From: PROCEEDINGS OF THE FOURTH ANNUAL WORKSHOP ON COMPUTATIONAL LEARNING THEORY (COLT'91), M.K. Warmuth, L.G. Valiant, eds.; Morgan Kaufmann Publishers (San Mateo, 1991), 167 - 175

On-line Learning with an Oblivious Environment and the Power of Randomization

Wolfgang Maass*
IIG
Technische Universität Graz
Klosterwiesgasse 32
A-8010 Graz, Austria
maass@iicm.tu-graz.ac.at

Abstract

A new model for on-line learning is introduced. In this model the environment is assumed to be oblivious to the learner: it supplies an arbitrary (not necessarily random) sequence of examples for the target concept which does not depend on the sequence of hypotheses of the learner. This model provides a framework for the design and analysis of on-line learning algorithms which acquire information not just from counterexamples, but also from examples which support their current hypotheses. It is shown that for various concept classes C an arbitrary target concept from C can be learned in this model by a randomized learning algorithm (which uses only hypotheses from C) with substantially fewer prediction errors than in the previously considered models for on-line learning. In particular any target-setting of weights and thresholds in a feed forward neural net can be learned by a randomized learning algorithm in this model with an expected number of prediction errors that is polynomial in the number of units of the neural net.

We also show that these positive results for randomized learning algorithms remain valid if the environment is only weakly oblivious, i.e. if the environment can let its choice of examples depend on earlier reactions of the learner, but is not able to predict future moves of the learner.

1 INTRODUCTION

Before we describe our model for learning with an oblivious environment, we first review the common model for on-line learning. In this model, the environment plays the role of an adversary ([A2], [L1], [LW], [GRS], [MT1]). One assumes that the learner proposes "hypotheses" H from a fixed "concept class" $C \subseteq 2^X$ over a finite domain X. The goal of the learner is to "learn" an unknown "target concept" $C_T \in C$ that has been fixed by the "environment". Whenever the learner proposes some hypothesis H with $H \neq C_T$, the environment responds with some "counterexample" $x \in H \triangle C_T := (C_T - H) \cup (H - C_T)$. x is called a "positive counterexample" if $x \in C_T - H$, and x is called a "negative counterexample" if $x \in H - C_T$. A learning algorithm for C is any algorithm A that produces new hypotheses

$$H_{i+1}^A := A(x_1, \ldots, x_i;\ H_1^A, \ldots, H_i^A)$$

in dependence of counterexamples $x_j \in H_j^A \triangle C_T$ for the preceding hypotheses H_j^A. One also refers to these hypotheses as "equivalence queries" [A1].

The "learning complexity" $\mathrm{LC}(A)$ of such a learning algorithm A is defined by

$$\mathrm{LC}(A) := \max \{\, i \in \mathbf{N} \mid \text{there is some } C_T \in C \text{ and some choice of counterexamples } x_j \in H_j^A \triangle C_T \text{ for } j = 1, \ldots, i-1 \text{ such that } H_i^A \neq C_T \,\}.$$

The "learning complexity" $\mathrm{LC}(C)$ of a concept class C is defined by

$$\mathrm{LC}(C) := \min \{\, \mathrm{LC}(A) \mid A \text{ is a learning algorithm for } C \text{ which only uses hypotheses from } C \,\}.$$

One sets

$$\text{LC-ARB}(C) := \min \{\, \mathrm{LC}(A) \mid A \text{ is a learning algorithm for } C \text{ which uses arbitrary subsets of the domain } X \text{ as hypotheses} \,\}.$$

In the preceding definition of $\mathrm{LC}(A)$ one considers the maximal number of errors of A for any choice of the target concept $C_T \in C$ and any choice of counterexamples. Furthermore the environment is allowed to choose its counterexamples in full knowledge of

*Parts of the research for this paper were carried out at the University of Illinois at Chicago, the Department of Computer Science of the Universität des Saarlandes in Saarbruecken (Germany) and the International Computer Science Institute in Berkeley.

all moves (including future moves) of the learner A. Hence in this model the environment is not only adaptive, but even "predictive". Furthermore the environment is seen as a malicious adversary since it is allowed to choose the "least informative" counterexample to the hypothesis. It is difficult to find learning situations (apart from cryptography) where this pessimistic view of the environment as an adaptive and malicious adversary (which even can predict future moves of the learner) is justified.

Frequently one can assume that the environment provides examples for the target concept according to some fixed time-invariant distribution. In this case Valiant's PAC-learning model [V] provides an adequate framework. However there are various learning situations where the environment is oblivious to the learner, but does not behave like a time-invariant stochastic process. For example visual inputs to an artificial or natural learning system are subject to systematic changes that result from differences in lighting (e.g. changes between day and night) or from movements of the system. In other learning systems (for example systems for speech recognition, concept learning, or language acquisition) the examples that are given to the learner are subject to systematic changes that result from different speakers and different contexts.

So far, the only theoretical framework for the investigation of on-line learning from arbitrary (non-stochastic) sequences S of examples is provided by the previously described LC-model. Formally, the LC-model does not quite fit into this learning situation, since it expects to receive only counterexamples from the environment. One has solved this problem by assuming that the learner processes the examples in the sequence S in an on-line manner, and that he throws away any example in S that supports (i.e. is consistent with) his current hypothesis H ([L1]).

This approach for modeling on-line learning from a non-random sequence S is unsatisfactory for two reasons. First, it is clear that in various practical learning situations not only counterexamples but also supporting examples (i.e. positive reinforcements) are helpful for the learner. Hence an adequate theoretical framework should support a quantitative analysis of learning algorithms that learn from both types of examples. Secondly, on the purely theoretical level, it has turned out that most results about learning in the LC-model are negative. For example it has been shown that a "huge" number of counterexamples is needed for learning finite automata, DNF-formulas [A2], rectangles in general position, or intersections of halfplanes [MT2]. Hence there is no theoretical justification for ignoring supporting examples that are available to the on-line learner.

