Mach Learn (2011) 83: 71–102
DOI 10.1007/s10994-010-5191-x

Learning noisy linear classifiers via adaptive and selective sampling

Giovanni Cavallanti · Nicolò Cesa-Bianchi · Claudio Gentile

Received: 13 January 2009 / Revised: 13 April 2010 / Accepted: 26 April 2010 /
Published online: 20 May 2010

Abstract We introduce efficient margin-based algorithms for selective sampling and filtering in binary classification tasks. Experiments on real-world textual data reveal that our algorithms perform significantly better than popular and similarly efficient competitors. Using the so-called Mammen-Tsybakov low noise condition to parametrize the instance distribution, and assuming linear label noise, we show bounds on the convergence rate to the Bayes risk of a weaker adaptive variant of our selective sampler. Our analysis reveals that, excluding logarithmic factors, the average risk of this adaptive sampler converges to the Bayes risk at rate $N^{-(1+\alpha)(2+\alpha)/2(3+\alpha)}$ where N denotes the number of queried labels, and $\alpha > 0$ is the exponent in the low noise condition. For all $\alpha > \sqrt{3} - 1 \approx 0.73$ this convergence rate is asymptotically faster than the rate $N^{-(1+\alpha)/(2+\alpha)}$ achieved by the fully supervised version of the base selective sampler, which queries all labels. Moreover, for $\alpha \to \infty$ (hard margin condition) the gap between the semi- and fully-supervised rates becomes exponential.

Keywords Active learning · Selective sampling · Adaptive sampling · Linear classification · Low noise

Editor: Avrim Blum.

Preliminary versions of this paper appeared in the proceedings of NIPS 2002 (Margin-based algorithms for information filtering), COLT 2003 (Learning probabilistic linear-threshold classifiers via selective sampling), and NIPS 2008 (Linear classification and selective sampling under low noise conditions). The authors gratefully acknowledge partial support by the PASCAL2 Network of Excellence under EC grant no. 216886. This publication only reflects the authors' views.

G. Cavallanti (✉) · N. Cesa-Bianchi
DSI, Università degli Studi di Milano, Milano, Italy
e-mail: cavallanti@dsi.unimi.it

N. Cesa-Bianchi
e-mail: cesa-bianchi@dsi.unimi.it

C. Gentile
DICOM, Università dell'Insubria, Varese, Italy
e-mail: claudio.gentile@uninsubria.it

1 Introduction

In the standard online learning protocol for binary classification the learner receives a sequence of instances generated by an unknown source. Each time a new instance is received the learner predicts its binary label, which is then immediately disclosed before the next instance is observed. This protocol is natural in many applications, for instance weather forecasting or stock market prediction, because Nature (or the market) is spontaneously revealing the true label after each learner's guess. However, in many other applications obtaining labels may be an expensive process.

In order to address this problem, selective sampling has been proposed as a more realistic variant of the basic online learning protocol. In this variant the true label of the current instance is never revealed unless the learner decides to issue an explicit query. The learner's performance is then measured with respect to both the number of mistakes (made on the entire sequence of instances) and the number of queries.

A natural sampling strategy is one that tries to identify labels which are likely to be useful to the algorithm, and then queries those labels only. This strategy needs to combine a measure of utility of examples with a measure of confidence. In the case of learning with linear functions a statistic that has often been used to quantify both utility and confidence is the margin.

In this work we follow the margin-based approach and define a selective sampling rule that queries the label whenever the margin of the corresponding instance, with respect to the current linear hypothesis, is smaller (in absolute value) than an adaptive threshold. Margins are computed using a linear learning algorithm based on a simple incremental version of regularized linear least-squares (RLS) for classification. This choice is motivated by the fact that RLS margins can be given a natural probabilistic interpretation, thus allowing a principled approach for setting the adaptive threshold.

We also investigate a slightly modified sampling criterion for solving online adaptive filtering tasks. In adaptive filtering the true binary label of an instance is revealed only if the learner makes a positive prediction. A natural application domain is document filtering, where instances represent documents and a positive prediction corresponds to forwarding the current document to a user. If a document is forwarded, then the user returns a binary relevance feedback (whether the document was interesting or not), which is assumed to be the document's true label. If the document is not forwarded, that is the filter makes a negative prediction, then its label remains unknown. Transforming our sampling rule into a filtering rule is simple. Since querying corresponds to forwarding, which is in turn equivalent to a positive prediction, the transformed rule forwards all instances with a positive margin getting their true labels as feedback. Moreover, the rule also forwards all instances whose negative margin is smaller than the same adaptive threshold used in selective sampling. By doing this, all the labels of small margin instances are obtained at the price of making some mistakes when forwarding instances with a negative margin.

1.1 Overview of results

The main goal of this research is the design of efficient algorithms with a good empirical behavior in selective sampling and filtering tasks. The experiments on a real-world dataset reported in Sect. 3 show that our algorithms compare favorably to other selective sampling and filtering procedures proposed in the literature (Cesa-Bianchi et al. 2006a; Dasgupta et al. 2005; Helmbold et al. 2000; Monteleoni and Kääriäinen 2007).

In order to complement these empirical results with theoretical performance guarantees, we introduce in Sect. 4 a stochastic model defining the distribution of examples $(\mathbf{X}, Y)$. In this model the label conditional distribution $\eta(\mathbf{x}) = \mathbb{P}(Y = 1 \mid \mathbf{X} = \mathbf{x})$ is a linear function determined by the fixed target vector $\mathbf{u} \in \mathbb{R}^d$. Following a standard approach in statistical learning, we parametrize the instance distribution via the Mammen-Tsybakov condition $\mathbb{P}(|\frac{1}{2} - \eta(\mathbf{X})| \le \varepsilon) = O(\varepsilon^\alpha)$.

In the standard online protocol, where the true label is revealed after each prediction, we prove in Theorem 1 that the fully supervised RLS converges to the Bayes risk at rate

$$\widetilde{O}\big(n^{-(1+\alpha)/(2+\alpha)}\big).$$

We then prove in Theorem 2 that an adaptive variant of our selective sampling algorithm converges to the Bayes risk at rate

$$\widetilde{O}\big(n^{-(1+\alpha)/(3+\alpha)}\big)$$

with labels being queried at rate

$$\widetilde{O}\big(n^{-\alpha/(2+\alpha)}\big).$$

When $\mathbb{P}(|\frac{1}{2} - \eta(\mathbf{X})| \le \varepsilon_0) = 0$ for a certain $\varepsilon_0 > 0$ (the hard margin case), we show that our sampling procedure converges to the Bayes risk at rate of order $(\ln n)/n$ with only a logarithmic number of queries, a phenomenon first observed in Freund et al. (1997) and also, under different and more general hypotheses, in Balcan et al. (2006, 2007), Castro and Nowak (2008), Dasgupta et al. (2005), Hanneke (2007).

1.2 Related work

Problems related to selective sampling and, more generally, to active learning are well represented in the statistical literature, in particular in the areas of adaptive sampling and sequential hypothesis testing (see the detailed account in Castro and Nowak (2008)). In statistical learning, the idea of selective sampling (sometimes also called uncertainty sampling) has been first introduced by Cohn et al. (1990, 1994)—see also Lewis and Gale (1994), Muslea et al. (2000).

Castro and Nowak (2008) study a framework in which the learner has the freedom to query arbitrary domain points whose labels are generated stochastically. They prove risk bounds in terms of nonparametric characterizations of both the regularity of the Bayes decision boundary and the behavior of the noise rate in its proximity.

The idea of querying small margin instances when learning linear classifiers has been explored many times in different active learning contexts. Campbell et al. (2000), and also Tong and Koller (2000), study a pool-based model of active learning, where the algorithm is allowed to interactively choose which labels to obtain from an i.i.d. pool of unlabeled instances. A landmark result in the selective sampling protocol is the query-by-committee algorithm of Freund et al. (1997). In the realizable (noise-free) case, and under strong distributional assumptions, this algorithm is shown to require exponentially fewer labels than instances when learning linear classifiers (see also Gilad-Bachrach et al. (2005) for a more practical implementation). An exponential advantage in the realizable case is also obtained with a simple variant of the Perceptron algorithm by Dasgupta et al. (2005), under the only assumption that instances are drawn from the uniform distribution over the unit ball in $\mathbb{R}^d$.

In the general statistical learning case, under no assumptions on the joint distribution of label and instances, selective sampling (or, more generally, active learning) bears no such exponential advantage. Indeed, Kääriäinen (2006) shows that, in order to approach the risk of the best linear classifier f^* within ε, at least order of $(\eta/\varepsilon)^2$ labels are needed, where η

is the risk of f^*. A much more general nonparametric lower bound for active learning is obtained by Castro and Nowak (2008).

The first active learning strategy for an arbitrary family $\mathscr{F}$ of classifiers in the general statistical learning case is the A^2 algorithm of Balcan et al. (2006). A^2 is provably never significantly worse than empirical risk minimization in passive learning. A precise characterization of the convergence rate of A^2 in terms of the *disagreement coefficient*, a quantity defined in terms of $\mathscr{F}$ and the joint distribution of examples, is due to Hanneke (2007). An algorithm that improves on A^2 in terms of convergence rate has been proposed by Dasgupta et al. (2008). Active learning under low noise conditions (such as the Mammen-Tsybakov condition considered in this paper) has been studied by Balcan et al. (2007) in the linear classification case, and by Hanneke (2009) for arbitrary hypothesis classes. In Sect. 5 we discuss the relationship between our results and some of these works.

Note the none of the above algorithms is computationally efficient when learning linear classifiers in the nonrealizable case. In this work, where our goal is to design practical algorithms, we achieve time-efficiency by assuming a specific label noise model—see the discussion in Sect. 4. The good empirical behavior of the algorithms designed under this noise model (Sect. 3) provides a reasonable empirical validation of the model itself.

Finally, we remark that the framework of *learning with queries* (see Angluin (2004) for a survey) radically differs from ours, as in the former the learner can query the labels of arbitrary instances of his choice.

2 The selective sampling and filtering algorithms

We consider the following online selective sampling protocol. At each step $t = 1, 2, \dots$ the sampling algorithm (or *selective sampler*) receives an instance $\mathbf{x}_t \in \mathbb{R}^d$ and outputs a binary prediction for the associated label $y_t \in \{-1, +1\}$. After each prediction, the algorithm has the option of "sampling" (issuing a query) to receive the label y_t. We call *example* the pair $(\mathbf{x}_t, y_t)$. If y_t is observed, the algorithm can choose whether or not to update its internal state using the new information encoded by $(\mathbf{x}_t, y_t)$. If the algorithm decides not to issue the query, the current label y_t remains unobserved.

In this respect we distinguish among three types of examples:

- *Queried examples*. Those examples $(\mathbf{x}_t, y_t)$ whose label y_t has been queried.
- *Stored examples*. Those examples $(\mathbf{x}_t, y_t)$ that have been used by the algorithm for an internal state update after their label y_t was queried (hence stored examples are a subset of queried examples).
- All remaining examples, i.e., those examples $(\mathbf{x}_t, y_t)$ whose labels y_t remain unknown to the algorithm.

Similarly, we define queried/stored instances, and stored labels.

The need for distinguishing between queried and stored examples will become clear in the following section, when we introduce a mistake-driven sampling algorithm that stores only those queried examples on which the current classifier makes a mistake. For now, we focus on the non-mistake-driven margin-based selective sampler described in Fig. 1, and use the words *queried* and *stored* interchangeably. The margin $\widehat{\Delta}_t = \mathbf{w}_t^\top \mathbf{x}_t$ at time t is based on the regularized least squares (RLS) estimator $\mathbf{w}_t$ defined over the set of previously stored examples. More precisely, let N_t be the number of examples stored in the first t steps, let $S_{t-1} = [\mathbf{x}'_1, \dots, \mathbf{x}'_{N_{t-1}}]$ be the $d \times N_{t-1}$ matrix of the instances whose label has been queried

```
Parameters: $K > 0$.
Initialization: weight vector $\mathbf{w} = \mathbf{0}$; query counter $N = 0$.

At each time $t = 1, 2, \ldots$ do the following:

1. Observe instance $\mathbf{x}_t \in \mathbb{R}^d$;
2. Predict the label $y_t \in \{-1, +1\}$ with $\widehat{y}_t = \textsc{sgn}(\mathbf{w}^\top \mathbf{x}_t)$;
3. If $(\mathbf{w}^\top \mathbf{x}_t)^2 \le \|\mathbf{x}_t\|^2 (K \ln t)/N$, then:
    – query the label $y_t$ of $\mathbf{x}_t$,
    – increment $N$,
    – update $\mathbf{w}$ using $(\mathbf{x}_t, y_t)$ as in (1).
```

Fig. 1 The selective sampling procedure

before step t begins, and let $\mathbf{y}_{t-1} = (y'_1, \ldots, y'_{N_{t-1}})$ be the vector of the corresponding labels. Then

$$\mathbf{w}_t = \left(I + S_{t-1} S_{t-1}^\top + \mathbf{x}_t \mathbf{x}_t^\top\right)^{-1} S_{t-1} \mathbf{y}_{t-1} \tag{1}$$

where I is the $d \times d$ identity matrix.[1] Note that $\mathbf{w}_t$ defined by (1) depends on the current instance $\mathbf{x}_t$. The regularized least squares estimator in this particular form has been first considered by Vovk (2001) and, independently, by Azoury and Warmuth (2001). In practice, putting $\mathbf{x}_t \mathbf{x}_t^\top$ into the inverse matrix reduces the variance of $\widehat{\Delta}_t$. In the sequel, we sometimes write $\widehat{\Delta}_{N_{t-1},t}$ instead of $\widehat{\Delta}_t$ to stress the dependence of $\widehat{\Delta}_t$ on the number of stored labels.

At each time step t the algorithm outputs a prediction $\textsc{sgn}(\widehat{\Delta}_t)$ for the label of instance $\mathbf{x}_t$. Then, the algorithm decides whether to issue a query to access the label y_t based on the margin $\widehat{\Delta}_t$ of the current instance, the number N_{t-1} of stored examples, and the current step number t. The intuition behind our sampling rule is the following: whenever our confidence on the prediction made at time t falls below a properly chosen adaptive threshold, that is when $\widehat{\Delta}_t^2 \le \|\mathbf{x}_t\|^2 (K \ln t)/N_{t-1}$, we ask for an additional label to refine our current hypothesis. Note that the threshold vanishes as the number of stored examples grows.

