JOURNAL OF APPROXIMATION THEORY 25, 369–383 (1979)

Linear Dynamic Approximation Theory

JAN MYCIELSKI*

Mathematics Department, University of Colorado, Boulder, Colorado 80309

Communicated by Oved Shisha

Received November 1, 1977

We study in detail the behavior of some known learning algorithms. We estimate the sum of the squares of the absolute relative errors of some general linear learning algorithms and the sum of the squares of the coefficients obtained by the perceptron algorithm. We prove the convergence of a statistical learning algorithm. The possibility of applications of this theory to biology is discussed.

1. INTRODUCTION

Perhaps the most important problem of classical approximation theory (see [5] and references therein) is the following. Given the values of a function f on a small set of points in the domain of f (which may be loaded with some errors) find a good model of f permitting its computation at all points of the domain.

Dynamic approximation theory (a term chosen here because of the analogy of this theory with dynamic programming [3]), which is a chapter of learning theory or prediction theory, looks at a slightly different problem. Points and the values of f at those points are given step by step. At each step, when a point is given, one has to predict the value of f at this point and then, as the true value is learned, one has to pay for the error committed. Thus one has to construct an algorithm for continuously improving the model of f rather than to construct a fixed model as in the classical theory.

In this paper I will consider only Chebyshev or linear models of f, i.e., a fixed sequence of functions $\varphi^1,\ldots,\varphi^d$ defined over the domain of f is chosen or given in advance and we look only at models of f of the form

$$\alpha_1\varphi^1 + \cdots + \alpha_d\varphi^d, \tag{1}$$

where α_k are constant coefficients. The algorithms which we consider here (with the exception of Section 7) serve only to find and update those coefficients. (For some nonlinear theory see [22].)

* This work was supported by the NSF.

0021-9045/79/040369–15$02.00/0

Copyright © 1979 by Academic Press, Inc.
All rights of reproduction in any form reserved.

This paper is partly expository. The linear algorithms considered here are of a well known kind, but our main Theorem 4.1, the second part of Theorem 6.1 and Theorem 7.1, although close to known facts, seem to be new. One generalization made here, which however is motivated only by the requirements of natural generality, is our treatment of (1) as an inner product $\langle \varphi, a \rangle$, where φ and a range over any Hilbert space H. (In all the applications which I know H has been finite dimensional.) A more important generalization drops the assumption of linearity, i.e., strict representability of f in the form (1) (Theorem 4.1).

In the final section we discuss the possibility of applying this mathematical theory to explain how living organisms learn some mechanical skills. (This was the original motivation of this paper.)

2. Notation

$\mathbb{R}$ and $\mathbb{C}$ are the *fields* of *real* and *complex* numbers respectively.

For $\alpha \in \mathbb{C}$, $\bar{\alpha}$ denotes the *complex conjugate* of α.

H is any *Hilbert space* over $\mathbb{R}$ or $\mathbb{C}$ (see [11]).

For any vectors $u, v \in H$, $\langle u, v \rangle$ denotes the *inner product* of u and v. Recall that $\langle v, u \rangle = \overline{\langle u, v \rangle}$.

$$\| v \| = \langle v, v \rangle^{1/2}.$$

$$v^\circ = v / \| v \| \qquad \text{for} \quad v \neq 0.$$

X is a non empty set.

$$\varphi \colon X \to H - \{0\}.$$

Given a sequence $x_0, x_1, \ldots, (x_i \in X)$ we put $\varphi_t = \varphi(x_t)$ for $t = 0, 1, \ldots$, $f \colon X \to \mathbb{R}$ or $f \colon X \to \mathbb{C}$ if H is real or complex respectively. For any sequence $\varphi_0, \varphi_1, \ldots$, of non zero vectors in H,

$$\varphi_0^\perp, \varphi_1^\perp, \ldots,$$

denotes the *Gram–Schmidt orthonormalization* of $\varphi_0, \varphi_1, \ldots$, i.e., $\varphi_n^\perp = 0$ if φ_n depends on $\varphi_0, \ldots, \varphi_{n-1}$ and

$$\varphi_n^\perp = \left(\varphi_n - \sum_{i<n} \langle \varphi_n, \varphi_i^\perp \rangle \varphi_i^\perp \right)^0$$

otherwise.

3. How to Learn Linear Functionals

Let f and φ be given and suppose that f is *linear in* φ, i.e.,

$$f(x) = \langle \varphi(x), a^* \rangle \tag{2}$$

for an $a^* \in H$ and all $x \in X$. (Examples are given in Section 5.)

At times $t = 0, 1, \ldots$, nature shows some points $x_t \in X$ and the scalars $f(x_t)$. For each t before knowing $f(x_t)$ the subject must guess a vector $a_t \in H$. The scalar

$$e_t = (f(x_t) - \langle \varphi_t , a_t \rangle) / \| \varphi_t \| = \langle \varphi_t^\circ, a^* - a_t \rangle \tag{3}$$

is called here the *relative error* at time t. Each time the subject pays $| e_t |^2$. His goal is to minimize $\sum_t | e_t |^2$.