In the following we define a model for on-line learning from arbitrary example sequences that allows us to investigate randomized on-line learning algorithms that exploit both counterexamples and supporting examples. We assume that the environment provides an arbitrary sequence S of positive and negative examples for the target concept independently of the learning algorithm that is used by the learner. Thus one may just as well assume that the environment has determined both this sequence S of examples and the target concept before the learning process begins. The learner (more precisely: the learning algorithm) processes these examples in an on-line fashion. Analogously as in the classical learning models for perceptrons ([R], [MP]) and neural networks ([N], [RM]) the learner is allowed to alter his hypothesis at each step where the current example provides a counterexample to his current hypothesis (one calls such an event a "prediction error", or simply an "error"). We refer to the other examples $\langle x, b\rangle$ in S (where the given classification $b = C_T(x)$ agrees with the "prediction" $H(x)$ of the current hypothesis H) as supporting examples. In the learning model defined below we assume that the learner does not change his hypothesis when he encounters a supporting example, but he may store any supporting example that he receives (as well as any counterexample) for later use.

It is obvious that for the case of a deterministic learning algorithm A it makes no difference whether the environment is adaptive or oblivious: the oblivious environment can predict all later reactions of a deterministic algorithm A, hence it can write down already at the beginning of the learning process a sequence S which consists of the "optimal" moves of an adaptive adversary in a learning process with this learning algorithm A. Therefore we consider in the following definition immediately the case of randomized learning algorithms.

Whenever we define the learning complexity for a model where randomized learning algorithms are permitted, we will write "RLC" instead of "LC". In order to distinguish the new model with an oblivious environment from the LC-model we use for the new model the suffix "OBL" (e.g. RLC-OBL($\mathcal{C}$)). We will always denote the domain of a concept class $\mathcal{C}$ by X, and we write $X^{\leq\infty}$ for the set of all finite and infinite sequences of elements of X. For any $C \in \mathcal{C}$ and $S = \langle x_1, x_2, \ldots\rangle \in X^{\leq\infty}$ we write S^C for the associated sequence $\langle \langle x_1, C(x_1)\rangle, \langle x_2, C(x_2)\rangle, \ldots\rangle$ of labeled examples for C (each concept C is identified with its characteristic function $\chi_C : X \to \{0,1\}$).

A deterministic learning algorithm A for a concept class $\mathcal{C}$ processes an arbitrary labeled sequence S^{C_T} (for some target concept $C_T \in \mathcal{C}$ and some $S \in X^{\leq\infty}$) as indicated above. In particular A computes a new hypothesis $H' \in \mathcal{C}$ (as a function of $\langle \langle x_1, C_T(x_1)\rangle, \ldots, \langle x_{t-1}, C_T(x_{t-1})\rangle \rangle$) at each step t where A makes a prediction error (i.e. $H(x_t) \neq C_T(x_t)$ for the current hypothesis $H \in \mathcal{C}$ of A). We write

$\text{Errors}(A, C_T, S)$ for the total number of prediction errors of A for the labeled sequence S^{C_T}.

Intuitively a randomized learning algorithm B will carry out some coin tosses (which are not visible to the environment) whenever he chooses another hypothesis. Formally, one may just as well assume that the learner carries out all these coin tosses at the beginning of the learning process. Hence one may assume that a randomized learning algorithm B for a concept class $\mathcal{C}$ is a probability distribution $Q_B(A)$ over deterministic learning algorithms A for $\mathcal{C}$. We set $\text{Errors}(B, C_T, S) := E_{A \in Q_B}(\text{Errors}(A, C_T, S))$,

$$\text{RLC-OBL}(B) := \max\{\, \text{Errors}(B, C_T, S) \mid C_T \in \mathcal{C}, S \in X^{\leq\infty} \,\},$$

$$\text{RLC-OBL}(\mathcal{C}) := \min\{\, \text{RLC-OBL}(B) \mid B \text{ is a randomized learning algorithm for } \mathcal{C} \text{ which only uses hypotheses from } \mathcal{C} \,\}.$$

It turns out that the positive results of this paper about the power of randomized on-line learning algorithms remain valid under a much weaker assumption, where the learner is only assumed to be "weakly oblivious". We call an environment weakly oblivious if it is allowed to take into consideration all reactions of the learner for the preceding examples $x_1, \ldots, x_{i-1}$ before it selects x_i (but it has no other information about the current hypothesis H of the learner, see Remark 2.2). This model with a weakly oblivious adversary is adequate even for various learning situations in cryptography, where the environment has to be viewed as an adaptive and malicious adversary.

In this paper we will focus on the expected number of prediction errors that can be achieved by an optimal learning algorithm, and we will ignore questions of computational efficiency. In section 2 we compare the error bounds that can be achieved in the RLC-OBL model with those of other models for on-line learning. In particular we introduce (in the proof of Theorem 2.1) the randomized learning algorithm GUESSING, which learns arbitrary target concepts from a concept class $\mathcal{C}$ with an expected number of $\leq \ln|\mathcal{C}|$ errors.

In section 3 we investigate the power of randomization for learning from an environment that is totally non-oblivious.

In section 4 we compare the performance of the here considered prediction algorithms with that of previously known algorithms. Furthermore, we prove in Theorem 4.1 a result that suggests that the RLC-OBL-model may be viewed as an interpolation between the LC-model and the PAC-model.

2 ERROR-BOUNDS FOR RANDOMIZED ON-LINE LEARNING ALGORITHMS WITH AN OBLIVIOUS ENVIRONMENT

Theorem 2.1. For any finite concept class $\mathcal{C}$

$$\text{RLC-OBL}(\mathcal{C}) \leq \ln|\mathcal{C}|.$$

Proof. Let $\text{GUESSING}_\mathcal{C}$ be the following randomized learning algorithm for $\mathcal{C}$: after any prediction error pick as next hypothesis uniformly random any concept $C \in \mathcal{C}$ which is consistent with all preceding examples (i.e. all previously seen supporting examples and counterexamples).