Together with the selective sampler of Fig. 1, we investigate a margin-based procedure for information filtering (or *filter*) described in Fig. 2—see Sculley (2008) for an account of recent advances in online filtering. At each time $t = 1, 2, \ldots$ the filter observes instance $\mathbf{x}_t$ and chooses whether or not to forward it according to its prediction about the relevance of $\mathbf{x}_t$. We say that $\mathbf{x}_t$ is relevant if its associated label y_t is $+1$, and not relevant if $y_t = -1$. If the observed instance is deemed relevant, and thus forwarded, its true label y_t is revealed to the filter; otherwise, no information about the current instance is disclosed. An example wrongly marked as relevant by the filter is called a false positive. Similarly, a relevant example that is not forwarded is called a false negative. Hence, a filtering algorithm should only forward truly relevant examples (so as to avoid false positives) and filter out only irrelevant ones (so as to avoid false negatives).

The filtering algorithm of Fig. 2 is a simple adaptation of the selective sampler described in Fig. 1. At each time step t our filter forwards instance $\mathbf{x}_t$ only if the margin achieved by its current weight vector $\mathbf{w}_t$ is greater than an adaptive *negative* threshold, that is, if $\widehat{\Delta}_t \ge -\|\mathbf{x}_t\| \sqrt{(K \ln t)/N_{t-1}}$. Whenever this condition occurs (note that in this case the sign of the margin $\widehat{\Delta}_t$ matters), instance $\mathbf{x}_t$ is forwarded and the true relevance label y_t is revealed to the algorithm, which then updates its current hypothesis through (1). As in the

[1] Adding the identity matrix I ensures the invertibility of $I + S_{t-1} S_{t-1}^\top + \mathbf{x}_t \mathbf{x}_t^\top$ at the price of adding a bias term to the margin estimator $\widehat{\Delta}_t$ (see Sect. 6.2).

```
Parameters: $K > 0$.
Initialization: weight vector $\mathbf{w} = \mathbf{0}$; query counter $N = 0$.

At each time $t = 1, 2, \ldots$ do the following:

1. Observe instance $\mathbf{x}_t \in \mathbb{R}^d$;
2. If $\mathbf{w}^\top \mathbf{x}_t + \|\mathbf{x}_t\| \sqrt{(K \ln t)/N} \ge 0$, then:
    – forward $\mathbf{x}_t$,
    – receive $y_t \in \{-1, +1\}$ and increment $N$,
    – update $\mathbf{w}$ using $(\mathbf{x}_t, y_t)$ as in (1);
3. else discard $\mathbf{x}_t$.
```

Fig. 2 The filtering procedure

selective sampling rule, the filtering threshold which the margin is compared to vanishes at rate $1/\sqrt{N_t}$.

2.1 Computational issues

The estimator (1) can be stored in space $\Theta(d^2)$, which we need for the inverse of matrix $I + S_{t-1} S_{t-1}^\top$. Moreover, using standard formulas for small-rank adjustments of inverses, we can compute updates and predictions in time $\Theta(d^2)$ as well.

The algorithms described in Figs. 1 and 2 can be also expressed in dual variable form. This is needed, for instance, when we want to use the feature expansion facility provided by kernel functions. In this case, the estimator (1) can be represented in space quadratic in the number of stored labels (in addition, however, all stored instances have to be explicitly maintained). The update time is also quadratic in the number of queries.

3 Experimental results

In this section, we present an empirical study of our algorithms and compare their performance to the one achieved by other margin-based selective samplers and filters in real-world scenarios. Unless stated otherwise our experiments were run on the first 20,000 newswire stories (in chronological order) from the Reuters Corpus Volume 1 (RCV1, NIST 2004) dataset. Every example in this dataset is encoded as a vector of real attributes computed through a standard TD-IDF bag-of-words processing of the original news stories, and is tagged with zero or more labels from the set of 102 Reuters news categories.

This text categorization setup is a natural framework for both selective sampling and filtering algorithms. In the former setting, a large amount of (unlabelled) data is usually available, as is typically the case for newswire stories, but it is expensive to have a human feedback after each prediction. Analogously, in a news filtering service a feed of news articles from a wide range of different topics is typically available, and only a given subset of these topics is of interest to the target recipient.

Following the common practice in text categorization applications, the classification performance is evaluated using the F-measure $2RP/(R+P)$, where P is the precision (fraction of correctly classified documents among all documents that were classified positive for the given topic) and R is the recall (fraction of correctly classified documents among all documents that are labelled with the given topic).

The experiments reported here involve algorithms employing both first-order and second-order update steps. In this context, the term first-order refers to a Perceptron-like linear classifier. We use the term second-order to denote linear classifiers based on incremental versions of the RLS procedure (see, e.g., Azoury and Warmuth 2001; Cesa-Bianchi et al. 2005; Vovk 2001). In this respect, both our selective sampler and filter are second-order procedures. All algorithms were evaluated using dual variable implementations and linear kernels.

3.1 Selective sampling

We start by investigating how variants of our original selective sampling scheme in Fig. 1 perform in a typical selective sampling setting. Unless otherwise specified, all results presented here are averaged over the 50 most frequent categories. This initial investigation also takes into account the impact of different parameter values on the final performance.

The following four variants were considered:

- The basic Selective Sampler (here referred to as SS). This is the algorithm outlined in Fig. 1.
- A variant of SS, referred to as SSNCI (Selective Sampling Not Current Instance), that uses the estimator $\mathbf{w}_t = (I + S_{t-1}\, S_{t-1}^\top)^{-1} S_{t-1}\mathbf{y}_{t-1}$. Note that, unlike (1), the current instance is not used for computing the prediction vector $\mathbf{w}_t$. When a dual variable representation is adopted, this results in a prediction step that is linear (instead of quadratic) in the number of stored examples. In fact, in dual variables $\mathbf{w}_t = \mathbf{y}_{t-1}^\top (I + S_{t-1}^\top S_{t-1})^{-1} S_{t-1}^\top$. Since $\mathbf{x}_t$ is not part of the weight vector $\mathbf{w}_t$, the vector $\mathbf{y}_{t-1}^\top (I + S_{t-1}^\top S_{t-1})^{-1}$ can be computed just after each update, so that only N_{t-1} inner products have to be carried out at each prediction step. Although it is true that this algorithm is faster than SS and predicts the same labels as SS whenever the stored examples are the same, it should be noted that the magnitude of the computed margins are smaller, thus directly affecting the selection of the examples to store.
- A mistake-driven version, here referred to as SSMD (Selective Sampling Mistake Driven), of the basic selective sampler. This can be obtained by replacing Line 3 in Fig. 1 with

  3′. If $(\mathbf{w}^\top \mathbf{x}_t)^2 \le \|\mathbf{x}_t\|^2 (K \ln t)/N$ then:
  – Query y_t;
  – If $y_t \mathbf{w}^\top \mathbf{x}_t < 0$ then increment N and update $\mathbf{w}$ as in (1).

  When a small margin is detected this variant queries the current label. Then, only in the case of sign disagreement, the current example $(\mathbf{x}_t, y_t)$ is stored. Because this algorithm works with a mistake-driven logic, the number of actual updates is much smaller than the number of queried labels. Therefore, SSMD runs in practice much faster than all other selective sampling variants. Note that in this version of the selective sampling algorithm N should be interpreted as the number of examples actually stored, rather than the number of queried labels, the former being typically much smaller than the latter.
- An *adaptive* sampling algorithm (here referred to as SSNL, Selective Sampling Next Label). This variant queries the label of the *next* instance (rather than the current one) whenever the observed margin falls below the threshold value. This is the only variant of selective sampling we have been able to analyze from a theoretical standpoint—see Sect. 5.

Evidence collected in the first series of experiments is summarized in Figs. 3 and 4. In order to study how effective our sampling procedures are, two non-sampling baselines, the Perceptron and the second-order Perceptron (Cesa-Bianchi et al. 2005) algorithms, here referred to as PERC and SOP, respectively, are included in both figures. These two algorithms are clearly designed for online classification setups where labels are revealed after each

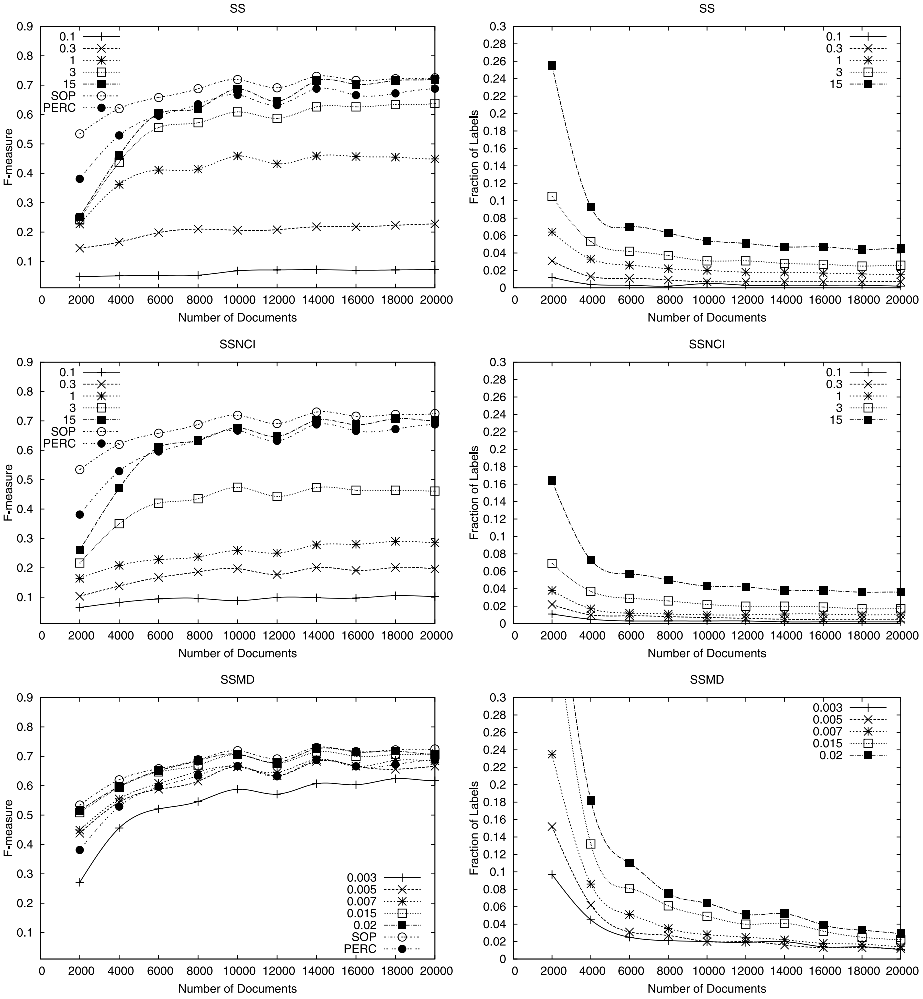


Fig. 3 Evolution of F-measure (*left*) and fraction of queried labels (*right*). At each point the F-measure is computed on the previous 2,000 examples as an average over the 50 most frequent categories. *Each row* of plots illustrates the behavior of a different algorithm (SS, SSNCI and SSMD) for different values of the parameter K. The standard Perceptron (PERC) and the second-order Perceptron (SOP) algorithms are included for comparison

prediction step (i.e., in a fully supervised fashion). Figures 3 and 4 give evidence that both SS and SSMD can be as effective as the fully supervised baselines, and yet use only a fraction of the labels. In fact, they actually surpass the F-measure achieved by PERC and match the one obtained by SOP when the parameter K is properly chosen.

For each algorithm two different plots are presented side by side. The one on the left shows the progression of the F-measure while the plot on the right tracks the fraction of queried labels. Both are computed as a function of the number of observed documents. Figure 3 shows that SS is slightly more effective than SSNCI for the same value of K but has otherwise the same behavior. In fact when the parameters are chosen in such a way that the number of queried labels are similar, the corresponding F-measure curves tend to overlap.

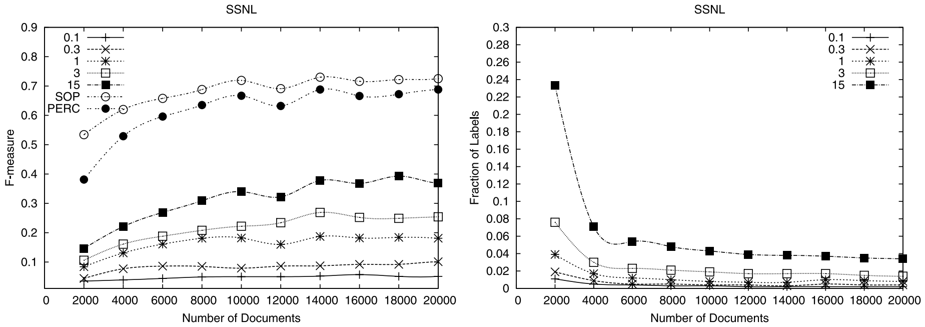


Fig. 4 Behavior of the adaptive sampler SSNL. As in Fig. 3 each point is computed on the previous 2,000 examples as an average over the 50 most frequent categories. The standard Perceptron (PERC) and the second-order Perceptron (SOP) algorithms are included for comparison

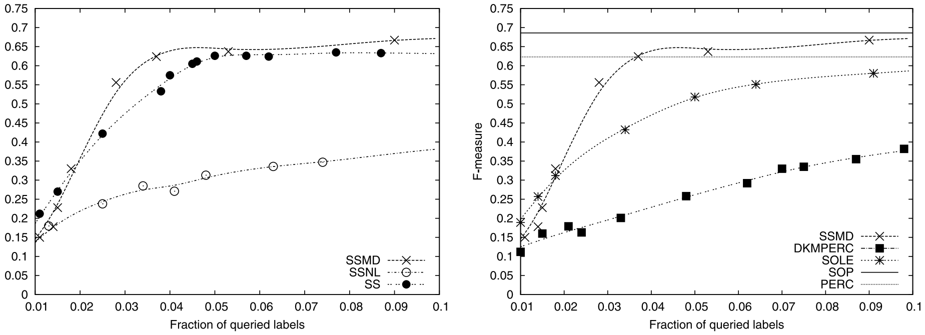


Fig. 5 F-measure obtained by different algorithms as a function of the number of observed labels. Each F-measure value is computed based on the algorithm's predictions output during a single run over 20,000 examples. Plots are obtained by repeatedly running each algorithm with different values of its parameter. *Trend lines* are computed as approximate cubic splines connecting consecutive points. Both SS and SSMD quickly converge to the F-measure obtained by the second-order Perceptron algorithm (SOP). On the other hand, DKM and SSNL exhibit a much slower convergence rate

As previously hinted, SSMD queries many more labels than SS for the same value of the parameter K. This is due to the fact that the threshold does not necessarily shrink whenever we query an example, since the mistake-driven mechanism prevents properly predicted queried examples to be actually stored. This is also the reason why, in order to obtain comparable plots, we had to select a different range of values for K.