We shall study first the following simple algorithm for the above problem.

$$\begin{aligned} &a_0 \text{ is the best guess of } a^*, \text{ and} \\ &a_{t+1} = a_t + \bar{e}_t \varphi_t^0. \end{aligned} \tag{L_1}$$

As easily seen a_{t+1} is the unique vector minimizing $\| a_{t+1} - a_t \|$ which satisfies

$$\langle \varphi_t , a_{t+1} \rangle = f(x_t);$$

in other words (see 3.1(ii) below) a_{t+1} is the perpendicular projection of a_t in a direction parallel to φ_t into a hyperplane containing a^*. The algorithm (L_1) is as conservative as possible when taking full advantage of the last information. Because of its remarkable computational simplicity (L_1) is suitable for applications. But first let us ask if it bounds $\sum | e_t |^2$ and if $a_t \to a^*$. The following theorem and comments answer these questions.

3.1. Theorem on (L_1). (i) $\sum_{t=0}^{\infty} | e_t |^2 \leqslant \| a^* - a_0 \|^2$;

(ii) $\| a^* - a_{t+1} \|^2 = \| a^* - a_t \|^2 - | e_t |^2$;

(iii) *if* $a_0 , a_1 , \ldots$, *converge to a vector* a *then* a *satisfies*

$$\lim_{t \to \infty} (f(x_t) - \langle \varphi_t , a \rangle) / \| \varphi_t \| = 0.$$

Proof. Each of the three implications

$$[(2) \,\&\, (3) \,\&\, (L_1)] \Rightarrow \text{(ii)} \Rightarrow \text{(i)}, \qquad \text{and} \qquad [(3) \,\&\, \text{(i)}] \Rightarrow \text{(iii)}$$

is an easy exercise. ∎

3.2. Comments. 1. The inequality (i) is the most important part of 3.1 since it secures a bound on the total loss. In particular $e_t \to 0$ follows. (i)

gives the best possible estimate of $\sum |e_t|^2$, since, if $\varphi_0 = a^* - a_0$, then $|e_0| = \|a^* - a_0\|$. A generalization of (i) will be proved in Section 4.

2. But the sequence $a_0, a_1, \ldots$, need not converge at all. Even if $H = \mathbb{R}^2$, given that $a_0 \neq a^*$, and given any $\alpha < \|a^* - a_0\|$ we can produce φ and a sequence $x_0, x_1, \ldots$, such that the polygonal line $\overline{a_0a_1} \cup \overline{a_1a_2} \cup \overline{a_2a_3} \cup \cdots$ given by (L_1) spirals infinitely many times around a^* within the ring $\{v: \alpha < \|a^* - v\| a^* - a_0\|\}$ and $\sum_{t=0}^{\infty} |e_t|^p = \infty$ for every $p < 2$. (Recall that, by (L_1), $\|a_{t+1} - a_t\| = |e_t|$ for all t.)

3. PROBLEM. Can one develop a continuous time analog of the above theory?

Now let us consider another algorithm for the same problem, which may seem more natural to the theoretician.

$$\begin{aligned} &b_0 \text{ is the best guess of } a^*, \text{ and} \\ &b_{t+1} = b_t + \bar{\eta}_t \varphi_t^{\perp}, \end{aligned} \tag{L_2}$$

where

$$\begin{aligned} \eta_t &= (f(x_t) - \langle \varphi_t, b_t \rangle) \Big/ \Big\| \varphi_t - \sum_{s<t} \langle \varphi_t, \varphi_s^{\perp} \rangle \varphi_s^{\perp} \Big\| \\ &= \langle \varphi_t^{\perp}, a^* - b_t \rangle. \end{aligned}$$

Here it is easy to see that b_{t+1} is the unique vector minimizing $\|b_{t+1} - b_t\|$ and satisfying

$$\langle \varphi_s, b_{t+1} \rangle = f(x_s) \qquad \text{for all} \quad s \leqslant t. \tag{4}$$

In fact by (L_2), we have

$$\langle \varphi_s^{\perp}, b_{t+1} \rangle = \langle \varphi_s^{\perp}, a^* \rangle \qquad \text{for all} \quad s \leqslant t.$$

Hence

$$\langle \varphi_s, b_{t+1} \rangle = \langle \varphi_s, a^* \rangle \qquad \text{for all} \quad s \leqslant t,$$

and, by (2), we get (4).

