Robust Selective Sampling from Single and Multiple Teachers

Ofer Dekel
Microsoft Research
oferd@microsoft.com

Claudio Gentile
DICOM, Università dell'Insubria
claudio.gentile@uninsubria.it

Karthik Sridharan
TTI-Chicago
karthik@tti-c.org

1 Introduction

We report on very recent (and unpublished) research activity in the context of *robust* selective sampling. A *selective sampling* algorithm [7, 10] is an online learning algorithm that at each time step t receives an instance $\mathbf{x}_t \in \mathbb{R}^d$, makes a prediction about the label y_t associated with $\mathbf{x}_t$, and decides whether or not to actually observe y_t. If the learner does observe y_t (we say that the algorithm has issued a *query* on the label at time t), then the label value can be used to improve future predictions. If the label is predicted but not queried, the learner never knows whether his prediction was correct. Hence, only queried labels are observed while all others remain unknown. We say that a selective sampling algorithm is *robust* if it works even when the instance sequence $\mathbf{x}_1, \mathbf{x}_2, ...$ is generated by an *adaptive adversary*. Robustness thereby implies a high level of adaptation to the learning environment.

Following [4, 6] (but see also [13]), we consider robust selective sampling algorithms where the above adversarial assumptions are combined with a simple linear noise model for the conditional label distribution. The general aim of this line of research is to provide algorithms of practical use having also provable performance guarantees. In fact, a selective sampling analysis involves both statistical considerations (e.g., the "information-theoretic" value of a label) and basic computational aspects of learning. For instance, in a typical practical scenario the number of observed labels does directly affect running time and memory requirements (this is certainly the case for our algorithms, see Sections 2 and 3). As a by-product of our adversarial analysis, we also obtain (tight) theoretical guarantees in the case when the instances $\mathbf{x}_t$ are generated i.i.d. according to a fixed and unknown distribution on the instance space.

Selective sampling is an online learning framework lying between *passive* learning (where the algorithm has no control over the learning sequence) and *active* learning (where the learning algorithm is allowed to select the instances $\mathbf{x}_t$). The literature on active learning is vast, and we can hardly do it justice here. Recent papers on this subject include [1, 2, 3, 8, 9, 11, 12]. All these papers consider the case where instances are drawn i.i.d. from a fixed distribution (either known or unknown). The results we present here are more in line with the worst-case analyses in [5, 13, 6], which discuss variants of Recursive Least Squares (RLS) algorithms that operate on arbitrary instance sequences. The work [5] is completely worst case: the authors make no assumptions whatsoever on the mechanism generating instances and labels; however, they are unable prove bounds on the label query rate. The setups in [13, 6] are closest to ours. In [13] the authors approximate the Bayes margin to within a given accuracy ϵ, and assume adversarial sequences of instances and the same linear conditional label distribution as we do here. However, their analysis yields the significantly worse bound $O(d^3/\epsilon^4)$ on the number of queries and, unlike ours, seems to work only in the finite dimensional ($d < \infty$) case. As far as comparison to [6] is concerned, we are able to work against adaptively adversarial strategies (the authors in [6] only consider *oblivious* adversaries) and moreover we obtain sharper results on both the cumulative regret and the number of queried labels. These results essentially solve questions left open in, e.g., [4, 6], and are as tight as those obtained in the full information setting (i.e., when all labels are observed).

Selective sampling, as presented above, is a special case of what we call the *multiple teacher* setting where, instead of a single source of labels, the learner has many to choose from. This was actually the initial inspiration for the research we are describing here. The multiple teacher setting is motivated by an Internet search company trying to improve the performance of its search engine by receiving human feedback. Large Internet search companies typically hire human editors (the "teachers") to manually label instances $\mathbf{x}_t$. These labels can then be pooled and used to train a classifier on $\mathbf{x}_t$. Each teacher may have a unique area of expertise, and not all teachers may be qualified to provide a label for a given $\mathbf{x}_t$. Ideally, the algorithm should identify the competent teachers for the current $\mathbf{x}_t$ (without any prior information) and query labels only from them. By suitably generalizing the robust selective sampling setting and algorithms, we obtain a regret analysis for the multiple teacher setting which, to the best of our knowledge, is novel. In order to show the practical utility of our algorithms, we are currently running extensive experiments on medium/large size datasets. The results of these experiments are not given here, but we expect to be able to report them at the workshop.

