, , 1–35 ()
© Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Structural Results About On-line Learning Models With and Without Queries

PETER AUER pauer@igi.tu-graz.ac.at
Institute for Theoretical Computer Science, Graz University of Technology, Klosterwiesgasse 32/2, A-8010 Graz, Austria.

PHILIP M. LONG plong@comp.nus.edu.sg
Department of Computer Science, National University of Singapore, Singapore 119260, Republic of Singapore.

Editor: Leonard Pitt and Lisa Hellerstein

Abstract. We solve an open problem of Maass and Turán, showing that the optimal mistake-bound when learning a given concept class without membership queries is within a constant factor of the optimal number of mistakes plus membership queries required by an algorithm that can ask membership queries. Previously known results imply that the constant factor in our bound is best possible. We then show that, in a natural generalization of the mistake-bound model, the usefulness to the learner of arbitrary "yes-no" questions between trials is very limited. We show that several natural structural questions about relatives of the mistake-bound model can be answered through the application of this general result. Most of these results can be interpreted as saying that learning in apparently less powerful (and more realistic) models is not much more difficult than learning in more powerful models.

Keywords: Computational learning theory, Learning with queries, Mistake bounds, Function learning, Learning with noise.

1. Introduction

In this paper, we present a new technique for proving structural results about on-line learning models, and describe a number of applications of this technique. For the most part, we will focus on the amount of information required for learning, and will ignore computation time. Many of the models considered in this paper are variants of the mistake-bound model, so we begin by describing it.

1.1. *The standard mistake-bound model*

In the standard mistake bound model [13, 3], learning is assumed to proceed in *trials*, where in the tth trial the learner

- is presented with an element x_t of some domain X,
- outputs a prediction $\hat{y}_t \in \{0, 1\}$
- discovers $y_t \in \{0, 1\}$ (called *reinforcement*).

If $\hat{y}_t \neq y_t$, the learner is said to have made a *mistake* on trial t, and the goal is to make few mistakes. It is further assumed that the learner knows of a set F of functions from X to $\{0,1\}$ containing a function f that satisfies $f(x_t) = y_t$ for all trials t. The performance of a learning algorithm is then measured by its worst-case number of mistakes, over all sequences $(x_1, y_1), (x_2, y_2), \ldots$ of elements of $X \times \{0,1\}$ for which there exists an $f \in F$ satisfying the above. Denote the optimal such performance by $\mathrm{opt}_{\mathrm{stand}}(F)$.[1]

1.2. Membership queries

In a heavily studied relative of the mistake-bound model [3], it is further assumed that, between trials, the learner may query $f(x)$ for elements x of its choosing. The performance of a learner for a particular sequence $\langle (x_t, y_t) \rangle_t$ is then measured by the sum of the number of its mistakes and its total number of queries between trials. Let us denote the optimal worst-case performance for a particular class F of functions from X to $\{0,1\}$ (defined analogously to the above) by $\mathrm{opt}_{\mathrm{memb}}(F)$.

We show that, for all F,

$$\mathrm{opt}_{\mathrm{memb}}(F) \geq (\log_2 4/3)\mathrm{opt}_{\mathrm{stand}}(F).$$

The VC-dimension of a class F is a common measure of the "richness" of F. As a direct consequence of the above bound, we obtain the following:

$$\mathrm{opt}_{\mathrm{memb}}(F) \geq (\log_2 4/3)\mathrm{VCdim}(F)$$

(note that $\log_2 4/3$ is approximately $1/2.41$). An example due to Maass and Turán shows that in neither of the above bounds can the constant be improved.

The previously best bounds, due to Maass and Turán [19], were

$$\begin{aligned} \mathrm{opt}_{\mathrm{memb}}(F) &\geq \frac{\mathrm{opt}_{\mathrm{stand}}(F)}{\log_2(1 + \mathrm{opt}_{\mathrm{stand}}(F))} \\ \mathrm{opt}_{\mathrm{memb}}(F) &\geq \frac{1}{7}\mathrm{VCdim}(F). \end{aligned}$$

We further show that if $F = \cup_s F_s$ and $X = \cup_n X_n$, then if there is an algorithm A that, given that the hidden function is taken from F_s and the x_t's come from X_n,

- makes its predictions in time pseudo-polynomial[2] in n and s
- makes at most polynomial in n and s mistakes
- asks polynomial in $\log n$ and $\log s$ membership queries,

then there is an algorithm A_0 that

- makes its predictions in time pseudo-polynomial in n and s
- makes at most polynomial in n and s mistakes

- asks *no* membership queries.

(Intuitively s measures the complexity of the function class F_s and n measures the length of the inputs $x_t \in X_n$.)

1.3. The strength of weak reinforcement

There are two very natural ways to generalize the standard mistake-bound model to the case in which the values to be predicted come from some finite set, possibly with more than two members [2]. At the end of a given trial t, either the algorithm could be told whether or not its prediction $\hat{y}_t$ was correct ("weak reinforcement") or it could be told the correct value y_t ("strong reinforcement"). Both types of reinforcement occur in nature. Notice that in the case in which the y_t's come from $\{0, 1\}$, both kinds of reinforcement are equivalent.

How much weaker is weak reinforcement? Suppose for a given set X and a finite set Y of at least two elements (from which the y_t's will be chosen), for a set F of functions from X to Y, we define[3] $\mathrm{opt}_{\mathrm{strong}}(F)$ and $\mathrm{opt}_{\mathrm{weak}}(F)$ in an analogous manner as $\mathrm{opt}_{\mathrm{stand}}(F)$, except replacing the standard reinforcement with strong and weak reinforcement respectively. We show that

$$\mathrm{opt}_{\mathrm{weak}}(F) \leq 1.39|Y|(\lceil 1 + \log_2(|Y| - 1)\rceil \mathrm{opt}_{\mathrm{strong}}(F) + 2).$$

A trivial lower bound shows that this bound is within an $O(\log |Y|)$ factor of the best possible.

1.4. Agnostic learning

For many applications, it is too optimistic to assume that there is an f from a reasonably small known class F that perfectly maps the x_t's to the corresponding y_t's in $\{0, 1\}$. A well established approach in such cases [26, 15, 14, 9, 16, 6, 7] is to assume nothing about the (x_t, y_t) pairs, and instead, for a given F, to give bounds on the number of mistakes made by a given learning algorithm in terms of the minimum over $f \in F$ of the number η of trials t for which $f(x_t) \neq y_t$. Learning models like this are often referred to as agnostic learning models[4] [12].

It is convenient to assume that the learner knows a bound on η before learning takes place, although this assumption can be removed with a slight degradation in the bounds via standard doubling techniques. In this case, informally, let $\mathrm{opt}_{\mathrm{agn}}(F, \eta)$ be the best bound on the number of mistakes that can be obtained given the assumption that there is an $f \in F$ such that the number of trials t for which $f(x_t) \neq y_t$ is at most η. As a special case of our main theorem (Theorem 4), we obtain the following bound:

$$\mathrm{opt}_{\mathrm{agn}}(F, \eta) \leq 4.82(\mathrm{opt}_{\mathrm{agn}}(F, 0) + \eta) + 1. \qquad (1)$$

Note that $\mathrm{opt}_{\mathrm{agn}}(F, 0) = \mathrm{opt}_{\mathrm{stand}}(F)$. Since, for many applications, one expects η to be much larger than $\mathrm{opt}_{\mathrm{agn}}(F, 0)$, optimizing the constant on the η term seems worthwhile. By applying the more refined Theorem 5, we can show that for all $\epsilon \leq 1/20$,

$$\mathrm{opt}_{\mathrm{agn}}(F,\eta) \leq \left(\frac{4}{\epsilon}\ln\frac{1}{\epsilon}\right)\mathrm{opt}_{\mathrm{agn}}(F,0) + \left(2+\frac{5}{2}\epsilon\right)\eta. \tag{2}$$

Littlestone and Warmuth [15] proved that for *any* F with $|F| > 1$,

$$\mathrm{opt}_{\mathrm{agn}}(F,\eta) \geq \mathrm{opt}_{\mathrm{agn}}(F,0) + 2\eta.$$

Thus, the bound of (1) is within a small constant factor of optimal *for each* (nontrivial) F. This reduces the problem of determining $\mathrm{opt}_{\mathrm{agn}}(F,\eta)$ to within a constant factor to that of determining $\mathrm{opt}_{\mathrm{agn}}(F,0)$ to within a constant factor. In other words, in a sense, it reduces the study of the agnostic learning model to the study of the standard mistake-bound model. (Notice, however, that this is without regard to *computational* complexity.) Furthermore, using (2), the constant on the η term can be brought arbitrarily close to the optimal 2, at the expense of increasing the constant on the other term.

Similar results about $\mathrm{opt}_{\mathrm{agn}}(F,\eta)$ were independently obtained by Cesa-Bianchi, Freund, Helmbold and Warmuth [7].

Littlestone and Warmuth [15], and independently Vovk [27], showed that for any F,

$$\mathrm{opt}_{\mathrm{agn}}(F,\eta) \leq 2.41(\log_2|F| + \eta). \tag{3}$$

Other refinements of this result, which retain the same flavor in that they are in terms of $\log_2|F|$ and η, but some of which concern probabilistic algorithms, which we don't study here, are described in [15, 14, 26, 6].[5] Due to the fact that for any finite F, $\mathrm{opt}_{\mathrm{agn}}(F,0) \leq \log_2|F|$ [13], our bound of (1) is always at most a small constant factor greater than (3). Furthermore, sometimes it is substantially less.

As an example, if SUBSP_n is the set of (indicator functions for) linear subspaces of $\mathbb{R}^n$, there is a trivial algorithm for learning given that a function in SUBSP_n maps x_t's (in $\mathbb{R}^n$) to corresponding y_t's that makes at most n mistakes [21], but SUBSP_n is infinite, so no bound on $\mathrm{opt}_{\mathrm{agn}}(\mathrm{SUBSP}_n,\eta)$ can be obtained from (3) and related results. However, a bound of $4.82(n+\eta)$ (as mentioned above, within a small constant factor of optimal) follows immediately from (1).

Finally, by adapting the proofs of Theorem 4 and Theorem 5, we may obtain (1) and (2) in the case that the predictions $\hat{y}_t$ and the true values y_t are chosen from any set Y, F is a set of functions from X to Y, and the goal is still to have few mistakes, i.e. trials in which $\hat{y}_t \neq y_t$.

1.5. Closure results

For many classes F of functions from some set X to $\{0,1\}$, one obtains a richer class by taking k-wise OR's of elements of F, i.e. by defining

$$\mathrm{OR}_k(F) = \{f_1 \vee \cdots \vee f_k : f_1, ..., f_k \in F\}$$

where $f_1 \vee \cdots \vee f_k$ has the obvious interpretation. How much harder can $\mathrm{OR}_k(F)$ be than F? By applying our Theorem 4, we can show that for all F,

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{OR}_k(F)) \leq 2.41k\lceil 1 + \log_2 k\rceil \mathrm{opt}_{\mathrm{stand}}(F) + 1.$$

A trivial lower bound shows that this bound is within an $O(\log k)$ factor of the best possible. While analogous results for the PAC model were obtained some time ago [10, 4], to the best of our knowledge, the question of whether there was any bound on $\mathrm{opt}_{\mathrm{stand}}(\mathrm{OR}_k(F))$ in terms of k and $\mathrm{opt}_{\mathrm{stand}}(F)$ had remained open. A more general result of this type is described in Section 4.4.

1.6. Temporal credit assignment

Sometimes on-line learning algorithms cannot expect to get reinforcement before having to predict again, and the reinforcement they get may be ambiguous, indicating that a mistake was made some time in the recent past.[6] We adapt the standard mistake-bound model to include such learning situations by assuming that after every certain number, say r, of trials, the learning algorithm is told whether any of the past r predictions were incorrect. (Of course, for applications, the number of trials between reinforcements seems bound to vary; however, we obtain an equivalent model if r is an upper bound on the number of trials between reinforcements.)

If, for a given class F of $\{0,1\}$-valued functions, we define $\mathrm{opt}_{\mathrm{amb,r}}(F)$ to be the worst-case number of mistakes made by the optimal algorithm in this model (where a mistake is said to be made if the algorithm was incorrect in *any* of its predictions before a particular reinforcement, see Section 4.5), we may obtain the following bound,

$$\mathrm{opt}_{\mathrm{amb,r}}(F) \leq 2(\ln 2r) \cdot 2^r \cdot \mathrm{opt}_{\mathrm{amb,1}}(F).$$

Note that $\mathrm{opt}_{\mathrm{amb,1}}(F) = \mathrm{opt}_{\mathrm{stand}}(F)$. We also describe a lower bound that shows that this bound cannot be significantly improved.

1.7. A unifying framework: the MB and MBQ models

All of the above results are direct consequences of a single theorem about more general models. These models (which we call the MB model and MBQ model) are relatives of the mistake-bound model [3, 13]. As in that model, we assume learning is an on-line process, proceeding in *trials.* During the tth trial,

1. the learner receives an *instance* x_t from some set X,
2. the learner outputs a *prediction* $\hat{y}_t$ in some set Y,
3. the learner receives a *response* $\overline{y}_t \in Y$ indicating that $y_t \neq \overline{y}_t$.

This type of response given in the MB model is a subtle point. Instead of receiving direct feedback to its prediction $\hat{y}_t$ the learner receives only some value $\overline{y}_t$ different from the correct y_t. For $|Y| = 2$ this is equivalent to giving the correct value y_t as a response since it can be inferred immediately. But for $|Y| > 2$ the environment is more flexible in giving feedback to the learner, even more flexible than just telling the learner if its prediction was correct or not as in the weak reinforcement model. The learner is said to have made a mistake if $\hat{y}_t = \overline{y}_t$, i.e. if the response $\overline{y}_t$ *implies* that the learner's prediction was incorrect. The learner is not charged for trials with $\hat{y}_t \neq y_t$ but $\overline{y}_t \neq \hat{y}_t$.

