Worst-Case Analysis of Selective Sampling for Linear Classification

Nicolò Cesa-Bianchi CESA-BIANCHI@DSI.UNIMI.IT
DSI, Università di Milano
via Comelico, 39
20135 Milano, Italy

Claudio Gentile CLAUDIO.GENTILE@UNINSUBRIA.IT
DICOM, Università dell'Insubria
via Mazzini, 5
21100 Varese, Italy

Luca Zaniboni ZANIBONI@DTI.UNIMI.IT
DTI, Università di Milano
via Bramante, 65
26013 Crema, Italy

Editor: Manfred Warmuth

Abstract

A selective sampling algorithm is a learning algorithm for classification that, based on the past observed data, decides whether to ask the label of each new instance to be classified. In this paper, we introduce a general technique for turning linear-threshold classification algorithms from the general additive family into randomized selective sampling algorithms. For the most popular algorithms in this family we derive mistake bounds that hold for individual sequences of examples. These bounds show that our semi-supervised algorithms can achieve, on average, the same accuracy as that of their fully supervised counterparts, but using fewer labels. Our theoretical results are corroborated by a number of experiments on real-world textual data. The outcome of these experiments is essentially predicted by our theoretical results: Our selective sampling algorithms tend to perform as well as the algorithms receiving the true label after each classification, while observing in practice substantially fewer labels.

Keywords: selective sampling, semi-supervised learning, on-line learning, kernel algorithms, linear-threshold classifiers

1. Introduction

A selective sampling algorithm (see, e.g., Cohn et al., 1990; Cesa-Bianchi et al., 2003; Freund et al., 1997) is a learning algorithm for classification that receives a sequence of unlabelled instances and decides whether to query the label of the current instance based on the past observed data. The idea is to let the algorithm determine which labels are most useful to its inference mechanism, and thus achieve a good classification performance while using fewer labels.

Natural real-world scenarios for selective sampling are all those applications where labels are scarce or expensive to obtain. For example, collecting web pages is a fairly automated process, but assigning them a label (e.g., from a set of possible *topics*) often requires time-consuming and costly human expertise. For this reason, it is clearly important to devise learning algorithms having the ability to exploit the label information as much as possible. An additional motivation for using selective sampling arises from the widespread use of kernel-based algorithms (Vapnik, 1998; Cristianini and Shawe-Taylor, 2001; Schölkopf and Smola, 2002). In this case, saving labels implies using fewer support vectors to represent the hypothesis, which in turn entails a more efficient use of the memory and a shorter running time in both training and test phases.

Many algorithms have been proposed in the literature to cope with the broad task of learning with partially labelled data, working under both probabilistic and worst-case assumptions for either on-line or batch settings. These range from active learning algorithms (Campbell et al., 2000; Tong and Koller, 2000), to the query-by-committee algorithm (Freund et al., 1997), to the adversarial "apple tasting" and label efficient algorithms investigated by Helmbold et al. (2000) and Helmbold and Panizza (1997), respectively. More recent work on this subject includes (Bordes et al., 2005; Dasgupta et al., 2005; Dekel et al., 2006).

In this paper we present a mistake bound analysis for selective sampling versions of Perceptron-like algorithms. In particular, we study the standard Perceptron algorithm (Rosenblatt, 1958; Block, 1962; Novikov, 1962) and the second-order Perceptron algorithm (Cesa-Bianchi et al., 2005). Then, we argue how to extend the above analysis to the general additive family of linear-threshold algorithms introduced by Grove et al. (2001) and Warmuth and Jagota (1997) (see also Cesa-Bianchi and Lugosi, 2003; Gentile, 2003; Gentile and Warmuth, 1999; Kivinen and Warmuth, 2001), and we provide details for a specific algorithm in this family, i.e., the (zero-threshold) Winnow algorithm (Littlestone, 1988, 1989; Grove et al., 2001).

Our selective sampling algorithms use a simple randomized rule to decide whether to query the label of the current instance. This rule prescribes that the label should be obtained with probability $b/(b+|\widehat{p}|)$, where $\widehat{p}$ is the (signed) margin achieved by the current linear hypothesis on the current instance, and $b > 0$ is a parameter of the algorithm acting as a scaling factor on $\widehat{p}$. Note that a label is queried with a small probability whenever the margin $\widehat{p}$ is large in magnitude. If the label is obtained, and it turns out that a mistake has been made, then the algorithm proceeds with its standard update rule. Otherwise, the algorithm's current hypothesis is left unchanged. It is important to remark that in our model we evaluate algorithms by counting their prediction mistakes also on those time steps when the true labels remain unknown. For each of the algorithms we consider a bound is proven on the expected number of mistakes made in an arbitrary data sequence, where the expectation is with respect to the randomized sampling rule.

Our analysis reveals an interesting phenomenon. In all algorithms we analyze, a proper choice of the scaling factor b in the randomized rule yields the same mistake bound as the one achieved by the original algorithm before the introduction of the selective sampling mechanism. Hence, in some sense, our technique exploits the margin information to select those labels that can be ignored without increasing (in expectation) the overall number of mistakes.

One may suspect that this gain is not real: it might very well be the case that the tuning of b preserving the original mistake bound forces the algorithm to query all but an insignificant number of labels. In the last part of the paper we present some experiments contradicting this conjecture. In particular, by running our algorithms on real-world textual data, we show that no significant decrease in the predictive performance is suffered even when b is set to values that leave a significant fraction of the labels unobserved.

The paper is organized as follows. In the remainder of this introduction we give the notation and the basic definitions used throughout the paper. In Section 2 we describe and analyze our

Perceptron-like selective sampling algorithms. In Section 3 we extend our margin-based argument to the zero-threshold Winnow algorithm. Empirical comparisons are reported in Section 4. Finally, Section 5 is devoted to conclusions and open problems.

Notation and basic definitions

An *example* is a pair (x,y), where $x \in \mathbb{R}^d$ is an *instance* vector and $y \in \{-1,+1\}$ is the associated binary label.

We consider the following selective sampling variant of the standard on-line learning model (Angluin, 1988; Littlestone, 1988). Learning proceeds in a sequence of *trials.* In the generic trial t the algorithm observes instance x_t, outputs a prediction $\widehat{y}_t \in \{-1,+1\}$ for the label y_t associated with x_t, and decides whether or not to ask the label y_t. No matter what the algorithm decides, we say that the algorithm has made a *prediction mistake* if $\widehat{y}_t \neq y_t$. We measure the performance of a linear-threshold algorithm by the total number of mistakes it makes on a sequence of examples (including the trials where the true label y_t remains unknown). The goal of the algorithm is to bound, on an *arbitrary* sequence of examples, the amount by which this total number of mistakes exceeds the performance of the best linear predictor in hindsight.

In this paper we are concerned with selective sampling versions of linear-threshold algorithms. When run on a sequence $(x_1,y_1),(x_2,y_2),\ldots$ of examples, these algorithms compute a sequence $w_0,w_1,\ldots$ of weight vectors $w_t \in \mathbb{R}^d$, where w_t can only depend on the past examples (x_1,y_1), ..., (x_t,y_t) but not on the future ones, (x_s,y_s) for $s > t$. In each trial $t = 1,2,\ldots$ the linear-threshold algorithm predicts y_t using[1] $\widehat{y}_t = \text{SGN}(\widehat{p}_t)$ where $\widehat{p}_t = w_{t-1}^\top x_t$ is the margin of w_{t-1} on the instance x_t. If the label y_t is queried, then the algorithm (possibly) uses y_t to compute a new weight w_t; on the other hand, if y_t remains unknown then $w_t = w_{t-1}$.

We identify an arbitrary linear-threshold classifier with its coefficient vector $u \in \mathbb{R}^d$. For a fixed sequence $(x_1,y_1),\ldots,(x_n,y_n)$ of examples and a given margin threshold $\gamma > 0$, we measure the performance of u by its cumulative *hinge loss* (Freund and Schapire, 1999; Gentile and Warmuth, 1999)

$$L_{\gamma,n}(u) = \sum_{t=1}^{n} \ell_{\gamma,t}(u) = \sum_{t=1}^{n} (\gamma - y_t u^\top x_t)_+$$

where we used the notation $(x)_+ = \max\{0,x\}$. In words, the hinge loss, also called *soft margin* in the statistical learning literature (Vapnik, 1998; Cristianini and Shawe-Taylor, 2001; Schölkopf and Smola, 2002), measures the extent to which the hyperplane u separates the sequence of examples with margin at least γ.

We represent the algorithm's decision of querying the label at time t through the value of a Bernoulli random variable Z_t, whose parameter is determined by the specific selection rule used by the algorithm under consideration. Though we make no assumptions on the source generating the sequence $(x_1,y_1),(x_2,y_2),\ldots$, we require that each example (x_t,y_t) be generated before the value of Z_t is drawn. In other words, the source cannot use the knowledge of Z_t to determine x_t and y_t. We use $\mathbb{E}_{t-1}[\cdot]$ to denote the conditional expectation $\mathbb{E}[\cdot \mid Z_1,\ldots,Z_{t-1}]$ and M_t to denote the indicator function of the event $\widehat{y}_t \neq y_t$, where $\widehat{y}_t$ is the prediction at time t of the algorithm under consideration.

[1] Here and throughout SGN denotes the signum function $\text{SGN}(x) = 1$ if $x > 0$ and $\text{SGN}(x) = -1$, otherwise.

```
Selective sampling Perceptron.
Parameters: $b > 0$.
Initialization: $w_0 = (0, \dots, 0)^\top$.
For each trial $t = 1, 2, \dots$
    (1) observe an instance vector $x_t \in \mathbb{R}^d$, and set $\widehat{p}_t = w_{t-1}^\top x_t$;
    (2) predict with $\widehat{y}_t = \mathrm{SGN}(\widehat{p}_t)$;
    (3) draw a Bernoulli random variable $Z_t \in \{0, 1\}$ of parameter $\frac{b}{b + |\widehat{p}_t|}$;
    (4) if $Z_t = 1$ then query label $y_t \in \{-1, +1\}$ and perform the standard Perceptron
        update: $w_t = w_{t-1} + M_t y_t x_t$;
    (5) else ($Z_t = 0$) set $w_t = w_{t-1}$.
```

Figure 1: A selective sampling version of the classical Perceptron algorithm.