Clearly (L_2) takes advantage of *all* the past information while (L_1) takes advantage of the last input only. But the computational cost of (L_2) is much larger since it requires orthonormalization and hence much more storage and retrieval than (L_1). Moreover the theorem and comments which follow show that in general the advantage of (L_2) over (L_1) is too small to justify so much more computation. The relative errors committed by (L_2) are

$$e_t' = (f(x_t) - \langle \varphi_t, b_t \rangle) / \|\varphi_t\| = \langle \varphi_t^0, a^* - b_t \rangle.$$

3.3. THEOREM ON (L_2). (0) $|e_t'| \leqslant |\eta_t|$;

(i) $\sum_{t=0}^{\infty} |\eta_t|^2 \leqslant \|a^* - b_0\|^2$;

(ii) $\|a^* - b_{t+1}\|^2 = \|a^* - b_t\|^2 - |\eta_t|^2$;

(iii) *the sequence* $b_0, b_1, \ldots$, *converges to a vector* b *such that*

$$\langle \varphi_t, b \rangle = f(x_t) \quad \text{for} \quad t = 0, 1, \ldots,$$

Proof. Of course

$$\|\varphi_t\| \geqslant \left| \varphi_t - \sum_{s<t} \langle \varphi_t, \varphi_s^{\perp} \rangle \varphi_s^{\perp} \right|$$

and (0) follows. The assertions (i) and (ii) follow from 3.1(i) and (ii) respectively applied in the case when $\varphi_t = \varphi_t^{\perp}$ for all t. To show (iii) notice that, by (L_2),

$$\|b_{t+n} - b_t\|^2 = \sum_{k=0}^{n-1} |\eta_{t+k}|^2 .$$

Hence, by (i), $b_0, b_1, \ldots$, is a Cauchy sequence and thus it has a limit $b \in H$. Then by (4) we get the equalities of (iii). ∎

3.4. COMMENTS. 1. The estimates (0) and (i) do not show any advantage of (L_2) over (L_1), and those estimates, like those in 3.1, are the best possible.

2. The only advantages of (L_2) over (L_1) are: (a) If H is finite dimensional, say n-dimensional, then there are at most n errors e_t' different from 0. (b) By 3.3 (iii), if we regard b_t as the state of our knowledge about f, it is nice to know that this state stabilizes, e.g., if we pay for the computations of b_t, we may stop modifying b_t when the errors seem to have decreased enough.

3. As already mentioned, in applications the computational disadvantages of (L_2) are likely to be prohibitive. In applications to theoretical biology (see Section 8) they *are* prohibitive.

4. How to Learn Near-to-Linear Functionals

To get closer to applications we must generalize the above theory to a wider class of f's. We can even permit some dependence of f on time. Thus let

$$f: X \times T \to \mathbb{R} \quad \text{or} \quad \mathbb{C},$$

where $T = \{0, 1, \ldots,\}$ is the discretized time axis and let H be a real or complex Hilbert space respectively. Again $\varphi: X \to H - \{0\}$ is given and we generalize the definition (3) of relative errors putting

$$e_t = (f(x_t, t) - \langle \Omega_t, a_t \rangle)/\|\varphi_t\|. \tag{5}$$

The game is the same as in Section 3, that is the subject must choose a_t before knowing $f(x_t, t)$ and his interest is to minimize $|e_t|$.

We put

$$N(f) = \inf_{a \in H} \sup_{x \in X, t \in T} |f(x, t) - \langle \varphi(x), a \rangle| / \|\varphi(x)\|.$$

Thus $N(f)$ is a measure of the nonlinearity of f and its dependence on t. We generalize the algorithm (L_1) as follows

Choose $\alpha \geqslant 0$ and $a_0 \in H$, and

$$a_{t+1} = \begin{cases} a_t + \bar{e}_t \varphi^0 & \text{if } |e_t| > \alpha, \\ a_t & \text{if } |e_t| \leqslant \alpha. \end{cases} \tag{L_3}$$

Now we shall prove that if $N(f) < \alpha/2$ then (L_3) secures

$$\sum_{|e_t| > \alpha} (|e_t| - \alpha) < \infty. \tag{6}$$

More exactly, we have the following theorem (announced in [21]).

4.1. THEOREM ON (L_3). *If $a^* \in H$ and*

$$|f(x_t, t) - \langle \varphi_t, a^* \rangle| / \|\varphi_t\| \leqslant \alpha/2 \quad \textit{for} \quad 0, 1, \ldots, \tag{7}$$

then

$$\sum_{|e_t| > \alpha} |e_t| (|e_t| - \alpha) \leqslant \|a^* - a_0\|^2. \tag{8}$$

Proof. Let a^* satisfy (7) and put $v_t = a^* - a_t$ and $\gamma_t = e_t - \langle \varphi_t^\circ, v_t \rangle$. Then, by (5),

$$\gamma_t = (f(x_t, t) - \langle \varphi_t, a^* \rangle) / \|\varphi_t\|,$$

and, by (7), $|\gamma_t| \leqslant \alpha/2$. Hence, by (L_3), for all t such that $|e_t| > \alpha$ we have

