A REVIEW OF RECENT RESULTS ON LEARNING WITH FINITE MEMORY

M. E. HELLMAN* and T. M. COVER**

Department of Electrical Engineering
Stanford University, Stanford, California 94305 U.S.A.

* Formerly with Massachusetts Institute of Technology, Cambridge, Massachusetts. The work of this author was supported by NSF Grant GK5800 and the Joint Services Electronics Program (Contracts DA28-043-AMC-02536(E) and N00014—67—A—0112—0044).

** The work of this author was supported by Contract F44620-69-C-0101.

1. INTRODUCTION

Since any system is in fact finite, the problem of learning with finite memory is of great importance. The real problem is to formulate a model which is both applicable to the real world and amenable to theoretical study.

In this paper we will be concerned with several models and, not unexpectedly, there is an inverse relation between theoretical simplicity and real world applicability. However these models are all related and are outgrowths of one advanced by Hellman and Cover [1, 2, 3]. This model was motivated by earlier works [4, 5] but was the first to yield an optimal solution.

We are given a sequence of independent, identically distributed observations $\{X_n\}_{n=1}^{\infty}$ where each observation X is drawn according to the probability measure P. There are two hypotheses H_0 and H_1 with *a priori* probabilities π_0 and $\pi_1 = 1 - \pi_0$, where under H_t, $P = P_t$ for $t = 0.1$. We assume that π_0, π_1, P_0 and P_1 are known, and that $P_0 \neq P_1$ almost everywhere.

Let $d_n \in \{H_0, H_1\}$ denote the decision made at time n. If d_n is allowed to depend on $X_1, X_2, \ldots, X_n$ then a standard likelihood ratio test yields a probability of error tending exponentially to zero in the sample size n. However the likelihood ratio is real valued and to store it exactly requires infinite memory. We could try to estimate the degradation introduced in this method by the use of finite memory, but prefer to take the more fundamental viewpoint discussed below.

A finite memory algorithm consists of the sextuple $\mathcal{A} = (\mathcal{X}, D, \mathcal{S}, d, f, T_0)$. $\mathcal{X}$ is the space of allowable observations (i.e. $X_n \in \mathcal{X}$), D is the space of allowable decisions (i.e. $d_n \in D = \{H_0, H_1\}$ in our example), $\mathcal{S}$ is the finite state space or memory, $d\colon \mathcal{S} \to D$ is the decision function, $f\colon \mathcal{S} \times \mathcal{X} \to \mathcal{S}$ is the state transition function, and $T_0 \in \mathcal{S}$ is the initial state of memory. The interpretation is that at time zero memory is in state T_0. At time one X_1 is observed causing a transition to state $T_1 = f(T_0, X_1)$ and a decision $d_1 = d(T_1)$. At time two X_2 is observed causing a transition to state $T_2 = f(T_1, X_2)$ and a decision $d_2 = d(T_2)$ and, in general,

$$\begin{aligned} T_n &= f(T_{n-1}, X) \in S \\ d_n &= d(T_n) \in D. \end{aligned} \tag{1.1}$$

We measure the size of memory by the number of states in $\mathcal{S}$. That is memory is of size m if $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$. We will prefer to represent $\mathcal{S}$ by $\{1, 2, \ldots, m\}$ for notational convenience.

Letting e_n equal 0 or 1 accordingly as $d_n = H_t$ or $d_n \neq H_t$, where H_t denotes the true hypothesis, define

$$P_n(\mathcal{A}) = Ee_n \tag{1.2}$$

and

$$P_\infty(\mathcal{A}) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} P_n(\mathcal{A}). \tag{1.3}$$

Further, for a given problem (i.e., π_0, π_1, P_0, P_1), memory size m and sample size $n \leq \infty$ define

$$P^*(m, n) = \inf_{f,d,T_*} P_n(\mathcal{A}) . \tag{1.4}$$

The value of $P^*(m, n)$ depends on the class of algorithms considered. For example, should randomized mappings f and d be allowed? At first it would seem that since randomized mappings tend to add noise, their use would only increase error probability. However this is not so in general, and we will explore the differences between deterministic algorithms and randomized algorithms. Of course, when a randomized algorithm is used, the randomization must be independent of the data to avoid hidden memory.

The infimum in (1.4) will be taken over all m-state algorithms, randomized and deterministic. Therefore we define

$$P_d^*(m, n) = \inf_{f,d,T_*} P_n(\mathcal{A}) \tag{1.5}$$

where in (1.5) the infimum is over the class of deterministic automata. Clearly for all problems, m and n

$$P^*(m, n) \leq P_d^*(m, n) . \tag{1.6}$$

In Section 2 we will review the work of Hellman and Cover [1, 3], finding explicit expressions for $P^*(m, \infty)$. We will see that for large memory sizes $P^*(m, \infty)$ goes to zero exponentially in m. This is a dual result to that for infinite memory but finite sample size. There, for large sample sizes, probability of error goes to zero exponentially in the sample size.

In Section 3 we will explore the differences between randomized and deterministic automata when $n = \infty$. Following Hellman and Cover [6] we will find that there exist problems for which randomized rules are arbitrarily better than deterministic rules. However we will then see [7] that deterministic rules are asymptotically optimal. These two statements seem contradictory but, in fact, are not when precisely stated.