Finally, whenever the distribution laws of $Z_1, Z_2, \dots$ and $M_1, M_2, \dots$ are clear from the context, we use the abbreviation

$$\overline{L}_{\gamma,n}(u) = \mathbb{E}\left[\sum_{t=1}^{n} M_t \, Z_t \, \ell_{\gamma,t}(u)\right] .$$

Note that $\overline{L}_{\gamma,n}(u) \le L_{\gamma,n}(u)$ trivially holds for all choices of γ, n, and u.

2. Selective Sampling Algorithms and Their Analysis

In this section we describe and analyze three algorithms: a selective sampling version of the classical Perceptron algorithm (Rosenblatt, 1958; Block, 1962; Novikov, 1962), a variant of the same algorithm with a dynamically tuned parameter, and a selective sampling version of the second-order Perceptron algorithm (Cesa-Bianchi et al., 2005). It is worth pointing out that, like any Perceptron-like update rule, each of the algorithms presented in this section can be efficiently run in any given reproducing kernel Hilbert space once the update rule is expressed in an equivalent dual-variable form (see, e.g., Vapnik, 1998; Cristianini and Shawe-Taylor, 2001; Schölkopf and Smola, 2002). Note that, in the case of kernel-based algorithms, label efficiency provides the additional benefit of a more compact representation of the trained classifiers. The experiments reported in Section 4 were indeed obtained using a dual-variable implementation of our algorithms.

2.1 Selective Sampling Perceptron

Our selective sampling variant of the classical Perceptron algorithm is described in Figure 1. The algorithm maintains a vector $w \in \mathbb{R}^d$ (whose initial value is zero). In each trial t the algorithm observes an instance vector $x_t \in \mathbb{R}^d$ and predicts the binary label y_t through the sign of the margin value $\widehat{p}_t = w_{t-1}^\top x_t$. Then the algorithm decides whether to query the label y_t through the randomized rule described in the introduction: a coin with bias $b/(b+|\widehat{p}_t|)$ is flipped; if the coin turns up heads ($Z_t = 1$ in Figure 1), then the label y_t is queried. If a prediction mistake is observed ($\widehat{y}_t \neq y_t$), then the algorithm updates vector w_t according to the usual Perceptron additive rule. On the other hand, if either the coin turns up tails or $\widehat{y}_t = y_t$ ($M_t = 0$ in Figure 1), then no update takes place.

The following theorem shows that our selective sampling Perceptron can achieve, in expectation, the same mistake bound as the standard Perceptron's, but using fewer labels.

Theorem 1 *If the algorithm of Figure 1 is run with input parameter $b > 0$ on a sequence (x_1,y_1), (x_2,y_2), $\ldots \in \mathbb{R}^d \times \{-1,+1\}$ of examples, then for all $n \geq 1$, all $u \in \mathbb{R}^d$, and all $\gamma > 0$,*

$$\mathbb{E}\left[\sum_{t=1}^{n} M_t\right] \leq \left(1 + \frac{X^2}{2b}\right) \frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{\|u\|^2 \left(2b + X^2\right)^2}{8b\gamma^2}$$

where $X = \max_{t=1,\ldots,n} \|x_t\|$. Furthermore, the expected number of labels queried by the algorithm equals $\sum_{t=1}^{n} \mathbb{E}\left[\frac{b}{b+|\widehat{p}_t|}\right]$.

The above bound depends on the choice of parameter b. In general, b might be viewed as a noise parameter ruling the extent to which a linear threshold model fits the data at hand. In principle, the optimal tuning of b is easily computed. Choosing

$$b = \frac{X^2}{2}\sqrt{1 + \frac{4\gamma^2}{||u||^2 X^2} \frac{\overline{L}_{\gamma,n}(u)}{\gamma}}$$

in Theorem 1 gives the following bound on the expected number of mistakes

$$\frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{\|u\|^2 X^2}{2\gamma^2} + \frac{\|u\|\, X}{\gamma}\sqrt{\frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{\|u\|^2 X^2}{4\gamma^2}}\ . \tag{1}$$

This is an expectation version of the mistake bound for the standard Perceptron algorithm (Freund and Schapire, 1999; Gentile, 2003; Gentile and Warmuth, 1999). Note that in the special case when the data are linearly separable with margin γ^* the optimal tuning simplifies to $b = X^2/2$ and yields the familiar Perceptron bound $\left(\|u\|\, X\right)^2/(\gamma^*)^2$. Hence, in the separable case, we obtain the somewhat counterintuitive result that the standard Perceptron bound is achieved by an algorithm whose label rate does not (directly) depend on how big the separation margin is.

As it turns out, (1) might be even sharper than its deterministic counterpart since, as already noted, $\overline{L}_{\gamma,n}(u)$ can be much smaller than $L_{\gamma,n}(u)$. However, since b is an input parameter of the selective sampling algorithm, the above setting implies that, at the beginning of the prediction process, the algorithm needs some extra information on the sequence of examples. In addition, unlike the bound of Theorem 1, which holds simultaneously for all γ and u, this refined bound can only be obtained for fixed choices of these quantities. Finally, observe that letting $b \to \infty$ in Figure 1 yields the standard Perceptron algorithm but, as a shortcoming, the corresponding bound in Theorem 1 gets vacuous. This is due to the fact that our simple proof produces a mistake bound where the constant ruling the (natural) trade-off between hinge loss term and margin term is directly related to the label sampling rate.

All of the above shortcomings will be fixed in Section 2.2, where we present an adaptive parameter version of the algorithm in Figure 1. Via a more involved analysis, we show that it is still possible to achieve a bound having the same form as (1) with no prior information.

That said, we are ready to prove Theorem 1.

Proof of Theorem 1. The proof extends the standard proof of the Perceptron mistake bound (see, e.g., Duda et al., 2000, Chap. 5) which is based on estimating the influence of an update on the distance $\|u-w_{t-1}\|^2$ between the current weight vector w_{t-1} and an arbitrary "target" hyperplane u. Our analysis uses a tighter estimate on this influence, and then uses a probabilistic analysis to turn this increased tightness into an expected saving on the number of observed labels. Since this probabilistic analysis only involves the terms that are brought about by the improved estimate, we are still able to recover (in expectation) the original Perceptron bound.

Fix an arbitrary sequence $(x_1,y_1),\ldots,(x_n,y_n) \in \mathbb{R}^d \times \{-1,+1\}$ of examples. Let t be an update trial, i.e., a trial such that $M_t\,Z_t = 1$. We can write

$$
\begin{aligned}
\gamma - \ell_{\gamma,t}(u) &= \gamma - (\gamma - y_t\, u^\top x_t)_+ \\
&\le y_t\, u^\top x_t \\
&= y_t (u - w_{t-1} + w_{t-1})^\top x_t \\
&= y_t\, w_{t-1}^\top x_t + \frac{1}{2}\|u-w_{t-1}\|^2 - \frac{1}{2}\|u-w_t\|^2 + \frac{1}{2}\|w_{t-1}-w_t\|^2 \\
&= y_t\, \widehat{p}_t + \frac{1}{2}\|u-w_{t-1}\|^2 - \frac{1}{2}\|u-w_t\|^2 + \frac{1}{2}\|w_{t-1}-w_t\|^2 \ .
\end{aligned}
$$

Since the above inequality holds for any $\gamma > 0$ and any $u \in \mathbb{R}^d$, we can replace γ by $\alpha\gamma$ and u by αu, where α is a constant to be optimized.

Rearranging, and using $y_t\,\widehat{p}_t \le 0$ implied by $M_t = 1$, yields

$$
\alpha\gamma + |\widehat{p}_t| \le \alpha\ell_{\gamma,t}(u) + \frac{1}{2}\|\alpha u - w_{t-1}\|^2 - \frac{1}{2}\|\alpha u - w_t\|^2 + \frac{1}{2}\|w_{t-1}-w_t\|^2 \ .
$$

Note that, instead of discarding the term $|\widehat{p}_t|$, as in the original Perceptron proof, we keep it around. This yields a stronger inequality which, as we will see, is the key to achieving our final result.

If t is such that $M_t\,Z_t = 0$ then no update occurs and $w_t = w_{t-1}$. Hence we conclude that, for any trial t,

$$
\begin{aligned}
M_t\,Z_t(\alpha\gamma + |\widehat{p}_t|) \le{}& M_t\,Z_t\,\alpha\,\ell_{\gamma,t}(u) \\
&+ \frac{1}{2}\|\alpha u - w_{t-1}\|^2 - \frac{1}{2}\|\alpha u - w_t\|^2 + \frac{M_t\,Z_t}{2}\|w_{t-1}-w_t\|^2 \ . \qquad (2)
\end{aligned}
$$

We now sum the above over t, use $\|w_{t-1}-w_t\|^2 \le X^2$ and recall that $w_0 = 0$. We get

$$
\sum_{t=1}^n M_t\,Z_t\left(\alpha\gamma + |\widehat{p}_t| - \frac{X^2}{2}\right) \le \alpha\sum_{t=1}^n M_t\,Z_t\,\ell_{\gamma,t}(u) + \frac{\alpha^2}{2}\|u\|^2 \ .
$$

Now choose $\alpha = (2b+X^2)/(2\gamma)$, where $b > 0$ is the algorithm's parameter. The above then becomes

$$
\sum_{t=1}^n M_t\,Z_t\left(b + |\widehat{p}_t|\right) \le \frac{2b+X^2}{2\gamma}\sum_{t=1}^n M_t\,Z_t\,\ell_{\gamma,t}(u) + \frac{\|u\|^2\,(2b+X^2)^2}{8\gamma^2} \ . \qquad (3)
$$

A similar inequality is also obtained in the analysis of the standard Perceptron algorithm. Here, however, we have added the random variable Z_t, associated with the selective sampling, and kept the term $|\widehat{p}_t|$. Note that this term also appears in the conditional expectation of Z_t, since we have defined $\mathbb{E}_{t-1} Z_t$ as $b/(b+|\widehat{p}_t|)$. This fact is exploited now, when we take expectations on both sides of (3). On the left-hand side we obtain

$$\mathbb{E}\left[\sum_{t=1}^{n} M_t Z_t \big(b+|\widehat{p}_t|\big)\right] = \mathbb{E}\left[\sum_{t=1}^{n} M_t \big(b+|\widehat{p}_t|\big)\,\mathbb{E}_{t-1} Z_t\right] = \mathbb{E}\left[\sum_{t=1}^{n} b M_t\right] ,$$

where the first equality is proven by observing that M_t and $\widehat{p}_t$ are determined by $Z_1,\ldots,Z_{t-1}$ (that is, they are both measurable with respect to the σ-algebra generated by $Z_1,\ldots,Z_{t-1}$). Dividing by b we obtain the claimed inequality on the expected number of mistakes.

