)$$

KernelPerceptron($\boldsymbol{\alpha}_0$)

$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}_0$ ▷ typically $\boldsymbol{\alpha}_0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
  Receive(x_t)
  $\widehat{y}_t \leftarrow \operatorname{sgn}(\sum_{s=1}^{T} \alpha_s y_s K(x_s, x_t))$
  Receive(y_t)
  if $(\widehat{y}_t \neq y_t)$ **then**
    $\alpha_t \leftarrow \alpha_t + 1$
return $\boldsymbol{\alpha}$

Fig. 2. Kernel Perceptron algorithm for PDS kernel K.

Note that this theorem does not require the loss function to be convex. Thus, if L is the zero-one loss, then the empirical loss term is precisely the average number of mistakes made by the algorithm. Plugging in any of the mistake bounds from the previous sections then gives us a learning guarantee with respect to the performance of the best hypothesis as measured by a margin-loss (or any γ-admissible loss if using Theorem 2). Let $\widehat{\mathbf{w}}$ denote the weight vector corresponding to the penalized risk minimizing Perceptron hypothesis chosen from all the intermediate hypotheses generated by the algorithm. Then, in view of Theorem 2, the following corollary holds.

Corollary 3. *Let I denote the set of rounds at which the Perceptron algorithm makes an update when processing a sequence of training instances $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$. For any vector $\mathbf{u} \in \mathbb{R}^N$ with $\|\mathbf{u}\| \leq 1$ and any γ-admissible loss function ϕ_γ, consider the vector of losses incurred by $\mathbf{u}$: $\mathbf{L}_{\phi_\gamma}(\mathbf{u}) = \left[\phi_\gamma(y_t(\mathbf{u} \cdot \mathbf{x}_t))\right]_{t \in I}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following generalization bound holds for the penalized risk minimizing Perceptron hypothesis $\widehat{\mathbf{w}}$:*

$$\Pr_{(x,y)\sim D}[y(\widehat{\mathbf{w}} \cdot \mathbf{x}) < 0]$$
$$\leq \inf_{\gamma > 0, \|\mathbf{u}\| \leq 1} \frac{\|\mathbf{L}_{\phi_\gamma}(\mathbf{u})\|_1}{\phi_\gamma(0)T} + \frac{\gamma\sqrt{\sum_{t \in I} \|\mathbf{x}_t\|^2}}{\phi_\gamma(0)T} + 6\sqrt{\frac{1}{T}\log\frac{2(T+1)}{\delta}}\,.$$

Any γ-admissible loss can be used to derive a more explicit form of this bound in special cases, in particular the hinge loss or the squared hinge loss. Using Theorem 3, we obtain the following L_2-norm generalization bound.

Corollary 4. *Let I denote the set of rounds at which the Perceptron algorithm makes an update when processing a sequence of training instances $\mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^N$. For any $\rho > 0$ and any $\mathbf{u} \in \mathbb{R}^N$ with $\|\mathbf{u}\| \leq 1$, consider the vector of ρ-hinge losses incurred by $\mathbf{u}$: $\mathbf{L}_\rho(\mathbf{u}) = \left[(1 - \frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho})_+\right]_{t \in I}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the*

following generalization bound holds for the penalized risk minimizing Perceptron hypothesis $\widehat{\mathbf{w}}$*:*

$$\Pr_{(x,y)\sim D}[y(\widehat{\mathbf{w}} \cdot \mathbf{x}) < 0]$$
$$\leq \inf_{\rho>0, \|\mathbf{u}\|\leq 1} \frac{1}{T}\left(\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2}{2} + \sqrt{\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2^2}{4} + \frac{\sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}}{\rho}}\right)^2 + 6\sqrt{\frac{1}{T}\log\frac{2(T+1)}{\delta}}\,.$$

5 Kernel Perceptron algorithm

The Perceptron algorithm of Figure 1 can be straightforwardly extended to define a non-linear separator using a positive definite kernel K [Aizerman et al., 1964]. Figure 2 gives the pseudocode of that algorithm known as the *kernel Perceptron algorithm*. The classifier $\operatorname{sgn}(h)$ learned by the algorithm is defined by $h\colon x \mapsto \sum_{t=1}^{T} \alpha_t y_t K(x_t, x)$. The results of the previous sections apply similarly to the kernel perceptron algorithm with $\|\mathbf{x}_t\|^2$ replaced with $K(x_t, x_t)$. In particular, the quantity $\sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}$ appearing in several of the learning guarantees can be replaced with the familiar trace $\operatorname{Tr}[\mathbf{K}]$ of the kernel matrix $\mathbf{K} = [K(x_i, x_j)]_{i,j\in I}$ over the set of points at which an update is made, which is a standard term appearing in margin bounds for kernel-based hypothesis sets.

Bibliography

Mark A. Aizerman, E. M. Braverman, and Lev I. Rozonoèr. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

Nicolò Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37:277–296, 1999.

Nick Littlestone. From on-line to batch learning. In *COLT*, pages 269–284, 1989.

Albert B.J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.