The learner's prior knowledge is modeled by assuming that the learner knows of a set $\mathcal{L} \subseteq (X \times Y)^*$ of sequences of pairs (x_t, y_t) containing those pairs encountered on any run of the algorithm.

In the MB model, the goal of the learner is simply to minimize the number of trials t in which it makes a mistake. For a particular set $\mathcal{L}$, we then define $\mathrm{opt}_{\mathrm{MB}}(\mathcal{L})$ to be the best bound on the number of mistakes that can be obtained by a learning algorithm given the assumption that the sequence $\langle (x_t, y_t) \rangle_t$ of (x_t, y_t) pairs is in $\mathcal{L}$.

In the MBQ model, the learner is allowed to ask arbitrary "yes-no" questions about the entire sequence $\langle (x_t, y_t) \rangle_t$ between trials to gain additional information. In this model, the performance of the learner is measured by the sum of the number of questions it asked and the number of mistakes, and $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})$ is defined to be the optimal performance given $\mathcal{L}$ in a similar manner to the above. More formal descriptions of both models are given in Section 3.1. All the models considered in this paper are summarized in Table 1.

We show for all $\mathcal{L} \subseteq (X \times Y)^*$ that

$$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \leq \begin{cases} 2|Y| - 1 + \frac{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) - \log_2 |Y|}{\log_2 \frac{2|Y|}{2|Y|-1}} & \text{if } |Y| \leq 2^{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})} \\ 2^{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})+1} & \text{otherwise} \end{cases}$$

which implies the looser but more suggestive bound

$$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \leq 1.39|Y| \left(\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) + 2\right).$$

These are the general results which imply the results mentioned in previous sections. We also show that this bound is within a constant factor of the best possible for all values of $|Y|$ and $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})$.

We also consider the natural variant of the MB and MBQ models where the response $\rho_t \in \{\mathrm{TRUE}, \mathrm{FALSE}\}$ does indicate whether the learner's prediction has been correct or not. These models are denoted by MBρ and MBQρ. Note again that for $|Y| = 2$ the MB and MBρ models are equivalent (as are the MBQ and MBQρ models) and that they are a generalization of the standard mistake-bound model from learning functions to learning sequences. For the relationship of the MBρ and MBQρ models we show the bound

$$\mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L}) \leq (|Y| \ln |Y|)\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}) + 130(|Y| \ln \ln |Y|)\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})$$

which is almost best possible.

Table 1. A summary of the learning models studied in this paper. In each trial t, the learner is presented with an element x_t of some domain X, outputs a prediction $\hat{y}_t$ from some set Y, then possibly gets some information about the correct y_t. In some models queries are allowed between trials; for these the algorithms are evaluated by summing the number of prediction errors and the number of queries. In different models, different types of assumptions about the relationship between the x_t's and y_t's are considered. We denote by F a class of functions from X to Y, we denote by $\mathcal{L} \subseteq (X \times Y)^*$ a set of sequences of pairs (x_t, y_t), and we denote by $Q \subseteq \mathcal{L}$ a subset of $\mathcal{L}$. Note that for $|Y| = 2$, $\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) = \mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L})$ and $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) = \mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})$.

notation for optimal	Y	relationship between x_t's and y_t's	information at end of trial	queries allowed
$\mathrm{opt}_{\mathrm{stand}}(F)$	$\{0, 1\}$	for some $f \in F$, for all t, $f(x_t) = y_t$	y_t	none
$\mathrm{opt}_{\mathrm{memb}}(F)$	$\{0, 1\}$	for some $f \in F$, for all t, $f(x_t) = y_t$	y_t	what is $f(x)$?
$\mathrm{opt}_{\mathrm{agn}}(F, \eta)$	$\{0, 1\}$	for some $f \in F$, $|\{t : f(x_t) \neq y_t\}| \leq \eta$	y_t	none
$\mathrm{opt}_{\mathrm{amb,r}}(F)$	$\{0, 1\}$	for some $f \in F$, for all t, $f(x_t) = y_t$	in every rth trial, was there a mistake in the past r trials?	none
$\mathrm{opt}_{\mathrm{strong}}(F)$	any finite set	for some $f \in F$, for all t, $f(x_t) = y_t$	y_t	none
$\mathrm{opt}_{\mathrm{weak}}(F)$	any finite set	for some $f \in F$, for all t, $f(x_t) = y_t$	is $y_t = \hat{y}_t$?	none
$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L})$	any finite set	$\langle (x_t, y_t) \rangle_t \in \mathcal{L}$	$\overline{y}_t \neq y_t$	none
$\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})$	any finite set	$\langle (x_t, y_t) \rangle_t \in \mathcal{L}$	$\overline{y}_t \neq y_t$	is $\langle (x_t, y_t) \rangle_t \in Q$?
$\mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L})$	any finite set	$\langle (x_t, y_t) \rangle_t \in \mathcal{L}$	is $y_t = \hat{y}_t$?	none
$\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})$	any finite set	$\langle (x_t, y_t) \rangle_t \in \mathcal{L}$	is $y_t = \hat{y}_t$?	is $\langle (x_t, y_t) \rangle_t \in Q$?

1.8. Related results and the organization of the paper

Our technique to prove the above results builds on the "weighted majority" technique of Littlestone and Warmuth [15]. The weighted majority technique uses a fixed set of specialized subalgorithms, and it uses a weighting scheme to combine the predictions of these algorithms. In contrast, our technique *dynamically* creates subalgorithms depending on information gathered during a particular run.

Kulkarni, Mitter and Tsitsiklis [11] studied PAC learning using *only* "yes-no" questions. Bshouty, Goldman, Hancock and Matar studied the use of membership queries to reduce the number of mistakes as much as possible [5].

The paper is organized as follows. In Section 2 we illustrate our main technique by showing how membership queries can be simulated by an algorithm which cannot ask membership queries. In Section 3 we present our general results from which most of the other results can be derived. In Section 4 we give various applications of our main result, and we conclude in Section 5. Appendix A contains several lower bound proofs.

2. Bounds on the usefulness of membership queries

In this section we illustrate the techniques of this paper with an example. We bound the number of mistakes in the standard mistake-bound model in terms of the number of queries and mistakes in the mistake-bound model with membership queries.

Choose a set X. In this subsection, we study a model due to Angluin [3]. (To make our notation and terminology more uniform throughout the paper, on the face of it, the model we describe looks somewhat different than Angluin's original model, but the two can be shown to be equivalent [13].) In this model, we assume that a function f from X to $\{0,1\}$ is hidden from the learner, and that learning proceeds in trials, where in the tth trial, the learner (a) receives $x_t \in X$ from the environment, (b) outputs a prediction $\hat{y}_t \in \{0,1\}$, (c) discovers $f(x_t)$. We further assume that, before each trial, the learner may determine $f(x)$ for different $x \in X$ of its choosing (*membership queries*). The performance of an algorithm on a particular run is the total of the number of mistakes and the number of membership queries, and the overall quality of an algorithm is measured by its worst-case performance. Then $\mathrm{opt}_{\mathrm{memb}}(F)$ is the optimal performance that can be obtained in this model, and $\mathrm{opt}_{\mathrm{stand}}(F)$ is the optimal performance that can be obtained with an algorithm that never asks membership queries.

Theorem 1 *Choose X, and a set F of functions from X to $\{0,1\}$. Then*

$$\mathrm{opt}_{\mathrm{stand}}(F) \leq \frac{\mathrm{opt}_{\mathrm{memb}}(F)}{\log_2(4/3)}.$$

The VC-dimension [25] of a class F is defined by

$$\mathrm{VCdim}(F) = \max\{d : \exists x_1, ..., x_d \in X, \{(f(x_1), ..., f(x_d)) : f \in F\} = \{0,1\}^d\}.$$

The fact that $\mathrm{opt}_{\mathrm{stand}}(F) \geq \mathrm{VCdim}(F)$ [13] trivially yields the following corollary.

THEOREM 2 *Choose X, and a set F of functions from X to $\{0,1\}$. Then*

$$\mathrm{opt}_{\mathrm{memb}}(F) \geq \log_2(4/3)\mathrm{VCdim}(F).$$

As discussed in the introduction, the following theorem due to Maass and Turán shows that the constant cannot be improved in either Theorem 1 or Theorem 2.

THEOREM 3 ([19]) *There is a family $\langle X_n \rangle_n$ of sets and a family $\langle F_n \rangle_n$ such that for each n, F_n is a set of functions from X_n to $\{0,1\}$ and*

$$\mathrm{opt}_{\mathrm{memb}}(F_n) \leq (\log_2(4/3) + o(1))\mathrm{VCdim}(F_n) \leq (\log_2(4/3) + o(1))\mathrm{opt}_{\mathrm{stand}}(F_n)$$

as $n \to \infty$.

Proof of Theorem 1: Let A^{memb} be an optimal learning algorithm which for all targets $f \in F$ and $x_1, x_2, ... \in X$ has its total number of mistakes and membership queries bounded by $\mathrm{opt}_{\mathrm{memb}}(F)$. We construct a learning algorithm A^{stand} which makes at most $\mathrm{opt}_{\mathrm{memb}}(F)/\log_2(4/3)$ mistakes, and asks no membership queries.

The algorithm A^{stand} runs copies A_i^{memb} of A^{memb} as subalgorithms and keeps a weight w_i for each copy. These weights indicate how "reliable" the corresponding copies are. Initially A^{stand} starts with one copy of A^{memb} and its weight is 1. To prove the theorem we (as observers of the algorithm A^{stand}) investigate how the total sum of all weights w_i changes, and we keep track of a special copy A_s^{memb} (and its weight) which performs in the same way as A^{memb} would perform if membership queries were available. Initially the single copy is the special one. During the tth trial, algorithm A^{stand} behaves as follows:

- As long as any copy A_i^{memb} wants to ask a membership query "$f(q) = 1$?", this copy is split into two copies, one copy receives the answer YES and the other copy receives the answer NO. The weight $w_i/2$ is assigned to both copies. Intuitively the weight is split between the two copies since it is unknown whether the YES or the NO answer is correct.

  Clearly the total sum of weights is not changed.

  If A_i^{memb} is the special copy then one of the new copies represents the correct answer to the query and this copy becomes the special one. Its weight is half the weight of the original special copy.

- Since we can assume that no copy asks more than $\mathrm{opt}_{\mathrm{memb}}(F)$ queries, eventually all copies are ready to make a prediction. When this happens, algorithm A^{stand} constructs its prediction $\hat{y}_t$ using a majority vote of the predictions $\hat{y}_{i,t}$ of the subalgorithms according to their weights,

$$\hat{y}_t = \begin{cases} 1 & \text{if} \quad \sum_{i:\hat{y}_{i,t}=1} w_i \geq \sum_{i:\hat{y}_{i,t}=0} w_i \\ 0 & \text{if} \quad \sum_{i:\hat{y}_{i,t}=1} w_i < \sum_{i:\hat{y}_{i,t}=0} w_i. \end{cases} \tag{4}$$

Then the correct answer y_t is passed to all copies A_i^{memb} of A^{memb}. If A^{stand} made a mistake, then those copies A_i^{memb} whose predictions $\hat{y}_{i,t}$ were the same as A^{stand}'s prediction $\hat{y}_t$ also made mistakes. The weights of all these copies are multiplied by 1/2 (since they seem less reliable). The copies that predicted correctly have their weights unchanged. If A^{stand} predicts correctly, for simplicity, none of the copies have their weights reduced.

Since $\sum_{i:\hat{y}_{i,t}=\hat{y}_t} w_i \geq \sum_{i:\hat{y}_{i,t}\neq\hat{y}_t} w_i$, arguing as in [15], we have for the modified weights w_i' that

$$\begin{aligned}
\sum_i w_i' &= \sum_{i:\hat{y}_{i,t}=\hat{y}_t} w_i' + \sum_{i:\hat{y}_{i,t}\neq\hat{y}_t} w_i' \\
&= \frac{1}{2}\sum_{i:\hat{y}_{i,t}=\hat{y}_t} w_i + \sum_{i:\hat{y}_{i,t}\neq\hat{y}_t} w_i \\
&= \frac{3}{4}\sum_i w_i - \frac{1}{4}\sum_{i:\hat{y}_{i,t}=\hat{y}_t} w_i + \frac{1}{4}\sum_{i:\hat{y}_{i,t}\neq\hat{y}_t} w_i \\
&\leq \frac{3}{4}\sum_i w_i.
\end{aligned}$$

Thus the total sum of weights decreases by at least a factor 3/4 if A^{stand} makes a mistake.

The weight of the special copy is multiplied by 1/2 only if it predicted incorrectly.