The value of $\mathbb{E}\left[\sum_{t=1}^{n} Z_t\right]$ (the expected number of queried labels) trivially follows from

$$\mathbb{E}\left[\sum_{t=1}^{n} Z_t\right] = \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{E}_{t-1} Z_t\right] .$$

This concludes the proof. ■

2.2 Selective Sampling Perceptron: Adaptive Version

In this section we show how to learn the best trade-off parameter b in an on-line fashion. Our goal is to devise a time-changing expression for this parameter that achieves a bound on the expected number of mistakes having the same form as (1)—i.e., with constant 1 in front of the cumulative hinge loss term—but relying on no prior knowledge whatsoever on the sequence of examples.

We follow the "self-confident" approach introduced by Auer et al. (2002) and Gentile (2001) though, as pointed out later, our self-confidence tuning here is technically different, since it does not rely on projections to control the norm of the weight (as in, e.g., Herbster and Warmuth, 2001; Auer et al., 2002; Gentile, 2001, 2003).

Our adaptive version of the selective sampling Perceptron algorithm is described in Figure 2. The algorithm still has a parameter $\beta > 0$ but, as we will see, any constant value for β leads to bounds of the form (1). Thus β has far less influence on the final bound than the b parameter in Figure 1.

The adaptive algorithm is essentially the same as the one in Figure 1, but for maintaining two further variables, X_t and K_t. At the *end* of trial t, variable X_t stores the maximal norm of the instance vectors involved in updates up to and including time t, while K_t just counts the number of such updates. Observe that b_t increases with (the square root of) this number, thereby implementing the easy intuition that the more updates are made by the algorithm the harder the problem looks, and the more labels are needed on average. However the reader should not conclude from this observation that the label rate $b_{t-1}/(b_{t-1}+|\widehat{p}_t|)$ converges to 1 as $t \to \infty$, since b_t does not scale with time t but with the number of *updates* made up to time t, which can be far smaller than t. At the same time, the margin $|\widehat{p}_t|$ might have an erratic behavior whose oscillations can also grow with the number of updates.

We have the following result.

```
Selective sampling Perceptron with adaptive parameter.
Parameters: $\beta > 0$.
Initialization: $w_0 = (0,\dots,0)^\top$, $X_0 = 0$, $K_0 = 0$.
For each trial $t = 1,2,\dots$
    (1) observe an instance vector $x_t \in \mathbb{R}^d$, and set $\widehat{p}_t = w_{t-1}^\top x_t$;
    (2) predict with $\widehat{y}_t = \mathrm{SGN}(\widehat{p}_t)$;
    (3) set $X' = \max\{X_{t-1}, \|x_t\|\}$;
    (4) draw a Bernoulli random variable $Z_t \in \{0,1\}$ of parameter
            $\frac{b_{t-1}}{b_{t-1} + |\widehat{p}_t|}$    where    $b_{t-1} = \beta\,(X')^2\sqrt{1+K_{t-1}}$ ;
    (5) if $Z_t = 1$ then
        (a) query label $y_t \in \{-1,+1\}$,
        (b) if $\widehat{y}_t \neq y_t$ (i.e., $M_t = 1$) then update
                $w_t = w_{t-1} + y_t\,x_t$
                $K_t = K_{t-1} + 1$
                $X_t = X'$;
    (6) else ($Z_t = 0$) set $w_t = w_{t-1}$,  $K_t = K_{t-1}$,  $X_t = X_{t-1}$.
```

Figure 2: Adaptive parameter version of the selective sampling Perceptron algorithm.

Theorem 2 *If the algorithm of Figure 2 is run with input parameter* $\beta > 0$ *on a sequence* (x_1,y_1), $(x_2,y_2)\ldots \in \mathbb{R}^d \times \{-1,+1\}$ *of examples, then for all* $n \geq 1$, *all* $u \in \mathbb{R}^d$, *and all* $\gamma > 0$,

$$\mathbb{E}\left[\sum_{t=1}^{n} M_t\right] \leq \frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{R}{2\beta} + \frac{B^2}{2} + B\sqrt{\frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{R}{2\beta} + \frac{B^2}{4}}$$

where

$$B = R + \frac{1+3R/2}{\beta} \qquad \text{and} \qquad R = \frac{\|u\|\,\left(\max_{t=1,\dots,n}\|x_t\|\right)}{\gamma}\ .$$

Moreover, the expected number of labels queried by the algorithm equals $\sum_{t=1}^{n}\mathbb{E}\left[\frac{b_{t-1}}{b_{t-1}+|\widehat{p}_t|}\right]$.

Before delving into the proof, it is worth observing the role of parameter β. As we have already said, if we set β to any constant value (no matter how small), we obtain a bound of the form (1). On the other hand, for $\beta \to \infty$ the algorithm reduces to the classical Perceptron algorithm, and the bound (unlike the one in Theorem 1) becomes the Perceptron bound, as given by Gentile (2003). Clearly, the larger is β the more labels are queried on average over the trials. Thus β has also an indirect influence on the hinge loss term $\overline{L}_{\gamma,n}(u)$. In particular, we might expect that a small value of β makes the number of updates shrink (note that in the limit when $\beta \to 0$ this number goes to 0).

Proof of Theorem 2. Fix an arbitrary sequence $(x_1,y_1),\dots,(x_n,y_n) \in \mathbb{R}^d \times \{-1,+1\}$ of examples and let $X = \max_{t=1,\dots,n} \|x_t\|$. The proof is a more involved version of the proof of Theorem 1. We start from the one-trial equation (2) established there, where we replace the (constant) stretching factor α by the time-varying factor c_{t-1}/γ, where $c_{t-1} \ge 0$ will be set later and $\gamma > 0$ is the free margin parameter of the hinge loss. This yields

$$\begin{aligned} M_t\, Z_t (c_{t-1} + |\widehat{p}_t|) &\le M_t\, Z_t\, \frac{c_{t-1}}{\gamma}\, \ell_{\gamma,t}(u) \\ &\quad + \frac{1}{2}\left\| \frac{c_{t-1}}{\gamma} u - w_{t-1} \right\|^2 - \frac{1}{2}\left\| \frac{c_{t-1}}{\gamma} u - w_t \right\|^2 + \frac{M_t\, Z_t}{2}\, \|w_{t-1} - w_t\|^2 \ . \end{aligned}$$

From the update rule in Figure 2 we have $(M_t\, Z_t/2)\, \|w_{t-1} - w_t\|^2 \ \le (M_t\, Z_t/2)\, \|x_t\|^2$. We rearrange and divide by b_{t-1}. This yields

$$\begin{aligned} M_t\, Z_t \left(\frac{c_{t-1} + |\widehat{p}_t| - \|x_t\|^2/2}{b_{t-1}} \right) &\le M_t\, Z_t\, \frac{c_{t-1}}{b_{t-1}} \frac{\ell_{\gamma,t}(u)}{\gamma} \\ &\quad + \frac{1}{2\, b_{t-1}} \left(\left\| \frac{c_{t-1}}{\gamma} u - w_{t-1} \right\|^2 - \left\| \frac{c_{t-1}}{\gamma} u - w_t \right\|^2 \right) \ . \qquad (4) \end{aligned}$$

We now transform the difference of squared norms in (4) into a pair of telescoping differences,

$$\begin{aligned} &\frac{1}{2\, b_{t-1}} \left(\left\| \frac{c_{t-1}}{\gamma} u - w_{t-1} \right\|^2 - \left\| \frac{c_{t-1}}{\gamma} u - w_t \right\|^2 \right) \\ &\qquad = \frac{1}{2\, b_{t-1}} \left\| \frac{c_{t-1}}{\gamma} u - w_{t-1} \right\|^2 - \frac{1}{2\, b_t} \left\| \frac{c_t}{\gamma} u - w_t \right\|^2 \\ &\qquad\quad + \frac{1}{2\, b_t} \left\| \frac{c_t}{\gamma} u - w_t \right\|^2 - \frac{1}{2\, b_{t-1}} \left\| \frac{c_{t-1}}{\gamma} u - w_t \right\|^2 \ . \qquad (5) \end{aligned}$$

If we set

$$c_{t-1} = \frac{1}{2} \big(\max\{X_{t-1}, \|x_t\|\} \big)^2 + b_{t-1}$$

we can expand the difference of norms (5) as follows

$$\begin{aligned} (5) &= \frac{\|u\|^2}{2\gamma^2} \left(\frac{c_t^2}{b_t} - \frac{c_{t-1}^2}{b_{t-1}} \right) + \frac{u^\top w_t}{\gamma} \left(\frac{c_{t-1}}{b_{t-1}} - \frac{c_t}{b_t} \right) + \frac{\|w_t\|^2}{2} \left(\frac{1}{b_t} - \frac{1}{b_{t-1}} \right) \\ &\le \frac{\|u\|^2}{2\gamma^2} \left(\frac{c_t^2}{b_t} - \frac{c_{t-1}^2}{b_{t-1}} \right) + \frac{\|u\|\, \|w_t\|}{\gamma} \left(\frac{c_{t-1}}{b_{t-1}} - \frac{c_t}{b_t} \right) \qquad (6) \end{aligned}$$

where in the last step we used $b_t \ge b_{t-1}$ and the inequality

$$\frac{c_{t-1}}{b_{t-1}} \ge \frac{c_t}{b_t}$$

which follows from $c_{t-1}/b_{t-1} = 1\big/\big(2\beta\sqrt{1+K_{t-1}}\big)+1$.