Finally, as expected, SSNL exhibits a less than ideal performance. Specifically, comparing SS and SSNL makes evident that the former can actually detect and exploit informative instances, thus effectively performing selective sampling, while the latter is only able to detect the need for new information encoded in the labels. See Sect. 5 for a more thorough discussion on this issue.

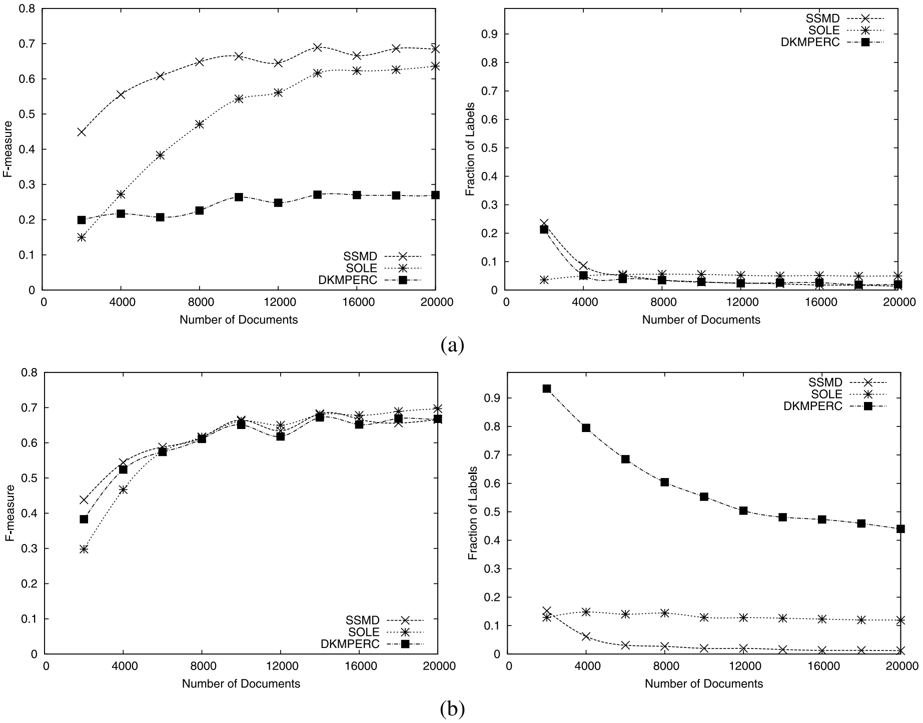
A more direct comparison among selective sampling variants is contained in Fig. 5 (left), where we show how SS compares to SSMD and to SSNL. In the right part of Fig. 5 we compare SSMD to other state-of-the-art algorithms, including the second-order version (here referred to as SOLE, Second-Order Label Efficient) of the label efficient classifier introduced in Cesa-Bianchi et al. (2006b), and to the DKMPERC variant (Monteleoni and Kääriäinen 2007) of the DKM sampler (Dasgupta et al. 2005). Unlike the SS-series introduced in this paper, the label efficient algorithms of Cesa-Bianchi et al. (2006b) involve an internal randomization designed to cope with (adversarial) situations, when no assumptions are made on the process generating the examples. Even though such algorithms are randomized, we did not take averages over multiple runs since the results we obtained exhibited low variance. The DKM Perceptron is a recently proposed margin-based algorithm that queries labels only when the observed margin is below a given time-changing threshold, and then performs mistake-driven updates. The threshold value is halved whenever the algorithm sees a certain number of consecutive correctly predicted examples whose margin is lower than the current threshold. This number is the algorithm's sole parameter. Whereas the original version of DKM features an update rule whose learning rate depends on the margin of the wrongly predicted instance, the version evaluated here uses the standard Perceptron update. We found this version to perform better than the basic one, as also confirmed by Monteleoni and Kääriäinen (2007).

Each point in the plots describes the F-measure achieved by a given algorithm run with a certain value of its parameter (this parameter is K for SSMD, b for SOLE, and μ for DKMPERC—see, e.g., Monteleoni and Kääriäinen (2007) for an explanation of b and μ). Since different parameter values result in different numbers of labels queried at the end of a run, we chose to actually record on the horizontal axis the percentage of queried labels rather than the parameter value. Trends were then obtained by running each algorithm with different values of their parameters, each run thus resulting in a certain query rate and cumulative F-measure. The plotted data show that when the number of observed labels rises above some value, the F-measures achieved by SS and SSMD stop increasing, or increase at a much reduced rate. It is reasonable to assume that in a selective sampling setup we are interested in the behavior of an algorithm when the fraction of observed labels stays below some threshold, say 10%. In this range SSMD outperforms all other algorithms. We also observe that, because of its mistake-driven behavior, SSMD is much faster than the other selective sampling variants. Under our test conditions DKMPERC proved ineffective probably because most tasks in the RCV1 dataset are not linearly separable. A similar behavior was observed in Monteleoni and Kääriäinen (2007). In particular, DKMPERC is also outperformed by SSNL, as immediately evinced by superimposing Fig. 5 (left) and (right).

In the three subsequent figures (Figs. 6, 7 and 8) we focus on SSMD only, since it appears to be the best selective sampling variant within our SS-series and, as such, the best candidate for real-world applications.

Figure 6(a) describes the performance of the three algorithms SSMD, SOLE, and DKMPERC when their parameters are chosen in such a way that the fraction of observed labels is around 5.0%, about the middle of the 0% to 10% range. The left plot shows that most of the learning takes place on the first part of the dataset. In particular, SSMD exhibits the shortest and most effective learning phase. In fact, its empirical performance turns out to be essentially unaffected by observing progressively fewer labels. As for querying rates, we see that the non-randomized algorithms have a higher rate on the first documents of the stream, whereas SOLE keeps querying a fixed number of labels throughout the entire run.

The curves in Fig. 6(b) somehow complement those contained in Fig. 6(a). Here we show how the F-measure and sampling rate of the evaluated algorithms change throughout a run when their parameters are set so as each algorithm has a cumulative F-measure value of 0.60 at the end of the run. This is the highest F-measure that can be achieved by the worst algorithm evaluated in our tests. It clearly appears that whereas the learning curves are similar, DKMPERC differs in the way it queries labels in order to achieve this common behavior. Specifically, both SSMD and SOLE ask a smaller number of labels (overall), with a slightly more intensive sampling rate for SSMD in the initial phase. On the other hand,

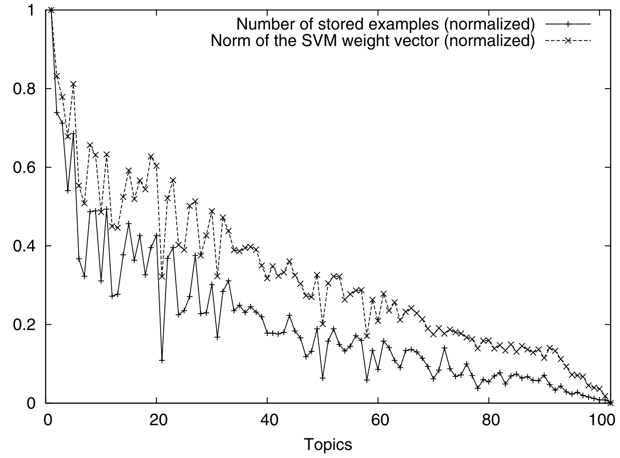


(a)

(b)

Fig. 6 F-measure achieved by different algorithms (*left*) and corresponding observed labels (*right*) when their parameters are chosen in such a way that: (**a**) the fraction of queried labels after 20,000 examples is around 5.0%; (**b**) the F-measure averaged over the predictions output on the last 2,000 examples is 0.60

Fig. 7 Correlation between the number of stored examples and the difficulty of each (binary) task, as measured by the norm of the SVM weight vector. Topics are sorted by decreasing frequency of positive examples



DKMPERC samples almost all labels in the initial stage of the run, and then slowly decreases the rate.

Finally, in order to investigate how different problems influence storage and sampling rate of SSMD, and in order to assess the impact of the number of positive examples on per-

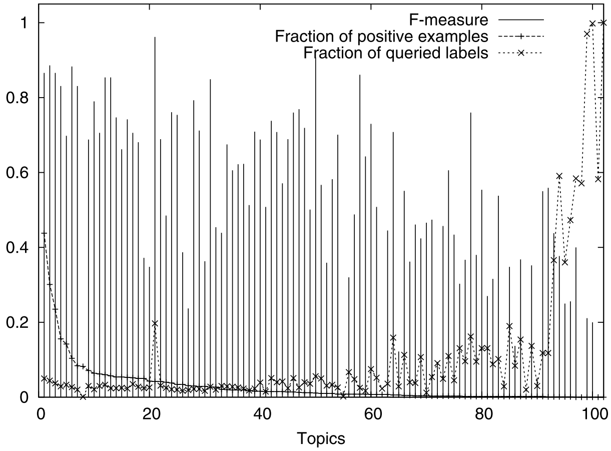


Fig. 8 F-measure achieved on different (binary) classification tasks compared to the number of positive examples in each topic, and to the fraction of queried labels. As in Fig. 7 topics are sorted by decreasing frequency of positive examples

formance, we report in Fig. 7 the number of stored examples on the different binary learning tasks (those associated with each topic), and in Fig. 8 the corresponding F-measure and fraction of queried labels. Data are gathered over the whole set of 102 categories. To account for uncommon categories we extended the learning phase to the first 40,000 RCV1 stories. In both plots topics are sorted by frequency with the most frequent topics appearing on the left. We represent the difficulty of a learning task as the norm of the weight vector obtained by running the C-SVM algorithm on that task.[2] Figure 7 clearly shows that SSMD rises its storage rate on problems that are more difficult. In particular, even if two different tasks have largely different numbers of positive examples, the storage rate achieved by SSMD on those tasks may be similar when the norm of the weight vectors computed by C-SVM is nearly the same. On the other hand, Fig. 8 makes evident that the achieved F-measure is fairly independent of the number of positive examples, but this independence is obtained at the cost of querying more and more labels. In other words, SSMD seems to realize the difficulty of learning infrequent topics and, in order to achieve a good F-measure performance, it compensates by querying many more labels.

3.2 Filtering

We now examine the empirical performance of our filtering algorithm (Fig. 2). In a filtering setting the interesting examples are usually a small fraction of the whole stream. For this reason, the results we report are averaged over the 32 topics whose frequency in the first 20,000 newswire stories of the RCV1 dataset is between 1% and 5%. In Fig. 9 we separately plot the precision and recall obtained by our filter for different values of parameter K. Each point in the plots is computed on the previous 2,000 examples. Figure 9 (left) shows that the recall does not considerably change throughout a run. In particular, as the value of K increases (thereby widening the threshold area), the number of false negatives gets smaller while achieving a higher recall. Instead, as K grows beyond 5 the number of false positives quickly outnumbers the true positives, and this prevents the filtering algorithm from matching the precision curve obtained by the second-order Perceptron algorithm (SOP).

[2] The actual values were computed using SVM-LIGHT (Joachims 1999) with default parameters. Since the examples in the Reuters Corpus Volume 1 are cosine normalized, the choice of default parameters amounts to indirectly setting the parameter C to approximately 1.0.

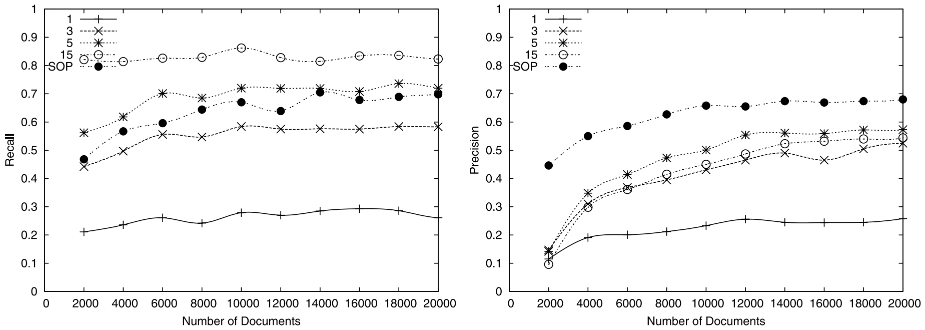


Fig. 9 Recall (*left*) and precision (*right*) obtained by our filtering algorithm for various choices of the parameter K. Each point is computed on the previous 2,000 examples as an average over those topics whose frequency is in the range 1%–5%

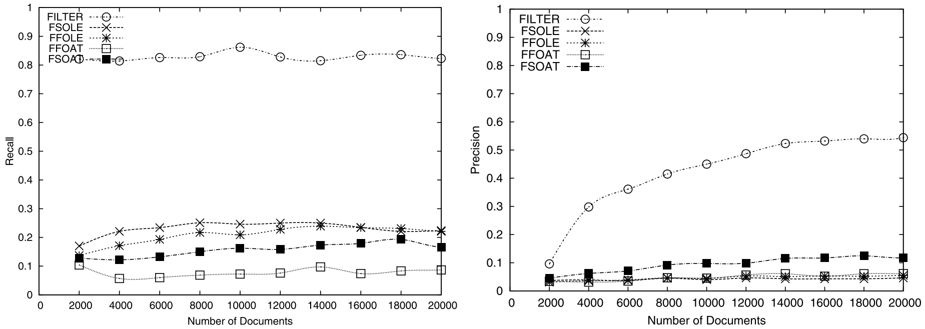


Fig. 10 Recall (*left*) and precision (*right*) obtained by different margin-based filtering algorithms. Each point is computed on the previous 2,000 examples as an average over those topics whose frequency is in the range 1%–5%. Results for the apple-tasting algorithms are averaged over three runs

In order to better understand the experimental behavior of our filtering algorithm, we performed a comparative investigation by testing our filter against the "apple-tasting" algorithm of Helmbold et al. (2000), and a straightforward adaptation to the filtering framework of the label efficient classifiers (Cesa-Bianchi et al. 2006b). Both these filtering procedures can be seen as standard classification algorithms with an additional mechanism specifically designed to let them operate within the constraints set by the filtering protocol. The apple-tasting approach prescribes that at each time step a prediction is issued using an underlying classification algorithm, and a positive prediction is forced (independent of the prediction output by the underlying classifier) with probability $\sqrt{(1+m)/t}$, being m the number of false positives occurred up to time t. On the other hand, a label efficient classifier can be used as a filter by forcing a positive prediction whenever a label is queried. Keeping up with the separation between first and second-order algorithms, we tested both procedures using first-order and second-order Perceptron algorithms as the underlying classifiers. Therefore, the following five filtering algorithms were considered (see Fig. 10):

- Our filtering procedure in Fig. 2 (FILTER);
- An apple-tasting filter with a first-order underlying classifier (FFOAT);
- An apple-tasting filter with a second-order underlying classifier (FSOAT);

- A label efficient filter with a first-order underlying classifier (FFOLE);
- A label efficient filter with a second-order underlying classifier (FSOLE).