To summarize, if A^{stand} has made M mistakes the total sum of all weights is at most $(3/4)^M$. On the other hand the weight of the special copy is always at least $(1/2)^{\mathrm{opt}_{\mathrm{memb}}(F)}$ since the number of mistakes plus the number of membership queries of the special copy is bounded by $\mathrm{opt}_{\mathrm{memb}}(F)$. By taking logarithms and solving for M, we get

$$M \leq \frac{\mathrm{opt}_{\mathrm{memb}}(F)}{\log_2 4/3}$$

which implies the theorem. □

To get a feel for how A^{stand} works, it is worthwhile to view its state as a tree, where the various copies of A^{memb} correspond to the leaves. For example, suppose A^{stand} is learning f, and that the single copy of A^{memb} would be ready to make a prediction. Then the tree at this point would consist of a single node labeled READY. The prediction of A^{stand} would then be just that of the single copy of A^{memb}. Suppose that A^{stand} made a mistake in the first trial. Then the single copy A^{memb} made a mistake on the first trial, too. This is reflected in the tree by giving the node corresponding to the single copy of A^{memb} a child (see Figure 1a). Suppose that the single copy of A^{memb} then wanted to ask a membership query q_1. Then A^{stand} would create two copies of A^{memb}, one which it would give the response YES, and the other which would get the response NO. If the copy that

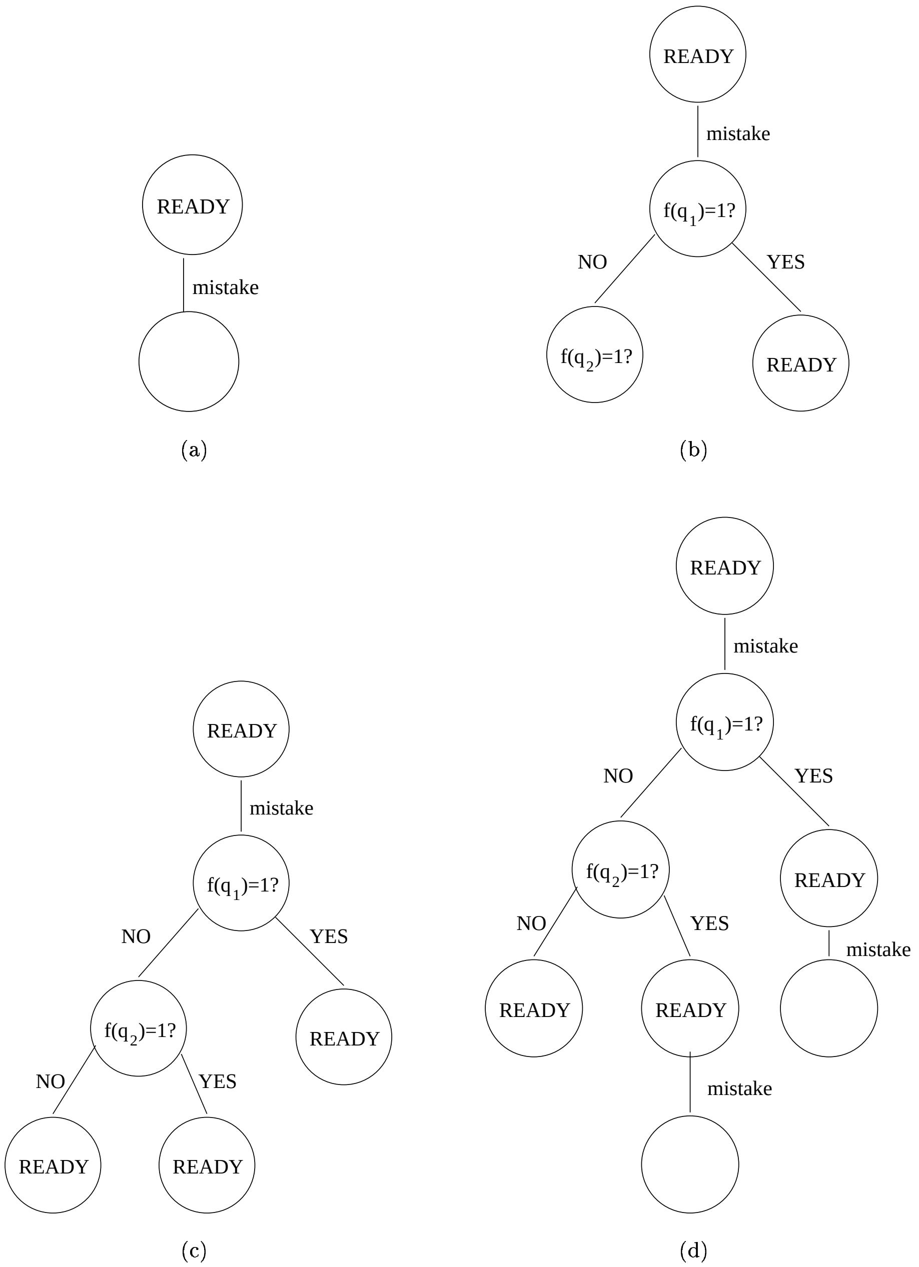

Figure 1. A succession of trees corresponding to states of the algorithm A^{stand}.

got the response YES did not want to ask another membership query, and the copy that got the response NO asked another membership query, call it q_2, then we can visualize the state of A^{stand} with the following tree in Figure 1b. Now, A^{stand} would "expand" the leaf on the left, again creating two copies, which would be given YES and NO respectively as answers to their most recent question. If neither of these copies wanted to ask a membership query, then the tree in Figure 1c would encode the state of A^{stand}. Now A^{stand} would be ready for the second trial. Its prediction $\hat{y}_2$ would be calculated as the weighted majority vote of the copies of A^{memb} in the leaves of the tree, see equation (4). The weight of each copy is simply 2^{-d} when d is the depth of the corresponding leaf in the tree. The leaves corresponding to those copies of A^{memb} which made a mistake would be given children, and the new tree would look for example like Figure 1d. The process would continue in this manner, with A^{stand} "expanding" all leaves whose copies of A^{memb} ask membership queries until there are no more such leaves, and then constructing its prediction using those of the copies on the leaves as described above.

3. The MB and MBQ models

In this section we present our general result from which the other results can be obtained.

3.1. *Definitions*

Choose sets X and Y, and let $\mathcal{L} \subseteq (X \times Y)^*$ be some set of sequences of elements of $X \times Y$ ($|Y| \geq 2$). A kind of subset of $(X \times Y)^*$ will be of particular interest. For a set F of functions from X to Y, let $\mathcal{L}_F$ consist of those sequences $\langle (x_t, y_t) \rangle_t$ of elements of $X \times Y$ for which there is an $f \in F$ such that for all t, $f(x_t) = y_t$. Our results, however, will hold for arbitrary sets of sequences of (x_t, y_t) pairs.

We consider the following MB model for on-line learning of sequences $\sigma = \langle (x_t, y_t) \rangle_t$ from $\mathcal{L}$. This model is included to provide the cleanest statement we can of a general result unifying our treatment of the applications in the paper; it is not intended itself as an accurate model of applied learning problems.

As in the standard mistake-bound model, we assume learning proceeds in *trials*. In the tth trial,

- the algorithm is given x_t,
- the algorithm outputs a prediction $\hat{y}_t$ of y_t
- the algorithm receives a response $\overline{y}_t \in Y$ with $\overline{y}_t \neq y_t$.

In the MBQ model, we further assume that the learner may ask arbitrary "yes-no" questions about σ between trials. Since for any "yes-no" question about σ one is equivalently asking whether σ is contained in the set of those elements of $\mathcal{L}$ for which the answer is "yes", a "yes-no" question can be formalized as asking "Is $\sigma \in \mathcal{L}'$?" for some $\mathcal{L}' \subseteq \mathcal{L}$.

A prediction of an algorithm is counted as mistake if $\overline{y}_t = \hat{y}_t$, i.e. an algorithm is only charged for a trial when evidence of a mistake is given. We measure the performance $M(\mathcal{L}, A)$ of an algorithm A for learning $\mathcal{L}$ in the MBQ model by the maximum, over $\sigma \in \mathcal{L}$ and any consistent responses, of the number of mistakes and queries made by A. We define $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})$ to be the minimum of $M(\mathcal{L}, A)$ over all learning algorithms A, and $\mathrm{opt}_{\mathrm{MB}}(\mathcal{L})$ to be the minimum of $M(\mathcal{L}, A)$ over learning algorithms A that do not ask queries.

For some of the applications, we will want to assign different costs to YES answers to queries, NO answers, and mistakes. Choose positive constants c_{YES}, c_{NO} and c_m, and let $\vec{c} = (c_{\mathrm{YES}}, c_{\mathrm{NO}}, c_m)$. Define $M(\mathcal{L}, A, \vec{c})$ to be the maximum, over $\sigma \in \mathcal{L}$ and consistent responses, of $c_{\mathrm{YES}} \cdot n_{\mathrm{YES}} + c_{\mathrm{NO}} \cdot n_{\mathrm{NO}} + c_m \cdot m$, where n_{YES}, n_{NO} and m are the number of A's queries answered YES, the number answered NO, and the number of A's mistakes. Define $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}, \vec{c})$ to be the minimum of $M(\mathcal{L}, A, \vec{c})$ over learning algorithms A.

3.2. Upper bounds

The following result limits the usefulness of "yes-no" questions.

THEOREM 4 *For any sets X and Y for which $|Y| \geq 2$, and any $\mathcal{L} \subseteq (X \times Y)^*$*

$$\begin{aligned} \mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \;&\leq\; \begin{cases} 2|Y| - 1 + \frac{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) - \log_2 |Y|}{\log_2 \frac{2|Y|}{2|Y|-1}} & \textit{if } |Y| \leq 2^{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})} \\ 2^{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})+1} & \textit{otherwise} \end{cases} \\ &\leq\; 1.39|Y| \left(\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) + 2\right). \end{aligned}$$

We also have the following result concerning different costs for the number of YES and NO answers and the number of mistakes.

THEOREM 5 *Choose $0 < \alpha, \beta, \gamma < 1$ such that $\alpha + \beta = 1$. Choose sets X and Y for which $|Y| \geq 2$, and some $\mathcal{L} \subseteq (X \times Y)^*$. Then for the weighted cost $M = \mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}, (\log_2 \frac{1}{\alpha}, \log_2 \frac{1}{\beta}, \log_2 \frac{1}{\gamma}))$,*

$$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \leq \begin{cases} 1 + \frac{|Y|-1}{1-\gamma} + \frac{M - \log_2 |Y|}{\log_2 \frac{|Y|}{|Y|-(1-\gamma)}} & \textit{if } |Y| \leq 2^M \\ \frac{2^M}{1-\gamma} & \textit{otherwise} \end{cases}$$

The first inequality[7] of Theorem 4 follows from Theorem 5 by setting $\alpha = \beta = \gamma = 1/2$. The proof of Theorem 5 is similar to the proof of Theorem 1 in that a master algorithm that does not ask questions keeps track of several copies of an algorithm that does, and generates its predictions from the copies using weighted voting. But the generality of the theorem gives rise to some new issues.

First, if $|Y| > 2$, if the master algorithm finds out that its prediction $\hat{y}_t$ on trial t is wrong, i.e. $\overline{y}_t = \hat{y}_t$, it cannot tell whether the predictions of those copies of the question-asking algorithm that didn't predict $\hat{y}_t$ were correct or wrong. But since in the MBQ model such feedback is not required, it is sufficient that the master algorithms gives response $\overline{y}_t$ to all the copies. (For the MBQρ model of Section 3.4, where such feedback is required, this problem has to be dealt with differently.) Another complication is that the weights are adjusted by factors other than 1/2. This is needed for some of the applications. Finally, the analysis for Theorem 5 is divided into two stages. In the first stage, we show that the total weight goes down by a certain factor, as we did in the proof of Theorem 1. In the second stage, we use an additive bound on the reduction of weight, which is sometimes tighter due to the fact that $|Y|$ can be large. This is apparently required to get bounds that are tight to within a constant factor.

Proof of Theorem 5: Choose an MBQ algorithm A^{MBQ} which is optimal with respect to costs $\log_2 \frac{1}{\alpha}$, $\log_2 \frac{1}{\beta}$, and $\log_2 \frac{1}{\gamma}$ for YES answers, NO answers, and mistakes respectively. Consider the MB algorithm A^{MB} which uses A^{MBQ} as a subroutine defined in Figure 2.

By induction, at any time during the execution of A^{MB} when learning some sequence σ with responses $\langle \overline{y}_t \rangle_t$, there is a special copy A_s^{MBQ} which corresponds to a state of A^{MBQ} when learning σ with responses $\langle \overline{y}_t \rangle_t$. This follows from the fact that both answers to queries are given to corresponding copies of A^{MBQ} and that all responses $\overline{y}_t$ are given to the copies. The weight $w_s = \alpha^{n_{\mathrm{YES}}(s)} \beta^{n_{\mathrm{NO}}(s)} \gamma^{m(s)}$ of the special copy satisfies $w_s \geq 2^{-M}$, where $M = \mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}, (\log_2 \frac{1}{\alpha}, \log_2 \frac{1}{\beta}, \log_2 \frac{1}{\gamma}))$.

Denote by $W = \sum_i w_i$ the total weight of all copies A_i^{MBQ} maintained by A^{MB}. First W is 1 when A^{MB} starts. Note further that since $\alpha + \beta = 1$, that W does not change when copies are duplicated and given both answers to "yes-no" questions during the simulation of queries.

Our proof proceeds by using W as a measure of progress. As mentioned earlier, the analysis is divided into two stages. The first stage consists of those trials t such that, before the beginning of trial t, $W > 2^{-M}|Y|$. The second stage consists of the remaining trials. In both stages we are ignoring the change of W during trials in which the master algorithm does not make a mistake since W never increases.