Recall now the standard way of bounding the norm of a Perceptron weight vector in terms of the number of updates,

$$\begin{aligned}\|w_t\|^2 &= \|w_{t-1}\|^2 + M_t Z_t y_t w_{t-1}^\top x_t + M_t Z_t \|x_t\|^2\\ &\le \|w_{t-1}\|^2 + M_t Z_t \|x_t\|^2\\ &\le \|w_{t-1}\|^2 + M_t Z_t X^2\end{aligned}$$

which, combined with $w_0 = 0$, implies

$$\|w_t\| \le X\sqrt{K_t} \qquad \text{for any } t. \tag{7}$$

Applying inequality (7) to (6) yields

$$(5) \le \frac{\|u\|^2}{2\gamma^2}\left(\frac{c_t^2}{b_t} - \frac{c_{t-1}^2}{b_{t-1}}\right) + \frac{\|u\|\,X\sqrt{K_t}}{\gamma}\left(\frac{c_{t-1}}{b_{t-1}} - \frac{c_t}{b_t}\right). \tag{8}$$

We continue by bounding from above the last term in (8). If t is such that $M_t Z_t = 1$ we have $K_t = K_{t-1}+1$. Thus we can write

$$\begin{aligned}\sqrt{K_t}\left(\frac{c_{t-1}}{b_{t-1}} - \frac{c_t}{b_t}\right) &= \frac{\sqrt{K_t}}{2\beta}\left(\frac{1}{\sqrt{1+K_{t-1}}} - \frac{1}{\sqrt{1+K_t}}\right)\\ &= \frac{\sqrt{K_t}}{2\beta}\left(\frac{1}{\sqrt{K_t}} - \frac{1}{\sqrt{1+K_t}}\right)\\ &= \frac{1}{2\beta}\,\frac{\sqrt{1+K_t}-\sqrt{K_t}}{\sqrt{1+K_t}}\\ &\le \frac{1}{4\beta}\,\frac{1}{\sqrt{K_t}\sqrt{1+K_t}}\\ &\qquad \text{(using } \sqrt{1+x}-\sqrt{x} \le \tfrac{1}{2\sqrt{x}}\text{)}\\ &\le \frac{1}{4\beta}\,\frac{1}{K_t}\,.\end{aligned}$$

On the other hand, if $M_t Z_t = 0$ we have $b_t = b_{t-1}$ and $c_t = c_{t-1}$. Hence, for any trial t we obtain

$$\sqrt{K_t}\left(\frac{c_{t-1}}{b_{t-1}} - \frac{c_t}{b_t}\right) \le \frac{M_t Z_t}{4\beta}\,\frac{1}{K_t}\,.$$

Putting together as in (4) and using $c_{t-1} - \|x_t\|^2/2 \ge b_{t-1}$ on the left-hand side yields

$$\begin{aligned}M_t\,Z_t\left(\frac{b_{t-1}+|\widehat{p}_t|}{b_{t-1}}\right) &\le M_t\,Z_t\,\frac{c_{t-1}}{b_{t-1}}\,\frac{\ell_{\gamma,t}(u)}{\gamma}\\ &\quad + \frac{1}{2b_{t-1}}\left\|\frac{c_{t-1}}{\gamma}u - w_{t-1}\right\|^2 - \frac{1}{2b_t}\left\|\frac{c_t}{\gamma}u - w_t\right\|^2\\ &\quad + \frac{\|u\|^2}{2\gamma^2}\left(\frac{c_t^2}{b_t} - \frac{c_{t-1}^2}{b_{t-1}}\right) + \frac{\|u\|\,X}{\gamma}\,\frac{M_t Z_t}{4\beta}\,\frac{1}{K_t}\end{aligned}$$

holding for any trial t, any $u \in \mathbb{R}^d$, and any $\gamma > 0$.

Now, as in the proof of Theorem 1, we sum over $t = 1, \dots, n$, use $w_0 = 0$, and simplify

$$\begin{aligned}\sum_{t=1}^{n} M_t Z_t \left(\frac{b_{t-1} + |\widehat{p}_t|}{b_{t-1}}\right) \quad &\le \quad \frac{1}{\gamma}\sum_{t=1}^{n} M_t Z_t \frac{c_{t-1}}{b_{t-1}} \ell_{\gamma,t}(u) &(9)\\ &\quad + \frac{c_n^2}{b_n}\frac{\|u\|^2}{2\gamma^2} - \frac{1}{2\,b_n}\left\|\frac{c_n}{\gamma}u - w_n\right\|^2 &(10)\\ &\quad + \frac{1}{4\beta}\frac{\|u\|\,X}{\gamma}\sum_{t=1}^{n}\frac{M_t Z_t}{K_t}\ .\end{aligned}$$

We now proceed by bounding separately the terms in the right-hand side of the above inequality. For (9) we get

$$\begin{aligned}\frac{1}{\gamma}\frac{c_{t-1}}{b_{t-1}}\ell_{\gamma,t}(u) \quad &= \quad \frac{1}{\gamma}\left(\frac{1}{2\beta\sqrt{1+K_{t-1}}} + 1\right)\ell_{\gamma,t}(u)\\ &\le \quad \frac{1}{\gamma}\frac{1}{2\beta\sqrt{1+K_{t-1}}}\big(\gamma + \|u\|\,X\big) + \frac{\ell_{\gamma,t}(u)}{\gamma}\\ &\qquad \text{(since } \ell_{\gamma,t}(u) \le \gamma + \|u\|\,X\text{)}\\ &= \quad \frac{1}{2\beta}\left(1 + \frac{\|u\|X}{\gamma}\right)\frac{1}{\sqrt{1+K_{t-1}}} + \frac{\ell_{\gamma,t}(u)}{\gamma}\ .\end{aligned}$$

For (10) we obtain

$$\begin{aligned}\frac{c_n^2}{b_n}\frac{\|u\|^2}{2\gamma^2} - \frac{1}{2b_n}\left\|\frac{c_n}{\gamma}u - w_n\right\|^2 \quad &= \quad \frac{c_n}{b_n}\frac{u^\top w_n}{\gamma} - \frac{\|w_n\|^2}{2b_n}\\ &\le \quad \frac{c_n}{b_n}\frac{u^\top w_n}{\gamma}\\ &\le \quad \frac{c_n}{b_n}\frac{\|u\|\,\|w_n\|}{\gamma}\\ &\le \quad \left(1 + \frac{1}{2\beta\sqrt{1+K_n}}\right)\frac{\|u\|\,X\sqrt{K_n}}{\gamma}\end{aligned}$$

where in the last step we used (7). Using these expressions to bound the left-hand side of (9) yields

$$\begin{aligned}&\sum_{t=1}^{n} M_t Z_t \left(\frac{b_{t-1} + |\widehat{p}_t|}{b_{t-1}}\right)\\ &\qquad \le \frac{1}{\gamma}\sum_{t=1}^{n} M_t Z_t\, \ell_{\gamma,t}(u) + \frac{1}{2\beta}\left(1 + \frac{\|u\|X}{\gamma}\right)\sum_{t=1}^{n}\frac{M_t Z_t}{\sqrt{1+K_{t-1}}} &(11)\\ &\qquad\quad + \left(1 + \frac{1}{2\beta\sqrt{1+K_n}}\right)\frac{\|u\|X\sqrt{K_n}}{\gamma} + \frac{1}{4\beta}\frac{\|u\|X}{\gamma}\sum_{t=1}^{n}\frac{M_t Z_t}{K_t}\ . &(12)\end{aligned}$$

Next, we focus on the second sum in (11) and the sum in (12). Since $M_t Z_t = 1$ implies $K_t = K_{t-1} + 1$ we can write

$$\sum_{t=1}^{n}\frac{M_t Z_t}{\sqrt{1+K_{t-1}}} = \sum_{t\,:\,M_t Z_t = 1}\frac{1}{\sqrt{K_t}} = \sum_{t=1}^{K_n}\frac{1}{\sqrt{t}} \le 2\sqrt{K_n}\ .$$

Similarly for the other sum, but using a more crude bound,

$$\sum_{t=1}^{n} \frac{M_t Z_t}{K_t} = \sum_{t\,:\,M_t\,Z_t=1} \frac{1}{K_t} \le \sum_{t\,:\,M_t\,Z_t=1} \frac{1}{\sqrt{K_t}} \le 2\sqrt{K_n}\ .$$

Recalling the short-hand $R = (\|u\|\, X)/\gamma$, we apply these bounds to (11) and (12). After a simple overapproximation this gives

$$\sum_{t=1}^{n} M_t\, Z_t \left(\frac{b_{t-1} + |\widehat{p}_t|}{b_{t-1}} \right) \le \frac{1}{\gamma} \sum_{t=1}^{n} M_t\, Z_t\, \ell_{\gamma,t}(u) + \sqrt{K_n} \left(R + \frac{1+3R/2}{\beta} \right) + \frac{R}{2\beta}\ .$$

We are now ready to take expectations on both sides. As in the proof of Theorem 1, since $\mathbb{E}_{t-1} Z_t = \frac{b_{t-1}}{b_{t-1}+|\widehat{p}_t|}$ and both M_t and b_{t-1} are measurable with respect to the σ-algebra generated by $Z_1,\dots,Z_{t-1}$, we obtain

$$\mathbb{E}\left[\sum_{t=1}^{n} M_t\, Z_t \left(\frac{b_{t-1} + |\widehat{p}_t|}{b_{t-1}} \right) \right] = \mathbb{E}\left[\sum_{t=1}^{n} M_t \right]\ .$$

In taking the expectation on the right-hand side, we first bound $K_n = \sum_{t=1}^{n} M_t\, Z_t$ as $K_n \le \sum_{t=1}^{n} M_t$, then exploit the concavity of the square root. This results in

$$\sum_{t=1}^{n} \mathbb{E}\, M_t \le \frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \left(R + \frac{1+3R/2}{\beta} \right) \sqrt{\sum_{t=1}^{n} \mathbb{E}\, M_t} + \frac{R}{2\beta}\ .$$

Solving the above inequality for $\sum_{t=1}^{n} \mathbb{E}\, M_t$ gives the stated bound on the expected number of mistakes.

Finally, as in the proof of Theorem 1, the expected number of labels queried by the algorithm trivially follows from

$$\mathbb{E}\left[\sum_{t=1}^{n} Z_t \right] = \mathbb{E}\left[\sum_{t=1}^{n} \mathbb{E}_{t-1} Z_t \right]$$

concluding the proof. ■

The proof of Theorem 2 is reminiscent of the analysis of the “self-confident” dynamical tuning used in Auer et al. (2002) and Gentile (2001). In those papers, however, the variable learning rate was combined with a re-normalization step of the weight. Here we use a different technique based on a time-changing stretching factor $\alpha_{t-1} = c_{t-1}/\gamma$ for the comparison vector u. This alternative approach is made possible by the boundedness of the hinge loss terms, as shown by the inequality $\ell_{\gamma,t}(u) \le \gamma + \|u\|\, X$.