In this experiment the parameters of FILTER and those of FFOLE and FSOLE are tuned. The apple-tasting method, on the other hand, does not depend on input parameters. Results plotted in Fig. 10 show that our filtering algorithm is by far superior to both apple-tasting and label efficient filters: it achieves a much higher recall while being able to considerably improve its precision as the number of observed documents increases.

4 Probabilistic model, Bayes classifier and regret

We now provide a formal framework within which we analyze a specific variant of the selective sampling algorithm described in the previous sections.

First, we make assumptions on the source of the examples $(\mathbf{x}_t, y_t)$. We assume instances $\mathbf{x}_t$ are realizations of i.i.d. random variables $\mathbf{X}_t$ drawn from an unknown distribution on the surface of the unit Euclidean sphere in $\mathbb{R}^d$, so that $\|\mathbf{X}_t\| = 1$ for all $t \geq 1$. We also assume that labels y_t are generated according to the following simple linear noise model: there exists a fixed and unknown vector $\mathbf{u} \in \mathbb{R}^d$, with Euclidean norm $\|\mathbf{u}\| = 1$, such that $\mathbb{E}[Y_t \mid \mathbf{X}_t = \mathbf{x}_t] = \mathbf{u}^\top \mathbf{x}_t$ for all $t \geq 1$, where Y_t is used to denote the random label at time t. Hence $\mathbf{X}_t = \mathbf{x}_t$ is labelled 1 with probability $(1 + \mathbf{u}^\top \mathbf{x}_t)/2 \in [0, 1]$. Note that $\text{SGN}(f^*)$, for $f^*(\mathbf{x}) = \mathbf{u}^\top \mathbf{x}$, is the Bayes optimal classifier under the described noise model.

Note that the linear noise assumption is less restrictive than it appears. Indeed, all of our learning algorithms can be reformulated in dual variables in order to learn functions from any given reproducing kernel Hilbert space (RKHS) as opposed to vectors from $\mathbb{R}^d$. When working in a RKHS $\mathscr{H}$, the noise model is defined through $\mathbb{E}[Y_t \mid \mathbf{X}_t = \mathbf{x}_t] = g(\mathbf{x}_t)$ where g is any element of $\mathscr{H}$ such that $g(\mathbf{X}_t) \in [-1, 1]$ w.p. 1. Hence, by suitably choosing $\mathscr{H}$ we can approximate any nonlinear continuous noise functions.

In what follows, all probabilities $\mathbb{P}$ and expectations $\mathbb{E}$ are understood with respect to the joint distribution of the i.i.d. data process $\{(\mathbf{X}_1, Y_1), (\mathbf{X}_2, Y_2), \ldots\}$. We use $\mathbb{P}_t$ to denote conditioning on $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_t, Y_t)$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be an arbitrary measurable function. The *instantaneous regret* $R(f)$ is the excess risk of $\text{SGN}(f)$ with respect to the Bayes risk, that is,

$$R(f) = \mathbb{P}(Y_1 f(\mathbf{X}_1) < 0) - \mathbb{P}(Y_1 f^*(\mathbf{X}_1) < 0).$$

Let $f_1, f_2, \ldots$ be a sequence of real functions where each f_t is measurable with respect to the σ-algebra generated by $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{t-1}, Y_{t-1}), \mathbf{X}_t$. When $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{t-1}, Y_{t-1})$ is understood from the context, we write f_t as a function of $\mathbf{X}_t$ only. Let $R_{t-1}(f_t)$ be the *conditional* instantaneous regret $R_{t-1}(f_t) = \mathbb{P}_{t-1}(Y_t f_t(\mathbf{X}_t) < 0) - \mathbb{P}_{t-1}(Y_t f^*(\mathbf{X}_t) < 0)$. Our goal is to bound the *cumulative (expected) regret*

$$\mathbb{E}\left[\sum_{t=1}^{n} R(f_t)\right] = \mathbb{E}\big[R_0(f_1) + R_1(f_2) + \cdots + R_{n-1}(f_n)\big]$$

as a function of n, and other relevant quantities. Observe that, although the learner's predictions can only depend on the observed instances and queried labels, the above regret is computed over *all* time steps, including those time steps t when the selective sampler did not issue a query.

As mentioned in previous sections, we consider algorithms that predict the value of Y_t through $\text{SGN}(\mathbf{W}_t^\top \mathbf{X}_t)$, where $\mathbf{W}_t \in \mathbb{R}^d$ is a dynamically updated weight vector which might

```
Initialization: weight vector $\mathbf{w} = \mathbf{0}$.

At each time $t = 1, 2, \ldots$ do the following:

1. Observe instance $\mathbf{x}_t \in \mathbb{R}^d$;
2. Predict the label $y_t \in \{-1, 1\}$ with $\mathrm{SGN}(\mathbf{w}^\top \mathbf{x}_t)$;
3. Query the label $y_t$ of $\mathbf{x}_t$;
4. Update $\mathbf{w}$ using $(\mathbf{x}_t, y_t)$ as in (1).
```

Fig. 11 The online regularized least-squares classifier

be intended as the current estimate for $\mathbf{u}$ (when the data source is stochastic, the estimator $\mathbf{w}_t$ of Sect. 2 becomes a random variable $\mathbf{W}_t$). We denote by $\widehat{\Delta}_t$ the margin $\mathbf{W}_t^\top \mathbf{X}_t$, whenever $\mathbf{W}_t$ is understood from the context, and by Δ_t the Bayes function $f^*(\mathbf{X}_t)$. Thus $\widehat{\Delta}_t$ is the current approximation to Δ_t. Note that $\widehat{\Delta}_t$ is measurable with respect to the σ-algebra generated by $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{t-1}, Y_{t-1}), \mathbf{X}_t$. Sometimes $\widehat{\Delta}_{N_{t-1},t}$ is used instead of $\widehat{\Delta}_t$ to stress the dependence of $\widehat{\Delta}_t$ on the number of stored labels.

We model the distribution of the instances around the hyperplane $\mathbf{u}^\top \mathbf{x} = 0$, using the popular *Mammen-Tsybakov low noise condition*:

Assumption 1 (Tsybakov 2004) There exist $c > 0$ and $\alpha \geq 0$ such that

$$\mathbb{P}\big(|f^*(\mathbf{X}_1)| < \varepsilon\big) \leq c\varepsilon^\alpha \quad \text{for all } \varepsilon > 0.$$

When the noise exponent α is 0 the low noise condition becomes vacuous. In order to study the case $\alpha \to \infty$, one can use the following equivalent formulation (e.g., Bartlett et al. 2006, Lemma 9) of Assumption 1: There exist $c > 0$ and $\alpha \geq 0$ such that

$$\mathbb{P}\big(f^*(\mathbf{X}_1) f(\mathbf{X}_1) < 0\big) \leq c R(f)^{\alpha/(1+\alpha)} \quad \text{for all measurable } f : \mathbb{R}^d \to \mathbb{R}.$$

With this formulation one can see that $\alpha \to \infty$ implies the *hard margin condition* $|f^*(\mathbf{X}_1)| \geq 1/(2c)$ with probability 1.

In order to provide a proper assessment of our noise model, we introduce here (but defer the proof to Sect. 6) a theoretical result that establishes a regret bound for a fully supervised sampling algorithm. This algorithm, described in Fig. 11, predicts using RLS and queries (and stores) the label of every observed instance. This result serves as a baseline against which we measure the performance of the selective sampling algorithm. Note that the regret bound is expressed in terms of the whole spectrum of the process correlation matrix $\mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^\top]$.

Theorem 1 *Assume the low noise condition* (*Assumption* 1) *holds with exponent* $\alpha \geq 0$ *and constant* $c > 0$. *Then the expected cumulative regret after* n *steps of the fully supervised algorithm in Fig.* 11 *is bounded by*

$$\mathbb{E}\left[\Big(4c\big(1 + \ln\big|I + S_n S_n^\top\big|\big)\Big)^{\frac{1+\alpha}{2+\alpha}}\right] n^{\frac{1}{2+\alpha}}.$$

This, in turn, is upper bounded by

$$\left[4c\left(1 + \sum_{i=1}^{d} \ln(1 + n\lambda_i)\right)\right]^{\frac{1+\alpha}{2+\alpha}} n^{\frac{1}{2+\alpha}} = O\left(\big(d \ln n\big)^{\frac{1+\alpha}{2+\alpha}} n^{\frac{1}{2+\alpha}}\right).$$

In the above $|\cdot|$ *denotes the determinant of the matrix at argument*, $S_n = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n]$ *is the* (*random*) *matrix containing all instances*, *and* λ_i *is the* i*th eigenvalue of* $\mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^\top]$.

Remark 1 When $\alpha = 0$ (corresponding to a vacuous noise condition) the bound of Theorem 1 reduces to $O(\sqrt{d\,n \ln n})$. When $\alpha \to \infty$ (the hard margin condition) the bound gives the logarithmic behavior $O(d \ln n)$. Note that $\sum_{i=1}^d \ln(1 + n\lambda_i)$ is substantially smaller than $d \ln n$ whenever the spectrum of $\mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^\top]$ is rapidly decreasing. In fact, the second bound is clearly meaningful even when $d = \infty$, while the third one only applies to the finite dimensional case.

Remark 2 Fast rates of convergence (i.e., rates faster than $n^{-1/2}$) have typically been proven for batch-style algorithms, such as empirical risk minimizers and SVM (see, e.g., Bartlett et al. 2006; Steinwart and Scovel 2007; Tsybakov 2004; see also Boucheron et al. 2005 for a survey) rather than for online algorithms. A reference closer to our paper is Ying and Zhou (2006), where the authors prove bounds for online linear classification using the low noise condition (1), though under different distributional assumptions.

Remark 3 When divided by the number n of steps, the bound of Theorem 1 is of the order $n^{-\frac{1+\alpha}{2+\alpha}}$. Despite the fact we do not have a lower bounding argument holding for our *specific* label noise model $\mathbb{E}[Y_t \,|\, \mathbf{X}_t] = \Delta_t$, we would like to stress that these convergence rates actually match, up to log-factors, the best known upper bounds holding under Assumption 1 (not involving labels Y_t). Hence, we tend to consider the cumulative rate $n^{\frac{1}{2+\alpha}}$ in Theorem 1 as a good reference result to compare against.

Remark 4 The second bound in Theorem 1 makes explicit the dependence on the spectrum $\lambda_1, \lambda_2, \ldots$ of the process correlation matrix $\mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^\top]$. As far as we can tell, this bound is novel. In the analysis of the adaptive sampling algorithm in the next section, we will not obtain such a clean dependence on the process spectrum.

5 Adaptive sampling

As mentioned in Sect. 2, there are two basic intuitions behind the selective sampler of Fig. 1. First, the observation of small margin instances should lead to more queries. Indeed, we know that the sampling rate should be roughly proportional to the squared inverse of the typical margin. Second, queries should be prevalently issued on small margin instances, as these belong to a high-noise region (where more samples are needed to correctly estimate the sign of the Bayes optimal classification).

A formal analysis of selective sampling within the stochastic framework introduced in the previous section should thus capture both of these intuitions. On the one hand, it should show that the selective sampler is able to adaptively determine the correct sampling rate as a function of the actual amount of noise in the source of data (quantified by the exponent α in Assumption 1). On the other hand, the analysis should capture the advantage brought by querying instances with a small margin.

Unfortunately, such a result for the sampler of Fig. 1 is not within reach of our analysis. Indeed, the use of standard concentration results in our stochastic framework is impeded by the fact that queried labels, when conditioned on their associated instances, are no longer (conditionally) independent random variables. This fact prevents us from controlling bias and variance of our estimator.

```
Parameters: $\lambda > 0$, $\rho_t > 0$ for each $t \ge 1$.
Initialization: weight vector $\mathbf{w} = \mathbf{0}$; query counter $N = 0$.

At each time $t = 1, 2, \dots$ do the following:

1. Observe instance $\mathbf{x}_t \in \mathbb{R}^d$: $\|\mathbf{x}_t\| = 1$;
2. Predict the label $y_t \in \{-1, 1\}$ with $\widehat{y}_t = \mathrm{SGN}(\mathbf{w}^\top \mathbf{x}_t)$;
3. If $N \le \rho_t$ then schedule the storage of $(\mathbf{x}_t, y_t)$;
4. Else if $(\mathbf{w}^\top \mathbf{x}_t)^2 \le (128 \ln t)/(\lambda N)$ then schedule the storage of $(\mathbf{x}_{t+1}, y_{t+1})$;
5. If $(\mathbf{x}_t, y_t)$ is scheduled to be stored then:
    – increment $N$,
    – update $\mathbf{w}$ using $(\mathbf{x}_t, y_t)$ as in (1).
```

Fig. 12 The adaptive sampling procedure (called "adaptive sampler" in the text)

In order to circumvent this key technical problem we slightly modify the sampling criterion: We retain the same query condition as the algorithm in Fig. 1 but, at the same time, we ensure that the stored labels are indeed a sequence of independent random variables. This is done by considering an *adaptive* (rather than *selective*) sampler that queries the label of the random instance received immediately *after* each small margin instance. As the analysis in the next section shows, the adaptive sampler is able to learn on the fly a correct sampling rate for the labels albeit without focusing its queries on the most informative instances. Indeed, our adaptive sampling analysis measures the advantage of sampling at the rate at which we observe small margins, whereas it says nothing about the advantage, confirmed by the experiments, of querying precisely the small margin instances.