The power of this simple learning algorithm is demonstrated by the following observation: Consider a learning process with $\text{GUESSING}_\mathcal{C}$ for some arbitrary $C_T \in \mathcal{C}$, $S = \langle x_1, x_2, \ldots\rangle \in X^{\leq\infty}$. Assume that $\text{GUESSING}_\mathcal{C}$ makes a prediction error for the t-th element x_t of S. Define

$$\mathcal{C}_t := \{C \in \mathcal{C} \mid C(x_i) = C_T(x_i) \text{ for } i = 1, \ldots, t\}.$$

Consider any linear order $\prec$ on $\mathcal{C}_t$ which is consistent with the order in which these concepts will be eliminated by the subsequent examples

$$\langle x_{t+1}, C_T(x_{t+1})\rangle,\ \langle x_{t+2}, C_T(x_{t+2})\rangle, \ldots$$

from S^{C_T}. With probability 1/2 the hypothesis $H \in \mathcal{C}_t$ which is chosen at step t by $\text{GUESSING}_\mathcal{C}$ occurs in the second half of $\prec$. If this happens, then at least half of the other concepts $C \in \mathcal{C}_t$ will have been eliminated by some example in S by the first step $t' > t$ where the algorithm makes the next prediction error (thus $t' := \min\{\tilde{t} > t \mid H(x_{\tilde{t}}) \neq C_T(x_{\tilde{t}})\}$).

For a precise proof of Theorem 2.1 set

$$G_n := \max\{\text{RLC-OBL}(\text{GUESSING}_\mathcal{C}) \mid |\mathcal{C}| = n\}.$$

Define T_n by

$$T_1 = 0$$
$$T_n = \frac{n-1}{n} + \frac{T_1 + \cdots + T_{n-1}}{n} \qquad \text{for } n > 1.$$

One can easily show ([K]) that

$$T_n = \sum_{i=2}^{n} \frac{1}{i} \leq \ln n \qquad \text{for } n > 1.$$

Hence it is sufficient to show by induction on n that $G_n \leq T_n$ for all $n \geq 1$. The case $n = 1$ is trivial. For the induction step fix some concept class $\mathcal{C}$ over a domain X with $|\mathcal{C}| = n$, some $C_T \in \mathcal{C}$ and a sequence $S = \langle x_1, x_2, \ldots\rangle \in X^{\leq\infty}$ such that

$$\text{RLC-OBL}(\text{GUESSING}_\mathcal{C}, C_T, S) = G_n.$$

Let $\prec$ be a linear order on $\mathcal{C}$ such that

$$\min\{i \mid C(x_i) \neq C_T(x_i)\} < \min\{i \mid C'(x_i) \neq C_T(x_i)\}$$

implies that $C \prec C'$.

Number the concepts in $\mathcal{C}$ in such a way that $C_1 \prec C_2 \prec \cdots \prec C_n$. Each $C_j \in \mathcal{C}$ is chosen with probability $1/n$ as first hypothesis of GUESSING$_{\mathcal{C}}$. For each $j \in \{1, \ldots, n\}$ set

$$i_j := \min\{i \mid C_j(x_i) \neq C_T(x_i)\}$$

(set $i_j = \infty$ if there exists no i with $C_j(x_i) \neq C_T(x_i)$). If C_j is chosen as first hypothesis of GUESSING$_{\mathcal{C}}$ and $i_j < \infty$, then at least one error will occur. In this case the remaining class $\widetilde{\mathcal{C}} \subseteq \mathcal{C}$ of all concepts that are consistent with $x_1, \ldots, x_{i_j}$ is of the form $\widetilde{\mathcal{C}} = \{C_{j+k+1}, \ldots, C_n\}$ for some $k \geq 0$ (we may have $k > 0$ since x_{i_j} may eliminate several concepts). By induction hypothesis we have that the expected number of further errors of GUESSING$_{\mathcal{C}}$ is bounded above by

$$\begin{aligned} \text{RLC-OBL}(\text{GUESSING}_{\widetilde{\mathcal{C}}}) &\leq G_{n-j-k} \\ &\leq T_{n-j-k} \\ &\leq T_{n-j}. \end{aligned}$$

Note that in the case where C_n is chosen as first hypothesis we have $i_j = \infty$, hence no error will occur in this case. Thus altogether we have

$$\begin{aligned} G_n &= \text{RLC-OBL}(\text{GUESSING}_{\mathcal{C}}) \\ &\leq \frac{1}{n}\Big(\sum_{j=1}^{n-1}(1 + T_{n-j})\Big) \\ &= T_n. \end{aligned}$$

It turns out that T_n is in fact an optimal upper bound for RLC-OBL(GUESSING$_{\mathcal{C}}$) for concept classes $\mathcal{C}$ of size n. For

$$\mathcal{C}_n = \text{SINGLETON}_n := \{\{i\} \mid i \in \{1, \ldots, n\}\}$$

one can show by induction on n that RLC-OBL(GUESSING$_{\mathcal{C}_n}$) $= T_n$. ∎

Remark 2.2. One can easily see that the upper bound for GUESSING$_{\mathcal{C}}$ in Theorem 2.1, and hence Corollaries 2.3 to 2.5, only require that the environment is weakly oblivious (we refer to this model as RLC-WOBL). In the model RLC-WOBL the environment is allowed to be adaptive and malicious, but it cannot predict future steps of the learner. As before, the environment is required to fix some target concept $C_T \in \mathcal{C}$ at the beginning of the learning process. However the environment is now allowed to let each of its examples $\langle x_i, C_T(x_i)\rangle$ depend on all preceding hypotheses of the learner, and on the predictions of the learner for all preceding examples. In other words: the only information that is not known to the environment when it generates its i-th example $\langle x_i, C_T(x_i)\rangle$ is the prediction $H(x_i)$ of the learner's current hypothesis H for this example x_i (and the future hypotheses of the learner). However this prediction $H(x_i)$ will be made available to the environment immediately after it has generated $\langle x_i, C_T(x_i)\rangle$. In terms of "private" versus "public" coin tosses one can characterize the model RLC-WOBL by specifying that the environment is told the outcomes of all coin tosses of the learner, except for the outcomes of those coin tosses that the learner made for the selection of his current hypothesis H (and for the selection of future hypotheses). Thus the only difference between this model RLC-WOBL and the model RLC (which is discussed in section 3) is the fact that in the RLC-model the environment knows in addition the current hypothesis H of the learner (i.e. it knows $H(x)$ even for those points H for which it has not yet "tested" H) and it knows all future hypotheses of the learner.

