

# **On-Line Learning with an Oblivious Environment and the Power of Randomization**

Wolfgang Maass<sup>1,2</sup>

TR-91-003

January, 1991

## **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 counter examples, but also from examples which support their current hypothesis. 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 Angluin's classical model for on-line learning with an adaptive worst-case environment. 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.

For comparison we also examine the power of randomization for Angluin's model for learning with an adaptive environment.

<sup>1</sup>Written under partial support by NSF Grant CCR 8903398. Parts of the research for this paper were carried out while the author was visiting the Department of Computer Science of the Universitaet des Saarlandes in Saarbruecken (Germany) and the International Computer Science Institute in Berkeley. The author would like to thank these two institutes for their hospitality.

<sup>2</sup>Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago. E-mail: U45381@UICVM.BITNET.



## 1. Introduction.

The most common computational model for on-line (or: “incremental”) learning is due to Angluin [A]. In this model the learner proposes “hypotheses”  $H$  from a fixed “concept class”  $\mathcal{C} \subseteq 2^X$  over a finite domain  $X$ . The goal of the learner is to “learn” an unknown “target concept”  $C_T \in \mathcal{C}$  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”  $x \in H \Delta C_T := (C_T - H) \cup (H - C_T)$ .  $x$  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  $\mathcal{C}$  is any algorithm  $A$  that produces new hypotheses

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

in dependence of counterexamples  $x_j \in H_j^A \Delta C_T$  for the preceding hypotheses  $H_j^A$ . (One also refers to these hypotheses as “equivalence queries” [A]).

The “learning complexity”  $LC(A)$  of such a learning algorithm  $A$  is defined by

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

The “learning complexity”  $LC(\mathcal{C})$  of a concept class  $\mathcal{C}$  is defined by

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

We set

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

One may argue that the previously defined learning model is quite “pessimistic”. The definition of  $LC(A)$  implicitly assumes that the environment is “adaptive” to the learner: the definition of  $LC(A)$  is based on the assumption that the environment “knows” the current hypothesis  $H$  of the learning algorithm  $A$  and that it supplies among all possible counterexamples  $x \in H \Delta C_T$  the “least informative” one. It is difficult to imagine a learning situation where this pessimistic view of the environment as an “adaptive” (and malicious) adversary is actually justified. Hence a large lower bound for  $LC(\mathcal{C})$  does not necessarily imply that  $\mathcal{C}$  is not on-line learnable in the presence of a non-probabilistic worst case environment.

We introduce in this paper a variation of Angluin’s learning model where we assume that the environment is “oblivious” to the activities of the learner. More precisely, 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 learning complexity in the new model with an oblivious environment from the previously given learning complexity in Angluin’s models with an adaptive environment 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  $S = \langle x_1, x_2, \dots \rangle \in X^{\leq \infty}$  we write  $S^C$  for the associated sequence  $\langle \langle x_1, C(x_1) \rangle, \langle x_2, C(x_2) \rangle, \dots \rangle$  of labeled examples for  $C$  (each concept  $C$  is identified with its characteristic function  $\chi_C : X \rightarrow \{0, 1\}$ ).

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  $\langle \langle x_1, C_T(x_1) \rangle, \dots, \langle x_{t-1}, C_T(x_{t-1}) \rangle \rangle$ ) 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  $S^{C_T}$ .

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  $\text{Errors}(B, C_T, S) := E_{A \in Q_B}(\text{Errors}(A, C_T, S))$ ,

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

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

A learning algorithm for this model RLC-OBL has to perform well even if the environment does not behave like a time-invariant stochastic process. Examples of learning situations where it is not adequate to view the environment as a time-invariant stochastic process are provided by some customary training methods for artificial neural nets [RM] and by systems for speech recognition and optical character recognition.

So far the investigation of on-line learning with a non-stochastic environment has focused on learning from counterexamples. However various natural and artificial learning systems draw information both from counterexamples and from examples which support their current hypotheses. The considered model for on-line learning with an oblivious environment provides a theoretical framework for the investigation of this more powerful type of learning algorithms.

## 2. Error-bounds for Randomized On-line Learning Algorithms with an Oblivious Environment.

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

$$\text{RLC-OBL}(\mathcal{C}) \leq \ln |\mathcal{C}| + O(1).$$

**Sketch of the 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  $C_T \in \mathcal{C}$ ,  $S = \langle x_1, x_2, \dots \rangle \in X^{\leq \infty}$ . Assume that  $\text{GUESSING}_{\mathcal{C}}$  makes a prediction error for the  $t$ -th element  $x_t$  of  $S$ . Define

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

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

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

from  $S^{C_T}$ . With probability  $\frac{1}{2}$  the hypothesis  $H \in \mathcal{C}_t$  which is chosen at step  $t$  by  $\text{GUESSING}_{\mathcal{C}}$  occurs in the second half of  $\prec_t$ . If this happens, then at least half of the other concepts  $C \in \mathcal{C}_t$  will have been eliminated by some example in  $S$  before 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 one uses the preceding argument in order to show by induction on  $n$  that for all  $n \geq 1$  and all concept classes  $\mathcal{C}$  of size  $n$

$$\text{RLC-OBL}(\text{GUESSING}_{\mathcal{C}}) \leq T_n,$$

where  $T_n$  is defined by

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

It is easy to show that  $T_n = \sum_{i=2}^n \frac{1}{i} = \ln n + O(1)$  (see [K]).