$$\begin{aligned} \|a^* - a_{t+1}\|^2 &= \|v_t - \bar{e}_t \varphi_t^\circ\|^2 \\ &= \|v_t\|^2 - e_t \langle v_t, \varphi_t^\circ \rangle - \bar{e}_t \langle \varphi_t^\circ, v_t \rangle + |e_t|^2 \\ &= \|v_t\|^2 - 2\mathrm{Re}(\bar{e}_t \langle \varphi_t^\circ, v_t \rangle) + |e_t|^2 \\ &= \|v_t\|^2 - 2\mathrm{Re}(\bar{e}_t (e_t - \gamma_t)) + |e_t|^2 \\ &= \|v_t\|^2 - |e_t|^2 + 2\mathrm{Re}(\bar{e}_t \gamma_t) \\ &\leqslant \|v_t\|^2 - |e_t|^2 + 2|e_t||\gamma_t| \\ &\leqslant \|a^* - a_t\|^2 - |e_t|(|e_t| - \alpha). \end{aligned}$$

Hence, by (L_3),

$$\| a^* - a_{t+1} \|^2 \leqslant \| a^* - a_0 \|^2 - \Sigma_{|e_s|>\alpha, s\leqslant t} \, | e_s | (| e_s | - \alpha),$$

and (8) follows. ∎

4.2. COMMENTS. For the first five comments we assume that $f(x, t) = f(x)$, i.e., f does not depend on t.

1. We do not know if there exists any algorithm which secures that

$$| e_t | > 2N(f)$$

occurs only finitely many times.

2. As easily seen, if $N(f) > \alpha/2$ then $\limsup_{t\to\infty} | e_t | \leqslant \alpha$ may fail even when $H = \mathbb{R}$. *Problem*: Does this inequality hold if $N(f) = \alpha/2$?

3. If H is finite dimensional or if for every vector $v \in H - \{0\}$ there exists a scalar c such that cv is in the range of φ then one can prove the existence of a vector $a^* \in H$ such that

$$| f(x) - \langle \varphi(x), a^* \rangle | / \| \varphi(x) \| \leqslant N(f) \qquad \text{for all} \quad x \in X. \tag{9}$$

(Hint: In the infinite dimensional case use the Theorem of Alaoglu.) Hence, under such suppositions, (7) is valid (and meaningful) even for $N(f) = \alpha/2$.

4. However, in general, $N(f) < \alpha/2$ is necessary for the validity of (6). Indeed if $X = \{2, 3, \ldots\}$, and $\varphi(1)$, $\varphi(2), \ldots$, is an orthonormal sequence and $f(n) = 1/\log n$ then $N(f) = 0$, but no a^* satisfying (9) exists. And if $\alpha = 0$, $a_0 = 0$ and $(x_0, x_1, \ldots) = (2, 3, \ldots)$ then $e_t = f(t + 2)$ and $\sum | e_t |^p = \infty$ for all $p > 0$. (Nevertheless this example satisfies $e_t \to 0$, whence the Problem in 4.2.2 is open.) *Problem*: Are there any such examples with $N(f) > 0$?

5. The above example shows that if f is not linear in φ (see Section 3) then it may be risky to choose $\alpha = 0$, i.e., to use the algorithm (L_1). Already for $H = \mathbb{R}^2$ Theorem 3.1 can be invalidated by very small deviations of f from linearity. In fact for every positive constants E and ϵ, for $X = \{1, 2, 3\}$ and $f(1) = f(2) = 0$, $f(3) = E + 1$ we can construct $\varphi: X \to \mathbb{R}^2 - \{(0, 0)\}$ such that $N(f) < \epsilon$ and there exists a sequence $x_0, x_1, \ldots, (x_i \in X)$ such that, if (L_1) is used, $| e_t | > E$ infinitely many times and with positive frequency. (No such example is possible if $H = \mathbb{R}^1$.) Of course (L_3) would secure (6) and (8) if $\alpha \geqslant 2\epsilon$ was chosen.

6. PROBLEM. Are there any interesting estimates of the mean error or the mean square error given by (L_1), in terms of $N(f)$?

7. PROBLEM. The estimates (6) and (8) are worst-case estimates. In many situations the following assumption is more realistic. X is a probability-measure space, and $x_0, x_1, \ldots$, are chosen independently at random in X. Then better estimates of the sequence of errors should be true. E.g.: does (L_3) secure

$$\sum_{|e_t|>\alpha} (|e_t| - \alpha)^p < \infty$$

with probability 1 for some $p < 1$?

8. A case when (L_3) will give good results, which is not covered by Theorem 4.1, occurs when there exists a large enough constant T such that for every t the quantity

$$\inf_{a\in H} \sup_{x\in X, t\leqslant s<t+T} |f(x, s) - \langle \varphi(x), a\rangle|/\|\varphi(x)\|$$

is small. Then the a_t's will follow the local close to linear behavior of f (see [9, 17]).