In order to adapt the proof of Theorem 2.1 to the model RLC-WOBL, one defines the linear order $\prec$ on $\mathcal{C}_t$ slightly differently. Similarly as at the beginning of the proof of Theorem 2.1, assume that the learner has made a prediction error for the t-th example supplied by the environment, and that $\mathcal{C}_t$ is the set of all concepts in $\mathcal{C}$ that are consistent with the first t examples. One now lets

$$\langle y_1, C_T(y_1)\rangle,\ \langle y_2, C_T(y_2)\rangle, \ldots$$

be the sequence of examples that the environment would subsequently provide in the case that none of them would cause a prediction error of the learner (i.e. if $H(y_i) = C_T(y_i)$ for $i = 1, 2, \ldots$; where H is the current hypothesis of the learner). One then defines $\prec$ as a linear order on $\mathcal{C}_t$ that is consistent with the order in which these concepts would be eliminated by the sequence $(\langle y_i, C_T(y_i)\rangle)_{i \in \mathbf{N}}$. After the first example y_i in this sequence for which the current hypothesis H makes a prediction error (i.e. $H(y_i) \neq C_T(y_i)$), the environment is no longer required to continue with the rest

$$\langle y_{i+1}, C_T(y_{i+1})\rangle,\ \langle y_{i+2}, C_T(y_{i+2})\rangle, \ldots$$

of these examples (this is a difference to the model RLC-OBL). Rather, the environment may now choose (after it has been shown the just refuted hypothesis H of the learner) a completely different sequence of examples. Hence one defines a new linear order $\prec'$ on the set

$$\mathcal{C}_{t+1} := \{C \in \mathcal{C}_t \mid C \text{ is consistent with all previously seen examples}\}$$

by considering the new sequence $(\langle z_j, C_T(z_j)\rangle)_{j \in \mathbf{N}}$ of examples that the environment would provide (after the refutation of hypothesis H by y_i) in case that none of them would cause another prediction error of the learner (i.e. if $H'(z_j) = C_T(z_j)$ for $j = 1, 2, \ldots$; where H' is the next hypothesis chosen by the learner). One defines $\prec'$ as a linear order on $\mathcal{C}_{t+1}$ that is consistent

with the order in which the concepts in $\mathcal{C}_{t+1}$ are eliminated by the new sequence $(\langle z_j, C_T(z_j)\rangle)_{j\in\mathbf{N}}$. This linear order $\prec'$ need not be structurally related to $\prec$ (unlike the situation in the proof of Theorem 2.1), but nevertheless one can repeat for $\prec'$ the same argument as for $\prec$.

Corollary 2.3. There is a randomized on-line learning algorithm for arbitrary feedforward nets (= circuits with "sharp" Boolean threshold gates) that is expected to make at most polynomially in the size of the net many prediction errors for an arbitrary oblivious environment:

Let G be an arbitrary directed acyclic graph with exactly one node of outdegree 0 and n nodes of indegree 0 (labeled by $1,\ldots,n$). Define the associated concept class of neural networks with graph G as follows:

$$\mathcal{C}_G := \Big\{ C \subseteq \{0,1\}^n \;\Big|\; \text{there is an assignment of weights from } \mathbf{R} \text{ to edges in } G \text{ and an assignment of thresholds from } \mathbf{R} \text{ to nodes of indegree } > 0 \text{ in } G \text{ such that the resulting feedforward neural net (with "sharp" Boolean threshold gates) computes } C \Big\}.$$

Then $\text{RLC-OBL}(\mathcal{C}_G) = O((\text{number of edges in } G)^2)$.

Idea of the proof. A Boolean threshold gate of indegree d can only compute $2^{O(d^2)}$ different Boolean functions (even if arbitrary reals are allowed as weights and threshold). Hence $\log|\mathcal{C}_G| = O((\text{number of edges in } G)^2)$.

Note that it is essential for a learning algorithm for a feedforward neural net G that it only uses hypotheses that belong to the concept class $\mathcal{C}_G$: otherwise its hypotheses cannot be realized by some intermediate settings of the weights and thresholds in the neural net G. ∎

Corollary 2.4. Let

$$\mathcal{C}_{k,n} := \Big\{ C \subseteq \{0,1\}^n \;\Big|\; C \text{ is definable by a monomial with at most } k \text{ literals over the Boolean variables } x_1,\ldots,x_n \Big\}.$$

Then $\text{RLC-OBL}(\mathcal{C}_{k,n}) = O(k\cdot\log n)$. ∎

Corollary 2.5. For an arbitrary polynomial $p(n)$ set

$$\mathcal{C}_{p,n} := \Big\{ C \subseteq \{0,1\}^n \;\Big|\; C \text{ is definable by a DNF-formula of length } \leq p(n) \text{ over the Boolean variables } x_1,\ldots,x_n \Big\}.$$

Then $\text{RLC-OBL}(\mathcal{C}_{p,n}) = O(p(n)\cdot\log n)$. ∎

The following lower bound result was first observed by Nick Littlestone [L2]. It improves an earlier result due to Kurt Mehlhorn and the author, who had shown that $\text{RLC}(\mathcal{C}) \geq \frac{1}{2}\cdot\text{LC-ARB}(\mathcal{C})$.

Theorem 2.6. (Littlestone [L2]) For any finite concept class $\mathcal{C}$

$$\text{RLC-OBL}(\mathcal{C}) \geq \frac{1}{2}\cdot\text{LC-ARB}(\mathcal{C}).$$

Proof. Consider any randomized learning algorithm B for $\mathcal{C} \subseteq 2^X$ and a decision tree T for $\mathcal{C}$ in which every leaf has depth $\geq \text{LC-ARB}(\mathcal{C})$ (such T exists by [L1], see also [MT1]). Construct in T a path S from the root to a leaf by recursion. If the so far constructed path S' ends at an internal node ν with label $x \in X$ let p_ν be the probability that B predicts that $C_T(x) = 1$ (after B has processed the sequence of labeled examples which is encoded by S'). Extend S' by one of the two edges that leave node ν according to the following rule: choose the edge with label "0" iff $p_\nu \geq 1/2$.

The constructed path S has length $\ell \geq \text{LC-ARB}(\mathcal{C})$ and ends at a leaf with some $C_T \in \mathcal{C}$ as label. By construction one has $\text{Errors}(B, C_T, S) \geq \ell/2$. ∎

Remark 2.7. The preceding lower bound is optimal insofar as there are concept classes $\mathcal{C}$ for which $\text{RLC-OBL}(\mathcal{C}) = \frac{1}{2}\cdot\text{LC-ARB}(\mathcal{C})$ (for example take $\mathcal{C} = 2^X$).