Let us assume as a first case that $|Y| < 2^M$. In this case, the first stage has at least one trial. We begin by bounding the number m_1 of mistakes made by A^{MB} in the first stage. Choose some trial t in the first stage. Suppose $\hat{y}_t$ is a mistake, i.e. $\overline{y}_t = \hat{y}_t$. Then

$$\begin{aligned}
W_{\mathrm{new}} &= \sum_i \alpha^{n_{\mathrm{YES,new}}(i)} \beta^{n_{\mathrm{NO,new}}(i)} \gamma^{m_{\mathrm{new}}(i)} \\
&= \sum_{i: A_i^{\mathrm{MBQ}}(x_t) \neq \hat{y}_t} \alpha^{n_{\mathrm{YES,old}}(i)} \beta^{n_{\mathrm{NO,old}}(i)} \gamma^{m_{\mathrm{old}}(i)} \\
&\qquad + \sum_{i: A_i^{\mathrm{MBQ}}(x_t) = \hat{y}_t} \gamma \alpha^{n_{\mathrm{YES,old}}(i)} \beta^{n_{\mathrm{NO,old}}(i)} \gamma^{m_{\mathrm{old}}(i)} \\
&\leq (1 - 1/|Y|) W_{\mathrm{old}} + \gamma W_{\mathrm{old}}/|Y| \\
&= \left(1 - \frac{1-\gamma}{|Y|}\right) W_{\mathrm{old}},
\end{aligned}$$

Notation:

Maintains a set of copies A_i^{MBQ} of A^{MBQ} where each copy corresponds to a subset $\mathcal{L}_i \subseteq \mathcal{L}$ which denotes the current information of A_i^{MBQ} about the target sequence σ. Each copy maintains its number of YES answers, NO answers, and mistakes received so far, denoted by $n_{\mathrm{YES}}(i), n_{\mathrm{NO}}(i), m(i)$. The weight of a copy A_i^{MBQ} is calculated as $w_i = \alpha^{n_{\mathrm{YES}}(i)} \cdot \beta^{n_{\mathrm{NO}}(i)} \cdot \gamma^{m(i)}$ (thus α weights YES answers, β weights NO answers, and γ weights mistakes in predictions). We assume that a copy A_i^{MBQ} terminates if $n_{\mathrm{YES}}(i) \log_2 \frac{1}{\alpha} + n_{\mathrm{NO}}(i) \log_2 \frac{1}{\beta} + m(i) \log_2 \frac{1}{\gamma} > \mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}, (\log_2 \frac{1}{\alpha}, \log_2 \frac{1}{\beta}, \log_2 \frac{1}{\gamma}))$.

Initialization:

Initially there is only a single copy A_1^{MBQ} with $\mathcal{L}_1 = \mathcal{L}$ and $n_{\mathrm{YES}}(1) = n_{\mathrm{NO}}(1) = m(1) = 0$.

Simulating queries:

As long as there is a copy A_i^{MBQ} which wants to ask a yes-no question this copy is duplicated giving a copy A_j^{MBQ} and the answer YES is given to copy A_i^{MBQ} and the answer NO is given to copy A_j^{MBQ}.

Making a prediction:

If no copy wants to ask a yes-no question x_t is received from the environment and the prediction

$$\hat{y}_t := \mathrm{argmax}_{y \in Y} \sum_{i: A_i^{\mathrm{MBQ}}(x_t) = y} w_i$$

is calculated as the value with the highest weight.

Update:

The response $\overline{y}_t$ is given to all copies A_i^{MBQ}.

The steps Simulating queries, Making a prediction, and Update are repeated as long as required.

Figure 2. Algorithm A^{MB} from the proof of Theorem 5

where "old" and "new" indicate whether the values of the variables are considered before or after trial t. The inequality follows from the fact that A^{MB} makes the prediction with the greatest weight, and therefore a fraction at least $1/|Y|$ of the weight is behind this prediction. By induction, after A^{MB} has made m mistakes in the first stage, we have that

$$W \leq \left(1 - \frac{1-\gamma}{|Y|}\right)^m . \tag{5}$$

Since the first stage is over if $W \leq 2^{-M}|Y|$ inequality (5) implies that

$$\left(1 - \frac{1-\gamma}{|Y|}\right)^{m_1 - 1} > 2^{-M}|Y|.$$

Solving for m_1 yields that

$$m_1 \leq 1 + \frac{M - \log_2 |Y|}{\log_2 \frac{|Y|}{|Y|-(1-\gamma)}}. \tag{6}$$

Now, we bound the number of mistakes in the second stage. For any trial in the second stage with a mistake,

$$\begin{aligned}
&W_{\mathrm{new}} - W_{\mathrm{old}} \\
&= \sum_i \left(\alpha^{n_{\mathrm{YES,new}}(i)} \beta^{n_{\mathrm{NO,new}}(i)} \gamma^{m_{\mathrm{new}}(i)} - \alpha^{n_{\mathrm{YES,old}}(i)} \beta^{n_{\mathrm{NO,old}}(i)} \gamma^{m_{\mathrm{old}}(i)}\right) \\
&= \sum_{i : A_i^{\mathrm{MBQ}}(x_t) \neq \hat{y}_t} \left(\alpha^{n_{\mathrm{YES,new}}(i)} \beta^{n_{\mathrm{NO,new}}(i)} \gamma^{m_{\mathrm{new}}(i)} - \alpha^{n_{\mathrm{YES,old}}(i)} \beta^{n_{\mathrm{NO,old}}(i)} \gamma^{m_{\mathrm{old}}(i)}\right) \\
&= (\gamma - 1) \sum_{i : A_i^{\mathrm{MBQ}}(x_t) \neq \hat{y}_t} \alpha^{n_{\mathrm{YES,old}}(i)} \beta^{n_{\mathrm{NO,old}}(i)} \gamma^{m_{\mathrm{old}}(i)} \\
&\leq (\gamma - 1) 2^{-M}
\end{aligned}$$

since there is a special copy A_s^{MBQ} with $w_s \geq 2^{-M}$, and A^{MB} made the prediction with the greatest weight. Since, prior to the start of the second stage, W was at most $2^{-M}|Y|$, and at any time the total weight is at least 2^{-M}, this implies that the number of mistakes in the second stage is at most $(|Y| - 1)/(1 - \gamma)$. Combining this with (6) completes the proof in the case that $|Y| < 2^M$.

The proof in the case that $|Y| \geq 2^M$ goes as above, except that there is no first stage in this case, and in the analysis of the second stage, in place of the assumption that the weight at the beginning of the second stage is at most $2^{-M}|Y|$, we use that it is at most 1. □

3.3. A lower bound

In this section we present a lower bound that matches Theorem 4 to within constant factors. The proof is given in Appendix A.1.

THEOREM 6 *Choose positive integers a and u such that $u \geq 2$. Then there are sets X, Y such that $|Y| = u$, and there is a set $\mathcal{L} \subseteq (X \times Y)^*$ such that* $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) \leq a$*, and*

$$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \geq \begin{cases} 2^a - 1 & \text{if } |Y| \geq 2^a \\ \frac{|Y|}{3}\left(1 + \frac{\ln 2}{2}(a - \log_2 |Y|)\right) & \text{otherwise.} \end{cases}$$

3.4. The MBρ and MBQρ models

As a natural variant of the MB and MBQ models we consider the MBρ and MBQρ models where the response to the learner is $\rho_t \in \{\mathrm{TRUE}, \mathrm{FALSE}\}$ (instead of $\overline{y}_t \in Y$) indicating whether $\hat{y}_t = y_t$ or $\hat{y}_t \neq y_t$. A prediction $\hat{y}_t$ is a mistake if $\hat{y}_t \neq y_t$ and we measure the performance $M_\rho(\mathcal{L}, A)$ of an algorithm A for learning $\mathcal{L}$ in the MBQρ model by the maximum, over $\sigma \in \mathcal{L}$, of the number of mistakes and queries of A when learning σ. We define $\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})$ to be the minimum of $M_\rho(\mathcal{L}, A)$ over all algorithms A, and $\mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L})$ to be the minimum of $M_\rho(\mathcal{L}, A)$ over algorithms A which do not ask queries.

For the relationship between the MBρ and MBQρ models we get a similar but slightly weaker result than for the MB and MBQ models and we show that this result is close to best possible.

THEOREM 7 *For any sets X and Y for which $|Y| \geq 3$, and any $\mathcal{L} \subseteq (X \times Y)^*$*

$$\mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L}) \leq (|Y| \ln |Y|)\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}) + 130(|Y| \ln \ln |Y|)\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}).$$

THEOREM 8 *Choose positive integers a and u such that $u \geq 2981$. Then there are sets X, Y such that $|Y| = u$, and there is a set $\mathcal{L} \subseteq (X \times Y)^*$ such that* $\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}) = 2a + \lceil 2 \log_2 |Y| \rceil$*, and*

$$\mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L}) \geq a\lfloor |Y| \ln |Y| / 4 \rfloor.$$

The proof of Theorem 8 is given in Appendix A.2.

The proof of Theorem 7 is similar to the proof of Theorem 5. The main difference is that in the proof of Theorem 7, the copies which didn't predict $\hat{y}_t$ are split into two copies each, one which is told that its prediction was correct, and another that is told its prediction was not.

Proof of Theorem 7: Choose an optimal MBQρ algorithm $A^{\mathrm{MBQ}\rho}$ and consider the MBρ algorithm $A^{\mathrm{MB}\rho}$ which uses $A^{\mathrm{MBQ}\rho}$ as a subroutine defined in Figure 3 and set $\gamma = \frac{1}{|Y| \ln |Y|}$.

The key difference between $A^{\mathrm{MB}\rho}$ and A^{MB} is in the update after a mistake. Loosely speaking, when $A^{\mathrm{MB}\rho}$ makes a mistake, reinforcement TRUE or FALSE must be given to all copies of $A^{\mathrm{MBQ}\rho}$. Those copies that we do not know whether they made a mistake are split into two copies, one which receives the reinforcement that it made a mistake, and one which receives the reinforcement that it did not.

Notation:

Maintains a set of copies $A_i^{\mathrm{MBQ}\rho}$ of $A^{\mathrm{MBQ}\rho}$ where each copy corresponds to a subset $\mathcal{L}_i \subseteq \mathcal{L}$ which denotes the current information of $A_i^{\mathrm{MBQ}\rho}$ about the target sequence σ. Each copy maintains its number of queries and mistakes denoted by $q(i)$ and $m(i)$.

The weight of a copy $A_i^{\mathrm{MBQ}\rho}$ is calculated as $w_i = 2^{-q(i)}\gamma^{m(i)}$.

We assume that a copy $A_i^{\mathrm{MBQ}\rho}$ terminates if $q(i)+m(i) > \mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})$.

Initialization:

Initially there is only a single copy $A_1^{\mathrm{MBQ}\rho}$ with $\mathcal{L}_1 = \mathcal{L}$ and $q(1) = m(1) = 0$.

Simulating queries:

As long as there is a copy $A_i^{\mathrm{MBQ}\rho}$ which wants to ask a yes-no question this copy is duplicated giving a copy $A_j^{\mathrm{MBQ}\rho}$ and the answer YES is given to copy $A_i^{\mathrm{MBQ}\rho}$ and the answer NO is given to copy $A_j^{\mathrm{MBQ}\rho}$.

Making a prediction:

If no copy wants to ask a yes-no question x_t is received from the environment and the prediction

$$\hat{y}_t := \mathrm{argmax}_{y \in Y} \sum_{i: A_i^{\mathrm{MBQ}\rho}(x_t) = y} w_i$$

is calculated as the value with the highest weight.

Update when $\hat{y}_t \neq y_t$:

If the prediction was wrong then all copies with $A_i^{\mathrm{MBQ}\rho}(x_t) = \hat{y}_t$ are told that they have made a mistake. Each copy $A_i^{\mathrm{MBQ}\rho}$ with $A_i^{\mathrm{MBQ}\rho}(x_t) \neq \hat{y}_t$ is duplicated giving a copy $A_j^{\mathrm{MBQ}\rho}$, the copy $A_i^{\mathrm{MBQ}\rho}$ is told that its prediction was correct, and the copy $A_j^{\mathrm{MBQ}\rho}$ is told that its prediction was wrong.

Update when $\hat{y}_t = y_t$:

If the prediction was correct then all copies with $A_i^{\mathrm{MBQ}\rho}(x_t) = \hat{y}_t$ are told that their prediction was correct, and all copies with $A_i^{\mathrm{MBQ}\rho}(x_t) \neq \hat{y}_t$ are told that their prediction was wrong.

The steps Simulating queries, Making a prediction, and Update are repeated as long as required.

Figure 3. Algorithm $A^{\mathrm{MB}\rho}$ from the proof of Theorem 7

Our proof proceeds by using $W = \sum_i 2^{-q(i)}\gamma^{m(i)}$ as a measure of progress. Initially W is 1, and W does not change when copies are duplicated and given both answers to "yes-no" questions during the simulation of queries.