9. Some other algorithms analogous to (L_1) can be briefly introduced as follows. Let $f(x, t) = g(x, t) + \nu(t)$, where $\nu(t)$ is a random noise with mean 0 and $N(g)$ is small, but $N(f)$ may be large. Now we want to predict $g(x_{t+1}, t+1)$ but the information given is $f(x_t, t)$ and x_{t+1}. Then we may use the algorithm

$$\begin{aligned} &\text{Choose } c > 0 \text{ and } a_0, \text{ and} \\ &a_{t+1} = a_t + c\tilde{e}_t\varphi_t{}^0, \end{aligned} \qquad (L_e)$$

where e_t is computed relative to $f(x_t, t)$ (the only available information). Then, if c is sufficiently small, ν will average itself out (for a fuller development and error estimates see [7, 9, 10, 15, 17, 24, 25, 26]).

5. THEORETICAL APPLICATIONS

The algorithm (L_3) can be used for the approximation of functions by polynomials. E.g. if X is a compact subset of $\mathbb{R}$, $f\colon X \to \mathbb{R}$ is a continuous function, $H = \mathbb{R}^{n+1}$ and $\varphi(x) = (1, x, x^2, \ldots, x^n)$ then, by the first Weierstrass approximation theorem (see [14]), $N(f) \to 0$ as $n \to \infty$. Thus, if n is large enough, (L_3) with large enough α will give good results.

Similarly if $X = \{x \in \mathbb{C}: |x| = 1\}$, $f\colon X \to \mathbb{C}$ is a continuous function, $H = \mathbb{C}^{2n+1}$ and $\varphi(x) = (x^{-n}, \ldots, x^{-1}, 1, x, \ldots, x^n)$ then, by the second Weierstrass approximation theorem (see [13]), $N(f) \to 0$ as $n \to \infty$. Hence again (L_3) may be applicable.

In practice the dimension d of H cannot be too large since a computer can not handle vectors with too many coordinates. This is the main limitation of (L_3) in multivariate approximation. In fact, say in the m real variables case, if we want to apply (L_3) to obtain an approximating polynomial P of degree n,

$$P = \sum_{k_1+\ldots+k_m \leqslant n, k_i \geqslant 0} \alpha_{k_1 \ldots k_m} x_1^{k_1} \cdots x_m^{k_m},$$

then $\varphi(x) = (x_1^{k_1} \cdots x_m^{k_m}: k_i \geqslant 0, k_1 + \cdots + k_m \leqslant n)$ and the number of monomials of P, i.e., the dimension d, equals $\binom{m+n}{n}$. Thus $\binom{m+n}{n}$ should not be too large. In the applications to theoretical biology which we discuss in Section 8 it is conjectured that the brain uses (L_3) and does all the necessary computations. Here the acceptable d's are probably smaller than those acceptable to computers. (The learning of approximating polynomials and related expressions is considered in [25].)

In some applications dimensional analysis (see [6]) may indicate how to prune the general polynomial P, i.e., remove the monomials which may be irrelevant, and thus decrease d. Other methods (see [1, 18] and references therein) like relaxation procedures, finite elements methods, and splines use the idea of partitioning X into sets such that good approximation of f in each set can be achieved by polynomials of small degree. Of course, here, the number of such sets may become a difficulty. (The author is not aware of any theoretical results comparing the amount of computation or the efficiency of various methods of this kind.)

It can be argued that, without any a priori knowledge, the best choice of a_0 is 0. The problem of choosing α should probably be decided on the basis of the largest acceptable error. Or course l'appétit vient en mangeant and α can be decreased as the errors diminish, and then increased again if the decrease proved too optimistic.

6. Perceptron Learning Algorithm

In the previous sections we have studied algorithms for learning real valued or complex valued functions. Now, for completeness, we present a similar well known algorithm of F. Rosenblatt for learning two-valued functions.

Let X^+ and X^- be two disjoint sets, $X = X^+ \cup X^-$, H is a *real* Hilbert space and $\varphi: X \to H - \{0\}$. Let there be some constants $R > r > 0$ and $\alpha \geqslant 0$ and a vector $a^* \in H$ such that

$$\begin{aligned} \| a^* \| = 1, \| \varphi(x) \| &\leqslant R \quad \text{for all} \quad x \in X, \\ \langle \varphi(x), a^* \rangle &\geqslant r \quad \text{for all} \quad x \in X^+, \\ \langle \varphi(x), a^* \rangle &\leqslant -r \quad \text{for all} \quad x \in X^-. \end{aligned}$$

Again at times $t = 0, 1, \ldots$, nature shows some points $x_t \in X$ but now it tells only $+$ or $-$ depending on whether $x_t \in X^+$ or $x_t \in X^-$. Again after receiving x_t but before learning the corresponding sign the subject has to guess a vector a_t and to pay one unit iff $\langle \varphi_t , a_t \rangle \leqslant \alpha$ and $x_t \in X^+$, or $\langle \varphi_t , a_t \rangle \geqslant -\alpha$ and $x_t \in X^-$.

The "perceptron learning algorithm" does this in the following way.