It turns out that  $T_n$  is in fact an optimal upper bound for  $\text{RLC-OBL}(\text{GUESSING}_C)$  for concept classes  $C$  of size  $n$ : For  $C_n := \text{SINGLETON}_n := \{\{i\} \mid i \in \{1, \dots, n\}\}$  one can show by induction on  $n$  that  $\text{RLC-OBL}(\text{GUESSING}_{C_n}) = T_n$ .  $\square$

**Corollary 2.2.** 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  $1, \dots, n$ ). Define the associated concept class as follows:

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

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

**Idea of the proof.** Exploit the fact that  $\log |C_G| = O((\text{number of edges in } G)^2)$ .

Note that it is essential for a learning algorithm for a feedforward neural net  $G$  that it only uses hypotheses  $H$  that can be represented by some setting of weights and thresholds in  $G$  (i.e.  $H \in C_G$ ).  $\square$

**Corollary 2.3.** Let  $C_{k,n} = \{C \subseteq \{0,1\}^n \mid C \text{ is definable by a monomial with at most } k \text{ literals over the Boolean variables } x_1, \dots, x_n\}$ .

Then  $\text{RLC-OBL}(C_{k,n}) = O(k \cdot \log n)$ .  $\square$

**Corollary 2.4.** For an arbitrary polynomial  $p(n)$  set  $C_{p,n} := \{C \subseteq \{0,1\}^n \mid C \text{ is definable by a DNF-formula of length } \leq p(n) \text{ over the Boolean variables } x_1, \dots, x_n\}$ .

Then  $\text{RLC-OBL}(C_{p,n}) = O(p(n) \cdot \log n)$ .  $\square$

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

**Theorem 2.5.** (Littlestone [L2]): For any finite concept class  $C$

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

**Proof.** Consider any randomized learning algorithm  $B$  for  $C \subseteq 2^X$  and a decision tree  $T$  for  $C$  in which every leaf has depth  $\geq \text{LC-ARB}(C)$  (such  $T$  exists by [L1], see also [MT]). 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  $\nu$  with label  $x \in X$  let  $p_\nu$  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  $\nu$  according to the following rule: choose the edge with label "0" iff  $p_\nu \geq \frac{1}{2}$ .

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  $\text{Errors}(B, C_T, S) \geq \ell/2$ .  $\square$

**Remark 2.6.** 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  $(\mathcal{C}_n)_{n \in \mathbb{N}}$  of concept classes  $B(\mathcal{C}_n)$  grows faster than any polynomial in  $A(\mathcal{C}_n)$ .

**Theorem 2.7.**  $\text{LC-ARB} \prec \text{RLC-OBL} \prec \text{LC}$ .

**Sketch of the proof.** In order to separate  $\text{RLC-OBL}$  from  $\text{LC-ARB}$  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 ([V], see also [LR], [Y]) to a matrix with rows indexed by arbitrary elements  $\langle C_T, S \rangle$  from  $\text{SINGLETON}_n \times \{1, \dots, n\}^{\leq n^2}$  and columns indexed by arbitrary deterministic learning algorithms  $A$  for  $\text{SINGLETON}_n$  (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  $\text{SINGLETON}_n \times \{1, \dots, n\}^{\leq n^2}$  such that for every deterministic learning algorithm  $A$  for  $\text{SINGLETON}_n$  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

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

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  $S_\pi$  one can associate with any deterministic learning algorithm  $A$  for  $\text{SINGLETON}_n$  another deterministic learning algorithm  $A'$  for  $\text{SINGLETON}_n$  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

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

The other claims of Theorem 2.7 are consequences of Theorem 2.1 and Theorem 2.5 (consider  $\text{SINGLETON}_n$  in order to separate  $\text{RLC-OBL}$  from  $\text{LC}$ ).  $\square$

**Remark 2.8.**

- (a) The preceding argument together with the proof of Theorem 2.1 shows that GUESSING is an optimal learning algorithm for  $\text{SINGLETON}_n$  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 of Angluin’s model where the environment is adaptive.

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 model of [A], [MT], see the definition of  $\text{LC}(A)$  in section 1). 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  $\mathcal{C}$ . Define for any  $C_T \in \mathcal{C}$

$$\begin{aligned} \text{Errors}(B, C_T) &:= E_{A \in Q_B}(\text{Errors}(A, C_T)), \\ \text{RLC}(B) &:= \max\{\text{Errors}(B, C_T) \mid C_T \in \mathcal{C}\}, \text{ and} \\ \text{RLC}(\mathcal{C}) &:= \min\{\text{RLC}(B) \mid B \text{ is a randomized learning algorithm} \\ &\quad \text{for } \mathcal{C} \text{ that only uses hypotheses from } \mathcal{C}\}. \end{aligned}$$

We show in Theorem 3.1 that  $\text{RLC}(\mathcal{C}) < \text{LC}(\mathcal{C})$  for certain concept classes  $\mathcal{C}$ . Theorem 3.2 provides a lower bound for the power of randomization in the here considered model.