Now choose some t. Obviously, if $\hat{y}_t$ is not a mistake, W only decreases after trial t, but we will ignore this decrease in our analysis. Then if $\hat{y}_t$ is a mistake, since each copy $A_i^{\mathrm{MBQ}\rho}$ for which $A_i^{\mathrm{MBQ}\rho}(x_t) \neq \hat{y}_t$ is split into two copies, one whose weight is multiplied by γ, and the other whose weight remains the same, and all copies for which $A_i^{\mathrm{MBQ}\rho}(x_t) = \hat{y}_t$ have their weights multiplied by γ, we have

$$\begin{aligned} W_{\mathrm{new}} &= \sum_i 2^{-q_{\mathrm{new}}(i)}\gamma^{m_{\mathrm{new}}(i)} \\ &= \sum_{i:A_i^{\mathrm{MBQ}\rho}(x_t)\neq\hat{y}_t} (1+\gamma)2^{-q_{\mathrm{old}}(i)}\gamma^{m_{\mathrm{old}}(i)} + \sum_{i:A_i^{\mathrm{MBQ}\rho}(x_t)=\hat{y}_t} \gamma 2^{-q_{\mathrm{old}}(i)}\gamma^{m_{\mathrm{old}}(i)} \\ &\leq (1+\gamma)(1-1/|Y|)W_{\mathrm{old}} + \gamma W_{\mathrm{old}}/|Y| \\ &= W_{\mathrm{old}}(1 + 1/|Y|\ln|Y| - 1/|Y|). \end{aligned}$$

By induction, after $A^{\mathrm{MB}\rho}$ has made m mistakes, we have

$$W \leq \left(1 + \frac{1}{|Y|\ln|Y|} - \frac{1}{|Y|}\right)^m \leq \exp\left(-\left(1 - \frac{1}{\ln|Y|}\right)\frac{m}{|Y|}\right). \tag{7}$$

Also by induction, at any time during the execution of $A^{\mathrm{MB}\rho}$, there is a special copy $A_s^{\mathrm{MBQ}\rho}$ with $q(s) + m(s) \leq \mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})$. Then $W \geq 2^{-q(s)}\gamma^{m(s)} \geq \gamma^{\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})}$, since $\gamma \leq 1/2$. Combining this with (7), we get

$$\exp\left(-\left(1 - \frac{1}{\ln|Y|}\right)\frac{m}{|Y|}\right) \geq \gamma^{\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L})},$$

and solving for m and substituting the value of γ yields

$$\begin{aligned} m &\leq \left(\frac{|Y|\ln(|Y|\ln|Y|)}{1 - \frac{1}{\ln|Y|}}\right)\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}) \\ &= (|Y|\ln|Y| + |Y|\ln\ln|Y|)\left(1 + \frac{1}{\ln|Y| - 1}\right)\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}) \\ &\leq |Y|\ln|Y|\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}) + 130|Y|\ln\ln|Y|\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}), \end{aligned}$$

since $|Y| \geq 3$. □

3.5. Relationship between MB,MBQ and MBρ,MBQρ models

As mentioned before the models are equivalent if $|Y| = 2$ since the correct value y_t can be immediately deduced from the response $\overline{y}_t$ or ρ_t. In this case Theorem 4 gives the better bound for the relationship between MBρ and MBQρ model.

For any Y it holds that

$$\mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L}) \leq \mathrm{opt}_{\mathrm{MB}}(\mathcal{L}), \tag{8}$$

$$\mathrm{opt}_{\mathrm{MBQ}\rho}(\mathcal{L}) \leq \mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}), \tag{9}$$

since any MB or MBQ algorithm can be transformed into an MBρ or MBQρ algorithm, respectively, by translating a response ρ_t = FALSE into $\overline{y}_t = \hat{y}_t$ and ρ_t = TRUE into some $\overline{y}_t \neq \hat{y}_t$. That the converse of equation (9) is not true follows from Theorem 8 together with Theorem 4 and equation (8). That the converse of equation (8) is not true follows from a similar proof as for Theorem 8.

The converse of equation (8) does hold for sets of sequences $\mathcal{L}_F \subseteq (X \times Y)^*$ derived from classes F of functions from X to Y: if $\mathcal{L}_F$ is the set of all sequences $\langle(x_t, y_t)\rangle_t$ such that there is an $f \in F$ with $y_t = f(x_t)$ for all t then

$$\mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L}_F) = \mathrm{opt}_{\mathrm{MB}}(\mathcal{L}_F).$$

This follows from the fact that the maximum number of mistakes of an optimal MBρ algorithm for $\mathcal{L}_F$ does not increase if it is made to ignore trials where it predicted correctly.[8] Then such an algorithm can be used in the MB model by ignoring trials with $\overline{y}_t \neq \hat{y}_t$. We also conjecture that for $\mathcal{L}_F$ the converse of equation (9) holds but we were unable to prove that.

4. Applications of the general results

In this section we describe applications of the general results of the previous section. These applications are obtained by applying Theorem 4 or Theorem 5 to particular sets $\mathcal{L}$. Essentially we will show that all models considered in Section 1 are special cases of the MB and MBQ model, respectively.

4.1. The usefulness of few membership queries

First note, that a membership query is a special case of a yes-no question; i.e., for any class F of functions from X to $\{0,1\}$ we have $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}_F) \leq \mathrm{opt}_{\mathrm{memb}}(F)$.[9] Furthermore, when learning $\mathcal{L}_F$, the MB model is equivalent to the standard mistake-bound model so that $\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}_F) = \mathrm{opt}_{\mathrm{stand}}(F)$. Thus, modulo a small additive constant, Theorem 1 is a special case of Theorem 4. By examining the proof of Theorem 1 more closely, we may draw conclusions regarding the usefulness of polylogarithmically many membership queries in generating *computationally* efficient algorithms.

THEOREM 9 *Choose $X, F \subseteq \{0,1\}^X$. Then if there is an algorithm A^{memb} that takes at most T time between trials to learn F, and A^{memb} asks at most q membership queries, then there is an efficient algorithm A^{stand} for learning F that makes no membership queries and requires $O(2^q T)$ time between trials.*

Proof: We construct A^{stand} from A^{memb} as in the proof of Theorem 1, except with the following change: Any copy of A^{memb} that asks more than q membership queries is terminated. This does not affect the proof of Theorem 1 since A^{memb} asks at most q membership queries when learning a function from F.

Since the time required by A^{stand} to make a prediction is bounded by the number of copies A_i^{memb} times the time for A^{memb} to make a prediction, all that needs to be shown is that the number of copies maintained by A^{stand} never exceeds 2^q.

To see this, it is useful to view the copies A_i^{memb} as the leaves of a binary tree as discussed after the proof of Theorem 1. Since a node has two children only if it corresponds to a membership query and since there are at most q such nodes on any path from the root to a leaf, the number of leaves is bounded by 2^q.

□

4.2. Function learning with weak and strong reinforcement

Here we consider two generalizations of the standard mistake-bound model to functions with range possibly larger than two that were previously studied in [2]. Choose some set X, a finite set Y of at least two elements, and a class F of functions from X to Y.

We begin with the *weak reinforcement model.* Here learning also proceeds in trials, where in the tth trial, the learner (a) receives $x_t \in X$ from the environment, (b) outputs a prediction $\hat{y}_t \in Y$, (c) gets a response true or false indicating whether $\hat{y}_t = f(x_t)$ or not where $f \in F$ is the function to be learned. For a learning algorithm A for F let $\mathrm{M}_{\mathrm{weak}}(A, F)$ be the maximum number of mistakes of A when learning a function in F with weak reinforcement, and let $\mathrm{opt}_{\mathrm{weak}}(F) = \min_A \mathrm{M}_{\mathrm{weak}}(A, F)$. Note that the weak reinforcement model is simply the MBρ model for learning $\mathcal{L}_F$.

Next, we define the *strong reinforcement model.* Here again learning proceeds in trials. In the tth trial, the learner (a) receives $x_t \in X$ from the environment, (b) outputs a prediction $\hat{y}_t \in Y$, (c) discovers $y_t = f(x_t)$. For a learning algorithm A let $\mathrm{M}_{\mathrm{strong}}(A, F)$ be the maximum number of mistakes of A when learning a function in F with strong reinforcement, and let $\mathrm{opt}_{\mathrm{strong}}(F) = \min_A \mathrm{M}_{\mathrm{strong}}(A, F)$. The following result bounds the relative strength of strong reinforcement.

THEOREM 10 *For any set F of functions from X to Y,*

$$\mathrm{opt}_{\mathrm{weak}}(F) \leq 1.39|Y|(\lceil 1 + \log_2(|Y| - 1)\rceil \mathrm{opt}_{\mathrm{strong}}(F) + 2).$$

Proof: We show that an MBQ algorithm can simulate an algorithm which receives strong reinforcement: the MBQ algorithm predicts with the strong reinforcement algorithm and after a mistake it determines y_t by asking $\log_2 \lceil |Y| - 1 \rceil$ yes-no questions. Thus $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}_F) \leq \lceil 1 + \log_2(|Y| - 1)\rceil \mathrm{opt}_{\mathrm{strong}}(F)$. Since $\mathrm{opt}_{\mathrm{weak}}(F) = \mathrm{opt}_{\mathrm{MB}\rho}(\mathcal{L}_F) \leq \mathrm{opt}_{\mathrm{MB}}(\mathcal{L}_F)$ (by equation (8)) the theorem follows from Theorem 4. □

The following trivial lower bound shows that the above cannot be improved by more than an $O(\log |Y|)$ factor.

THEOREM 11 *For each positive integer a, and each integer $u \geq 2$, there is a set X, a set Y of u elements, and a set F of functions from X to Y such that* $\mathrm{opt}_{\mathrm{strong}}(F) = a$ *and*

$$\mathrm{opt}_{\mathrm{weak}}(F) \geq (|Y| - 1)\mathrm{opt}_{\mathrm{strong}}(F).$$

Proof: Choose a and u. Consider the set F of all functions from $\{1, ..., a\}$ to $\{1, ..., u\}$.

Trivially, $\mathrm{opt}_{\mathrm{strong}}(F)$ is a, since, with strong reinforcement an algorithm never need make a mistake on the same element of the domain twice.

To see that $\mathrm{opt}_{\mathrm{weak}}(F) \geq (|Y| - 1)a$, consider an adversary that first sets $x_1 = \cdots = x_{|Y|-1} = 1$, and tells the algorithm that all its predictions are wrong, then sets $x_{|Y|} = \cdots = x_{2(|Y|-1)} = 2$, and so on. Since the algorithm makes at most $|Y| - 1$ predictions on each element of the domain, there is some function from $\{1, ..., a\}$ to $\{1, ..., u\}$ consistent with the adversary's responses. This completes the proof.

□

4.3. Agnostic learning

In the agnostic learning model the learner again has to learn a function from X to $\{0, 1\}$ from some class F on-line, but some of the reinforcements given to the learner might be noisy. In the tth trial, the learner (a) receives $x_t \in X$ from the environment, (b) outputs a prediction $\hat{y}_t \in \{0, 1\}$, (c) discovers $y_t \in \{0, 1\}$. If $\hat{y}_y \neq y_t$ the learner has made a mistake. Denote by $M(A, F, \eta)$ the maximum number of mistakes of a learning algorithm A when the reinforcements y_t are such that there is an $f \in F$ with $|\{t : f(x_t) \neq y_t\}| \leq \eta$, i.e. at most η reinforcements are noisy. Finally, let $\mathrm{opt}_{\mathrm{agn}}(F, \eta) = \min_A M(A, F, \eta)$. We have the following result.

THEOREM 12 *For all sets X, for all sets F of functions from X to $\{0, 1\}$, for all nonnegative integers η, and for all $0 < \epsilon \leq 1/20$,*

$$\begin{aligned} \mathrm{opt}_{\mathrm{agn}}(F, \eta) &\leq 4.82(\mathrm{opt}_{\mathrm{agn}}(F, 0) + \eta) + 1 \\ \mathrm{opt}_{\mathrm{agn}}(F, \eta) &\leq \tfrac{4}{\epsilon}(\ln \tfrac{1}{\epsilon})\mathrm{opt}_{\mathrm{agn}}(F, 0) + (2 + \tfrac{5}{2}\epsilon)\eta. \end{aligned}$$

Proof: We show that an MBQ algorithm can simulate an algorithm for the standard mistake-bound model without noise. Let $\mathcal{L}_{F,\eta} \subseteq (X \times \{0, 1\})^*$ consist of those sequences $\langle (x_t, y_t) \rangle_t$ such that there exists an $f \in F$ with $|\{t : f(x_t) \neq y_t\}| \leq \eta$ (there may be many such f for the same sequence). Note that $\mathcal{L}_{F,\eta}$ is closed under subsequences. Now let A be a standard mistake-bound algorithm for F. We construct an MBQ algorithm B for $\mathcal{L}_{F,\eta}$ as follows. Algorithm B maintains a list of correct reinforcements $z_t \in \{0, 1\}$. In each trial it predicts with algorithm A. If $\hat{y}_t = y_t$ both algorithms ignore this trial. If $\hat{y}_t \neq y_t$ algorithm B determines if the reinforcement was noisy by asking "Is $\sigma = \langle (x_\tau, y_\tau) \rangle_\tau$ such that there is an $f \in F$ with $f(x_\tau) = z_\tau$ for $\tau < t$, $f(x_t) = y_t$, and $|\{\tau : f(x_\tau) \neq y_\tau\}| \leq \eta$?" (It is worth emphasizing at this point that this question is about the sequence σ of examples.) If the answer is YES algorithm B sets $z_t = y_t$, otherwise it sets $z_t = 1 - y_t$, and it passes z_t to algorithm A. By induction, there is an $f \in F$ such that for all trials t, $f(x_t) = z_t$, and $|\{t : f(x_t) \neq y_t\}| \leq \eta$. Therefore the number of trials t on which $\hat{y}_t \neq z_t$ is at most $\mathrm{opt}_{\mathrm{stand}}(F) = \mathrm{opt}_{\mathrm{agn}}(F, 0)$ and the number of trials on which $\hat{y}_t \neq y_t$ is at most $\mathrm{opt}_{\mathrm{agn}}(F, 0) + \eta$. Finally, since B asks a question after each mistake, we get $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}_{F,\eta}) \leq 2(\mathrm{opt}_{\mathrm{agn}}(F, 0) + \eta)$. Since

$\mathrm{opt}_{\mathrm{agn}}(F,\eta) = \mathrm{opt}_{\mathrm{MB}}(\mathcal{L}_{F,\eta})$, the first bound of Theorem 4 gives the first bound of the theorem.