2 The single teacher case: setting, algorithm and results

We consider an online binary classification problem where at each time step $t = 1, 2, \ldots$ the learner receives input $\mathbf{x}_t \in \mathbb{R}^d$ and predicts $\hat{y}_t \in \{-1, +1\}$ for the associated unknown label $y_t \in \{-1, +1\}$. The true label y_t is revealed to the learner only after the learner has made its prediction and has issued a query for the label. The goal of the selective sampling algorithm is to learn to predict the labels with as few queries as possible. We assume that for any given $\mathbf{x}_t$, the corresponding label y_t is generated stochastically using the law $P(y_t = 1 \,|\mathbf{x}_t\,) = (1 + \mathbf{u}^\top \mathbf{x}_t)/2$ where $\mathbf{u} \in \mathbb{R}^d$ is some vector such that $\|\mathbf{u}\| \leq 1$. We also assume that for each t, $\|\mathbf{x}_t\| \leq 1$. Note that the current $\mathbf{x}_t$ could be chosen adversarially depending on previous $\mathbf{x}$'s and y's. The learner uses hyperplanes for prediction during each round. That is, on round t the learner predicts $\hat{y}_t = \mathrm{sign}(\hat{\Delta}_t)$, where $\hat{\Delta}_t = {\mathbf{w}_{t-1}}^\top \mathbf{x}_t$. Let P_t denote the conditional probability $P_t(\cdot) := \mathbb{P}(\cdot|\, \mathbf{x}_1, \ldots, \mathbf{x}_t, y_1, ..., y_{t-1})$. We are interested in simultaneously bounding with high probability the cumulative regret R_T and the number of queried labels N_T, defined as

$$R_T \;=\; \textstyle\sum_{t=1}^T \Big(P_t(y_t \hat{\Delta}_t < 0) - P_t(y_t \Delta_t < 0) \Big), \qquad\qquad N_T = \textstyle\sum_{t=1}^T Z_t,$$

where Z_t is 1 if we query at round t, and 0 otherwise. The algorithm we analyze is a margin-based selective sampling procedure with query condition $Z_t := \mathbb{1}\Big\{ \hat{\Delta}_t^2 \leq \theta_t^2 \Big\}$, where θ_t is an adaptive (and data-dependent) threshold which will be specified below. The algorithm ("Selective Sampler") keeps a weight vector $\mathbf{w}$ and a correlation data matrix A.

Algorithm 1 Selective Sampler

$\mathbf{w}_0 \leftarrow \mathbf{0}$, $A_0 \leftarrow I$
for $t = 1$ to T **do**
  Receieve $\mathbf{x}_t$, calculate $\hat{\Delta}_t = {\mathbf{w}_{t-1}}^\top \mathbf{x}_t$, predict $\hat{y}_t = \mathrm{sgn}(\hat{\Delta}_t) \in \{-1, +1\}$
  Set $\theta_t^2 = \mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t \Big(1 + 4\, g(t-1) + 36 \log(t/\delta) \Big)$, compute $Z_t = \mathbb{1}\Big\{ \hat{\Delta}_t^2 \leq \theta_t^2 \Big\} \in \{0, 1\}$
  if $Z_t = 1$ **then**
    Query y_t

$$\mathbf{w}'_{t-1} = \begin{cases} \mathbf{w}_{t-1} - \left(\frac{|{\mathbf{w}_{t-1}}^\top \mathbf{x}_t| - 1}{\mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t} \right) A_{t-1}^{-1} \mathbf{x}_t & \text{if } |\hat{\Delta}_t| > 1 \\ \mathbf{w}_{t-1} & \text{otherwise} \end{cases}$$

    $A_t = A_{t-1} + \mathbf{x}_t \mathbf{x}_t^\top, \quad \mathbf{w}_t = A_t^{-1}(A_{t-1} \mathbf{w}'_{t-1} + y_t \mathbf{x}_t)$
  else
    $A_t = A_{t-1},\ \mathbf{w}_t = \mathbf{w}_{t-1}$
  end if
end for

The most important aspect is the magnitude of the threshold θ_t in the query condition "$\hat{\Delta}_t^2 \leq \theta_t^2$". In the pseudocode, $g(t) = \sum_{i=1}^t Z_i r_i$, where r_t is the quadratic form $r_t := \mathbf{x}_t^\top A_t^{-1} \mathbf{x}_t$. Hence $g(t-1)$ is the sum of the r_i up to time $t-1$ on rounds when we did actually issue a query. One can show that $g(t)$ grows mildly (logarithmically) with N_t (number of queries so far). On the other hand, the other factor $\mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t$ in θ_t^2 tends to shrink as $1/N_t$, as long as the current $\mathbf{x}_t$ lies along the direction of previously stored instances. As a result, θ_t^2 is either small (roughly $(\log t)/N_t$) or independent of the interaction between the current instance $\mathbf{x}_t$ and the previous instances and the number of labels observed so far.