Set $a_0 = 0$, and

$$a_{t+1} = \begin{cases} a_t + \varphi_t \text{ if } \langle \varphi_t , a_t \rangle \leqslant \alpha \text{ and } x_t \in X^+, \\ a_t - \varphi_t \text{ if } \langle \varphi_t , a_t \rangle \geqslant -\alpha \text{ and } x_t \in X^-, \\ a_t \text{ otherwise.} \end{cases} \tag{L_1}$$

The following theorem is a slight refinement of the classical "perceptron convergence theorem" which was stated in [19].

6.1. THEOREM ON (L_5). *(L_5) secures that the number of adjustments, i.e., the total loss, will not exceed $(2\alpha + R^2)/r^2$ and $\| a_t \| \leqslant (2\alpha + R^2)/r$ for all t.*

Proof. We can assume without loss of generality that $a_0 \neq a_1 \neq a_2 \neq \cdots$ until (if ever) the sequence becomes constant. We can also assume without loss of generality that X^- is empty, by substituting $X^+ \cup X^-$ for X^+ and $-\varphi(x)$ for $\varphi(x)$ when $x \in X^-$. By the Schwartz inequality, and since $\| a^* \| = 1$, we have

$$\langle a_t , a^* \rangle \leqslant \| a_t \| \qquad \text{for all} \quad t. \tag{10}$$

By (L_5) and $X^- = \varnothing$, whenever $a_{t+1} \neq a_t$, we have

$$\| a_{t+1} \|^2 = \| a_t \|^2 + 2\langle \varphi_t , a_t \rangle + \| \varphi_t \|^2 \leqslant \| a_t \|^2 + 2\alpha + R^2$$

and

$$\langle a_{t+1} , a^* \rangle = \langle a_t , a^* \rangle + \langle \varphi_t , a^* \rangle \geqslant \langle a_t , a^* \rangle + r$$

Hence if $a_0 \neq a_1 \neq \cdots \neq a_N$ then

$$\| a_N \|^2 \leqslant N(2\alpha + R^2) \qquad \text{and} \qquad \langle a_N , a^* \rangle \geqslant Nr.$$

Therefore, by (10), we have $N^2 r^2 \leqslant N(2\alpha + R^2)$, i.e., $N \leqslant (2\alpha + R^2)/r^2$, and hence also $\| a_N \| \leqslant (2\alpha + R^2)/r$. ∎

6.2. COMMENTS. 1. The above proof is an inessential modification of the proof given in [16], and the history of the theorem and the proof is related there and in [4]. (L_5) belongs to a group of similar algorithms used in pattern recognition and related engineering problems [4, 7, 9, 16, 24, 26].

2. PROBLEM. Are the upper bounds in 6.1 sharp (especially in the case when $H = \mathbb{R}^d$)?

3. A natural modification of the algorithm, similar to (L_2), may reuse the data until $\langle \varphi_s , a_{t+1} \rangle$ is right for all $s = 0,..., t$. Notice that the total number of modifications remains bounded by $(2\alpha + R^2)/r^2$.

4. As in Section 5, the main applications of (L_5) involve $X = [a, b]^m$ or $X = \{x \in \mathbb{C}: | x | = 1\}^m$ and monomial vector functions φ.

7. A PROBABILISTIC LEARNING ALGORITHM

We take this opportunity to prove one more theorem on a learning algorithm (L_6) announced in [20]. Unlike the algorithms discussed in the previous sections, (L_6) seems essentially useless (both in theory and practice) but it is pretty and may inspire worthier things. It applies to the prediction of r-valued (r is any positive integer) random variables f. (L_6) generalizes some algorithms given in [8] in as much as it avoids any suppositions on f. (The reader interested in statistical prediction theory may consult [1, 17].)

Let $\langle X, \mathscr{B}, \mu \rangle$ be a probability measure space; only finite additivity of the Boolean algebra $\mathscr{B}$ and of the measure μ is stipulated. Let $f: X \to \{1,..., r\}$ be a random variable, i.e., $f^{-1}(v) \in \mathscr{B}$ for $v = 1,..., r$. m sets $A_1 ,..., A_m \in \mathscr{B}$ are given. Nature picks at random relative to μ, independently, $n + 1$ points $x_1 ,..., x_n$ and x in X and it shows $f(x_1),..., f(x_n)$. The subject may guess $f(x)$. If his guess is correct he wins α, if it is incorrect he loses β and if he does not guess the payoff is γ. He knows only $A_1 ,..., A_m$ a priori.

We consider the following simple algorithm for this problem.

(L_6) If there exist $k \in \{1,..., m\}$ and $v \in \{1,..., r\}$ such that $x \in A_k$ and $f(x_i) = v$ for all $x_i \in A_k$ then pick the least such k and the corresponding v (if $\{x_1,..., x_n\} \cap A_k = \varnothing$ pick $v = 1$) and guess $f(x) = v$. Otherwise do not guess.

Let p be the probability that a wrong guess was made.