2.3 Selective Sampling Second-Order Perceptron

We now consider a selective sampling version of the second-order Perceptron algorithm introduced by Cesa-Bianchi et al. (2005). The second-order Perceptron algorithm might be seen as running the standard (first-order) Perceptron algorithm as a subroutine. Let v_{t-1} denote the weight vector computed by the standard Perceptron algorithm. In trial t, instead of using the sign of $v_{t-1}^\top x_t$ to predict the current instance x_t, the second-order algorithm predicts through the sign of the margin

$$\widehat{p}_t = \left(M^{-1/2} v_{t-1}\right)^\top \left(M^{-1/2} x_t\right) = v_{t-1}^\top M^{-1} x_t\ .$$

Here $M = I + \sum_s x_s x_s^\top + x_t x_t^\top$ is a (full-rank) positive definite matrix, where I is the $d \times d$ identity matrix, and the sum $\sum_s x_s x_s^\top$ runs over the mistaken trials s up to time $t-1$. If, when using the above prediction rule, the algorithm makes a mistake in trial t, then v_{t-1} is updated according to the standard Perceptron rule and t is included in the set of mistaken trials. Hence the second-order algorithm differs from the standard Perceptron algorithm in that, before each prediction, a linear transformation $M^{-1/2}$ is applied to both the current Perceptron weight v_{t-1} and the current instance x_t. This linear transformation depends on the correlation matrix defined over mistaken instances, including the current one. As explained in Cesa-Bianchi et al. (2005), this linear transformation has the effect of reducing the number of mistakes whenever the instance correlation matrix $\sum_s x_s x_s^\top + x_t x_t^\top$ has a spectral structure that causes an eigenvector with small eigenvalue to correlate well with a good linear approximator u of the entire data sequence. In such situations, the mistake bound of the second-order Perceptron algorithm can be shown to be significantly better than the one for the first-order algorithm.

In what follows, we use A_{t-1} to denote $I + \sum_s x_s x_s^\top$ where the sum ranges over the mistaken trials between trial 1 and trial $t-1$. We derive a selective sampling version of the second-order algorithm in much the same way as we did for the standard Perceptron algorithm: The selective sampling second-order Perceptron algorithm predicts and then decides whether to ask for the label y_t using the same randomized rule as the one in Figure 1. In Figure 3 we provide a pseudo-code description and introduce the notation used in the analysis.

The analysis follows the same pattern as the proof of Theorem 1. A key step is a one-trial progress equation developed by Forster (1999) for a regression framework. See also Azoury and Warmuth (2001). As before, the comparison between the second-order Perceptron's bound and the one contained in Theorem 3 reveals that the selective sampling algorithm can achieve, in expectation, the same mistake bound using fewer labels.

Theorem 3 *If the algorithm of Figure 3 is run with parameter $b > 0$ on a sequence (x_1, y_1), (x_2, y_2), $\ldots \in \mathbb{R}^d \times \{-1, +1\}$ of examples, then for all $n \geq 1$, all $u \in \mathbb{R}^d$, and all $\gamma > 0$,*

$$\mathbb{E}\left[\sum_{t=1}^{n} M_t\right] \leq \frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{b}{2\gamma^2} u^\top \mathbb{E}[A_n] u + \frac{1}{2b} \sum_{i=1}^{d} \mathbb{E} \ln(1+\lambda_i)$$

where $\lambda_1, \ldots, \lambda_d$ are the eigenvalues of the (random) correlation matrix $\sum_{t=1}^n M_t Z_t x_t x_t^\top$ and $A_n = I + \sum_{t=1}^n M_t Z_t x_t x_t^\top$ (thus $1 + \lambda_i$ is the i-th eigenvalue of A_n). Moreover, the expected number of labels queried by the algorithm equals $\sum_{t=1}^n \mathbb{E}\left[\frac{b}{b+|\widehat{p}_t|}\right]$.

Again, the above bound depends on the algorithm's parameter b. Setting

$$b = \gamma \sqrt{\frac{\sum_{i=1}^{d} \mathbb{E} \ln(1+\lambda_i)}{u^\top \mathbb{E}[A_n] u}}$$

in Theorem 3 we are led to the bound

$$\mathbb{E}\left[\sum_{t=1}^{n} M_t\right] \leq \frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{1}{\gamma} \sqrt{(u^\top \mathbb{E}[A_n] u) \sum_{i=1}^{d} \mathbb{E} \ln(1+\lambda_i)} \,. \tag{13}$$

This is an expectation version of the mistake bound for the (deterministic) second-order Perceptron algorithm, as proven by Cesa-Bianchi et al. (2005). As for the first-order algorithms, this

```
**Selective sampling second-order Perceptron.**
**Parameter:** $b > 0$.
**Initialization:** $A_0 = I$, $v_0 = (0,\dots,0)^\top$.
**For** each trial $t = 1,2,\dots$

  (1) observe an instance vector $x_t \in \mathbb{R}^d$, and set $\widehat{p}_t = v_{t-1}^\top (A_{t-1} + x_t x_t^\top)^{-1} x_t$;

  (2) predict with $\widehat{y}_t = \mathrm{SGN}(\widehat{p}_t)$;

  (3) draw a Bernoulli random variable $Z_t \in \{0,1\}$ of parameter $\frac{b}{b+|\widehat{p}_t|}$;

  (4) **if** $Z_t = 1$ **then** query label $y_t \in \{-1,+1\}$ and perform the standard second-order
      Perceptron update:

          $v_t = v_{t-1} + M_t y_t x_t$ ,
          $A_t = A_{t-1} + M_t x_t x_t^\top$ ;

  (5) **else** ($Z_t = 0$) set $v_t = v_{t-1}$ and $A_t = A_{t-1}$.
```

Figure 3: A selective sampling version of the second-order Perceptron algorithm.

bound might be sharper than its deterministic counterpart, since the magnitude of the three quantities $\overline{L}_{\gamma,n}(u)$, $u^\top \mathbb{E}[A_n] u$, and $\sum_{i=1}^d \mathbb{E} \ln(1+\lambda_i)$ is ruled by the size of the random set of updates $\{t : M_t Z_t = 1\}$, which is typically smaller than the set of mistaken trials of the deterministic algorithm.

However, as for the algorithm in Figure 1, this parameter tuning turns out to be unrealistic, since it requires preliminary information on the structure of the sequence of examples. Unlike the first-order algorithm, we have been unable to devise a meaningful adaptive parameter version for the algorithm in Figure 3.

Proof of Theorem 3. The proof proceeds along the same lines as the proof of Theorem 1, thus we only emphasize the main differences. In addition to the notation given there, we define Φ_t to be the (random) function

$$\Phi_t(u) = \frac{1}{2}\|u\|^2 + \sum_{s=1}^{t} \frac{M_s Z_s}{2} (y_s - u^\top x_s)^2 .$$

The quantity $\Phi_t(u)$, which is the regularized cumulative square loss of u on the past mistaken trials, plays a key role in the proof. Indeed, we now show that the algorithm incurs on each mistaken trial a square loss $\big(y_t - \widehat{p}_t\big)^2$ bounded by the difference $\inf_v \Phi_{t+1}(v) - \inf_v \Phi_t(v)$ plus a quadratic term involving A_t^{-1}. When we sum over mistaken trials, the difference telescopes and the sum of quadratic terms can be bounded using known results. Finally, the margin we use in the probabilistic analysis is obtained as cross-term when the square loss is expanded.

When trial t is such that $M_t\,Z_t = 1$ we can exploit a result proven by Forster (1999) for linear regression (proof of Theorem 3 therein), where it is essentially shown that choosing $\widehat{p}_t = v_{t-1}^\top(A_{t-1}+x_t x_t^\top)^{-1}x_t$ (as in Figure 3) yields

$$\frac{1}{2}\left(\widehat{p}_t - y_t\right)^2 = \inf_v \Phi_{t+1}(v) - \inf_v \Phi_t(v) + \frac{1}{2}x_t^\top A_t^{-1}x_t - \frac{1}{2}\left(x_t^\top A_{t-1}^{-1}x_t\right)\widehat{p}_t^2 \ .$$

On the other hand, if trial t is such that $M_t\,Z_t = 0$ we have $\inf_{v\in\mathbb{R}^d}\Phi_{t+1}(v) = \inf_{v\in\mathbb{R}^d}\Phi_t(v)$. Hence the equality

$$\frac{M_t Z_t}{2}\left(\widehat{p}_t - y_t\right)^2 = \inf_v \Phi_{t+1}(v) - \inf_v \Phi_t(v) + \frac{M_t\,Z_t}{2}x_t^\top A_t^{-1}x_t - \frac{M_t\,Z_t}{2}\left(x_t^\top A_{t-1}^{-1}x_t\right)\widehat{p}_t^2$$

holds for all trials t. We drop the term $-M_t Z_t\left(x_t^\top A_{t-1}^{-1}x_t\right)\widehat{p}_t^2/2$, which is nonpositive (since A_{t-1} is positive definite), and sum over $t = 1,\ldots,n$. Observing that $\inf_v \Phi_1(v) = 0$, we obtain

$$\begin{aligned}
\frac{1}{2}\sum_{t=1}^n M_t Z_t\left(\widehat{p}_t - y_t\right)^2 &\le \inf_v \Phi_{n+1}(v) - \inf_v \Phi_1(v) + \frac{1}{2}\sum_{t=1}^n M_t\,Z_t\,x_t^\top A_t^{-1}x_t \\
&\le \Phi_{n+1}(u) + \frac{1}{2}\sum_{t=1}^n M_t\,Z_t\,x_t^\top A_t^{-1}x_t \\
&\le \frac{1}{2}\|u\|^2 + \frac{1}{2}\sum_{t=1}^n M_t\,Z_t\left(u^\top x_t - y_t\right)^2 + \frac{1}{2}\sum_{t=1}^n M_t\,Z_t\,x_t^\top A_t^{-1}x_t
\end{aligned}$$

holding for any $u \in \mathbb{R}^d$.