Finally, note that the trick of querying the next label for ensuring conditionally independent label sequences cannot be applied to the algorithm for adaptive filtering of Fig. 2. In fact, in the filtering setting querying a label implies that the corresponding instance is forwarded to the user, and we cannot hope to prove good bounds for a filter that defers the forwarding of requests (thus essentially forwarding randomly drawn instances). This is the reason why the algorithm in Fig. 2 is evaluated only empirically, with no accompanying theoretical statements.

We now turn to the description of our adaptive sampling algorithm. The algorithm, shown in Fig. 12, queries all labels (and stores all examples) during an initial stage of length at least $(16d)/\lambda^2$, where λ denotes the smallest nonzero eigenvalue of the process correlation matrix $\mathbb{E}[\mathbf{X}_1\mathbf{X}_1^\top]$. When this transient regime is over, the sampler issues a query at time t based on both the query counter N_{t-1} and the (signed) margin $\widehat{\Delta}_t$. Specifically, if evidence is collected that the number N_{t-1} of stored labels is smaller than our current estimate of $1/\Delta_t^2$, that is if $\widehat{\Delta}_t^2 \le (128 \ln t)/(\lambda N_{t-1})$, then we query (and store) the label of the next instance $\mathbf{X}_{t+1}$. Observe that once an example is scheduled to be stored, the algorithm cannot change its mind on the next time step (because, say, the new margin now happens to be larger than the current threshold).

Our algorithm is able to adaptively optimize the sampling rate by exploiting the additional information provided by the examples having small margin. The appropriate rate clearly depends on the (unknown) amount of noise α which the algorithm implicitly learns on the fly.

Before turning to our main theoretical result, it is worth discussing some of the crucial issues that the adaptive behavior brings about. First and foremost, by dropping the selective sampling step we get around a technical problem which prevents the analysis of the sampler of Fig. 1. On the other hand, it allows us to state a theoretical guarantee for a computationally efficient algorithm and one that casts light on the closely related sampling counterpart. We find the efficiency a key aspect of this algorithm. In particular, the issue of obtaining efficient implementations of state-of-the-art selective sampling algorithms (e.g., Dasgupta et al. 2008; Hanneke 2009) is, as far as we know, an open problem which is not a straightforward consequence of published results. For instance, even when $P(DIS(V))$ and $P(R)$ in algorithm A^2 are replaced by empirical estimates, we do not see how their computation can be generally carried out in an efficient way for all kinds of input distributions. In fact, as analyzed in Dasgupta et al. (2008), Hanneke (2009), A^2 seems to rely on a routine (denoted there by LEARN_H) which is unclear how to make it work efficiently. Efficiently maintaining (or even sampling from) a version space which is possibly infinite-dimensional appears to be a common problem in all these papers. However, the fact that existing methods have no efficient implementation (up to special cases) seems to follow from the more general perspective these methods are taking, rather than being related to the selective vs. adaptive choice.

Theorem 2 *Assume the low noise condition* (*Assumption* 1) *holds with unknown exponent* $\alpha \ge 0$, *and assume the adaptive sampler of Fig.* 12 *is run with* $\rho_t = \frac{16}{\lambda^2}\max\{d, \ln t\}$. *Then the expected cumulative regret after n steps is bounded by*

$$O\left(\frac{d+\ln n}{\lambda^2} + \left(\frac{\ln n}{\lambda}\right)^{\frac{1+\alpha}{3+\alpha}} n^{\frac{2}{3+\alpha}}\right)$$

whereas the expected number of queried labels (*including the stored ones*) *is bounded by*

$$O\left(\frac{d+\ln n}{\lambda^2} + \left(\frac{\ln n}{\lambda}\right)^{\frac{\alpha}{2+\alpha}} n^{\frac{2}{2+\alpha}}\right).$$

The proof (given in Sect. 6) hinges on showing that $\widehat{\Delta}_t$ is an estimate of the true margin Δ_t, and relies on known concentration properties of i.i.d. processes. In particular, we show that our sampling algorithm is able to adaptively estimate the number of queries needed to ensure a negligible regret when a query is not issued (more precisely, we show that when a query is not issued at time t the regret increases by at most $1/t$). Before turning to the proofs, we would like to make a few remarks.

Remark 5 As expected, when we compare our semi-supervised adaptive sampler (Theorem 2) to its fully supervised counterpart (Theorem 1), we see that the average instantaneous regret of the former vanishes at a significantly slower rate than the latter, i.e., $n^{-\frac{1+\alpha}{3+\alpha}}$ vs. $n^{-\frac{1+\alpha}{2+\alpha}}$ excluding log factors. Note, however, that the instantaneous regret of the semi-supervised algorithm vanishes faster than the fully-supervised algorithm when both regrets are expressed in terms of the number N of issued queries. To see this consider first the case $\alpha \to \infty$ (the hard margin case). Then both algorithms have an average regret of order $(\ln n)/n$. However, since the semi-supervised algorithm makes only $N = O(\ln n)$ queries, we have that, as a function of N, the average regret of the semi-supervised algorithm is of order N/e^N whereas the fully supervised has only $(\ln N)/N$. We have thus recovered the exponential advantage observed in previous works. When $\alpha = 0$ (vacuous noise conditions), the average regret rates in terms of N become (excluding logarithmic factors) of order $N^{-1/3}$ in the semi-supervised case and of order $N^{-1/2}$ in the fully supervised case. Hence, there is a critical value of α where the semi-supervised bound becomes better. In order to find this critical value we write the rates of the average instantaneous regret for $0 \le \alpha < \infty$ obtaining

$N^{-\frac{(1+\alpha)(2+\alpha)}{2(3+\alpha)}}$ (semi-supervised algorithm) and $N^{-\frac{1+\alpha}{2+\alpha}}$ (fully supervised algorithm). By comparing the two exponents we find that, asymptotically, the semi-supervised rate is better than the fully supervised one for all values of $\alpha > \sqrt{3} - 1$. This indicates that adaptive sampling is advantageous when the noise level (as modeled by the Mammen-Tsybakov condition) is not too high.

Finally, we note in passing that under the assumption of prior knowledge on the noise level α it would be possible to achieve a regret performance similar to the one given in Theorem 2 by simply replacing the margin-criterion used in step 4 of Fig. 12 by a coin-flipping mechanism whose bias is properly tuned as a function of α.

Remark 6 Hanneke (2009) shows for the algorithm A^2 a tail bound on the instantaneous regret after N queries of order $N^{-(1+\alpha)/2}$ irrespective of the label noise model. For all values of $0 \le \alpha < \infty$ this is better than the instantaneous regret rate of $N^{-\frac{(1+\alpha)(2+\alpha)}{2(3+\alpha)}}$ implied by Theorem 2. But as we already pointed out, no efficient implementation of A^2 is known for linear classification when working on arbitrary distributions over the instance domain.

Remark 7 Note that the adaptively adjusted margin threshold used by the algorithm of Fig. 12 explicitly depends, through λ, on additional information about the data-generating process. This additional information is needed because, unlike the fully supervised classifier of Theorem 1, the adaptive sampler queries labels at random steps. This prevents us from bounding the sum of conditional variances of the RLS estimator through $\ln|I + S_n S_n^\top|$, as we do when proving Theorem 1 (see Sect. 6). Instead, we have to individually bound each conditional variance term via the smallest empirical eigenvalue of the correlation matrix, and this causes the bound of Theorem 2 to depend (inversely) on the smallest process eigenvalue, rather than the whole process eigenspectrum as in Theorem 1. The transient regime in Fig. 12 is needed precisely to ensure that this smallest empirical eigenvalue gets close enough to λ.

Remark 8 Observe that the way it is stated now, the bound of Theorem 2 only applies to the finite-dimensional ($d < \infty$) case. It turns out this is a fixable artifact of our analysis, rather than an intrinsic limitation of the adaptive sampling scheme in Fig. 12. See Sect. 6.3.

Remark 9 We stress that it is fairly straightforward to add to the algorithm of Fig. 12 a mistake-driven rule for storing examples. Such a rule prescribes that, when a small margin is detected, a query is issued (and the next example is stored) only if $\mathrm{SGN}(\widehat{\Delta}_t \neq y_t)$, i.e., when the current prediction is wrong. This modification would make the algorithm more similar to the algorithm SSMD empirically tested in Sect. 3. It is easy to adapt our analysis to obtain for this algorithm the same regret bound as the one established in Theorem 2. However, in this case we can only give guarantees on the expected number of *stored* examples. Although this can be much smaller than the actual number of *queried* labels, it provides a good indication of actual running times. We again refer the reader to Sect. 6.3.

6 Analysis

This section contains the proofs of Theorems 1 and 2.

We denote by $\{a\}$ the indicator function of the event or predicate a, and we repeatedly use simple facts related to indicator functions, such as $\{a \vee b\} = \{a\} + \{b \wedge \neg a\} \le \{a\} + \{b\}$ and $\{a\} = \{a \wedge b\} + \{a \wedge \neg b\} \le \{a \wedge b\} + \{\neg b\}$, where b is another predicate.

6.1 Proof of Theorem 1

The proof proceeds by relating the classification regret of the algorithm to its square loss regret via a "comparison theorem". The square loss regret is then controlled by applying a known pointwise bound for the RLS regression function.

For all measurable $f : \mathbb{R}^d \to \mathbb{R}$, introduce the square loss regret

$$R_\phi(f) = \mathbb{E}\big[\big(1 - Y_1 f(\mathbf{X}_1)\big)^2 - \big(1 - Y_1 f_\phi^*(\mathbf{X}_1)\big)^2\big]$$

along with its conditional version $R_{t-1,\phi}$, where $\phi(z) = (1-z)^2$ and f_ϕ^* is the Bayes optimal function for the square loss.

Lemma 1 *If Assumption 1 holds with exponent* $\alpha \ge 0$ *and constant* $c > 0$, *then for all measurable* f

$$R(f) \le \big(4cR_\phi(f)\big)^{\frac{1+\alpha}{2+\alpha}}.$$

Proof Note that the square loss $\phi(z)$ is *classification-calibrated* in the sense of Bartlett et al. (2006) because it is differentiable in 0 and $\phi'(0) < 0$, see Bartlett et al. 2006, Theorem 4. We can therefore apply Theorem 10 in Bartlett et al. (2006) to ϕ obtaining, for all functions f,

$$cR(f)^{\frac{\alpha}{1+\alpha}}\,\psi\left(\frac{R(f)^{\frac{1}{1+\alpha}}}{2c}\right) \le R_\phi(f)$$

where $\psi(z) = z^2$ is the transformation function associated with $\phi(z)$, see Bartlett et al. 2006, Theorem 4. Solving for $R(f)$ gives the desired result. □

Now, observe that, conditioned on $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_{t-1}, Y_{t-1})$, $\widehat{\Delta}_t = \mathbf{W}_t^\top \mathbf{X}_t$ is a deterministic (nonlinear) function $f_t : \mathbb{R}^d \to \mathbb{R}$. Hence we can write

$$\begin{aligned}
&\sum_{t=1}^n \Big(\mathbb{P}\big(Y_t \widehat{\Delta}_t < 0\big) - \mathbb{P}\big(Y_t\, \Delta_t < 0\big)\Big) \\
&\quad = \mathbb{E}\left[\sum_{t=1}^n R_{t-1}(f_t)\right] \\
&\quad \le \mathbb{E}\left[\sum_{t=1}^n \Big(4cR_{t-1,\phi}(f_t)\Big)^{\frac{1+\alpha}{2+\alpha}}\right] \qquad \text{(by Lemma 1)} \\
&\quad \le \mathbb{E}\left[n\left(\frac{4c}{n}\sum_{t=1}^n R_{t-1,\phi}(f_t)\right)^{\frac{1+\alpha}{2+\alpha}}\right] \qquad \text{(by Jensen's inequality).}
\end{aligned}$$

Further, it is easy to verify that in our probabilistic model $f^*(\mathbf{x}) = \mathbf{u}^\top \mathbf{x}$ is Bayes optimal for the square loss as well; i.e., $f_\phi^* = f^*$. Hence

$$\sum_{t=1}^n R_{t-1,\phi}(f_t) = \sum_{t=1}^n \big(Y_t - \mathbf{W}_t^\top \mathbf{X}_t\big)^2 - \sum_{t=1}^n \big(Y_t - \mathbf{u}^\top \mathbf{X}_t\big)^2.$$

The right-hand side is bounded pointwise (see, e.g., Cesa-Bianchi and Lugosi 2006, Theorem 11.8) by $1 + \ln|I + S_n S_n^\top|$. Combining and simplifying yields

$$\sum_{t=1}^{n}\Big(\mathbb{P}\big(Y_t\,\widehat{\Delta}_t < 0\big) - \mathbb{P}\big(Y_t\,\Delta_t < 0\big)\Big) \le \mathbb{E}\left[\Big(4c\big(1+\ln\big|I + S_n S_n^\top\big|\big)\Big)^{\frac{1+\alpha}{2+\alpha}}\right] n^{\frac{1}{2+\alpha}}$$

i.e., the first bound in Theorem 1. Next, we take this bound and apply Jensen's inequality twice, first to the concave function $(\cdot)^{\frac{1+\alpha}{2+\alpha}}$ of a real argument, and then to the concave function $\ln|\cdot|$ of a (positive definite) matrix argument. Observing that $\mathbb{E}[S_n S_n^\top] = \mathbb{E}[\sum_{t=1}^n \mathbf{X}_t\mathbf{X}_t^\top] = n\mathbb{E}[\mathbf{X}_1\mathbf{X}_1^\top]$ yields the second bound. The third bound derives from the second one by using $\lambda_i \le 1$.

6.2 Proof of Theorem 2

We first introduce some preliminary results and formal definitions used in our analysis.

For any choice of s, $(\mathbf{x}'_1, y'_1), \dots, (\mathbf{x}'_s, y'_s)$ and $\mathbf{x}_t$, let $\mathbb{E}_{s,t}[\,\cdot\,]$ be the conditional expectation

$$\mathbb{E}_{s,t}[\,\cdot\,] = \mathbb{E}\big[\cdot \,\big|\, N_{t-1} = s, \mathbf{X}'_1 = \mathbf{x}'_1, \dots, \mathbf{X}'_s = \mathbf{x}'_s, \mathbf{X}_t = \mathbf{x}_t\big].$$

Since the analysis of our algorithm relies on proving that, conditioned on the past s queried labels, $\widehat{\Delta}_t$ is a good estimator of the corresponding margin Δ_t, we need to consider both the bias and the variance of $\widehat{\Delta}_t$.