In the following theorem we compare for arbitrary concept classes $\mathcal{C}$ the learning complexities $\text{LC-ARB}(\mathcal{C})$, $\text{RLC-OBL}(\mathcal{C})$, $\text{LC}(\mathcal{C})$. We write $A \prec B$ if $\forall\mathcal{C}(A(\mathcal{C}) = O(B(\mathcal{C})))$ and for some family $(\mathcal{C}_n)_{n\in\mathbf{N}}$ of concept classes $B(\mathcal{C}_n)$ grows faster than any polynomial in $A(\mathcal{C}_n)$.

Theorem 2.8.

$$\text{LC-ARB} \prec \text{RLC-OBL} \prec \text{LC}$$

Sketch of the proof. In order to separate RLC-OBL from LC-ARB we show that $\text{RLC-OBL}(\text{SINGLETON}_n) = \Omega(\log n)$ (it is obvious that $\text{LC-ARB}(\text{SINGLETON}_n) = 1$). We apply in this lower bound argument Von Neumann's minimax theorem ([Vo], see also [LR], [Y]) to a matrix with rows indexed by arbitrary elements $\langle C_T, S\rangle$ from $\text{SINGLETON}_n \times \{1,\ldots,n\}^{\leq n^2}$ and columns indexed by arbitrary deterministic learning algorithms A for SINGLETON_n (restricted to example sequences S of length $\leq n^2$). The matrix entry for row $\langle C_T, S\rangle$ and column A is $\text{Errors}(A, C_T, S)$. The minimax theorem implies that in order to prove that $\text{RLC-OBL}(\text{SINGLETON}_n) = \Omega(\log n)$, it is sufficient to show that there exists some distribution $\mathcal{P}_n$ over $\text{SINGLETON}_n \times \{1,\ldots,n\}^{\leq n^2}$ such that for every deterministic learning algorithm A for SINGLETON_n one has

$$E_{\mathcal{P}_n(\langle C_T,S\rangle)}(\text{Errors}(A, C_T, S)) = \Omega(\log n).$$

We will show that the following distribution $\mathcal{P}_n$ has the desired property. $\mathcal{P}_n$ is the uniform distribution over

$D_n :=$

$\{ \langle \{\pi(n)\}, S_\pi \rangle \mid \pi$ is a permutation of $\{1,\ldots,n\}$ and S_π is an associated sequence (with repetitions) that begins with n copies of $\pi(1)$, and in which n copies of the subsequence $\langle \pi(1),\ldots,\pi(i)\rangle$ are followed by n copies of the subsequence $\langle \pi(1),\ldots,\pi(i+1)\rangle$, $i = 1,\ldots,n-1 \}$.

We set $\mathcal{P}_n(\langle C_T, S\rangle) = 0$ for $\langle C_T, S\rangle \notin D_n$.

Because of the repetitions in the sequence S_π one can associate with any deterministic learning algorithm A for SINGLETON_n another deterministic learning algorithm A' for SINGLETON_n with $\text{Errors}(A, C_T, S) \geq \text{Errors}(A', C_T, S)$ for all $\langle C_T, S\rangle \in D_n$ such that A' is consistent (i.e. each hypothesis of A' is consistent with all previously seen examples). Hence it is sufficient to show for an arbitrary consistent deterministic learning algorithm A that

$$T_n^A = \Omega(\log n),$$

where

$$T_n^A := E_{\mathcal{P}_n(\langle C_T,S\rangle)}(\text{Errors}(A, C_T, S)).$$

This lower bound follows from the observation that

$$T_n^A = \frac{n-1}{n} + \frac{T_1^A + \ldots + T_{n-1}^A}{n}.$$

The other claims of Theorem 2.8 are consequences of Theorem 2.1 and Theorem 2.6 (consider SINGLETON_n in order to separate RLC-OBL from LC). ∎

Remark 2.9.

(a) The preceding argument together with the proof of Theorem 2.1 shows that GUESSING is an optimal learning algorithm for SINGLETON_n in the model RLC-OBL.

(b) It is not the case that for all concept classes $\mathcal{C}$ one has $\text{RLC-OBL}(\text{GUESSING}_\mathcal{C}) = \Theta(\text{RLC-OBL}(\mathcal{C}))$. For example for $\mathcal{C}_n := \text{SINGLETON}_n \cup \{\emptyset\}$ one has $\text{RLC-OBL}(\mathcal{C}_n) \leq \text{LC}(\mathcal{C}_n) = 1$, but $\text{RLC-OBL}(\text{GUESSING}_{\mathcal{C}_n}) = \Theta(\log n)$.

(c) Apparently there exists a trade-off for on-line learning with an oblivious environment between the number of random bits that are used by a learning algorithm and the "simplicity" of its hypotheses. The algorithms GUESSING and the halving algorithm lie at opposite ends of this spectrum.

3 THE POWER OF RANDOMIZATION FOR ON-LINE LEARNING WITH AN ADAPTIVE ENVIRONMENT

It is not clear from the results of the previous section how much of the performance of the considered learning algorithms should be credited to the use of randomized algorithms, and how much is due to the assumption that the environment is oblivious. We show in this section that randomized learning algorithms can achieve only a substantially smaller improvement in the error bound (compared with the best deterministic learning algorithm) in the case where the environment is as powerful as in the LC-model (and can see the outcomes of all coin tosses of the learner). However even in this model we assume that the environment determines the target concept at the beginning of the learning process (this gives a randomized learner some advantage over a deterministic one).

For any deterministic learning algorithm A for a concept class $\mathcal{C}$ and any target concept $C_T \in \mathcal{C}$ let $\text{Errors}(A, C_T)$ be the maximal length of a learning process of algorithm A if C_T is the target concept (assuming that the counterexamples to hypotheses of A are chosen by an adaptive adversary as in the LC-model). Thus $\text{Errors}(A, C_T) = \max\{\text{Errors}(A, C_T, S) \mid S \in X^{\leq\infty}\}$. Let B be a randomized learning algorithm for $\mathcal{C}$, i.e. B is a distribution $Q_B(A)$ over deterministic learning algorithm A for $\mathcal{C}$. For any $C_T \in \mathcal{C}$ we set $\text{Errors}(B, C_T) := E_{A\in Q_B}(\text{Errors}(A, C_T))$,

$\text{RLC}(B) :=$

$\max \{ \text{Errors}(B, C_T) \mid C_T \in \mathcal{C} \}$,

$\text{RLC}(\mathcal{C}) :=$

$\min \{ \text{RLC}(B) \mid B$ is a randomized learning algorithm for $\mathcal{C}$ that only uses hypotheses from $\mathcal{C} \}$.