Expanding the squares and performing trivial simplifications we arrive at the following inequality

$$\begin{aligned}
&\frac{1}{2}\sum_{t=1}^n M_t Z_t\left(\widehat{p}_t^2 - 2y_t\widehat{p}_t\right) \\
&\le \frac{1}{2}\left[\|u\|^2 + \sum_{t=1}^n M_t Z_t\left(u^\top x_t\right)^2\right] - \sum_{t=1}^n M_t Z_t\,y_t\,u^\top x_t + \frac{1}{2}\sum_{t=1}^n M_t Z_t\,x_t^\top A_t^{-1}x_t \ . \qquad (14)
\end{aligned}$$

We focus on the right-hand side of (14). We rewrite the first term and bound from above the last term. For the first term we have

$$\frac{1}{2}\left[\|u\|^2 + \sum_{t=1}^n M_t\,Z_t\left(u^\top x_t\right)^2\right] = \frac{1}{2}u^\top\left(I + \sum_{t=1}^n x_t\,x_t^\top\,M_t\,Z_t\right)u = \frac{1}{2}u^\top A_n u \ . \qquad (15)$$

For the third term, we use a property of the inverse matrices A_t^{-1} (see, e.g., Lai and Wei, 1982; Azoury and Warmuth, 2001; Forster, 1999; Cesa-Bianchi et al., 2005),

$$\begin{aligned}
\frac{1}{2}\sum_{t=1}^{n} M_t Z_t x_t^\top A_t^{-1} x_t &= \frac{1}{2}\sum_{t=1}^{n}\left(1-\frac{|A_{t-1}|}{|A_t|}\right)\\
&\le \frac{1}{2}\sum_{t=1}^{n}\ln\frac{|A_t|}{|A_{t-1}|}\\
&= \frac{1}{2}\ln\frac{|A_n|}{|A_0|}\\
&= \frac{1}{2}\ln|A_n|\\
&= \frac{1}{2}\sum_{i=1}^{d}\ln(1+\lambda_i)
\end{aligned}$$

where we recall that $1+\lambda_i$ is the i-th eigenvalue of A_n.

Replacing back, observing that $-y_t\widehat{p}_t \le 0$ whenever $M_t = 1$, dropping the term involving $\widehat{p}_t^2$, and rearranging yields

$$\sum_{t=1}^{n} M_t Z_t \left(|\widehat{p}_t| + y_t u^\top x_t\right) \le \frac{1}{2}u^\top A_n u + \frac{1}{2}\sum_{i=1}^{d}\ln(1+\lambda_i)\ .$$

At this point, as in the proof of Theorem 1, we introduce hinge loss terms and stretch the comparison vector u to $\frac{b}{\gamma}u$, where b is the algorithm's parameter. We obtain

$$\sum_{t=1}^{n} M_t Z_t \left(|\widehat{p}_t| + b\right) \le \frac{b}{\gamma}\sum_{t=1}^{n} M_t Z_t\, \ell_{\gamma,t}(u) + \frac{b^2}{2\gamma^2}u^\top A_n u + \frac{1}{2}\sum_{i=1}^{d}\ln(1+\lambda_i)\ .$$

We take expectations on both sides. Recalling that $\mathbb{E}_{t-1} Z_t = b/(b+|\widehat{p}_t|)$, and proceeding similarly to the proof of Theorem 1 we get the claimed bounds on $\sum_{t=1}^{n}\mathbb{E} M_t$ and $\sum_{t=1}^{n}\mathbb{E} Z_t$. ■

3. Selective Sampling Winnow

The techniques used to prove Theorem 1 can be readily extended to analyze selective sampling versions of algorithms in the general additive family of Grove et al. (2001), Warmuth and Jagota (1997), and Kivinen and Warmuth (2001). The algorithms in this family—which includes Winnow (Littlestone, 1988), the p-norm Perceptron (Grove et al., 2001; Gentile, 2001), and others—are parametrized by a strictly convex and differentiable *potential function* $\Psi : \mathbb{R}^d \to \mathbb{R}$ obeying some additional regularity properties. We now show a concrete example by analyzing the selective sampling version of the Winnow algorithm (Littlestone, 1988), a member of the general additive family based on the exponential potential $\Psi(u) = e^{u_1} + \cdots + e^{u_d}$.

In its basic version, Winnow uses weights that belong to the probability simplex in $\mathbb{R}^d$. The update rule for the weights is multiplicative, and is followed by a normalization step which projects the updated weight vector back to the simplex. Introducing the intermediate weight w_t', we define

```
Selective sampling Winnow.
Parameters: $\eta, b > 0$.
Initialization: $w_0 = (1/d, \dots, 1/d)^\top$.
For each trial $t = 1, 2, \dots$
    (1) observe an instance vector $x_t \in \mathbb{R}^d$, and set $\widehat{p}_t = w_{t-1}^\top x_t$;
    (2) predict with $\widehat{y}_t = \mathrm{SGN}(\widehat{p}_t)$;
    (3) draw a Bernoulli random variable $Z_t \in \{0,1\}$ of parameter $\frac{b}{b + |\widehat{p}_t|}$;
    (4) if $Z_t = 1$ then query label $y_t \in \{-1,+1\}$ and perform the standard Winnow
        update:
            $w'_{i,t} = w_{i,t-1} \, e^{M_t \eta y_t x_{i,t}}$ ,
            $w_{i,t} = w'_{i,t}/(w'_{1,t} + \cdots + w'_{d,t}) \qquad i = 1, \dots, d$;
    (5) else ($Z_t = 0$) set $w_t = w_{t-1}$.
```

Figure 4: A selective sampling version of the Winnow algorithm.

the update rule as follows:

$$\begin{aligned} w'_{i,t} &= w_{i,t-1} \, e^{\eta y_t x_{i,t}} \\ w_{i,t} &= \frac{w'_{i,t}}{\sum_{j=1}^d w'_{j,t}} \end{aligned} \qquad \text{for } i = 1, \dots, d.$$

The theory behind the analysis of general additive family of algorithms shows that, notwithstanding their apparent diversity, Winnow and Perceptron are actually instances of the same additive algorithm.

To obtain a selective sampling version of Winnow we proceed exactly as we did in the previous cases: we query the label y_t with probability $b/(b + |\widehat{p}_t|)$, where $|\widehat{p}_t|$ is the margin computed by the algorithm. The complete pseudo-code is described in Figure 4.

The mistake bound we prove for selective sampling Winnow is somewhat atypical since, unlike the Perceptron-like algorithms analyzed so far, the choice of the learning rate η given in this theorem is the same as the one suggested by the original Winnow analysis (see, e.g., Littlestone, 1989; Grove et al., 2001). Furthermore, since a meaningful bound in Winnow requires η be chosen in terms of γ, it turns out that in the selective sampling version there is no additional tuning to perform, and we are able to obtain the same mistake bound as the original version. Thus, unlike the other cases, the selective sampling mechanism does not weaken in any respect the original mistake bound, apart from turning a deterministic bound into an expected one.

Theorem 4 *If the algorithm of Figure 4 is run with parameters*

$$\eta = \frac{2(1-\alpha)\gamma}{X_\infty^2} \qquad \text{and} \qquad b = \alpha\gamma \qquad \text{for some } \alpha \in (0,1)$$

on a sequence $(x_1,y_1),\dots,(x_n,y_n) \in \mathbb{R}^d \times \{-1,+1\}$ *of examples such that* $\|x_t\|_\infty \le X_\infty$ *for all* $t = 1,\dots,n$*, then for all* $u \in \mathbb{R}^d$ *in the probability simplex,*

$$\mathbb{E}\left[\sum_{t=1}^n M_t\right] \le \frac{1}{\alpha}\,\frac{\overline{L}_{\gamma,n}(u)}{\gamma} + \frac{1}{2\alpha(1-\alpha)}\,\frac{X_\infty^2 \ln d}{\gamma^2}\ .$$

As before, the expected number of labels queried by the algorithm equals $\sum_{t=1}^n \mathbb{E}\left[\frac{b}{b+|\widehat{p}_t|}\right]$.

Proof Similarly to the proof of Theorem 1, we estimate the influence of an update on the distance between the current weight w_{t-1} and an arbitrary “target” hyperplane u, where in this case both vectors live in the probability simplex. Unlike the Perceptron analysis, based on the squared Euclidean distance, the analysis of Winnow uses the Kullback-Leibler divergence, or relative entropy, $\mathrm{KL}(\cdot,\cdot)$ to measure the progress of w_{t-1} towards u. The relative entropy of any two vectors u,v belonging to the probability simplex on $\mathbb{R}^d$ is defined by

$$\mathrm{KL}(u,v) = \sum_{i=1}^d u_i \ln \frac{u_i}{v_i}\ .$$

Fix an arbitrary sequence $(x_1,y_1),\dots,(x_n,y_n) \in \mathbb{R}^d \times \{-1,+1\}$ of examples. As in the proof of Theorem 1, we have that $M_t Z_t = 1$ implies

$$\begin{aligned}
\eta\big(\gamma - \ell_{\gamma,t}(u)\big) &= \eta\big(\gamma - (\gamma - y_t, u^\top x_t)_+\big) \\
&\le \eta\, y_t\, u^\top x_t \\
&= \eta\, y_t (u - w_{t-1} + w_{t-1})^\top x_t \\
&= \eta\, y_t (u - w_{t-1})^\top x_t + \eta y_t\, w_{t-1}^\top x_t\ .
\end{aligned}$$

Besides, exploiting a simple identity (as in the proof of Theorem 11.3 in Cesa-Bianchi and Lugosi, 2006, Chap. 5), we can rewrite the term $\eta\, y_t (u - w_{t-1})^\top x_t$ as

$$\eta\, y_t (u - w_{t-1})^\top x_t = \mathrm{KL}(u, w_{t-1}) - \mathrm{KL}(u, w_t) + \ln\left(\sum_{j=1}^d w_{j,t-1} e^{\eta y_t v_j}\right)$$

where $v_j = x_j - w_{t-1}^\top x_t$. This equation is similar to the one obtained in the analysis of the selective sampling Perceptron algorithm, but for the relative entropy replacing the squared Euclidean distance. Note, however, that the last term in the right-hand side of the above equation is not a relative entropy. To bound this last term, we consider the random variable X taking value $x_{i,t} \in [-X_\infty, X_\infty]$ with probability $w_{i,t-1}$. Then, from the Hoeffding inequality (Hoeffding, 1963) applied to X,

$$\ln\left(\sum_{j=1}^d w_{j,t-1} e^{\eta y_t v_j}\right) = \ln \mathbb{E}\left[e^{\eta y_t (X - \mathbb{E}X)}\right] \le \frac{\eta^2}{2} X_\infty^2\ .$$

We plug back, rearrange and note that $w_t = w_{t-1}$ whenever $M_t Z_t = 0$. This gets

$$M_t\, Z_t\, \eta \left(\gamma + |\widehat{p}_t| - \frac{\eta}{2} X_\infty^2\right) \le M_t\, Z_t\, \eta\, \ell_{\gamma,t}(u) + \mathrm{KL}(u, w_{t-1}) - \mathrm{KL}(u, w_t) \ ,$$

holding for any t. Summing over $t = 1, \ldots, n$ and dividing by η yields

$$\sum_{t=1}^{n} M_t Z_t \left(\gamma + |\widehat{p}_t| - \frac{\eta}{2} X_\infty^2\right) \le \sum_{t=1}^{n} M_t\, Z_t\, \ell_{\gamma,t}(u) + \frac{\mathrm{KL}(u, w_0)}{\eta} - \frac{\mathrm{KL}(u, w_n)}{\eta} \ .$$