Let $S = [\mathbf{x}'_1, \dots, \mathbf{x}'_s]$, $\mathbf{Y} = (y'_1, \dots, y'_s)$, and $\mathbf{W}_t = (I + SS^\top + \mathbf{X}_t\mathbf{X}_t^\top)^{-1} S\mathbf{Y}$ as in (1). Recalling that $\mathbb{E}_{s,t}[\mathbf{Y}] = S^\top\mathbf{u}$ we have

$$\begin{aligned}
\mathbb{E}_{s,t}[\widehat{\Delta}_{s,t}] &= \mathbb{E}_{s,t}\big[\mathbf{W}_t^\top\mathbf{x}_t\big] \\
&= \mathbb{E}_{s,t}[\mathbf{Y}^\top]S^\top\big(I + SS^\top + \mathbf{x}_t\mathbf{x}_t^\top\big)^{-1}\mathbf{x}_t \\
&= \mathbf{u}^\top SS^\top(I + SS^\top + \mathbf{x}_t\mathbf{x}_t^\top)^{-1}\mathbf{x}_t \\
&= \Delta_t - \mathbf{u}^\top(I + \mathbf{x}_t\mathbf{x}_t^\top)\big(I + SS^\top + \mathbf{x}_t\mathbf{x}_t^\top\big)^{-1}\mathbf{x}_t \\
&= \Delta_t - B_{s,t}
\end{aligned}$$

where $B_{s,t} = \mathbf{u}^\top(I + \mathbf{x}_t\mathbf{x}_t^\top)(I + SS^\top + \mathbf{x}_t\mathbf{x}_t^\top)^{-1}\mathbf{x}_t$ is the (additive) bias. Note also that $\widehat{\Delta}_{s,t}$ can be rewritten as

$$\widehat{\Delta}_{s,t} = \sum_{k=1}^{s} Y'_k Z_k$$

where Y'_k is the label of instance $\mathbf{X}'_k$ and $\mathbf{Z} = (Z_1, \dots, Z_s)^\top = S^\top(I + SS^\top + \mathbf{x}_t\mathbf{x}_t^\top)^{-1}\mathbf{x}_t$ so that $\|\mathbf{Z}\|^2$ bounds the conditional variance of $\widehat{\Delta}_{s,t}$. Bias and variance can be both bounded in terms of quadratic forms. In particular, setting $r_{s,t} = \mathbf{x}_t^\top(I + SS^\top + \mathbf{x}_t\mathbf{x}_t^\top)^{-1}\mathbf{x}_t$ we can use, e.g., Lemmas 7 and 8 in Cesa-Bianchi et al. (2006a) to conclude

$$|B_{s,t}| \le \sqrt{\mathbf{x}_t^\top(I + SS^\top + \mathbf{x}_t\mathbf{x}_t^\top)^{-2}\mathbf{x}_t} + r_{s,t}, \tag{2}$$

$$\|\mathbf{Z}\|^2 \le r_{s,t}. \tag{3}$$

In turn, $r_{s,t}$ can be (crudely) bounded from above as follows:

$$\begin{aligned}
r_{s,t} &= \mathbf{x}_t^\top(I + S\,S^\top + \mathbf{x}_t\mathbf{x}_t^\top)^{-1}\mathbf{x}_t \\
&\le \mathbf{x}_t^\top(I + S\,S^\top)^{-1}\mathbf{x}_t
\end{aligned}$$

$$\begin{aligned}&\le \left\|(I + S\,S^\top)^{-1}\right\|\\&\le \frac{1}{1+\widehat{\lambda}_s}\end{aligned}$$

where $\widehat{\lambda}_s$ denotes the smallest eigenvalue of the empirical correlation matrix $SS^\top$. Similarly, the quadratic form $\mathbf{x}_t^\top(I + S\,S^\top + \mathbf{x}_t\mathbf{x}_t^\top)^{-2}\mathbf{x}_t$ occurring in (2) can be bounded by $(1+\widehat{\lambda}_s)^{-2}$. Hence, we end up with

$$|B_{s,t}| \le \frac{2}{1+\widehat{\lambda}_s}, \tag{4}$$

$$\|\mathbf{Z}\|^2 \le \frac{1}{1+\widehat{\lambda}_s}. \tag{5}$$

These are the actual bounds on bias and variance we will be using throughout.

The following lemma is of preliminary importance for bounding the instantaneous regret of our algorithm. This allows us to factor out a small margin term which is effectively controlled by the Mammen-Tsybakov noise condition.

Lemma 2 *Assume the low noise condition* (*Assumption* 1) *holds with exponent* $\alpha \ge 0$ *and constant* $c > 0$. *Then, for all* $\varepsilon > 0$ *and all random variables* $\widehat{\Delta}_t$,

$$\mathbb{P}(Y_t\widehat{\Delta}_t < 0) - \mathbb{P}(Y_t\Delta_t < 0) \le c\varepsilon^{1+\alpha} + \mathbb{P}(\widehat{\Delta}_t\Delta_t \le 0, |\Delta_t| \ge \varepsilon).$$

Proof We have

$$\begin{aligned}\{Y_t\widehat{\Delta}_t < 0\} - \{Y_t\Delta_t < 0\} &= \{Y_t\widehat{\Delta}_t < 0, |\Delta_t| \ge \varepsilon\} - \{Y_t\Delta_t < 0, |\Delta_t| \ge \varepsilon\}\\&\quad + \{Y_t\widehat{\Delta}_t < 0,\ |\Delta_t| < \varepsilon\} - \{Y_t\Delta_t < 0, |\Delta_t| < \varepsilon\}\end{aligned}$$

where

$$\begin{aligned}\{Y_t\widehat{\Delta}_t < 0, |\Delta_t| \ge \varepsilon\} &= \{Y_t\widehat{\Delta}_t < 0, Y_t\Delta_t < 0, |\Delta_t| \ge \varepsilon\} + \{Y_t\widehat{\Delta}_t < 0, Y_t\Delta_t \ge 0, |\Delta_t| \ge \varepsilon\}\\&\le \{Y_t\Delta_t < 0, |\Delta_t| \ge \varepsilon\} + \{\widehat{\Delta}_t\Delta_t \le 0, |\Delta_t| \ge \varepsilon\}.\end{aligned}$$

Combining and taking expectations of both sides yields

$$\begin{aligned}\mathbb{P}(Y_t\widehat{\Delta}_t < 0) - \mathbb{P}(Y_t\Delta_t < 0) &\le \mathbb{P}(\widehat{\Delta}_t\Delta \le 0, |\Delta_t| \ge \varepsilon)\\&\quad + \mathbb{P}(Y_t\widehat{\Delta}_t < 0, |\Delta_t| < \varepsilon) - \mathbb{P}(Y_t\Delta_t < 0, |\Delta_t| < \varepsilon).\end{aligned} \tag{6}$$

Moreover, our label noise model implies

$$\mathbb{P}(Y_t\widehat{\Delta}_t < 0, |\Delta_t| < \varepsilon) = \mathbb{P}\big(Y_t\widehat{\Delta}_t < 0 \,\big|\, |\Delta_t| < \varepsilon\big)\mathbb{P}(|\Delta_t| < \varepsilon) \le \frac{1+\varepsilon}{2}\mathbb{P}(|\Delta_t| < \varepsilon).$$

Likewise,

$$\mathbb{P}(Y_t\Delta_t < 0, |\Delta_t| < \varepsilon) = \mathbb{P}\big(Y_t\Delta_t < 0 \,\big|\, |\Delta_t| < \varepsilon\big)\mathbb{P}(|\Delta_t| < \varepsilon) \ge \frac{1-\varepsilon}{2}\mathbb{P}(|\Delta_t| < \varepsilon).$$

Finally, from Assumption 1 we get $\mathbb{P}(|\Delta_t| < \varepsilon) \le c\varepsilon^\alpha$. Combining as in (6) gives the claimed result. □

The next lemma establishes a concentration property about the smallest eigenvalue of the empirical correlation matrix towards the smallest eigenvalue of the correlation matrix describing the underlying stochastic process. It easily follows from Blanchard et al. (2007), Theorem 4.2 (see also the earlier reference results in Shawe-Taylor et al. (2005)).

Lemma 3 *Let S be the $d \times s$ matrix $[\mathbf{X}_1, \dots, \mathbf{X}_s]$, where $\mathbf{X}_1, \dots, \mathbf{X}_s$ are i.i.d. samples from a process whose correlation matrix $\mathbb{E}[\mathbf{X}_1 \mathbf{X}_1^\top]$ has minimal eigenvalue $\lambda > 0$. If $s \ge \frac{16d}{\lambda^2}$ and $\widehat{\lambda}_s$ is the smallest eigenvalue of $SS^\top$, then*

$$\mathbb{P}\left(\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2}\right) \le e^{-\lambda^2 s/8}. \tag{7}$$

The final ancillary result we need is contained in the next lemma. This lemma (whose proof is provided in the Appendix) is essential in that it allows to use large deviation bounds on i.i.d. variables.

Lemma 4 *Recall the adaptive sampling mechanism of the algorithm in Fig. 12. For each $i \ge 1$ let $T_i + 1$ be the time at which we query an instance for the i-th time and let $Z_{T_i} = (\mathbf{X}_{T_i}, Y_{T_i})$ be the example whose margin caused instance $\mathbf{X}_{T_i+1}$ to be queried. Then $Z_{T_1+1}, Z_{T_2+1}, \dots$ are independent random variables distributed as Z_1.*

We are now ready to prove Theorem 2.

We start by applying Lemma 2 and then further manipulate the resulting terms:

$$\begin{aligned}
&\sum_{t=1}^{n} \big(\mathbb{P}(Y_t \widehat{\Delta}_{N_{t-1},t} < 0) - \mathbb{P}(Y_t \Delta_t < 0)\big) \\
&\quad \le cn\varepsilon^{1+\alpha} + \sum_{t=1}^{n} \mathbb{P}\big(\Delta_t \widehat{\Delta}_{N_{t-1},t} \le 0, |\Delta_t| \ge \varepsilon\big) \\
&\quad \le cn\varepsilon^{1+\alpha} + \underbrace{\sum_{t=1}^{n} \mathbb{P}\left(N_{t-1} \le \rho_t\right)}_{\text{(I)}} \\
&\qquad + \underbrace{\sum_{t=1}^{n} \mathbb{P}\left(\widehat{\Delta}^2_{N_{t-1},t} \le \frac{128 \ln t}{\lambda N_{t-1}}, N_{t-1} > \rho_t, |\Delta_t| \ge \varepsilon\right)}_{\text{(II)}} \\
&\qquad + \underbrace{\sum_{t=1}^{n} \mathbb{P}\left(\Delta_t \widehat{\Delta}_{N_{t-1},t} \le 0, \widehat{\Delta}^2_{N_{t-1},t} > \frac{128 \ln t}{\lambda N_{t-1}}, N_{t-1} > \rho_t\right)}_{\text{(III)}}.
\end{aligned}$$

Term (I) bounds the regret on those steps t that trigger the storage of the current example (because $N_{t-1} \le \rho_t$). Term (II) bounds the regret on those steps on which we schedule a query for the next example (because $\widehat{\Delta}_t^2$ is smaller than the threshold at time t) despite the true margin Δ_t is not small (because $|\Delta_t| \ge \varepsilon$). Finally, term (III) bounds the regret on those steps that do not trigger any queries at all (i.e., over the non-sampled examples). We proceed by bounding the three terms separately.

In order to bound (I), we simply observe that $N_0 = 0$, and $N_{t-1} \le \rho_t$ implies $N_t = N_{t-1} + 1$. Therefore

$$(\mathrm{I}) = \mathbb{E}\left[\sum_{t=1}^{n} \{N_{t-1} \le \rho_t\}\right] \le \rho_n \tag{8}$$

just because $\rho_n \ge \rho_t$ for all $t \le n$.

To bound (II) we set

$$L_\varepsilon = \sum_{t=1}^{n} \left\{ N_{t-1} \le \frac{128 \ln t}{\lambda \widehat{\Delta}^2_{N_{t-1},t}}, N_{t-1} > \rho_t, |\Delta_t| \ge \varepsilon \right\}$$

so that (II) $= \mathbb{E} L_\varepsilon$. Then, for any positive integer M, we can write

$$L_\varepsilon \le M + \sum_{t=1}^{n} \left\{ N_{t-1} \le \frac{128 \ln t}{\lambda \widehat{\Delta}^2_{N_{t-1},t}}, N_{t-1} > \rho_t, |\Delta_t| \ge \varepsilon, N_{t-1} \ge M \right\}.$$

We proceed by applying t he union bound on each term of the sum,

$$\begin{aligned}
&\left\{ N_{t-1} \le \frac{128 \ln t}{\lambda \widehat{\Delta}^2_{N_{t-1},t}}, N_{t-1} > \rho_t, |\Delta_t| \ge \varepsilon, N_{t-1} \ge M \right\} \\
&\quad \le \sum_{s=\max\{\rho_t+1, M\}}^{t-1} \left\{ s \le \frac{128 \ln t}{\lambda \widehat{\Delta}^2_{s,t}}, |\Delta_t| \ge \varepsilon \right\} \\
&\quad \le \sum_{s=\max\{\rho_t+1, M\}}^{t-1} \left(\left\{ s \le \frac{4 \times 128 \ln t}{\lambda \Delta_t^2}, |\Delta_t| \ge \varepsilon \right\} + \left\{ |\widehat{\Delta}_{s,t}| \le \frac{|\Delta_t|}{2}, |\Delta_t| \ge \varepsilon \right\} \right) \\
&\quad \le \sum_{s=\max\{\rho_t+1, M\}}^{t-1} \left(\left\{ s \le \frac{4 \times 128 \ln t}{\lambda \varepsilon^2} \right\} + \left\{ |\widehat{\Delta}_{s,t} - \Delta_t| \ge \frac{|\Delta_t|}{2}, |\Delta_t| \ge \varepsilon \right\} \right) \\
&\quad \le \sum_{s=\max\{\rho_t+1, M\}}^{t-1} \left(\left\{ s \le \frac{4 \times 128 \ln t}{\lambda \varepsilon^2} \right\} + \left\{ |\widehat{\Delta}_{s,t} + B_{s,t} - \Delta_t| \ge \frac{\varepsilon}{2} - |B_{s,t}| \right\} \right) \\
&\quad \le \sum_{s=\max\{\rho_t+1, M\}}^{t-1} \left(\left\{ s \le \frac{4 \times 128 \ln t}{\lambda \varepsilon^2} \right\} + \left\{ |\widehat{\Delta}_{s,t} + B_{s,t} - \Delta_t| \ge \frac{\varepsilon}{4} \right\} + \left\{ |B_{s,t}| \ge \frac{\varepsilon}{4} \right\} \right).
\end{aligned}$$