It is obvious that $\text{LC}(\mathcal{C}) \geq \text{RLC}(\mathcal{C}) \geq \text{RLC-OBL}(\mathcal{C})$ for any concept class $\mathcal{C}$. We show in Theorems 3.1 and 3.2 that $\text{LC}(\mathcal{C}) > \text{RLC}(\mathcal{C})$ for certain concept classes $\mathcal{C}$. Theorem 3.3 provides a general lower bound for $\text{RLC}(\mathcal{C})$ which implies for many concept classes $\mathcal{C}$ that $\text{RLC}(\mathcal{C})$ is not much smaller than $\text{LC}(\mathcal{C})$. In particular for $\mathcal{C} = \text{SINGLETON}_n$ this lower bound implies that $\text{RLC}(\mathcal{C})$ is exponentially larger than $\text{RLC-OBL}(\mathcal{C})$.

Theorem 3.1. For $\mathcal{C} = 2^{\{1,\ldots,n\}}$,

$$\text{RLC}(\mathcal{C}) = \text{RLC-OBL}(\mathcal{C}) = \frac{1}{2} \cdot \text{LC}(\mathcal{C}) = \frac{n}{2}.$$

Proof. It is obvious that $\text{LC}(\mathcal{C}) = \text{LC-ARB}(\mathcal{C}) = n$. Hence by Theorem 2.6 one has $\text{RLC-OBL}(\mathcal{C}) \geq n/2$.

In order to prove the upper bound for $\text{RLC}(\mathcal{C})$ we

consider a probabilistic learning algorithm B whose first hypothesis H is a randomly chosen subset of $\{1,\ldots,n\}$. Subsequently B changes its hypothesis only on those points where it has made an error on one of the previously seen examples.

If H is the initial hypothesis of B, then B makes $|H \triangle C_T|$ errors in the worst case. Obviously the expected size of $H \triangle C_T$ is $n/2$. ∎

Theorem 3.2.

$$\begin{aligned} \mathrm{RLC}(\mathrm{SINGLETON}_n) &= \frac{1}{2} \cdot \mathrm{LC}(\mathrm{SINGLETON}_n) \\ &= \frac{n-1}{2}. \end{aligned}$$

Proof. In order to show that $\mathrm{RLC}(\mathrm{SINGLETON}_n) \geq (n-1)/2$ one applies Von Neumann's minimax theorem [Vo] to the uniform distribution over $\mathcal{C}$.

For the upper bound one considers a probabilistic learning algorithm B that chooses uniformly randomly any permutation π of $\{1,\ldots,n\}$, and then issues hypotheses in the order $\{\pi(1)\},\{\pi(2)\},\ldots$ (unless it gets a positive counterexample). For any fixed $C_T \in \mathrm{SINGLETON}_n$ the expected number of errors of B is $(n-1)/2$. ∎

Theorem 3.3. For any finite concept class $\mathcal{C}$ with $|\mathcal{C}| > 1$,

$$\mathrm{RLC}(\mathcal{C}) \geq \frac{\mathrm{LC}(\mathcal{C})}{2\lceil \log |\mathcal{C}| \rceil}.$$

Idea of the proof. We first observe that it is sufficient to consider in the definition of $\mathrm{RLC}(\mathcal{C})$ only randomized learning algorithms B with the property that $Q_B(A) > 0$ only if A is a consistent deterministic learning algorithm for $\mathcal{C}$ (A is called consistent if it always outputs hypotheses that are consistent with the preceding counterexamples). There are only finitely many such algorithms A, and hence we can apply Von Neumann's minimax theorem [Vo] (again we only need its "easy" inequality). However we apply it here for a different matrix than in the proof of Theorem 2.8. Here the columns are labeled by consistent deterministic learning algorithms for $\mathcal{C}$ and the rows are labeled by the concepts $C \in \mathcal{C}$. The matrix entry for column A and row C is $\mathrm{Errors}(A,C)$. The minimax theorem implies that for any distribution $\mathcal{P}$ over $\mathcal{C}$ there is a deterministic learning algorithm $A_{\mathcal{P}}$ for $\mathcal{C}$ such that $E_{\mathcal{P}(C)}(\mathrm{Errors}(A_{\mathcal{P}},C)) \leq \mathrm{RLC}(\mathcal{C})$. We exploit this fact for distributions $\mathcal{P}_i$, $i \in \{1,\ldots,\lceil \log |\mathcal{C}| \rceil\}$ over $\mathcal{C}$ which are defined as follows. Each $\mathcal{P}_i$ is uniform on some subclass $\mathcal{C}_i \subseteq \mathcal{C}$ and identically zero on $\mathcal{C} - \mathcal{C}_i$. Set $\mathcal{C}_1 := \mathcal{C}$. Let $\mathcal{C}_{i+1}$ be the class of all $C \in \mathcal{C}_i$ such that $\mathrm{Errors}(A_{\mathcal{P}_i},C) \geq 2 \cdot \mathrm{RLC}(\mathcal{C})$. The definitions of $\mathcal{P}_i$ and $A_{\mathcal{P}_i}$ imply that $|\mathcal{C}_{i+1}| \leq \frac{|\mathcal{C}_i|}{2}$. The desired deterministic learning algorithm A with $\mathrm{LC}(A) \leq \lceil \log |\mathcal{C}| \rceil \cdot 2 \cdot \mathrm{RLC}(\mathcal{C})$ executes in alternation one step in each of the algorithms $A_{\mathcal{P}_i}$. $i = 1,\ldots,\lceil \log |\mathcal{C}| \rceil$. A succeeds for any target concept $C_T \in \mathcal{C}$ after $\leq \lceil \log |\mathcal{C}| \rceil \cdot 2 \cdot \mathrm{RLC}(\mathcal{C})$ steps since one of the algorithms $A_{\mathcal{P}_i}$ identifies C_T after $\leq 2 \cdot \mathrm{RLC}(\mathcal{C})$ steps. ∎

4 COMPARISONS WITH OTHER PREDICTION MODELS AND ALGORITHMS

The LC-model differs in three essential aspects from the prediction model of [HKLW], [HLW1] with a stochastic environment, which is closely related to Valiant's model for PAC-learning [V] (we refer to the prediction model of [KHLW], [HLW1] in the following as "PAC prediction model"):

(a) the environment is represented in the LC-model by a worst case adaptive adversary, whereas it is represented in the PAC prediction model by a worst case probability distribution over the domain (in both models one considers the worst case choice of a target concept $C_T \in \mathcal{C}$)

(b) in the LC-model one measures the performance of a learning algorithm in terms of its total number of errors, whereas in the PAC prediction model one is interested in the expected number of errors for the first m examples

(c) in the LC-model the current hypothesis of the learning algorithm is always required to be from the same concept class $\mathcal{C}$ as the target concept, whereas the hypothesis in the PAC prediction model need not be from $\mathcal{C}$.