To get the second bound, note that at most η of B's questions are answered NO and at most $\mathrm{opt}_{\mathrm{agn}}(F,0)$ are answered YES. Applying Theorem 5 with $\alpha = \epsilon^2$, $\beta = 1-\epsilon^2$, $\gamma = 1-\epsilon$, gives the result after some calculations. □

The proofs of Theorem 5 and Theorem 12 can be modified to obtain the same bounds for agnostically learning sets of functions from an arbitrary set X to an arbitrary set Y with strong reinforcement.

For comparison, we give the following lower bound of Littlestone and Warmuth.

THEOREM 13 ([15]) *For any X, and any set F of at least two functions from X to $\{0,1\}$,*

$$\mathrm{opt}_{\mathrm{agn}}(F,\eta) \geq \mathrm{opt}_{\mathrm{agn}}(F,0) + 2\eta.$$

4.4. Closure Results

Now we return to the standard mistake-bound model. Choose an integer $k \geq 2$ and a set X. If $f_1, ..., f_k$ are functions from X to $\{0,1\}$, and g is a function from $\{0,1\}^k$ to $\{0,1\}$, then define the function $g(f_1,...,f_k)$ from X to $\{0,1\}$ by

$$(g(f_1,...,f_k))(x) = g(f_1(x),...,f_k(x)).$$

For any fixed $g : \{0,1\}^k \rightarrow \{0,1\}$, and any sets $F_1,...,F_k$ of functions from x to $\{0,1\}$, define

$$\mathrm{COMPOSE}(F_1,...,F_k,g) = \{g(f_1,...,f_k) : f_1 \in F_1,...,f_k \in F_k\}$$

and for any set G of functions from $\{0,1\}^k$ to $\{0,1\}$, let

$$\mathrm{COMPOSE}(F_1,...,F_k,G) = \cup_{g\in G}\mathrm{COMPOSE}(F_1,...,F_k,g).$$

THEOREM 14 *For any sets $F_1,...,F_k$ of functions from X to $\{0,1\}$, for any function g from $\{0,1\}^k$ to $\{0,1\}$, and for any set G of such functions*

$$\begin{aligned} &\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1,...,F_k,g)) \\ &\quad \leq\ 2.41\lceil 1+\log_2 k\rceil \sum_{i=1}^{k} \mathrm{opt}_{\mathrm{stand}}(F_i) + 1, \\ &\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1,...,F_k,G)) \\ &\quad \leq\ 2.41\lceil 1+\log_2(k+1)\rceil \left(\mathrm{opt}_{\mathrm{stand}}(G) + \sum_{i=1}^{k}\mathrm{opt}_{\mathrm{stand}}(F_i)\right) + 1. \end{aligned}$$

Proof: We begin with the first bound. Suppose, for a known g, functions $f_1 \in F_1,...,f_k \in F_k$ are unknown to the learner, who is trying to learn $g(f_1,...,f_k)$. A harder problem is to try to predict, for each trial t, the vector $(f_1(x_t), ..., f_k(x_t))$ in the weak reinforcement model above. This problem becomes easy, however, if after each mistake, the learner can determine a component of its prediction that was incorrect: The learner can then simply run separate algorithms for learning each of $f_1, ..., f_k$. Any time the master algorithm makes a mistake, it can make one of the subroutine algorithms make a mistake (all other subalgorithms ignore that trial), and therefore the number of mistakes made by the master algorithm is at most $\sum_{i=1}^{k} \mathrm{opt}_{\mathrm{stand}}(F_i)$ if optimal algorithms are used for the subalgorithms. Since an MBQ learner can determine a component of its prediction that was incorrect through $\lceil \log_2 k \rceil$ "yes-no" questions, an MBQ learner can obtain a performance guarantee of $(1+\lceil \log_2 k \rceil) \sum_{i=1}^{k} \mathrm{opt}_{\mathrm{stand}}(F_i)$. Applying the first bound of Theorem 4 then yields the first bound of this theorem.

For the second bound, we do the analogous thing, except using the value of

$$(f_1(x_t), ..., f_k(x_t), g(f_1(x_t), ..., f_k(x_t))).$$

Whenever the master algorithm makes a mistake it determines the least component of the prediction of the above which was incorrect through $\lceil \log_2(k+1) \rceil$ questions. If it was of an $f_i(x_t)$, it simulates for the corresponding subalgorithm the trial with x_t, the subalgorithm's prediction, and $f_i(x_t)$. If the only incorrect component of the prediction was of $g(f_1(x_t), ..., f_k(x_t))$ then the algorithm simulates for the subalgorithm learning g the trial consisting of $(f_1(x_t), ..., f_k(x_t))$, the subalgorithm's prediction, and $g(f_1(x_t), ..., f_k(x_t))$. Since such trials are only simulated when all predictions of $f_1(x_t), ..., f_k(x_t)$ are correct, the trials given to the algorithm for learning g are consistent with g. Continuing as in the previous paragraph yields the second bound. □

The following lower bound shows that Theorem 14 is within an $O(\log k)$ factor of optimal. The proof is given in Appendix A.3. From the proof one can also easily see that corollaries obtained by applying Theorem 14 with many natural concrete g are also within this $O(\log k)$ factor of optimal. (Of course, there are exceptions, e.g. $g \equiv 0$.)

THEOREM 15 *Choose an integer $k \geq 2$ and positive integers $a_1, ..., a_k$. Then there is a set X and sets $F_1, ..., F_k$ of functions from X to $\{0,1\}$ such that for all i, $\mathrm{opt}_{\mathrm{stand}}(F_i) = a_i$, and there is a $g : \{0,1\}^k \to \{0,1\}$ such that*

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, g)) \geq \sum_{i=1}^{k} \mathrm{opt}_{\mathrm{stand}}(F_i).$$

Choose a positive integer $a_{k+1} \leq 2^k$. Then there is a set X and sets $F_1, ..., F_k$ of functions from X to $\{0,1\}$ such that for all i, $\mathrm{opt}_{\mathrm{stand}}(F_i) = a_i$, and there is a set G of functions from $\{0,1\}^k$ to $\{0,1\}$ such that $\mathrm{opt}_{\mathrm{stand}}(G) = a_{k+1}$ and

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, G)) \geq \frac{1}{2}\left(\mathrm{opt}_{\mathrm{stand}}(G) + \sum_{i=1}^{k} \mathrm{opt}_{\mathrm{stand}}(F_i)\right).$$

The restriction $a_{k+1} \leq 2^k$ is needed since for any set G of functions from $\{0,1\}^k$ to $\{0,1\}$, $\mathrm{opt}_{\mathrm{stand}}(G) \leq 2^k$.

4.5. Mistake bounds with delayed, ambiguous reinforcement

Finally, we formally define what we call the *delayed, ambiguous reinforcement model.*

In this model the learner again has to learn a function f from a class F of functions from X to $\{0,1\}$, but it receives no immediate reinforcement. Learning proceeds in *rounds*, where in each round t the learner is given $x_{t,1} \in X$, outputs a prediction $\hat{y}_{t,1}$, $\ldots$, is given $x_{t,r} \in X$, outputs a prediction $\hat{y}_{t,r}$, then receives reinforcement FALSE or TRUE indicating whether any of the predictions $\hat{y}_{t,1}, \ldots, \hat{y}_{t,r}$ was incorrect, i.e. the reinforcement is FALSE iff $\hat{y}_{t,i} \neq f(x_{t,i})$ for any $i \in \{1, \ldots, r\}$. Denote by $\mathrm{M}_{\mathrm{amb},r}(A,F)$ the maximum number of false rounds of an algorithm A when learning a function $f \in F$ and let $\mathrm{opt}_{\mathrm{amb,r}}(F) = \min_A \mathrm{M}_{\mathrm{amb},r}(A,F)$. Note that $\mathrm{opt}_{\mathrm{amb,1}}(F) = \mathrm{opt}_{\mathrm{stand}}(F)$.

Note that before the algorithm outputs $\hat{y}_{t,i}$, it does not know the values of $x_{t,j}$ for $j > i$. A natural question is if knowing these values could help the algorithm. If this were not the case, then learning in the r-trial delayed ambiguous feedback model would reduce to learning in the weak reinforcement model as follows. For some set X and some set F of functions from X to $\{0,1\}$, we might set $X' = X^r$ and define $F' = \mathrm{CART}_r(F)$ to be all functions f' from X' to $\{0,1\}^r$ such that there exists $f \in F$ for which for all $(x_1, ..., x_r) \in X^r$, $f'(x_1, ..., x_r) = (f(x_1), ..., f(x_r))$. If it didn't help the algorithm to know $x_{t,1}, ..., x_{t,r}$, then we could assume without loss of generality that $x_{t,1}, ..., x_{t,r}$ were all given at the beginning of the round, and it would be the case that $\mathrm{opt}_{\mathrm{amb,r}}(F) = \mathrm{opt}_{\mathrm{weak}}(\mathrm{CART}_r(F))$. The following theorem shows that this is not the case. The proof is given in Appendix A.5.

THEOREM 16 *There exists a set X and a set F of functions from X to $\{0,1\}$ such that*

$$\mathrm{opt}_{\mathrm{weak}}(\mathrm{CART}_2(F)) < \mathrm{opt}_{\mathrm{amb},2}(F).$$

The following result bounds the relative difficulty of learning with ambiguous reinforcement.

THEOREM 17 *For any set F of functions from X to $\{0,1\}$,*

$$\mathrm{opt}_{\mathrm{amb,r}}(F) \leq 2(\ln 2r) \cdot 2^r \cdot \mathrm{opt}_{\mathrm{amb,1}}(F).$$

Proof: If, after each round in which it makes a mistake, a learning algorithm is told of a trial during that round in which its prediction was incorrect, then by ignoring the other trials of those rounds, an algorithm can make at most $\mathrm{opt}_{\mathrm{amb,1}}(F)$ mistakes. Similar to the proof of Theorem 5 knowledge of the incorrect trials can be simulated by splitting into r copies, each given one of the trials as a mistake. Since the master algorithm can choose its predictions such that at least a fraction of $1/2^r$ of the total weight predicted the same on all r trials of a round the bound follows analogously as in the proof of Theorem 5. □

Finally, we describe a polynomially related lower bound. The proof is given in Appendix A.4.

THEOREM 18 *For any integers $a, r \geq 1$, there is a class F of functions such that* $\mathrm{opt}_{\mathrm{amb},1}(F) = a$ *and*

$$\mathrm{opt}_{\mathrm{amb,r}}(F) \geq \min\left\{\frac{1}{2r}(2^r - 1)\mathrm{opt}_{\mathrm{amb},1}(F), \left(\sum_{i=0}^{\mathrm{opt}_{\mathrm{amb},1}(F)} \binom{r}{i}\right) - 1\right\}.$$

5. Conclusions and future directions

In this paper, we have presented a new method for simulating on-line learning algorithms which have access to queries by algorithms that have no such access, and presented applications of this simulation concerning structural questions about several natural on-line learning models.

An interesting open question is to try to find a more efficient simulation, in particular with respect to computational requirements. Significant progress in this direction would result in a strengthening of Theorem 9. A more computationally efficient simulation which achieved a worse mistake-bound would be potentially interesting.

An anonymous referee asked whether arbitrary boolean queries are significantly more powerful than membership queries for learning $\{0,1\}$-valued functions.

Finally, many of the bounds of Section 4 have small gaps that it would be nice to remove. Furthermore, it would be interesting to try to find computationally efficient algorithms for learning in the models described in Section 4.

Acknowledgments

We thank Wolfgang Maass for his advice and encouragement, and for his comments on an earlier draft of this paper. We are also grateful to Manfred Warmuth for posing the problem solved in Theorem 12. We thank Nick Littlestone for discussions relating to the delayed ambiguous reinforcement model treated in Theorem 17. Finally, we are very grateful to two anonymous referees for their thoughtful reviews.

Appendix A

A.1. Proof of Theorem 6

First, we restate Theorem 6 for easy reference: *Choose positive integers a and u such that $u \geq 2$. Then there are sets X, Y such that $|Y| = u$, and there is a set $\mathcal{L} \subseteq (X \times Y)^*$ such that* $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) = a$, *and*

$$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \geq \begin{cases} 2^{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})} - 1 & \text{if } |Y| \geq 2^{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})} \\ \frac{|Y|}{3}\left(1 + \frac{1}{2\ln 2}(\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) - \log_2 |Y|)\right) & \text{otherwise.} \end{cases}$$

This theorem is proved through a pair of lemmas. For any positive integers u, v, let $\mathrm{SVAR}_{u,v}$ be the set of all functions $f : \{1, ..., u\}^v \to \{1, ..., u\}$ such that there exists i for which for all $\vec{x} \in \{1, ..., u\}^v$, $f(\vec{x}) = x_i$.

Lemma 1 *For any nonnegative integer a and any positive integer u,*

$$\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}_{\mathrm{SVAR}_{u,2^a}}) \leq a.$$

Proof: There are at most 2^a elements of $\mathrm{SVAR}_{u,2^a}$. Therefore, by asking for the bits of the index of the function mapping the x_t's to the y_t's before the first trial (a questions), this MBQ algorithm never makes a mistake. □

Lemma 2 *For any positive integer v and any positive integer $u \geq 2$,*

$$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}_{\mathrm{SVAR}_{u,v}}) \geq \begin{cases} v - 1 & \text{if } v \leq u \\ \frac{u}{3}(1 + \frac{1}{2}\ln\frac{v}{u}) & \text{otherwise.} \end{cases}$$

Proof: If $u = 2$, then the theorem follows from the fact [14] that $\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}_{\mathrm{SVAR}_{2,v}}) = \lfloor \log_2 v \rfloor$. Assume from here on that $u > 2$.