We then treat the bias term $\{|B_{s,t}| \ge \frac{\varepsilon}{4}\}$ through (4) and isolate the term referring to the concentration of eigenvalues:

$$\begin{aligned}
\left\{ |B_{s,t}| \ge \frac{\varepsilon}{4} \right\} &\le \left\{ \frac{2}{1 + \widehat{\lambda}_s} \ge \frac{\varepsilon}{4} \right\} \\
&\le \left\{ \frac{1}{1 + \lambda s/2} \ge \frac{\varepsilon}{8} \right\} + \left\{ \widehat{\lambda}_s \ge \frac{\lambda s}{2} \right\} \\
&\le \left\{ s \le \frac{16}{\lambda\, \varepsilon} \right\} + \left\{ \frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2} \right\}.
\end{aligned}$$

This gives

$$L_\varepsilon \le M + \sum_{t=1}^{n} \sum_{s=\max\{\rho_t+1,M\}}^{t-1} \left(\left\{ s \le \frac{4 \times 128 \ln t}{\lambda \varepsilon^2} \right\} + \left\{ |\widehat{\Delta}_{s,t} + B_{s,t} - \Delta_t| \ge \frac{\varepsilon}{4} \right\} \right.$$
$$\left. + \left\{ s \le \frac{16}{\lambda\,\varepsilon} \right\} + \left\{ \frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2} \right\} \right). \tag{9}$$

Now, if $M \ge \frac{4\times 128}{\lambda\varepsilon^2} \ln n = \frac{512}{\lambda\varepsilon^2} \ln n$, then the first and the third term inside the double sum (9) vanish (observe that $s \ge M$ in the inner sum). Thus, under this condition, we end up with

$$L_\varepsilon \le M + \sum_{t=1}^{n} \sum_{s=\max\{\rho_t+1,M\}}^{t-1} \left(\left\{ |\widehat{\Delta}_{s,t} + B_{s,t} - \Delta_t| \ge \frac{\varepsilon}{4} \right\} + \left\{ \frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2} \right\} \right).$$

We want to apply expectations to both sides of the last inequality. Lemma 4 states that the queried examples $(\mathbf{X}'_1, Y'_1), \ldots, (\mathbf{X}'_s, Y'_s)$ are a sequence of independent random variables distributed as $(\mathbf{X}_1, Y_1)$. Hence, we drop the primes and simply write $(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_s, Y_s)$. Since the process $\{(\mathbf{X}_t, Y_t) : t = 1, 2, \ldots\}$ is independent we have that

$$\mathbb{P}(Y_1, \ldots, Y_{t-1} \mid \mathbf{X}_1, \ldots, \mathbf{X}_{t-1}) = \mathbb{P}(Y_1 \mid \mathbf{X}_1) \times \cdots \times \mathbb{P}(Y_{t-1} \mid \mathbf{X}_{t-1}).$$

Hence $Y_1, \ldots, Y_s$ are independent when conditioned on the queried instances. Recalling (5), we have

$$\widehat{\Delta}_{s,t} = \sum_{k=1}^{s} Y_k Z_k \quad \text{and} \quad \sum_{k=1}^{s} Z_k^2 \le r_{s,t} \le \frac{1}{1+\widehat{\lambda}_s}.$$

We thus apply Chernoff-Hoeffding bounds conditioned on $X_1, \ldots, X_s$ and X_t. This gives.

$$\begin{aligned}
\text{(II)} &\le M + \mathbb{E}\left[\sum_{t=1}^{n} \sum_{s=\max\{\rho_t+1,M\}}^{t-1} \left(\left\{ |\widehat{\Delta}_{s,t} + B_{s,t} - \Delta_t| \ge \frac{\varepsilon}{4} \right\} + \left\{ \frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2} \right\} \right) \right] \\
&\le M + \sum_{t=1}^{n} \sum_{s=\max\{\rho_t+1,M\}}^{t-1} \left(\mathbb{E}\left[2\exp\left(-\frac{\varepsilon^2}{32 r_{s,t}} \right) \right] + \mathbb{P}\left(\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2} \right) \right) \\
&\le M + \sum_{t=1}^{n} \sum_{s=\max\{\rho_t+1,M\}}^{t-1} \left(\mathbb{E}\left[2\exp\left(-\frac{\varepsilon^2}{32}(1+\widehat{\lambda}_s) \right) \right] + \mathbb{P}\left(\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2} \right) \right) \\
&\le M + \sum_{t=1}^{n} \sum_{s=\max\{\rho_t+1,M\}}^{t-1} \left(2\exp\left(-\frac{\varepsilon^2}{32}\left(1 + \frac{\lambda s}{2}\right) \right) + 3\mathbb{P}\left(\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2} \right) \right) \\
&\le M + \sum_{t=1}^{n} \sum_{s=\max\{\rho_t+1,M\}}^{t-1} \left(2\exp\left(-\frac{\varepsilon^2 \lambda s}{64} \right) + 3\exp\left(-\frac{\lambda^2 s}{8} \right) \right)
\end{aligned}$$

the last inequality deriving from an application of Lemma 3 (this lemma can be applied since $\rho_t \ge \frac{16d}{\lambda^2}$ and the queried instances are independent).

We further observe that our assumption $M \geq \frac{512}{\lambda\varepsilon^2}\ln n$ makes the first exponential inside the double sum be at most $2/n^8$ (this is because $s \geq M$ in the inner sum), while our initial assumption $\rho_t = \frac{16}{\lambda^2}\max\{d, \ln t\}$ for all t makes the second exponential be at most $3/t^2$ (due to $s > \rho_t$). Thus, under the above assumptions on M and ρ_t,

$$\text{(II)} \leq M + \sum_{t=1}^{n}\sum_{s=\max\{\rho_t+1, M\}}^{t-1}\left(\frac{2}{n^8}+\frac{3}{t^2}\right) \leq M + 1 + 3\ln(n+1). \tag{10}$$

We now turn to the remaining term (III). We have, for any[3] $t \geq 23$,

$$\begin{aligned}
&\left\{\widehat{\Delta}_{N_{t-1},t}\Delta_t \leq 0,\, \widehat{\Delta}^2_{N_{t-1},t} > \frac{128\ln t}{\lambda N_{t-1}},\, N_{t-1} > \rho_t\right\}\\
&\quad\leq \left\{|\widehat{\Delta}_{N_{t-1},t} - \Delta_t| \geq |\Delta_t|,\, |\widehat{\Delta}_{N_{t-1},t}| > \sqrt{\frac{128\ln t}{\lambda N_{t-1}}},\, N_{t-1} > \rho_t\right\}\\
&\quad\leq \left\{|\widehat{\Delta}_{N_{t-1},t} - \Delta_t| \geq \frac{1}{2}\sqrt{\frac{128\ln t}{\lambda N_{t-1}}},\, N_{t-1} > \rho_t\right\}\\
&\quad\leq \sum_{s=\rho_t+1}^{t-1}\left\{|\widehat{\Delta}_{s,t} - \Delta_t| \geq \frac{1}{2}\sqrt{\frac{128\ln t}{\lambda s}}\right\}\\
&\quad\leq \sum_{s=\rho_t+1}^{t-1}\left(\left\{|\widehat{\Delta}_{s,t} + B_{s,t} - \Delta_t| \geq \frac{1}{4}\sqrt{\frac{128\ln t}{\lambda s}}\right\} + \left\{|B_{s,t}| \geq \frac{1}{4}\sqrt{\frac{128\ln t}{\lambda s}}\right\}\right).
\end{aligned}$$

Now, using (4) again, we can argue that

$$\left\{|B_{s,t}| \geq \frac{1}{4}\sqrt{\frac{128\ln t}{\lambda s}}\right\} \leq \left\{s < \frac{16}{\lambda\sqrt{\frac{128\ln t}{\lambda s}}}\right\} + \left\{\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2}\right\}. \tag{11}$$

Since we are summing over $s > \rho_t$, it is easy to verify that $s > \rho_t$ combined with $t \geq 23$ certainly make the first term in the right-hand side of (11) vanish.

We sum over $t = 23, \dots, n$ and take expectations. As before, we apply Chernoff-Hoeffding bounds and Lemma 3. This results in the following chain of inequalities

$$\begin{aligned}
\text{(III)} &\leq \sum_{t=23}^{n}\sum_{s=\rho_t+1}^{t-1}\left(\mathbb{E}\left[2\exp\left(-\frac{4\ln t}{\lambda s r_{s,t}}\right)\right] + \mathbb{P}\left(\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2}\right)\right)\\
&\leq \sum_{t=23}^{n}\sum_{s=\rho_t+1}^{t-1}\left(\mathbb{E}\left[2\exp\left(-\frac{(1+\widehat{\lambda}_s)4\ln t}{\lambda s}\right)\right] + \mathbb{P}\left(\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2}\right)\right)\\
&\leq \sum_{t=23}^{n}\sum_{s=\rho_t+1}^{t-1}\left(\mathbb{E}\left[2\exp\left(-\frac{(1+\lambda s/2)4\ln t}{\lambda s}\right)\right] + 3\mathbb{P}\left(\frac{\widehat{\lambda}_s}{s} < \frac{\lambda}{2}\right)\right)
\end{aligned}$$

[3] Note that $N_{t-1} \leq \rho_t$ certainly holds when $t \leq 22$.

$$< \sum_{t=23}^{n} \sum_{s=\rho_t+1}^{t-1} \left(2e^{-2\ln t} + 3\exp\left(-\frac{\lambda^2 s}{8}\right)\right)$$

$$\le \sum_{t=1}^{n} \sum_{s=\rho_t}^{t-1} \left(\frac{2}{t^2} + \frac{3}{t^2} \right)$$

$$\le 5\ln(n+1). \tag{12}$$

Combining together the bounds in (8), (10), and (12), and recalling the assumption on M, we obtain

$$\begin{aligned}
&\sum_{t=1}^{n} \big(\mathbb{P}(Y_t \widehat{\Delta}_{N_{t-1},t} < 0) - \mathbb{P}(Y_t \Delta_t < 0)\big) \\
&\quad \le cn\varepsilon^{1+\alpha} + (\mathrm{I}) + (\mathrm{II}) + (\mathrm{III}) \\
&\quad \le cn\varepsilon^{1+\alpha} + \rho_n + M + 1 + 8\ln(n+1) \\
&\quad \le cn\varepsilon^{1+\alpha} + \frac{16}{\lambda^2}\max\{d, \ln n\} + \frac{512}{\lambda\varepsilon^2}\ln n + 1 + 8\ln(n+1).
\end{aligned} \tag{13}$$

We can now optimize (13) with respect to ε. The optimal ε is

$$\varepsilon^* = \left(\frac{1024\ln n}{\lambda n c\alpha}\right)^{1/(2+\alpha)}.$$

Substituting back gives desired regret bound.

The expected number of labels queried by the algorithm is bounded by $(\mathrm{I}) + \mathbb{E}L_0$, being

$$L_0 = \sum_{t=1}^{n} \left\{ \widehat{\Delta}^2_{N_{t-1},t} \le \frac{128\ln t}{\lambda N_{t-1}}, N_{t-1} > \rho_t \right\}.$$

This can be treated similarly as before. Using Assumption 1 we can write

$$\begin{aligned}
\mathbb{E}L_0 &\le cn\varepsilon^{\alpha} + \sum_{t=1}^{n} \mathbb{P}\left(\widehat{\Delta}^2_{N_{t-1},t} \le \frac{128\ln t}{\lambda N_{t-1}}, N_{t-1} > \rho_t, |\Delta_t| \ge \varepsilon \right) \\
&= cn\varepsilon^{\alpha} + (\mathrm{II})
\end{aligned}$$

holding for any $\varepsilon > 0$. Putting together we end up with the upper bound $cn\varepsilon^{\alpha} + (\mathrm{I}) + (\mathrm{II})$. We exploit the bounds in (8) and (10) and again optimize for ε, yielding the claimed bound on the number of labels.

6.3 Improvements

In this section we make further comments on the technical aspects of our analysis. In particular, we would like to stress the many improvements one can achieve over the basic proof we presented in Sect. 6.2. These improvements are interesting in that they allow us to either sharpen the bounds of Theorem 2 or extend its applicability. The main reason why we decided to stick to the simpler analysis leading to Theorem 2, as currently presented, is to avoid cluttering the bounds with inessential details. In fact, the bounds resulting from the application of these improvements would be harder to read, and would somehow obscure the understanding of regret and sampling rate behavior as a function of n.