The following result shows that the new model RLC-OBL for on-line learning with an oblivious environment may be viewed as an interpolation between the LC-model and the PAC prediction model: it is equivalent to a learning model which agrees in point (a) with the PAC prediction model and in points (b) and (c) with the LC-model. In order to make this equivalence precise we introduce the following notation.

Consider an arbitrary concept class $\mathcal{C}$ over a domain X (i.e. $\mathcal{C} \subseteq 2^X$) and an arbitrary distribution D over X. For $S \in X^\infty$ we write $S \in D^\infty$ to indicate that S results from independent random drawings from X according to D. For any deterministic learning algorithm A for $\mathcal{C}$ and any $C_T \in \mathcal{C}$ we define:

$$\mathrm{Errors}(A,C_T,D) := E_{S \in D^\infty}(\mathrm{Errors}(A,D_T,S)),$$

and for any randomized learning algorithm B

$$\mathrm{Errors}(B,C_T,D) := E_{A \in Q_B}(\mathrm{Errors}(A,C_T,D)).$$

Finally we define

$$\mathrm{RLC\text{-}PAC}(B) := \max\{\, \mathrm{Errors}(B,C_T,D) \mid C_T \in \mathcal{C} \text{ and } D \text{ is a distribution over } X \,\},$$

$$\text{RLC-PAC}(\mathcal{C}) := \min\{\, \text{RLC-PAC}(B) \mid B \text{ is a randomized learning algorithm for } C \text{ which uses only hypotheses from } \mathcal{C} \,\}.$$

We have added the suffix "PAC" in "RLC-PAC" to indicate that with regard to the assumption about the environment (point (a) in the preceding discussion) this model agrees with the PAC prediction model. Note however that with regard to points (b) and (c) RLC-PAC agrees with the LC-model (and with RLC-OBL).

The following theorem shows that in the here considered context the assumption of an arbitrary worst case oblivious environment is equivalent to that of a stochastic environment with a worst case distribution.

Theorem 4.1. For every concept class $\mathcal{C}$:

$$\text{RLC-OBL}(\mathcal{C}) = \text{RLC-PAC}(\mathcal{C}).$$

Idea of the proof. "$\geq$" is trivial. In order to prove "$\leq$" one associates with any sequence $S = \langle x_1, x_2, \ldots\rangle$ of elements (without repetitions) a suitable distribution D_S over X such that for arbitrary random drawings $\widetilde{S}$ according to D_S the first occurrence of elements of X in $\widetilde{S}$ is likely to be in the same order as in S (i.e. $D_S(x_1) \gg D_S(x_2) \gg \ldots$). Let B be any randomized learning algorithm with $\text{RLC-PAC}(B) = \text{RLC-PAC}(\mathcal{C})$. One defines for any $\delta > 0$ a learning algorithm B_δ with $\text{RLC-OBL}(B_\delta) \leq (1+\delta)\cdot\text{RLC-PAC}(B)$ which generates (internally) for the prediction for the t-th element x_t of any given sequence $S = \langle x_1, x_2, \ldots\rangle \in X^{\leq\infty}$ the associated distribution $D_{(x_1,\ldots,x_t)}$. B_δ predicts "$x_t \in C_T$" with probability p_t, where p_t is defined as the probability that B predicts "$x_t \in C_T$" for the first occurrence of x_t in arbitrary sequences $\widetilde{S}$ that result from random drawings according to $D_{(x_1,\ldots,x_t)}$ (note that B_δ might give different responses for the first occurrence of x_t in $\widetilde{S}$ in dependence on the number of repetitions of preceding elements in $\widetilde{S}$). ∎

In the following we will compare the prediction performance of the very simple randomized algorithm GUESSING (which was introduced in the proof of Theorem 2.1) with the performance of other prediction algorithms (we view in this context the notions "learning algorithms" and "prediction algorithms" as being equivalent). Since $\text{RLC-OBL}(\text{GUESSING}_\mathcal{C}) = O(\log|\mathcal{C}|)$, $\text{GUESSING}_\mathcal{C}$ will make for all $\mathcal{C}$ with $\log|\mathcal{C}| \ll \text{LC}(\mathcal{C})$ substantially fewer errors in a learning situation with an oblivious environment than the best known prediction algorithm with hypotheses from $\mathcal{C}$ in the LC-model.

The expected number of errors of $\text{GUESSING}_\mathcal{C}$ is bounded above by the same parameter $O(\log|\mathcal{C}|)$ as the worst case number of error of the well-known halving algorithm (see [A1], [L1], [MT1]). The latter algorithm performs well even against an adaptive environment and it requires no random bits, but it uses hypotheses which do not belong to $\mathcal{C}$ (which are in general difficult to compute). Haussler, Littlestone and Warmuth [HLW2] introduced the "1-inclusion graph prediction algorithm" which also uses hypotheses that do not belong to $\mathcal{C}$, and which is expected to make at most $O(\text{VC-dim}(\mathcal{C})\cdot\log m)$ prediction errors for m examples (but it requires that the examples result from independent random drawings). This bound is smaller than $\log|\mathcal{C}|$ for certain $\mathcal{C}$ and certain values of m. A similar bound has been achieved for a probabilistic environment by Schapire [S] for any PAC-learnable $\mathcal{C}$ with a computationally feasible prediction algorithm (this algorithm also uses hypotheses which do not belong to $\mathcal{C}$). Other prediction algorithms which use hypotheses that do not belong to $\mathcal{C}$ result from the work by Littlestone and Warmuth [LW] on the weighted majority algorithm (typically these algorithms use "nicer" hypotheses outside of $\mathcal{C}$ than the halving algorithm, but they may make more errors than the halving algorithm).