As a first case, assume $v \leq u$. Choose an MB algorithm A. Let $\hat{y}_1, ..., \hat{y}_{v-1}$ be the predictions made by A when given $x_1 = \cdots = x_{v-1} = (1, 2, ..., v)$ on-line with response $\overline{y}_t = \hat{y}_t$ at the end of each trial. Choose $y \in (\{1, ..., u\} - \{\hat{y}_1, ..., \hat{y}_{v-1}\})$. Thus if $\sigma = ((x_1, y), ..., (x_{v-1}, y))$, A makes $v - 1$ mistakes on σ, and $\sigma \in \mathcal{L}_{\mathrm{SVAR}_{u,v}}$.

Now, assume $v > u$. Construct a sequence $\sigma \in (X \times Y)^*$ using an adversary as follows. The adversary operates in two stages. The adversary maintains a list of functions in $\mathrm{SVAR}_{u,v}$ which map previous x_t's to y_t's (or equivalently a list of the coordinates defining those functions). Let l_t be the number of elements in this list before the tth trial ($l_1 = v$). The first stage ends when $l_t < u$. During the first stage, on each trial, the adversary divides up the l_t remaining coordinates into u nearly equal sized groups, each consisting of either $\lceil l_t/u \rceil$ or $\lfloor l_t/u \rfloor$ members. Then x_t is chosen so that the coordinates in the first group take the value 1, the coordinates in the second group take the value 2, and so on. Whatever the algorithm's prediction it is given same value as the response (resulting in a mistake), and the "live" coordinates which evaluated to the algorithm's prediction are no longer so.

During the first stage, we have $l_1 = v$, and

$$\begin{aligned} l_{t+1} &\geq l_t - \lceil l_t/u \rceil \qquad (A.1) \\ &\geq l_t - \frac{2l_t}{u} \\ &= l_t(1 - 2/u). \end{aligned}$$

Thus, by induction, for any trial t in the first stage $l_{t+1} \geq v(1-2/u)^t$. Thus, the number of trials (and therefore mistakes) in the first stage is at least

$$\begin{aligned}
&\max\{q : v(1-2/u)^{q-1} \geq u\} \\
&\geq \max\{q : v \exp\left(\frac{-2(q-1)/u}{1-2/u}\right) \geq u\} \quad \text{since } u > 2 \\
&\geq (u/2-1)\ln\frac{v}{u}. \qquad \text{(A.2)}
\end{aligned}$$

The number of "live" coordinates $l_{t'}$ before the first trial t' of the second stage is at most u, so the adversary may force the algorithm to make $l_{t'}-1$ mistakes similarly as in the first paragraph of the proof. We claim that $l_{t'} = u-1$ which is seen from (A.1): If $l_{t'-1} = u$ then $l_{t'} = u-1$. If $l_{t'-1} \geq u+1$ then $l_{t'} \geq (u+1)(1-2/u) \geq u-1-2/3$ since $u \geq 3$.

Thus, the number of "live" coordinates prior to the onset of stage two is at least $u-1$, and therefore there are at least $u-2$ mistakes during the second stage. Combining with the lower bound of (A.2) on the number of mistakes during the first stage, we arrive at a total of

$$\begin{aligned}
(u/2-1)\ln\frac{v}{u} + (u-2) &= (u-2)\left(1+\frac{1}{2}\ln\frac{v}{u}\right) \\
&\geq \frac{u}{3}\left(1+\frac{1}{2}\ln\frac{v}{u}\right)
\end{aligned}$$

completing the proof. □

Theorem 6 is an immediate consequence of Lemma 1 and Lemma 2.

A.2. Proof of Theorem 8

We restate Theorem 8 for reference: *Choose positive integers a and u such that $u \geq 2981$. Then there are sets X, Y such that $|Y| = u$, and there is a set $\mathcal{L} \subseteq (X \times Y)^*$ such that* $\mathrm{opt}_{\mathrm{MBQ}_\rho}(\mathcal{L}) = 2a + \lceil 2\log_2|Y|\rceil$, *and*

$$\mathrm{opt}_{\mathrm{MB}_\rho}(\mathcal{L}) \geq a\lfloor |Y| \ln |Y|/4 \rfloor.$$

Proof: For any positive integers u, v, let $\mathrm{SVAR}_{u,v}$ be the set of all functions $f_i : \{1,...,u\}^v \to \{1,...,u\}$ with $f_i(\vec{x}) = x_i$, $i \in \{1,\ldots,v\}$. Informally, this is the set of all functions which "pick out some component" of their input.

Now let $Y = \{1,\ldots,u\}$, $v = u^2$, and $X = Y^v$. The set $\mathcal{L}$ consists of sequences of length ar, $r = \lfloor |Y| \ln |Y|/4 \rfloor$, where each sequence

$$\sigma = \langle (x_{1,1}, y_{1,1}), \ldots, (x_{1,r}, y_{1,r}), \ldots, (x_{a,1}, y_{a,1}), \ldots, (x_{a,r}, y_{a,r}) \rangle \in \mathcal{L}$$

consists of a subsequences of length r. Each subsequence is consistent with one of the functions in $\mathrm{SVAR}_{u,v}$ except for two elements of the subsequence, i.e. there are $i_1, \ldots, i_a \in \{1,\ldots,v\}$, $s_1, \ldots, s_a \in \{1,\ldots,u\}$, and $t_1, \ldots, t_a \in \{u+1,\ldots,2u\}$

with $y_{\phi,\psi} = f_{i_\phi}(x_{\phi,\psi})$ for $\psi \notin \{s_\phi, t_\phi\}$ and $y_{\phi,\psi} \neq f_{i_\phi}(x_{\phi,\psi})$ for $\psi \in \{s_\phi, t_\phi\}$, $\phi \in \{1, \ldots, a\}$. Furthermore, s_ϕ and t_ϕ encode the function consistent with the next subsequence, i.e. $i_{\phi+1} = u \cdot (s_\phi - 1) + (t_\phi - r)$ for $\phi \in \{1, \ldots, a\}$ (assume $i_{a+1} = 1$). Observe that such a coding is possible since $r \geq 2u$ for $u \geq 2981$.

An MBQ algorithm can ask $\lceil \log_2 v \rceil$ yes-no questions to determine i_1. Then it will predict with f_{i_1} for the first subsequence. The elements for which it makes a mistake determine s_2 and t_2. Continuing this way the algorithm will make two mistakes for each subsequence which gives $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) \leq \lceil 2 \log_2 |Y| \rceil + 2a$.

To get a lower bound for any MB algorithm we define an adversary strategy. For each subsequence the adversary maintains a list of functions in $\mathrm{SVAR}_{u,v}$ (or equivalently coordinates) which are consistent with the previous trials of this subsequence. Let l_τ be the number of elements in this list before processing the τth element of the subsequence ($l_1 = v$). On each trial the adversary divides the l_τ remaining coordinates into u nearly equally sized groups, each consisting of either $\lceil l_\tau / u \rceil$ or $\lfloor l_\tau / u \rfloor$ members. Then x_τ is chosen so that the coordinates in the first group take the value 1, the coordinates in the second group take the value 2, and so on. Whatever the algorithm's prediction is it is given the reinforcement "false", and the "live" coordinates which evaluated to the algorithm's prediction are no longer so, yielding by induction that

$$l_{\tau+1} \geq l_\tau - \lceil l_\tau / u \rceil \geq l_\tau - \frac{2 l_\tau}{u} = l_\tau (1 - 2/u)$$

and

$$l_{r+1} \geq l_1 (1 - 2/u)^r \geq v \exp(-\ln |Y|) \geq u.$$

Thus for all $\phi = 1, \ldots, a$ there is a function $f_{i_\phi} \in \mathrm{SVAR}_{u,v}$ which is consistent with the r trials of the subsequence.

Now we show that after all ar trials there is a sequence in $\mathcal{L}$ consistent with all the reinforcements given by the adversary. For $\phi = 1, \ldots, a$ let s_ϕ and t_ϕ be such that $i_{\phi+1} = u \cdot (s_\phi - 1) + (t_\phi - u)$. We set $y_{\phi,\psi} = f_{i_\phi}(x_{\phi,\psi})$ for $\psi \notin \{s_\phi, t_\phi\}$, y_{ϕ,s_ϕ} to a value different from $f_{i_\phi}(x_{\phi,s_\phi})$ and $\hat{y}_{\phi,s_\phi}$, and y_{ϕ,t_ϕ} to a value different from $f_{i_\phi}(x_{\phi,t_\phi})$ and $\hat{y}_{\phi,t_\phi}$. Since $u \geq 3$ this is always possible. Thus $\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \geq ar$.

□

Remark. By a more careful analysis the constants in the theorem can be improved. Furthermore, along the same line it can be shown that for positive integers a and u there are X, Y, and $\mathcal{L}$, such that $|Y| = u$, $\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L}) \leq a$, and

$$\mathrm{opt}_{\mathrm{MB}}(\mathcal{L}) \geq a|Y|(\ln |Y| + \ln \ln |Y| - C)$$

for $a \geq \log^3 |Y|$ and some constant C. Thus the upper bound in Theorem 7 has the correct constant at the first order term and the correct magnitude of the second order term.

A.3. Proof of Theorem 15

LEMMA 3 *Choose finite sets X_1 and X_2 such that $X_1 \subseteq X_2$, an integer a such that $a \leq |X_2|$, and a function f_1 from X_1 to $\{0,1\}$. Then there is a function f_2 from X_2 to $\{0,1\}$ such that for all $x \in X_1$, $f_1(x) = f_2(x)$, and there is a set F of functions from X_2 to $\{0,1\}$ such that $f_2 \in F$ and* $\mathrm{opt}_{\mathrm{stand}}(F) = a$.

Proof: Extend f_1 to f_2 arbitrarily. Trivially, $\mathrm{opt}_{\mathrm{stand}}(\{f_2\}) = 0$. Furthermore, if P is the set of all functions from X_2 to $\{0,1\}$, $\mathrm{opt}_{\mathrm{stand}}(P) = |X_2| \geq a$. Also, for any $G \subseteq P$ and any $g \in P$,

$$\mathrm{opt}_{\mathrm{stand}}(G) \leq \mathrm{opt}_{\mathrm{stand}}(G \cup \{g\}) \leq \mathrm{opt}_{\mathrm{stand}}(G) + 1.$$

Therefore, if we start with $F = \{f_2\}$ and add the elements of P to F one by one, $\mathrm{opt}_{\mathrm{stand}}(F)$ goes from being 0 to $|X_2|$, increasing by at most one each time we add an element to F. Since $a \leq |X_2|$, there must be a time when $\mathrm{opt}_{\mathrm{stand}}(F) = a$. □

Here is a restatement of Theorem 15: *Choose an integer $k \geq 2$ and positive integers $a_1, ..., a_k$. Then there is a set X and sets $F_1, ..., F_k$ of functions from X to $\{0,1\}$ such that for all i,* $\mathrm{opt}_{\mathrm{stand}}(F_i) = a_i$*, and there is a $g : \{0,1\}^k \to \{0,1\}$ such that*

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, g)) \geq \sum_{i=1}^{k} \mathrm{opt}_{\mathrm{stand}}(F_i).$$

Choose a positive integer $a_{k+1} \leq 2^k$. Then there is a set X and sets $F_1, ..., F_k$ of functions from X to $\{0,1\}$ such that for all i, $\mathrm{opt}_{\mathrm{stand}}(F_i) = a_i$*, and there is a set G of functions from $\{0,1\}^k$ to $\{0,1\}$ such that* $\mathrm{opt}_{\mathrm{stand}}(G) = a_{k+1}$ *and*

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, G)) \geq \frac{1}{2}\left(\mathrm{opt}_{\mathrm{stand}}(G) + \sum_{i=1}^{k} \mathrm{opt}_{\mathrm{stand}}(F_i)\right).$$

Proof: We begin with the first bound. Let $X_1 = \{1, ..., a_1\}$, $X_2 = \{a_1+1, ..., a_2\}$,..., $X_k = \{1 + \sum_{i=1}^{k-1} a_i, ..., \sum_{i=1}^{k} a_i\}$. Let $X = \cup_{i=1}^{k} X_i = \{1, ..., \sum_{i=1}^{k} a_i\}$. For each i, let F_i be the set of all functions from X to $\{0,1\}$ that are zero everywhere in $X - X_i$. Then for each i, $\mathrm{opt}_{\mathrm{stand}}(F_i) = |X_i| = a_i$. Let $g : \{0,1\}^k \to \{0,1\}$ evaluate to the disjunction of its arguments. That is $g(b_1, ..., b_k) = 1$ if and only if $1 \in \{b_1, ..., b_k\}$.

We claim that $\mathrm{COMPOSE}(F_1, ..., F_k, g)$ is the set of all functions from X to $\{0,1\}$. Choose a function f from X to $\{0,1\}$. For each i, let $f_i \in F_i$ be defined by

$$f_i(x) = \begin{cases} f(x) & \text{if } x \in X_i \\ 0 & \text{otherwise.} \end{cases}$$

Then, trivially, $f = g(f_1, ..., f_k)$. Since f was chosen arbitrarily, $\mathrm{COMPOSE}(F_1, ..., F_k, g)$ is the set of all functions from X to $\{0,1\}$, and therefore,

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, g)) = |X| = \sum_{i=1}^{k} a_i,$$

completing the proof of the first bound.

Now for the second bound. We will distinguish two cases, $a_{k+1} \geq \sum_{i=1}^{k} a_i$ and $a_{k+1} \leq \sum_{i=1}^{k} a_i$, proving that $\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, \ldots, F_k, G)) \geq a_{k+1}$ and $\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, \ldots, F_k, G)) \geq \sum_{i=1}^{k} a_i$, respectively.