1. The first improvement one can obtain is getting rid of the linear dependence on the dimension d in Theorem 2. This dependence derives from a direct application of the concentration results contained in Shawe-Taylor et al. (2005) (also recalled in Blanchard et al. 2007). In fact, it is possible to take into account in a fairly precise manner the way the process spectrum decreases (see, e.g., Blanchard et al. 2007; Braun 2006), thereby extending the analysis to the infinite-dimensional case. This improvement should be then combined with the full-rank assumption $\lambda > 0$ on the process correlation matrix $\mathbb{E}[\mathbf{X}_1\mathbf{X}_1^\top]$ we implicitly make in Theorem 2: The way it is presented now, the proof makes substantial use of the approximation $\widehat{\lambda}_s/s \approx \lambda$, being $\widehat{\lambda}_s$ the smallest eigenvalue of the current empirical correlation matrix. If $\lambda = 0$ (because the underlying process is not full rank) then the concentration property $\widehat{\lambda}_s/s \approx \lambda$ would no longer allow us to argue that the bias and variance upper bounds in (4) and (5) converge to 0 as $1/s$; this is just because if $\widehat{\lambda}_s/s \to 0$ the quantity $\frac{1}{1+\widehat{\lambda}_s}$ cannot converge to 0 as $1/s$. One way around this (seemingly) degenerate behavior is to directly deal with the concentration properties of all process eigenvalues at once. For instance, if we consider the quadratic forms in (2) and (3) as (sharper) bounds on bias and variance, then a standard SVD would single out all eigenvalues of the empirical correlation matrix along with the corresponding eigenvector projections. One can then apply the results in Blanchard et al. (2007), Shawe-Taylor et al. (2005) on the concentration properties of projections. This approach would remove the constraint $\lambda > 0$ at the cost of making the proof far more complicated.
2. As anticipated in Sect. 5, it is not hard to adapt the presented analysis to a more time-efficient (mistake-driven) algorithm that schedules storage of the next example only if the current prediction is wrong. One can turn the algorithm in Fig. 12 into a mistake-driven algorithm by replacing line 4 with

   4′. Else if $(\mathbf{w}^\top\mathbf{x}_t)^2 \le \frac{128 \ln t}{\lambda N}$ then query y_t. If $y_t\mathbf{w}^\top\mathbf{x}_t < 0$ then schedule storage of $(\mathbf{x}_{t+1}, y_{t+1})$;

   Hence, when a small margin is detected, we query the current label and, only in the case of sign disagreement, we schedule to store the next example $(\mathbf{x}_{t+1}, y_{t+1})$. In this case y_t is queried but the example $(\mathbf{x}_t, y_t)$ *is not stored*. The analysis is very similar to the one in Sect. 6, and is based on the simple observation that the one-step regret in Lemma 2 can actually be sharpened as

$$\mathbb{P}(Y_t\widehat{\Delta}_t < 0) - \mathbb{P}(Y_t\Delta_t < 0) \le c\varepsilon^{1+\alpha} + \mathbb{P}(\widehat{\Delta}_t\Delta_t \le 0,\, y_t\widehat{\Delta}_t < 0,\, |\Delta_t| \ge \varepsilon).$$

   This allows us to propagate the additional condition $y_t\widehat{\Delta}_t < 0$ throughout the proof (in particular, to Term (II)), leading to exactly the same regret bound as the one in Theorem 2. On the other hand, the counter N would now serve as accumulator for the number of stored examples, which can be quite smaller than the number of labels queried by the mistake-driven algorithm. Since N is intimately related to the actual running time of the algorithm (see Sect. 2.1), this new proof would yield a bound on the (expected) running time, rather than on the number of queried labels.

7 Conclusions and open problems

We have investigated new selective sampling and filtering algorithms for learning noisy classifiers. The algorithms use least-squares estimates to learn a classifier, and compare the margin of this classifier to dynamically adjusted thresholds.

We reported on an extensive experimental study on a medium-size text categorization benchmark. The experimental results show that our selective sampling and filtering algorithms can effectively exploit the additional information provided by small margin examples, to the extent that our algorithms outperform known competitors. Adding a mistake-driven mechanism does not have any negative impact on the F-measure obtained by the sampling algorithm. Rather, it appears to slightly improve the overall performance while reducing running times.

On the theoretical side, we formulated a probabilistic model for the data-generating process based on a low-noise assumption combined with linear label noise (sometimes called cosine label noise). We proved a bound on the expected cumulative regret of a fully-supervised algorithm, and used this bound as a yardstick for the subsequent analysis. Our main theoretical result is a regret analysis for an adaptive sampling variant of one of the algorithms used in the experiments. We showed that both the expected cumulative regret and the expected number of labels are bounded by quantities depending on the (unknown) amount of noise in the data. In the case of hard margins, such quantities can be combined to recover the standard negative exponential behavior of the average regret in terms of the number of queried labels.

The adaptive sampling algorithm has a deferring mechanism for querying the labels. This is used to provably learn the amount of noise in the data without querying the most informative labels (i.e., the algorithm learns *how many* labels are needed rather than *which ones*). As expected, our experiments confirm that the need of deferring queries is an artifact of the analysis and is detrimental in practice. Still, on our text categorization experiments the adaptive algorithm significantly outperformed one of the competitors.

We close by mentioning two open questions.

1. Our analysis in Sect. 6.2 works under the assumption that the smallest empirical eigenvalue of the data correlation matrix is close enough to the corresponding process eigenvalue λ. As we already said, this is the reason why the algorithm in Fig. 12 undergoes an initial regime when it queries all labels. As stated in Theorem 2, the duration of this regime is independent of the amount of noise in the data (the α exponent), but it does still depend on both λ and the input dimension d. In fact, the way it is stated, Theorem 2 only holds in the finite dimensional case. As we explained in Sect. 6.3, the latter issue can be faced at the cost of shifting to more involved results on the concentration of eigenvalues in infinite-dimensional spaces. The question whether it is possible to remove the transient regime in Fig. 12, while still getting the same regret bound, remains open.
2. Analyzing the algorithms we used in the experiments does not seem to be easy, since a direct concentration analysis does not seem to be applicable. On one hand, it would be definitely interesting to evolve the current analysis so as to state results similar to the one given in Theorem 2, regardless of the fact that the samples gathered by the algorithm are conditionally dependent. However, this entails a technical problem which is not straightforward to get around. On the other hand, a possible intermediate step might be to give up some of the accuracy and/or efficiency of the algorithm of Fig. 12 in order to obtain theoretical guarantees similar to those contained in Theorem 2. For instance, would it possible to modify the SS algorithm, without hurting its performance, so as the queried labels become conditionally independent? Indeed if the query selection criterion depends only on the past instances, rather than on both past instances and past labels, then the resulting sampling algorithm does clearly rely on conditionally independent labels. A good starting point to tackle this problem might be the recent paper (Strehl and Littman 2008).

Acknowledgements We thank all reviewers for their helpful comments. Thanks to Alex Conconi for his contributions to earlier versions of this work. Thanks also to Tong Zhang and Nina Balcan for useful comments.

Appendix

This appendix contains the proof of Lemma 4. Before proceeding with the proof, we need a few definitions and a technical lemma.

An integer-valued random variable T is a stopping time w.r.t. a random process $Z_1, Z_2, \ldots$ if, for each $k \geq 1$, $\{T = k\}$ belongs to the σ-algebra $\sigma(Z_1, \ldots, Z_k)$ generated by the random variables $Z_1, \ldots, Z_k$. A stopping time T is finite if $\mathbb{P}(T = \infty) = 0$. The next result proves an elementary property of stopping times.

Lemma 5 *If T is a finite stopping time w.r.t. the i.i.d. random variables $Z_1, Z_2, \ldots$, then Z_{T+1} is independent of $Z_1, \ldots, Z_T$ and is distributed as Z_1.*

Proof Choose any $A \in \sigma(Z_1, \ldots, Z_T)$ and choose any subset B of the range of Z_{T+1} such that $\{Z_{T+1} \in B\}$ is measurable. We have

$$\begin{aligned}\mathbb{P}(A \cap \{Z_{T+1} \in B\}) &= \sum_{j=1}^{\infty} \mathbb{P}\big(A \cap \{T = j\} \cap \{Z_{j+1} \in B\}\big) \\ &= \sum_{j=1}^{\infty} \mathbb{P}(A \cap \{T = j\})\, \mathbb{P}(Z_{j+1} \in B) \\ &\qquad \text{(since } A \cap \{T = j\} \text{ is } \sigma(Z_1, \ldots, Z_j)\text{-measurable)} \\ &= \sum_{j=1}^{\infty} \mathbb{P}(A \cap \{T = j\})\, \mathbb{P}(Z_1 \in B) = \mathbb{P}(A)\mathbb{P}(Z_1 \in B). \end{aligned} \tag{14}$$

Hence, taking $A = \Omega$, we get $\mathbb{P}(Z_{T+1} \in B) = \mathbb{P}(Z_1 \in B)$, showing that Z_{T+1} is distributed as Z_1. Consequently, from (14) we get that $\mathbb{P}(A \cap \{Z_{T+1} \in B\}) = \mathbb{P}(A)\mathbb{P}(Z_{T+1} \in B)$, as desired. □

We are now ready to prove Lemma 4 in the main text.

The random variables $T_1, T_2, \ldots$ are finite stopping times with respect to the i.i.d. process $Z_1, Z_2, \ldots$ according to which the examples are generated. In fact, $\{T_i = k\}$ is completely determined by the values taken by $Z_1, \ldots, Z_k$. Furthermore, $\mathbb{P}(T_i = \infty) = 0$ since, for each $i \geq 1$ and for t large enough, $\widehat{\Delta}^2_{N_{t-1},t} \leq \frac{128 \ln t}{\lambda N_{t-1}}$, where $N_{t-1} = i - 1$, will hold.

Pick any $i \geq 1$. By Lemma 5, Z_{T_i+1} is independent of $Z_1, \ldots, Z_{T_i}$ and distributed as Z_1. Since the random variables $Z_{T_1+1}, \ldots, Z_{T_{i-1}+1}$ are $\sigma(Z_1, \ldots, Z_{T_i})$-measurable (this is guaranteed by the fact that $T_{i-1} < T_i$ always holds), we get that Z_{T_i+1} is also independent of $Z_{T_1+1}, \ldots, Z_{T_{i-1}+1}$. As i was chosen arbitrarily, we get that $Z_{T_1+1}, Z_{T_2+1}, \ldots$ are independent random variables distributed as Z_1.

References

Angluin, D. (2004). Queries revisited. *Theoretical Computer Science*, *313*(2), 175–194.

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

Balcan, M. F., Beygelzimer, A., & Langford, J. (2006). Agnostic active learning. In *Proceedings of the 23rd international conference on machine learning (ICML)* (pp. 65–72). Omnipress.

Balcan, M. F., Broder, A., & Zhang, T. (2007). Margin-based active learning. In *Proceedings of the 20th annual conference on learning theory (COLT)* (pp. 35–50). Berlin: Springer.

Bartlett, P. L., Jordan, M. I., & McAuliffe, J. D. (2006). Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, *101*(473), 138–156.

Blanchard, G., Bousquet, O., & Zwald, L. (2007). Statistical properties of kernel principal component analysis. *Machine Learning*, *66*(2–3), 259–294.

Boucheron, S., Bousquet, O., & Lugosi, G. (2005). Theory of classification: a survey of some recent advances. *ESAIM Probability and Statistics*, *9*, 323–375.

Braun, M. L. (2006). Accurate error bounds for the eigenvalues of the kernel matrix. *Journal of Machine Learning Research*, *7*, 2303–2328.

Campbell, C., Cristianini, N., & Smola, A. (2000). Query learning with large margin classifiers. In *Proceedings of the 17th international conference on machine learning (ICML)* (pp. 111–118). San Mateo: Morgan Kaufmann.

Castro, R., & Nowak, R. D. (2008). Minimax bounds for active learning. *IEEE Transactions on Information Theory*, *54*(5), 2339–2353.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge: Cambridge University Press.

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

Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006a). Incremental algorithms for hierarchical classification. *Journal of Machine Learning Research*, *7*, 31–54.

Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006b). Worst-case analysis of selective sampling for linear-threshold algorithms. *Journal of Machine Learning Research*, *7*, 1205–1230.

Cohn, R., Atlas, L., & Ladner, R. (1990). Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems (NIPS)*, 1989. New York: MIT Press.

Cohn, R., Atlas, L., & Ladner, R. (1994). Improving generalization with active learning. *Machine Learning*, *15*(2), 201–221.

Dasgupta, S., Kalai, A. T., & Monteleoni, C. (2005). Analysis of Perceptron-based active learning. In *Proceedings of the 18th conference on learning theory (COLT 2005)* (pp. 249–263). Berlin: Springer.

Dasgupta, S., Hsu, D., & Monteleoni, C. (2008). A general agnostic active learning algorithm. In *Advances in neural information processing systems (NIPS)* (Vol. 21, pp. 353–360). New York: MIT Press.

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

Gilad-Bachrach, R., Navot, A., & Tishby, N. (2005). Query by committee made real. In *Advances in neural information processing systems (NIPS)* (Vol. 19). New York: MIT Press.

Hanneke, S. (2007). A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on machine learning (ICML)* (pp. 353–360). Omnipress.

Hanneke, S. (2009). Adaptive rates of convergence in active learning. In *Proceedings of the 22nd conference on learning theory (COLT 2009)*. Omnipress.

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

Joachims, T. (1999). Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, & A. Smola (Eds.), *Advances in kernel methods: support vector learning*. New York: MIT Press.

Kääriäinen, M. (2006). Active learning in the non-realizable case. In *Proceedings of the 17th international conference on algorithmic learning theory (ALT 2006)* (pp. 63–77). Berlin: Springer.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM-SIGIR conference on research and development in information retrieval* (pp. 3–12). Berlin: Springer.

Monteleoni, C., & Kääriäinen, M. (2007). Practical online active learning for classification. In *IEEE computer society conference on computer vision and pattern recognition* (pp. 249–263). New York: IEEE Computer Society.

Muslea, I., Minton, S., & Knoblock, C. A. (2000). Selective sampling with redundant views. In *Proceedings of the national conference on artificial intelligence (AAAI 2000)* (pp. 621–626). New York: MIT Press.

NIST (2004). trec.nist.gov/data/reuters/reuters.html.

Sculley, D. (2008). *Advances in online learning-based spam filtering*. PhD Thesis in Computer Science, Tufts University. August.

Shawe-Taylor, J., Williams, C. K. I., Cristianini, N., & Kandola, J. (2005). On the eigenspectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*, *51*(7), 2510–2522.

Steinwart, I., & Scovel, C. (2007). Fast rates for support vector machines using Gaussian kernels. *Annals of Statistics*, *35*, 575–560.

Strehl, A. L., & Littman, M. L. (2008). Online linear regression and its application to model-based reinforcement learning. In *Advances in neural information processing systems (NIPS)* (Vol. 21, pp. 1417–1424). New York: MIT Press.

Tong, S., & Koller, D. (2000). Support vector machine active learning with applications to text classification. In *Proceedings of the 17th international conference on machine learning (ICML)* (pp. 999–1006). San Mateo: Morgan Kaufmann.

Tsybakov, A. (2004). Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, *32*(1), 135–166.

Vovk, V. (2001). Competitive on-line statistics. *International Statistical Review*, *69*, 213–248.

Ying, Y., & Zhou, D. X. (2006). Online regularized classification algorithms. *IEEE Transactions on Information Theory*, *52*, 4775–4788.