7.1. THEOREM ON (L_6). $p < mr/ne$, where $e = \sum_0^\infty 1/k!$.

Proof. Let $\mu^{(n+1)}$ be the product measure in X^{n+1} and $\mathbb{R}$ be the set of all random variables $g: X \to \{1,..., r\}$. Then

$$\begin{aligned} p &\leqslant \mu^{(n+1)}\{(x_1 ,..., x_n , x) : \exists k\ \exists v[x \in A_k , f(x_i) = v \text{ for} \\ &\qquad \text{all } x_i \in A_k \text{ and } f(x) \neq v]\} \\ &\leqslant m \max_{\mathscr{A} \in \mathscr{B}} \max_{g \in \mathscr{R}} \mu^{(n+1)}\{(x_1 ,..., x_n , x) : \exists v[x \in A, g(x) \neq v \text{ and} \\ &\qquad g(x_i) = v \text{ for all } x_i \in A]\} \\ &\leqslant m \max_{0 \leqslant a \leqslant 1} \max_{a_1 + \cdots + a_r = a, a_i \geqslant 0} \sum_{v=1}^{r} (a - a_v)(1 - (a - a_v))^n. \end{aligned}$$

Since the function $t(1-t)^n$ attains it maximum over [0, 1] at the point $t = 1/(n+1)$, it follows that the above maximum is attained if

$$a - a_v = 1/(n+1) \qquad \text{for} \quad v = 1,..., r.$$

Hence

$$p \leqslant \frac{mr}{n+1}\left(1 - \frac{1}{n+1}\right)^n = \frac{mr}{n}\left(1 + \frac{1}{n}\right)^{-n-1} < \frac{mr}{ne},$$

where the last inequality is classical (see e.g. [12]).

7.2. COMMENTS. 1. If $n \leqslant mr/e$ the theorem is vacuous, but since $mr/ne \to 0$ as $n \to \infty$ hence, with large enough n and appropriate payoffs α, β and γ, (L_6) yields a meaningful learning algorithm.

2. PROBLEM. Can one significantly improve the estimate of p if the sequence $A_1,..., A_m$ is closed under complementation?

3. The probability that (L_6) leads to the "no guess" decision does not exceed

$$\sum_{v=1}^{r} \mu\left(f^{-1}(v) - \bigcup \{A_k : A_k \subseteq f^{-1}(v)\}\right).$$

Thus, if f is "regular" relative to $A_1,..., A_m$, (L_6) will usually advise some guess.

4. We can modify (L_6) in many ways, e.g., we can choose k and v such that $x \in A_k$ and the ratio

$$\frac{\text{card}\{i : x_i \in A_k \text{ and } f(x_i) = v\} + 1}{\text{card}(\{x_1,..., x_n\} \cap A_k) + 2}$$

is maximal, and guess $f(x) = v$. But we have not found any estimates for this natural algorithm.

8. APPLICATIONS TO THEORETICAL BIOLOGY

Assume that in smooth movements the accelerations i.e. the forces applied are constant. E.g. in a fast turn on skis in good conditions the forces applied during the turn should be constant; when an experienced carpenter drives a nail the acceleration of the hammer during one hit is rather constant; a good driver brakes as follows, smoothly he decreases the acceleration to a desired negative level and keeps it constant for most of the time of the maneuver and only at the end smoothly he raises it back to 0. In many other single move-

ments of walking, work and sports the above assumption of constancy of force seems close enough to reality to justify the consideration of the following problem.

How does the brain learn the function computing the appropriate level of the force or the rate of firing of some efferent nerves, from the real variables of a given situation?

To answer this question we shall still assume that the appropriate forces are continuous functions of the relevant variables, and that there exists a good error estimating mechanism. (In fact an additional error correcting feedback mechanism is present which often permits us to attain tolerable results at the cost of greater attention and effort even before the right functions have been learned. This complicates the estimation of the error.)

8.1. CONJECTURE. *The brain learns the functions which determine the right forces by means of some algorithm similar to* (L_3).

Proof of plausibility. The great variety of functions which can be learned rules out the existence of an inherited baggage of functions which works for all purposes. Hence there ought to be a "universal" learning algorithm. (L_3) is the simplest learning algorithm which we know. The computation of linear forms $\alpha_1\varphi^1 + \cdots + \alpha_d\varphi^d$ with rather large d (perhaps even integrals $\int \alpha_u\varphi^u\, du$) and the modifications of the coefficients α required by (L_3) seem quite feasible in the brain. Moreover the exact nature of the functions φ is unimportant as long as their linear span sufficiently approximates the functions which are to be learned. The brain looks like an enormous analog computer, processing information in the form of rates of firing of neurons. But since very little is known about the kinds of computations which are going on, further speculation would be premature. (It seems that reinforcements or weakenings of synaptic connections could represent the modification of the α's, (see e.g. [23] and references therein, or [16] Section 12.4.7), but the true form of the φ's is probably still hidden; perhaps they are monomials of degree $\leqslant 2$.) ▮

8.2. COMMENTS. 1. Perhaps it is possible to conjecture in a more definite way, and consistently with the facts which are already known, that the cerebellum is the site of some simple algorithms like those discussed in this paper. The cerebellum's relatively regular structure would lend strong support to any conjecture which explains the role of this structure.