Assume as the first case that $a_{k+1} \geq \sum_{i=1}^{k} a_i$. For each i, let $f_i : \{0,1\}^k \to \{0,1\}$ simply output the ith component of its argument, i.e. $f_i(\vec{x}) = x_i$. Let X' be a set containing all the elements of $\{0,1\}^k$ which has a total of at least $\max_i a_i$ elements. Apply Lemma 3 to obtain functions $f'_1, ..., f'_k$ from X' to $\{0,1\}$ and sets $F_1, ..., F_k$ of functions from X' to $\{0,1\}$ such that for all $i \leq k$,

- for all $\vec{x} \in \{0,1\}^k$, $f'_i(\vec{x}) = x_i$
- $f'_i \in F_i$
- $\mathrm{opt}_{\mathrm{stand}}(F_i) = a_i$.

Since for any $\vec{x} \in \{0,1\}^k$, $(f'_1(\vec{x}), ..., f'_k(\vec{x})) = \vec{x}$, even if a learning algorithm knows $f'_1, ..., f'_k$, learning $g(f'_1, ..., f'_k)$ is at least as hard as learning g. Therefore

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, G)) \geq \mathrm{opt}_{\mathrm{stand}}(G) = a_{k+1} \geq \frac{1}{2} \sum_{i=1}^{k+1} a_i,$$

completing the proof of the second bound in the case $a_{k+1} \geq \sum_{i=1}^{k} a_i$.

To establish the second bound in the case $a_{k+1} \leq \sum_{i=1}^{k} a_i$, again apply Lemma 3 to obtain a set G of functions from $\{0,1\}^k$ to $\{0,1\}$ such that G contains the function g_d computing the disjunction of its arguments and that $\mathrm{opt}_{\mathrm{stand}}(G) = a_{k+1}$. Using the argument for the first bound, there exist $F_1, ..., F_k$ such that

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, g_d)) \geq \sum_{i=1}^{k} a_i,$$

and therefore

$$\mathrm{opt}_{\mathrm{stand}}(\mathrm{COMPOSE}(F_1, ..., F_k, G)) \geq \sum_{i=1}^{k} a_i \geq \frac{1}{2} \sum_{i=1}^{k+1} a_i,$$

completing the proof. □

A.4. Proof of Theorem 18

Recall the statement of Theorem 18: *For any integers $a, r \geq 1$, there is a class F of functions such that* $\mathrm{opt}_{\mathrm{amb},1}(F) = a$ *and*

$$\mathrm{opt}_{\mathrm{amb,r}}(F) \geq \min\left\{\frac{1}{2r}(2^r-1)\mathrm{opt}_{\mathrm{amb,1}}(F), \left(\sum_{i=0}^{\mathrm{opt}_{\mathrm{amb,1}}(F)} \binom{r}{i}\right)-1\right\}.$$

Proof: The first term in the min holds in the case $a \geq r$. In this case, let F be the set of all functions from $\{1, ..., a\}$ to $\{0,1\}$. Trivially, $\mathrm{opt}_{\mathrm{amb,1}}(F) = a$. To show that $\mathrm{opt}_{\mathrm{amb,r}}(F) \geq \frac{1}{2r}(2^r - 1)a$, we construct an adversary to generate a hard sequence for any learner.

Choose a learning algorithm A. The adversary gives $x_{t,1} = 1, ..., x_{t,r} = r$ for $2^r - 1$ rounds, then gives $x_{t,1} = r+1, ..., x_{t,r} = 2r$ for $2^r - 1$ rounds, and does this $\lfloor \frac{a}{r} \rfloor$ times. It always answers FALSE. The total number of mistakes is

$$\left\lfloor \frac{a}{r} \right\rfloor (2^r - 1) \geq \frac{2^r - 1}{2r} a.$$

For each $\{(i-1)r+1, ..., ir\}$ there is some sequence of r elements of $\{0,1\}$ that was not guessed by A. If we define f to take on those values, then the resulting sequence is consistent with f.

When $a \leq r$, let F be the set of all functions from $\{1, ..., r\}$ to $\{0,1\}$ which map at most a elements to 1. Then $\mathrm{opt}_{\mathrm{amb,1}}(F) = a$, see e.g. [19]. The adversary sets $x_{t,1} = 1, ..., x_{t,r} = r$, for $t = 1, ..., \left(\sum_{i=0}^{a} \binom{r}{i}\right) - 1$. The reinforcement FALSE is given in all rounds. Again, for any algorithm, there must be some sequence of r predictions with at most a 1's that the algorithm didn't make on any of those rounds, and therefore there is a function in F consistent with all those rounds.

□

A.5. Proof of Theorem 16

We restate Theorem 16: *There exists a set X and a set F of functions from X to $\{0,1\}$ such that*

$$\mathrm{opt}_{\mathrm{weak}}(\mathrm{CART}_2(F)) < \mathrm{opt}_{\mathrm{amb,2}}(F).$$

Proof: Let $X = \{1,2,3\}$, and consider the set $F = \{f_1, ..., f_4\}$ of functions from X to $\{0,1\}$ defined in the following table.

x	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
1	0	0	1	1
2	0	0	0	1
3	0	1	1	1

First, we claim that $\mathrm{opt}_{\mathrm{amb,2}}(F) \geq 3$. To see this, imagine an adversary that sets $x_{1,1} = 1$. If the algorithm's prediction $\hat{y}_{1,1} = 1$, it sets $x_{1,2} = 2$, otherwise it sets $x_{1,2} = 3$. In either case the reinforcement for the first round is FALSE.

If $\hat{y}_{1,1} = \hat{y}_{1,2} = 1$, then any of f_1, f_2, f_3 are consistent with the information of the first round. In this case, the adversary can set $x_{2,1} = 1, x_{2,2} = 3$. No matter how the algorithm predicts, the adversary can give reinforcement FALSE, and has two functions remaining, trivially enabling it to force a mistake in the third round.

If $\hat{y}_{1,1} = 1, \hat{y}_{1,2} = 0$, then any of f_1, f_2, f_4 are consistent with the information of the first round. In this case, the adversary can also set $x_{2,1} = 1, x_{2,2} = 3$. No matter how the algorithm predicts, the adversary can give reinforcement FALSE, and has two functions remaining, again trivially enabling it to force a mistake in the third round.

If $\hat{y}_{1,1} = 0, \hat{y}_{1,2} = 1$ (recall that in this case $x_{1,3} = 3$), then any of f_1, f_3, f_4 are consistent with the information of the first round. In this case, the adversary can set $x_{2,1} = 1, x_{2,2} = 2$. No matter how the algorithm predicts, the adversary can give reinforcement FALSE, and has two functions remaining, also trivially enabling it to force a mistake in the third round.

Finally, if $\hat{y}_{1,1} = 0, \hat{y}_{1,2} = 0$ (again, $x_{1,3} = 3$), then any of f_2, f_3, f_4 are consistent with the information of the first round. In this case, the adversary also can set $x_{2,1} = 1, x_{2,2} = 2$. No matter how the algorithm predicts, the adversary can give reinforcement FALSE, and has two functions remaining, enabling it to force a mistake in the third round. This completes the proof that $\mathrm{opt}_{\mathrm{amb},2}(F) \geq 3$.

Next, we claim that $\mathrm{opt}_{\mathrm{weak}}(\mathrm{CART}_2(F)) = 2$. Consider the following algorithm in the weak reinforcement model. If $x_1 \in \{(1,2),(2,1)\}$, the algorithm predicts $(0,0)$. If $x_1 = (2,3)$, it predicts $(0,1)$. If $x_1 = (3,2)$, it predicts $(1,0)$. If $x_1 \in \{(1,3),(3,1)\}$, the algorithm predicts $(1,1)$.

In any of those cases, by inspection, after the first trial, there are at most two functions in F' consistent with the information received. Therefore, if the algorithm predicts with some consistent function for the second trial, it can ensure that it will make at most two mistakes. □

Notes

1. This quantity is called opt in [13, 14] and LC-ARB in [17, 18, 19].
2. Recall that pseudo-polynomial is commonly defined to be $\exp(\mathrm{poly}(\log n))$.
3. $\mathrm{opt}_{\mathrm{strong}}(F)$ was denoted by LC-ARB(F) in [2].
4. The model studied in [12] is considerably different than the model considered here. The common aspect is measuring the performance of a learning algorithm by comparison with the best function in F.
5. These results can also be viewed as bounding $\mathrm{opt}_{\mathrm{agn}}(\mathrm{SVAR}_n, \eta)$ and related quantities (for the randomized algorithms), where SVAR_n is the set of all functions f from $\{0,1\}^n$ to $\{0,1\}$ that output a single variable; i.e., such that there is an i such that for all $\vec{x} \in \{0,1\}^n$, $f(\vec{x}) = x_i$.
6. Some theorems have been proved about a popular approach to combat this problem, called *temporal difference* [20, 22, 23, 28, 8], but they rely on probabilistic assumptions about the environment of the learner, unlike the worst-case analysis done here for our new approach. Recently, Schapire and Warmuth [24] proved worst-case results about temporal difference learning in conjunction with the Widrow-Hoff rule in a model different from that of this section.
7. Getting the second is easy if $|Y| > 2^{\mathrm{opt}_{\mathrm{MBQ}}(\mathcal{L})}$; otherwise, since for all positive x, $\ln(1+x) \geq x/(1+x)$, we have $\log_2 \frac{2|Y|}{2|Y|-1} \geq \frac{1}{(2\ln 2)|Y|}$, which implies the second.

8. To see this, consider that as long as possible the environment might present x_t for which the algorithm predicts incorrectly. Presenting in between x_t for which the algorithm predicts correctly only helps the algorithm by providing additional information at no cost. Thus by ignoring trials for which it predicted correctly the algorithm ignores this additional information but does not increase the maximum number of mistakes for the worst possible sequence from $\mathcal{L}_F$. Note that this argument only holds since $\mathcal{L}_F$ is closed under permutations. For arbitrary $\mathcal{L}$ the position of a pair (x_t, y_t) in the sequence might encode information that is lost if the corresponding trial is ignored, for example see the proof of Theorem 8.
9. To simulate a membership query "what is $f(x)$?" while learning $\mathcal{L}_F$ in the MBQ model, one may ask "is the target sequence such that there is an $f \in F$ with $f(x) = 1$ and which is consistent with the target sequence and all previous queries?"

References

1. P. Auer and P.M. Long. Simulating access to hidden information while learning. *Proceedings of the 26th ACM Symposium on the Theory of Computing*, pages 263–272, 1994.
2. P. Auer, P.M. Long, W. Maass, and G.J. Woeginger. On the complexity of function learning. *Machine Learning*, 18(2):187–236, 1995.
3. D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.
4. A. Blumer, A. Ehrenfeucht, D. Haussler, and M.K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *JACM*, 36(4):929–965, 1989.
5. Nader H. Bshouty, Sally A. Goldman, Thomas R. Hancock, and Sleiman Matar. Asking questions to minimize errors. *J. of Comput. Syst. Sci.*, 52(2):268–286, 1996. Earlier version in 6th COLT, 1993.
6. Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the Association for Computing Machinery*, 44(3):427–485, May 1997.
7. N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, and M. K. Warmuth. On-line prediction and conversion strategies. *Machine Learning*, 25:71–114, 1996.
8. P. Dayan. The convergence of $TD(\lambda)$ for general λ. *Machine Learning*, 8:341–362, 1992.
9. M. Feder, N. Merhav, and M. Gutman. Universal prediction of individual sequences. *IEEE Transactions of Information Theory*, 38:1258–1270, 1992.
10. M. Kearns, M. Li, L. Pitt, and L.G. Valiant. On the learnability of Boolean formulae. *Proceedings of the 19th Annual Symposium on the Theory of Computation*, pages 285–295, 1987.
11. S.R. Kulkarni, S.K. Mitter, and J.N. Tsitsiklis. Active learning using arbitrary binary valued queries. *Machine Learning*, 11(1):23–36, 1993.
12. M.J. Kearns, R.E. Schapire, and L.M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.
13. N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.
14. N. Littlestone. *Mistake Bounds and Logarithmic Linear-threshold Learning Algorithms.* PhD thesis, UC Santa Cruz, 1989.
15. N. Littlestone and M.K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108:212–261, 1994.
16. N. Merhav and M. Feder. Universal schemes for sequential decision from individual data sequences. *IEEE Trans. Inform. Theory*, 39(4):1280–1291, 1993.
17. W. Maass and G. Turán. On the complexity of learning from counterexamples. *Proceedings of the 30th Annual Symposium on the Foundations of Computer Science*, pages 262–267, 1989.
18. W. Maass and G. Turán. On the complexity of learning from counterexamples and membership queries. *Proceedings of the 31st Annual Symposium on the Foundations of Computer Science*, pages 203–210, 1990.
19. W. Maass and G. Turán. Lower bound methods and separation results for on-line learning models. *Machine Learning*, 9:107–145, 1992.

20. A.L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal on Research and Development*, pages 210–229, 1959.
21. H. Shvaytser, 1988. Manuscript.
22. R.S. Sutton. *Temporal credit assignment in reinforcement learning.* PhD thesis, University of Massachusetts, Amherst, 1984.
23. R.S. Sutton. Learning to predict by methods of temporal difference. *Machine Learning*, 3:9–44, 1988.
24. R.E. Schapire and M.K. Warmuth. On the worst-case analysis of temporal-difference learning algorithms. *Machine Learning*, 95–121, 1996.
25. V.N. Vapnik and A.Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.
26. V. Vovk. Aggregating strategies. In *Proceedings of the 3nd Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.
27. V. Vovk. Universal forecasting algorithms. *Information and Computation*, 96(2):245–277, 1992.
28. C.I.C.H. Watkins. *Learning from delayed rewards.* PhD thesis, University of Cambridge, 1989.