The above algorithm has a quadratic running time per round, where quadratic is $O(d^2)$ if it is run in primal form, and $O(N_t^2)$ if it is run in dual form (i.e., in a RKHS). Notice that the algorithm updates only when $Z_t = 1$, hence the number of labels N_t here corresponds to the number of "support vectors" (i.e., to the sparsity of the current hypothesis). The bounds we give depend on how many of the inputs $\mathbf{x}_t$ are close to being complete noise. To capture this dependence, we define for any $\epsilon > 0 \quad T_\epsilon = \sum_{t=1}^T \mathbb{1}\{|\Delta_t| \leq \epsilon\}$ Note that if $|\Delta_t| \leq \epsilon$ then $P_t(y_t = 1) \in [1/2 + \epsilon, 1/2 - \epsilon]$. In short, T_ϵ is a "hardness" parameter which is fully controlled by the adversary.[1] In the theorem below we emphasize both the data-dependent and the time-dependent aspects of our bounds.

Theorem 1. *Let Algorithm 1 be run with confidence parameter $\delta > 0$. Then w.p. $\geq 1 - \delta$ it holds that for all $T > 0$*

$$R_T \;\leq\; \inf_{\epsilon > 0} \Big\{ \epsilon\, T_\epsilon + \frac{2 + 8 \log|A_T| + 144 \log(T/\delta)}{\epsilon} \Big\} = \inf_{\epsilon > 0} \Big\{ \epsilon\, T_\epsilon + O\Big(\frac{d\, \log T + \log(T/\delta)}{\epsilon} \Big) \Big\}$$

$$N_T \;\leq\; \inf_{\epsilon > 0} \Big\{ T_\epsilon + O\Big(\frac{\log|A_T| \log(T/\delta) + \log^2|A_T|}{\epsilon^2} \Big) \Big\} = \inf_{\epsilon > 0} \Big\{ T_\epsilon + O\Big(\frac{d^2 \log^2(T/\delta)}{\epsilon^2} \Big) \Big\},$$

[1] This need not be the case when data are i.i.d. In fact, the bound in Theorem 1 can easily be specialized to the i.i.d. setting, just by controlling the contribution due to T_ϵ (details omitted due to space limitations).

where $|A_T|$ is the determinant of A_T.
The bound on N_T is tight w.r.t. ϵ (see the lower bound in [6]) but need not be tight w.r.t. the dimension d.

3 The multiple teacher case: setting and sketch of results

The problem is still online binary classification, where at each time step $t = 1, 2, \ldots$ the learner receives input $\mathbf{x}_t \in \mathbb{R}^d$, and is required to predict the label corresponding to $\mathbf{x}_t$. But now the learner can choose to query one or more of K teachers, with a fixed price per each query. If teacher j is queried at time t, it provides the binary label $y_{j,t}$ to the algorithm. We assume that for any $j \in [K]$, the j-th teacher stochastically generates label $y_{j,t}$ according to the law $P(y_{j,t} = 1 \,|\mathbf{x}_t\,) = (1 + \mathbf{u}_j{}^\top \mathbf{x}_t)/2$ where $\mathbf{u}_j \in \mathbb{R}^d$ is some vector such that $\|\mathbf{u}_j\| \leq 1$. We also assume that for each t, $\|\mathbf{x}_t\| \leq 1$. Again, the $\mathbf{x}_t$'s could be chosen adversarially depending on previous $\mathbf{x}$'s and y_j's.