2. We did not attempt to explain in this paper how the brain chooses the right functions to be used in a given movement nor how it switches from movement to movement. Some ideas on those "higher" functions and on the stream of consciousness are considered in [22].

3. In the early days of enthusiastic experimentation with perceptrons and related machines (based on algorithms related to (L_5)) there were hopes that it would explain all or most classification and recognition abilities which the brain can learn. The beautiful book [16] of Minsky and Papert threw a lot of cold water on such naive optimism (perhaps too much, see [4, 19]). And so we believe today that the current linear theory of pattern recognition, and all the algorithms considered in this paper, are quite inadequate for explaining this ability of the brain. Different ideas, unrelated to linear methods, which may be relevant to this problem are considered in [22].

4. On the other hand, considering the mathematical developments originated by Chebyshev, I believe that linear algorithms such as (L_3) are the best tools for learning continuous regular functions of relatively few variables, such as the learnable functions discussed in this section, and a limited class of pattern recognition problems with similar characteristics. If this is true then those algorithms are the natural tools for explaining a little part of intelligence.

5. It may be too difficult today to confirm the conjecture 8.1, by the methods of neurophysiology. However there is perhaps a chance of confirming it (with a lesser degree of certainty) by an appropriate study of the behavior of errors in real learning, and by exploring systematically the variety of functions which can be learned.

REFERENCES

1. S. AGMON, The relaxation method for linear inequalities, *Canad. J. Math.* **6** (1954), 382–392.
2. J. AITCHISON AND I. R. DUNSMORE, "Statistical Prediction Analysis," Cambridge University Press, London, 1975.
3. R. BELLMAN, "Dynamic Programming," Princeton University Press, Princeton, N. J., 1957.
4. H. D. BLOCK, Review of [16], *Inform. Contr.* **17** (1970), 501–522.
5. P. J. DAVIS, "Interpolation and Approximation," Dover, New York, 1975.
6. S. DROBOT, On the foundations of dimensional analysis, *Studia Math.* **14** (1953), 84–99.
7. R. O. DUDA AND P. E. HART, "Pattern Classification and Scene Analysis," Wiley, New York, 1973.
8. A. EHRENFEUCHT AND J. MYCIELSKI, Interpolation of functions over a measure space and conjectures about memory, *J. Approximation Theory* **9** (1973), 218–236.
9. K. S. FU (Ed.), "Pattern Recognition and Machine Learning," Plenum, New York, 1971.
10. A. GERSHO, "Adaptive Equalization of Highly Dispersive Channels for Data Transmission, I," Bell Telephone Laboratories Technical Memorandum, MM 68–1386–3, April 1968.
11. P. R. HALMOS, "Introduction to Hilbert Space," Chelsea, New York, 1972.
12. R. F. JOHNSONBAUGH, Another proof of an estimate for e, *Amer. Math. Monthly* **81** (1974), 1011–1012.

13. Y. KATZNELSON, "An Introduction to Harmonic Analysis," Dover, New York, 1976.
14. G. G. LORENTZ, "Approximation of Functions," Holt, Rinehart & Winston, New York, 1966.
15. R. W. LUCKY, Techniques for adaptive equalizations for digital communication systems, *Bell System Tech. J.* **45** (1966), 255–286.
16. M. MINSKY AND S. PAPERT, "Perceptrons, An Introduction to Computational Geometry," MIT Press, Cambridge, Mass., 1972.
17. N. MORRISON, "Introduction to Sequential Smoothing and Prediction," McGraw–Hil New York, 1969.
18. T. S. MOTZKIN AND I. J. SHOENBERG, The relaxation method for linear inequalities *Canad. J. Math.* **6** (1954), 393–404.
19. J. MYCIELSKI, Review of [16], *Bull. Amer. Math. Soc.* **78** (1972), 12–15.
20. J. MYCIELSKI, Monte Carlo interpolation over a measure space, *Notices Amer. Math. Soc.* **20** (1973), A–269.
21. J. MYCIELSKI, A linear learning theorem, *Notices Amer. Math. Soc.* **21** (1978).
22. J. MYCIELSKI, Toward a mathematical theory of memory, in preparation.
23. M. N. NASS AND L. N. COOPER, A theory for the development of feature detecting cells in visual cortex, *Biol. Cybernetics* **19** (1975), 1–18.
24. N. J. NILLSON, "Learning Machines, Foundations of Trainable Pattern-Classifying Systems," McGraw–Hill, New York, 1965.
25. R. J. ROY AND J. S. SHERMANN, A learning technique for Volterra series representation, *IEEE Trans. Automatic Control* (Dec. 1967), 761–764.
26. J. T. TOU AND R. C. GONZALEZ, "Pattern Recognition Principles," Addison–Wesley, Reading, Mass., 1974.