We drop the last term (which is nonpositive), and use $\mathrm{KL}(u, w_0) \le \ln d$ holding for any u in the probability simplex whenever $w_0 = (1/d, \ldots, 1/d)$. Then the above reduces to

$$\sum_{t=1}^{n} M_t\, Z_t \left(\gamma + |\widehat{p}_t| - \frac{\eta}{2} X_\infty^2\right) \le \sum_{t=1}^{n} M_t\, Z_t\, \ell_{\gamma,t}(u) + \frac{\ln d}{\eta} \ .$$

Substituting our choice for η and b yields

$$\sum_{t=1}^{n} M_t\, Z_t \left(b + |\widehat{p}_t|\right) \le \sum_{t=1}^{n} M_t\, Z_t\, \ell_{\gamma,t}(u) + \frac{X_\infty^2 \ln d}{2(1-\alpha)\gamma} \ .$$

To conclude, it suffices to exploit $\mathbb{E}_{t-1} Z_t = b/(b + |\widehat{p}_t|)$ and proceed as in the proof of the previous theorems. ■

4. Experiments

To investigate the empirical behavior of our algorithms we carried out a series of experiments on the first (in chronological order) 40,000 newswire stories from the Reuters Corpus Volume 1 (Reuters, 2000). Each story in this dataset is labelled with one or more elements from a set of 101 categories. In our experiments, we associated a binary classification task with each one of the 50 most frequent categories in the dataset, ignoring the remaining 51 (this was done mainly to reduce the effect of unbalanced datasets). All results presented in this section refer to the average performance over these 50 binary classification tasks. Though all of our algorithms are randomized, we did not compute averages over multiple runs of the same experiment, since we empirically observed that the variances of our statistics are quite small for the sample size taken into consideration.

To evaluate the algorithms we used the F-measure (harmonic average between precision and recall) since this is the most widespread performance index in text categorization experiments. Replacing F-measure with classification accuracy yields results that are qualitatively similar to the ones shown here.

We focused on the following three algorithms: the selective sampling Perceptron algorithm of Figure 1 (here abbreviated as SEL-P), its adaptive version of Figure 2 (abbreviated as SEL-ADA), and the selective sampling second-order Perceptron algorithm of Figure 3 (abbreviated as SEL-2ND).

In Figure 5 we check whether our margin-based sampling technique achieves a better performance than the baseline sampling strategy of querying each label with constant probability. In particular, we fixed 7 different sampling rates (from 29.2% to 71.8%) and run SEL-P each time with the parameter b chosen so as to obtain the desired sampling rate. Then we compared the achieved

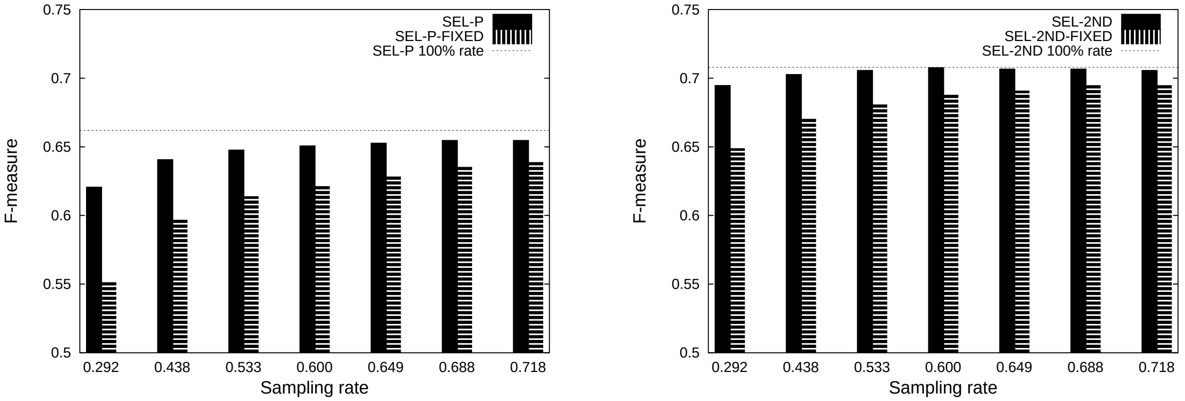

Figure 5: Comparison between margin-based sampling and random sampling with pre-specified sampling rate for the Perceptron algorithm (left) and the second-order Perceptron algorithm (right). The dotted lines show the performance obtained by querying all labels.

performance to the performance obtained by sampling each label with constant probability, i.e., the case when the Bernoulli random variables Z_t in step (3) of Figure 1 have constant parameter equal to the desired sampling rate. We call this variant SEL-P-FIXED. The same experiment was repeated using SEL-2ND and its fixed probability variant SEL-2ND-FIXED.

The following table shows the values of parameter b leading to the fixed sampling rates for both experiments.

SAMPLING RATE	b (SEL-P)	b (SEL-2ND)
0.292	0.250	0.040
0.438	0.500	0.085
0.533	0.750	0.125
0.600	1.000	0.168
0.649	1.250	0.210
0.688	1.500	0.236
0.718	1.750	0.240

Note that in both cases the margin-based sampling technique is clearly dominating. Also, as expected, the difference between the two techniques tends to shrink as the sampling rate gets larger. In Figure 6 we illustrate the sensitivity of performance and sampling rate to different choices of the input parameter b for the two algorithms SEL-P and SEL-2ND. This experiment supports Theorems 1 and 3 in two ways: First, it shows that the choice of b achieving a performance comparable to the one obtained by sampling all labels can save a significant fraction of labels; second, this choice is not unique. Indeed, in a sizeable interval of values for parameter b, the sampling rate decreases significantly with b while the performance level is essentially constant. In Figure 7 we directly compare the performance of SEL-P, SEL-2ND, and SEL-ADA for different values of their average sampling rate (obtained, as before, via suitable choices of their input parameters b and β). This experiment confirms that SEL-2ND is the algorithm offering the best trade-off between performance and sampling rate. On the other hand, the fact that SEL-ADA performs slightly worse than SEL-P, together with the results of Figure 6, appears to indicate that our adaptive choice of b can only be motivated on theoretical grounds.

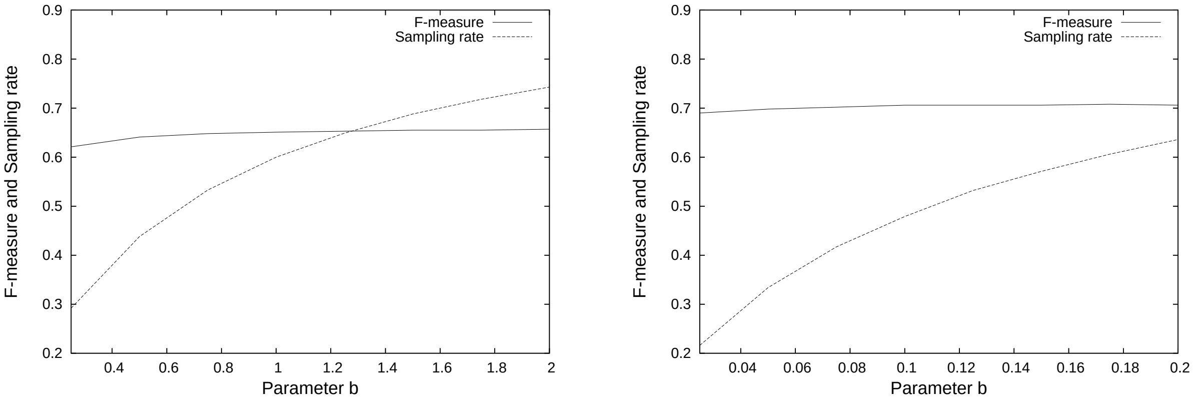

Figure 6: Dependence of performance and sampling rate on the *b* parameter for the Perceptron algorithm (left) and the second-order Perceptron algorithm (right).

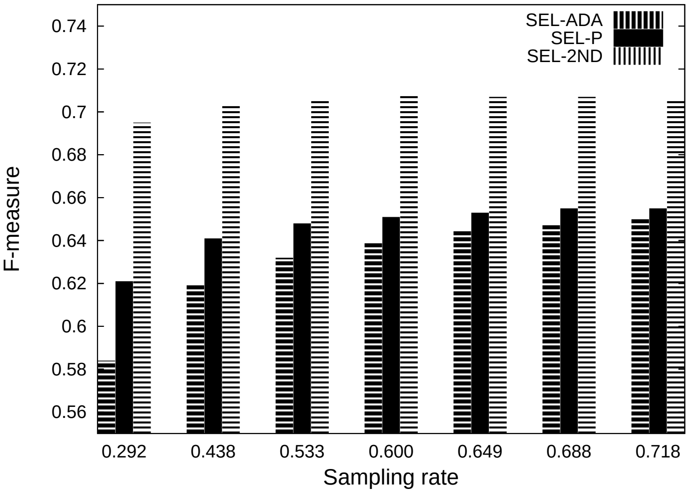

Figure 7: Performance level of SEL-P, SEL-2ND, and SEL-ADA at different sampling rates.