Since we have multiple labels per input instance, none of which is considered to be the ground truth, it is no longer clear how to evaluate our performance. To this end, we use the intuition that each teacher has his/her own field of expertise. The expertise of the teacher for a question is captured by the confidence of the teacher's answer for that question. We would like the learner to compete at each round against the cumulative opinion of the most confident teachers (who are experts for that question) for that round. To formalize this intuition, we let P_t denote the conditional probability $P_t(\cdot) := \mathbb{P}(\,\cdot|\,\mathbf{x}_1, y_{1,1} \ldots, y_{K,1}, \mathbf{x}_2, y_{1,2} \ldots, y_{K,2}, \ldots \mathbf{x}_{t-1}, y_{1,t-1}, \ldots y_{K,t-1}, \mathbf{x}_t)$. Also for any $N \subset [K]$, set $\Delta_{N,t} = \frac{1}{|N|} \sum_{i \in N} \Delta_{i,t}$. Let $j_t^* = \text{argmax}_j |\Delta_{j,t}|$ and define $C_t = \{i : |\Delta_{i,t}| \geq |\Delta_{j_t^*,t}| - \tau\}$ as the set of confident teachers at time t, where $\tau \geq 0$ is a margin threshold parameter. We are interested in bounding with high probability the cumulative regret

$$R_T^\tau \;:=\; \sum_{t=1}^T \Big(P_t(y_t \hat{\Delta}_t < 0) - P_t(y_t \Delta_{C_t,t} < 0) \Big) \;,$$

being $\text{SGN}(\hat{\Delta}_t)$ the prediction of the learner, and y_t stochastically selected using the law $P(y_t = 1|\mathbf{x}_t) = (1 + \Delta_{C_t,t})/2$. Notice that this is equivalent to generate y_t by first picking $j \in C_t$ uniformly at random and then using $y_t = y_{j,t}$. This notion of regret captures the intuition that the appropriate label we would like to predict is the one generated by the *average* opinion over the *confident* teachers.

Our algorithm keeps a weight vector $\mathbf{w}_j$ per teacher and learns on the fly the corresponding $\mathbf{u}_j$. The prediction rule queries only the teachers that are deemed competent on $\mathbf{x}_t$. A proxy of the set C_t is maintained which is based on estimated margins. A theorem similar[2] to Theorem 1 above holds in the multiple teacher case as well, though the statement is somewhat complicated by the interaction among teachers. The statement of this theorem has to quantify to what extent the average opinions of the confident teachers is of low confidence. This is the scenario where the most confident teachers are themselves either not very confident or conflict with each other (details omitted due to space limitations).

References

[1] Balcan, M., Beygelzimer, A., & Langford, J. (2006). Agnostic active learning. In *23rd ICML* (pp. 6572).

[2] Balcan, M., Broder, A., & Zhang, T. (2007). Margin-based active learning. In *20th COLT* (pp. 3550).

[3] R. Castro and R.D. Nowak. Minimax bounds for active learning. *IEEE Trans. IT*, 54(5):2339–2353, 2008.

[4] Cavallanti, G., Cesa-Bianchi, N., & Gentile, C. (2009). Linear classification and selective sampling under low noise conditions. In *NIPS 21*.

[5] Cesa-Bianchi, N., Gentile, C., & Zaniboni, L. (2006). Worst-case analysis of selective sampling for linear classification. *JMLR*, *7*, 1025–1230.

[6] Cesa-Bianchi, N., Gentile, C., & Orabona, F. (2009). Robust Bounds for Classification via Selective Sampling In Proc. *26th ICML*.

[7] Cohn, R., Atlas, L., & Ladner, R. (1990). Training connectionist networks with queries and selective sampling. In *NIPS 2*.

[8] S. Dasgupta, D. Hsu, and C. Monteleoni (2008). A general agnostic active learning algorithm. In *NIPS 21*.

[9] S. Dasgupta, A. T. Kalai, & C. Monteleoni (2005). Analysis of perceptron-based active learning. In *18th COLT*.

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

[11] S. Hanneke (2007). A bound on the label complexity of agnostic active learning. In *24th ICML*, pages 353–360.

[12] S. Hanneke (2009). Adaptive rates of convergence in active learning In *22nd COLT*.

[13] Strehl, A., and Littman, M. (2008). Online linear regression and its application to model-based reinforcement learning. In *NIPS 20*.

[2] One might wonder whether an adaptively adversarial model of learning might somehow be overkill for the motivating Internet search problem. Again, i.i.d. results can be derived as direct corollaries of our adversarial model. As a matter of fact, the way our algorithm works makes the adaptively adversarial analysis *strictly necessary* even for deriving such i.i.d. results.