**Theorem 3.1.**  $\text{RLC}(\mathcal{C}) = \frac{1}{2} \cdot \text{LC}(\mathcal{C})$  for  $\mathcal{C} = \text{SINGLETON}_n$  and  $\mathcal{C} = 2^{\{1, \dots, n\}}$ . □

**Theorem 3.2.** For any finite concept class  $\mathcal{C}$  with  $|\mathcal{C}| > 1$ ,

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

**Idea of the proof.** We first observe that it is sufficient to consider in the definition of  $\text{RLC}(\mathcal{C})$  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 [V] (again we only need its "easy" inequality). However we apply it here for a different matrix than in the proof of Theorem 2.7. 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  $\text{Errors}(A, C)$ . 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  $E_{\mathcal{P}(C)}(\text{Errors}(A_{\mathcal{P}}, C)) \leq \text{RLC}(\mathcal{C})$ . We exploit this fact for distributions  $\mathcal{P}_i$ ,  $i \in \{1, \dots, \lceil \log |\mathcal{C}| \rceil\}$  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  $\text{Errors}(A_{\mathcal{P}_i}, C) \geq 2 \cdot \text{RLC}(\mathcal{C})$ . 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  $\text{LC}(A) \leq \lceil \log |\mathcal{C}| \rceil \cdot 2 \cdot \text{RLC}(\mathcal{C})$  executes in alternation one step in each of the algorithms  $A_{\mathcal{P}_i}$ ,  $i = 1, \dots, \lceil \log |\mathcal{C}| \rceil$ .  $A$  succeeds for any target concept  $C_T \in \mathcal{C}$  after  $\leq \lceil \log |\mathcal{C}| \rceil \cdot 2 \cdot \text{RLC}(\mathcal{C})$  steps since one of the algorithms  $A_{\mathcal{P}_i}$  identifies  $C_T$  after  $\leq 2 \cdot \text{RLC}(\mathcal{C})$  steps.  $\square$

#### 4. Comparisons with other Prediction Models and Algorithms.

Angluin's model for on-line learning [A] (this is the model LC which is defined at the beginning of section 1) 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 [HKLW], [HLW1] in the following as "PAC prediction model"):

- (a) the environment is represented in Angluin's 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 Angluin's 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 Angluin's 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 Angluin's 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 Angluin's 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:

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

and for any randomized learning algorithm  $B$

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

Finally we define

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

and

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

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 Angluin’s 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  $S = \langle x_1, x_2, \dots \rangle$  of elements (without repetitions) a suitable distribution  $D_S$  over  $X$  such that for arbitrary random drawings  $\tilde{S}$  according to  $D_S$  the first occurrence of elements of  $X$  in  $\tilde{S}$  is likely to be in the same order as in  $S$  (i.e.  $D_S(x_1) \gg D_S(x_2) \gg \dots$ ). 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  $B_\delta$  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  $S = \langle x_1, x_2, \dots \rangle \in X^{\leq \infty}$  the associated distribution  $D_{\langle x_1, \dots, x_t \rangle}$ .  $B_\delta$  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  $\tilde{S}$  that result from random drawings according to  $D_{\langle x_1, \dots, x_t \rangle}$  (note that  $B_\delta$  might give different responses for the first occurrence of  $x_t$  in  $\tilde{S}$  in dependence on the number of repetitions of preceding elements in  $\tilde{S}$ ).  $\square$

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 Angluin’s 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 [A], [L1], [MT]). 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 2.5).

A typical concept class  $\mathcal{C}$  with  $\text{LC-ARB}(\mathcal{C}) \ll \log |\mathcal{C}|$  is  $\text{SINGLETON}_n$ . 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  $\text{SINGLETON}_n$  which only uses hypotheses from  $\text{SINGLETON}_n$  and which is expected to make at most  $O(\min(\frac{m}{n}, \log n))$  prediction errors for the first  $m$  examples of any given (oblivious) sequence  $S \in \{1, \dots, n\}^{\leq \infty}$ .

**Idea of the proof.** The algorithm  $\text{GUESSING}$  (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  $\text{GUESSING}$  with another randomized prediction algorithm  $\text{BLIND-}$

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.  $\square$

### Acknowledgements.

We would like to thank David Haussler, Nick Littlestone, Michael Luby, Kurt Mehlhorn and Manfred Warmuth for stimulating discussions.

## REFERENCES

- [A] D. Angluin, Queries and concept learning, *Machine Learning*, **2**, 1988, 319-342.
- [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).
- [MT] 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.
- [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.