In the last experiment we fixed a target value (0.65) for the F-measure averaged over all 50 categories and we tuned all algorithms to achieve that performance after training on the entire sequence of $40{,}000$ examples. Then, we compared the sampling rates that each algorithm needed to attain the target performance. To get a more accurate picture of the behavior of each algorithm, each time a block of $4{,}000$ training examples was completed, we plotted the average F-measure and sampling rate achieved over that block. The results are reported in Figure 8. Note that SEL-P uses an average sampling rate of about 60%, while SEL-ADA needs a larger (and growing with time) sampling rate of about 74%. On the other hand, SEL-2ND uses only about 9% of the labels. Note also that the sampling rate of SEL-P and SEL-2ND decreases with time, thus indicating that in both cases the margin tends to grow in magnitude. The small sampling rate exhibited by SEL-2ND compared to SEL-P

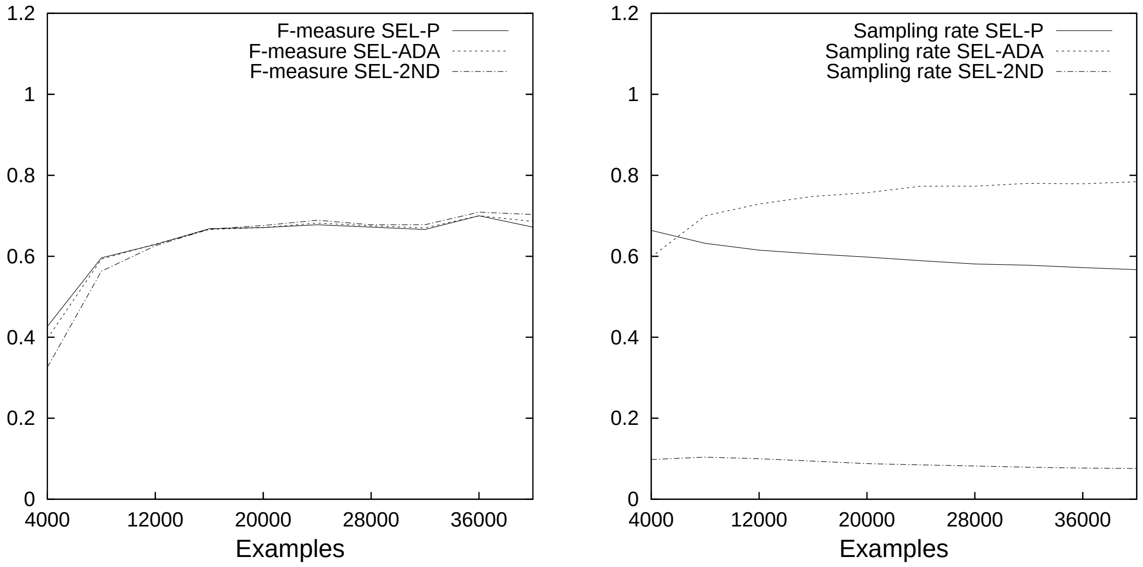

Figure 8: The right plot shows the sampling rates required by different algorithms to achieve a given target performance value (shown in the left plot).

(and SEL-ADA) might be an indication that the second-order Perceptron tends to achieve a larger margin than the standard Perceptron, but we do not have a clear explanation for this phenomenon.

5. Conclusions and Open Problems

We have introduced a general technique for turning linear-threshold algorithms from the general additive family into selective sampling algorithms. We have analyzed these algorithms in a worst-case on-line learning setting, providing bounds on the expected number of mistakes. Our theoretical investigation naturally arises from the traditional way margin-based algorithms are analyzed in the mistake bound model of on-line learning (Littlestone, 1988; Grove et al., 2001; Gentile and Warmuth, 1999; Freund and Schapire, 1999; Gentile, 2003; Cesa-Bianchi et al., 2005). This investigation suggests that our semi-supervised algorithms can achieve, on average, the same accuracy as that of their fully supervised counterparts, but allowing a substantial saving of labels. When applied to (kernel-based) Perceptron-like algorithms, label saving directly implies higher sparsity for the computed classifier which, in turn, yields a running time saving in both training and test phases.

Our theoretical results are corroborated by an empirical comparison on textual data. In these experiments we have shown that proper choices of the scaling parameter b yield a significant reduction in the rate of queried labels without causing an excessive degradation of the classification performance. In addition, we have also shown that by fixing ahead of time the total number of label observations, the margin-driven way of distributing these observations over the training set is largely more effective than a random one.

The choice of the scaling parameter b might affect performance in a significant way. Thus we have also provided a theoretical analysis for an adaptive parameter version of the (first-order) selective sampling Perceptron algorithm. This analysis shows that it is still possible to obtain, with no prior information, a bound on the expected number of mistakes having the same form as the one achieved by choosing the “best” b in hindsight. Now, it is intuitively clear that the number of prediction mistakes and the number of queried labels can be somehow traded-off against each other. Within this trade-off, the above “best” choice is only aimed at minimizing mistakes, rather than queried labels. In fact, the practical utility of this adaptive algorithm seems, at present, fairly limited.

There are many ways this work could be extended. Perhaps the most important is being able to quantify the expected number of requested labels as a function of the problem parameters (margin of the data and so on). It is worth observing that for the adaptive version of the selective sampling Perceptron (Figure 2) we can easily derive a *lower* bound on the label sampling rate. Assume for simplicity that $\|x_t\| = 1$ for all t. Then we can write

$$
\begin{aligned}
\frac{b_{t-1}}{b_{t-1} + |\widehat{p}_t|} &= \frac{\beta\sqrt{1+K_{t-1}}}{\beta\sqrt{1+K_{t-1}} + |w_{t-1}^\top x_t|} \\
&\geq \frac{\beta\sqrt{1+K_{t-1}}}{\beta\sqrt{1+K_{t-1}} + \|w_{t-1}\|} \\
&\geq \frac{\beta\sqrt{1+K_{t-1}}}{\beta\sqrt{1+K_{t-1}} + \sqrt{K_{t-1}}} \qquad \text{(using Inequality (7))} \\
&\geq \frac{\beta}{\beta+1}
\end{aligned}
$$

holding for any trial t. Is it possible to obtain a meaningful *upper* bound? At first glance, this requires a lower bound on the margin $|\widehat{p}_t|$. But since there are no guarantees on the margin the algorithm achieves (even in the separable case), this route does not look profitable. Would such an argument work for on-line large margin algorithms, such as those by Li and Long (2002) and Gentile (2001)?

As a related issue, our theorems do not make any explicit statement about the number of weight updates (i.e., support vectors) computed by our selective sampling algorithms. We would like to see a theoretical argument that enables us to combine the bound on the number of mistakes with a bound on the number of labels, resulting in an informative upper bound on the number of updates.

Finally, the adaptive parameter version of Figure 2 centers on inequalities such as (7) to determine the current label request rate. It seems these inequalities are too coarse to make the algorithm effective in practice. Our experiments basically show that this algorithm tends to query more labels than needed. It turns out there are many ways one can modify this algorithm to make it less “cautious”, though this gives rise to algorithms which seem to escape a crisp mathematical analysis. We would like to devise an adaptive parameter version of the selective sampling Perceptron algorithm that both lends itself to formal analysis and is competitive in practice.

Acknowledgments

The authors would like to thank the anonymous reviewers for their insightful comments that greatly helped to improve the presentation of this paper. In particular, we thank one of them for finding a mistake in our initial version of the proof of Theorem 4. We would also like to thank the action editor for his timely work.

The authors gratefully acknowledge partial support by the PASCAL Network of Excellence under EC grant no. 506778. This publication only reflects the authors' views.

References

D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64:48–75, 2002.

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential familiy of distributions. *Machine Learning*, 43(3):211–246, 2001.

H. D. Block. The Perceptron: A model for brain functioning. *Reviews of Modern Physics*, 34: 123–135, 1962.

A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning reserarch*, 6:1579–1619, 2005.

C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proceedings of the 17th International Conference on Machine Learning*, pages 111–11. Morgan Kaufman, 2000.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. Learning probabilistic linear-threshold classifiers via selective sampling. In *Proceedings of the 16th Annual Conference on Learning Theory, LNAI 2777*, pages 373–386. Springer, 2003.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. A second-order Perceptron algorithm. *SIAM Journal on Computing*, 43(3):640–668, 2005.

N. Cesa-Bianchi and G. Lugosi. Potential-based algorithms in on-line prediction and game theory. *Machine Learning*, 51(3):239–261, 2003.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

R. Cohn, L. Atlas, and R. Ladner. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems 2*. MIT Press, 1990.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2001.

S. Dasgupta, A. T. Kalai, and C. Monteleoni. Analysis of Perceptron-based active learning. In *Proceedings of the 18th Annual Conference on Learning Theory, LNAI 2777*, pages 249–263. Springer, 2005.

O. Dekel, S. Shalev-Shwartz, and Y. Singer. The Forgetron: a kernel-based Perceptron on a fixed budget. In *Advances in Neural Information Processing Systems 18*, pages 259–266. MIT Press, 2006.

R. Duda and P. Hart, and D. Stork. *Pattern classification, second edition*. Wiley Interscience, 2000.

J. Forster. On relative loss bounds in generalized linear regression. In *Proceedings of the 12th International Symposium on Fundamentals of Computation Theory, LNCS 1684*, pages 269–280. Springer, 1999.

Y. Freund and R. Schapire. Large margin classification using the Perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

Y. Freund, S. Seung, E. Shamir, and N. Tishby. Selective sampling using the query by committee algorithm. *Machine Learning*, 28(2/3):133–168, 1997.

C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.

C. Gentile. The robustness of the *p*-norm algorithms. *Machine Learning*, 53(3):265–299, 2003.

C. Gentile and M. Warmuth. Linear hinge loss and average margin. In *Advances in Neural Information Processing Systems 10*, pages 225–231. MIT Press, 1999.

A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001.

D. P. Helmbold, N. Littlestone, and P. M. Long. Apple tasting. *Information and Computation*, 161 (2):85–139, 2000.

D. P. Helmbold and S. Panizza. Some label efficient learning results. In *Proceedings of the 10th Annual Conference on Computational Learning Theory*, pages 218–230. ACM Press, 1997.

M. Herbster and M. K. Warmuth. Tracking the Best Linear Predictor. *Journal of Machine Learning Research*, 1: 281–309, 2001.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

J. Kivinen and M. K. Warmuth. Relative loss bounds for multidimensional regression problems. *Machine Learning*, 45(3):301–329, 2001.

T. L. Lai and C. Z. Wei. Least squares estimates in stochastic regression models with applications to identification and control of dynamic systems. *The Annals of Statistics*, 10(1):154–166, 1982.

Y. Li and P. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46:361–387, 2002.

N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

N. Littlestone. *Mistake Bounds and Logarithmic Linear-threshold Learning Algorithms*. PhD thesis, University of California Santa Cruz, 1989.

A. B. J. Novikov. On convergence proofs on Perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata, vol. XII*, pages 615–622, 1962.

Reuters. Reuters corpus vol. 1, 2000.
URL `about.reuters.com/researchandstandards/corpus/`.

F. Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

S. Tong and D. Koller. Support vector machine active learning with applications to text classification. In *Proceedings of the 17th International Conference on Machine Learning*, pages 999–1006. Morgan Kaufmann, 2000.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

M. K. Warmuth and A. K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *Electronic proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, 1997.