In the full version of [LW] one can also find a discussion of a randomized version of the weighted majority algorithm which uses only hypotheses from $\mathcal{C}$ and which works well even in the case of an adaptive environment, but which requires to change the hypothesis after each example (not only after prediction errors).

So far we have examined in this paper only the expected total number of errors for randomized prediction algorithms in our new model with an arbitrary oblivious environment. The preceding discussion showed that with regard to this measure $\text{GUESSING}_\mathcal{C}$ is not surpassed by other known prediction algorithms that use only hypotheses from $\mathcal{C}$. It turns out that with regard to another measure, the expected number of errors for the first m examples for any oblivious sequence S of examples, one can design for certain concept classes $\mathcal{C}$ a variation of $\text{GUESSING}_\mathcal{C}$ which performs better than $\text{GUESSING}_\mathcal{C}$. However this is only possible for concept classes $\mathcal{C}$ with $\text{LC-ARB}(\mathcal{C}) \ll \log|\mathcal{C}|$, since even if the environment is oblivious one can construct for any randomized learning algorithm B a target concept $C_T \in \mathcal{C}$ and an oblivious sequence S of examples such that B is likely to make for any $m \leq \text{LC-ARB}(\mathcal{C})$ at least $m/2$ prediction errors for the first m examples in S (use the construction in the proof of Theorem 2.6).

A typical concept class $\mathcal{C}$ with $\text{LC-ARB}(\mathcal{C}) \ll \log|\mathcal{C}|$ is SINGLETON_n. The following result shows that for this concept class $\mathcal{C}$ one can in fact design another randomized prediction algorithm with hypotheses from $\mathcal{C}$ which is expected to make fewer prediction errors than $\text{GUESSING}_\mathcal{C}$ for the first m examples (for any $m < n\log n$ and any oblivious sequence S of examples).

Theorem 4.2. There is a randomized prediction algorithm for SINGLETON_n which only uses hypotheses from SINGLETON_n and which is expected to make at most $O(\min(m/n, \log n))$ prediction errors for the first m examples of any given (oblivious) sequence $S \in \{1, \ldots, n\}^{\leq \infty}$.

Idea of the proof. The algorithm GUESSING (see the proof of Theorem 2.1) does not achieve this error bound since for many sequences S of $m = n$ examples it is expected to make $\log n$ errors. Therefore we combine GUESSING with another randomized prediction algorithm BLIND-GUESSING, which chooses after each error uniformly random any $C \in \text{SINGLETON}_n$ as next hypotheses (C need not be consistent with all previous examples). The claimed relative error bound is achieved by the randomized prediction algorithm B that calls after the k-th prediction error the algorithm $\text{GUESSING}_{\text{SINGLETON}_n}$ if k is even, and $\text{BLIND-GUESSING}_{\text{SINGLETON}_n}$ if k is odd. ∎

Acknowledgements

We would like to thank David Haussler, Nick Littlestone, Michael Luby, Kurt Mehlhorn, Robert H. Sloan, Manfred Warmuth and an anonymous referee for helpful comments. Written under partial support by NSF-Grant CCR 89033398.

References

[A1] D. Angluin, Queries and concept learning, *Machine Learning*, **2**, 1988, 319–342.

[A2] D. Angluin, Negative results for equivalence queries, *Machine Learning*, **5**, 1990, 121–150.

[GRS] S. A. Goldman, R. L. Rivest, R. E. Schapire, Learning binary relations and total orders, Proc. of the 30th IEEE Symposium on Foundations of Computer Science 1989, 46–51.

[HKLW] D. Haussler, M. Kearns, N. Littlestone, M. K. Warmuth, Equivalence of models for polynomial learnability, to appear in *Information and Computation* (see the Proc. of COLT 1988 for an extended abstract).

[HLW1] D. Haussler, N. Littlestone, M. Warmuth, Expected mistake bounds for on-line learning algorithms, unpublished manuscript (April 1987).

[HLW2] D. Haussler, N. Littlestone, M. Warmuth, Predicting $\{0,1\}$-functions on randomly drawn points, Proc. of COLT 1988, 280–296.

[K] D. E. Knuth, The Art of Computer Programming, vol. 1, Addison-Wesley (Reading, 1973).

[L1] N. Littlestone, Learning quickly when irrelevant attributes abound: a new linear threshold learning algorithm, *Machine Learning*, **2**, 1987, 285–318.

[L2] N. Littlestone, private communication.

[LW] N. Littlestone, M. Warmuth, The weighted majority algorithm, Proc. of the 30th IEEE Symposium on Foundations of Computer Science 1989, 256–261.

[LR] R. D. Luce, H. Raiffa, Games and Decisions, John Wiley & Sons (New York, 1957).

[MT1] W. Maass, G. Turan, On the complexity of learning from counterexamples, Proc. of the 30th IEEE Symposium on Foundations of Computer Science 1989, 262–267.

[MT2] W. Maass, G. Turan, Algorithms and lower bounds for on-line learning of geometrical concepts, 1991, submitted for publication.

[MP] M. Minsky, S. Papert, Perceptrons: An Introduction to Computational Geometry, Expanded edition, MIT Press, 1988.

[N] N. Nilsson, Learning Machines, McGraw-Hill (New York, 1965).

[R] F. Rosenblatt, Principles of Neurodynamics, Spartan Books (New York, 1962).

[RM] D. E. Rumelhart, J. L. McClelland, Parallel Distributed Processing, MIT Press (Cambridge, 1986).

[S] R. E. Schapire, The strength of weak learnability, preprint (Oct. 1989).

[V] L. G. Valiant, A theory of the learnable, *Comm. of the ACM*, vol. 27, 1984, 1134–1142.

[Vo] J. Von Neumann, Zur Theorie der Gesellschaftsspiele, *Math. Annalen*, **100**, 1928, 295–320.

[Y] A. C. Yao, Probabilistic computations: towards a unified measure of complexity, Proc. of the 18th IEEE Symposium on Foundations of Computer Science, 1977, 222–227.