The first statement becomes: For any $m < \infty$ and $\delta > 0$ there exists a problem such that $P^*(2, \infty) \leq \delta$, while $P_d^*(m, \infty) \geq 1/2 - \delta$.

The second statement becomes: For any problem there exists a $b < \infty$ such that for all m, $P_d^*(m2^b, \infty) \leq P^*(m, \infty)$. That is, adding b bits to memory makes deterministic rules competitive with randomized rules.

Then in Section 4 we review recent work of Flower [8] and Freedman [9] on the behaviour of $P^*(m, n)$ as a function of sample size n.

2. BEHAVIOUR OF $P^*(m, \infty)$

In [1] Hellman and Cover show that

$$P^*(m, \infty) = \min \left\{ \frac{2\sqrt{\pi_0\pi_1\gamma^{m-1} - 1}}{\gamma^{m-1} - 1}, \pi_0, \pi_1 \right\} \tag{2.1}$$

where γ is a measure of the distance between H_0 and H_1. When $\pi_0 = \pi_1 = 1/2$ we have

$$P^*(m, \infty) = \frac{1}{\gamma^{(m-1)/2} + 1} . \tag{2.2}$$

The parameter γ is defined by

$$\gamma = \bar{l}/\underline{l} > 1 \tag{2.3}$$

where

$$\bar{l} = \sup \frac{P_0(A)}{P_1(A)} \tag{2.4}$$

$$\underline{l} = \inf \frac{P_0(A)}{P_1(A)} \tag{2.5}$$

and the supremum and infimum are over all measurable sets A such that $P_0(A) + P_1(A) > 0$. That is, $\bar{l}$ is the essential supremum on the likelihood ratio $l(X)$, while $\underline{l}$ is the essential infimum. Clearly $\bar{l} > 1$, $\underline{l} < 1$ and $\gamma > 1$.

Since $\gamma > 1$ we see that

$$P^*(m, \infty) \sim r^m \tag{2.6}$$

where $r = \gamma^{-1/2} < 1$. Thus $P^*(m, \infty)$ goes to zero exponentially in m.

The form of the optimal machine is of interest and is derived in [1]. Here we will merely examine its structure. Let

$$\mathcal{H}_\varepsilon = \{x \in \mathcal{X} : l(x) \geq [(1/\bar{l}) + \varepsilon]^{-1}\} \tag{2.7}$$

and

$$\mathcal{I}_\varepsilon = \{x \in \mathcal{X} : l(x) \leq \underline{l} + \varepsilon\} . \tag{2.8}$$

Thus for small ε, $\mathcal{H}_\varepsilon$ and $\mathcal{I}_\varepsilon$ have likelihood ratios close to $\bar{l}$ and $\underline{l}$ respectively. Furthermore $P_0(\mathcal{H}_\varepsilon) > 0$ and $P_1(\mathcal{I}_\varepsilon) > 0$ by the definitions of $\bar{l}$ and $\underline{l}$.

Consider the machine which transits from state i to $i + 1$ if $X \in \mathcal{H}_\varepsilon$ and $i \leq m - 1$; from i to $i - 1$ if $X \in \mathcal{I}_\varepsilon$ and $i \geq 2$; and stays in the same state otherwise. This machine changes state only on a subsequence of high information observations, thereby making maximal use of its limited memory to store information. However it is seen that states 1 and m are the states in which we are most certain of our decisions. Therefore once in an end state we would like the machine to stay there for a long time before leaving. Using randomization we can effect this.

If in state 1 and $X \in \mathcal{H}_\varepsilon$ move to state 2 with small probability δ (and stay in state 1 with probability $1 - \delta$). If in state m and $X \in \mathcal{I}_\varepsilon$ move to state $m - 1$ with probability $k\delta$ (and stay in state m with probability $1 - k\delta$) Leave all other transitions as they were.

The purpose of not fixing $k = 1$ is to allow asymmetries in the structure of the machine to compensate for asymmetries in the statistics (e.g., $\pi_0 \neq \pi_1$, etc.). For symmetric problems the optimal value is $k = 1$.

In [1] it is shown that with k properly chosen, as $\varepsilon, \delta \to 0$, $P_\infty(\mathcal{A}) \to \to P^*(m, \infty)$ so that this is an optimal class of algorithms. The simple structure of this class is pleasing, and somewhat unexpected, since no constraints were placed on the "complexity" of the mapping f.

Note that randomization is necessary to approach $P^*(m, \infty)$ if $\mathcal{X}$ is a finite space. If $\mathcal{X}$ is continuous, randomization is not necessary (i.e., $P^*(m, \infty) = P_d^*(m, \infty)$) since then we can obtain low probability deterministic transitions.

3. RANDOMIZATION

We have just noted that randomization is not necessary if $\mathcal{X}$ is a continuous space. Here we examine the differences between randomized and deterministic rules when $\mathcal{X}$ is discrete.

First let us show that randomized rules are arbitrarily better in the sense that for any $m < \infty$ and $\delta > 0$ there exists a problem such that $P^*(2, \infty) \leq \delta$ while $P_d^*(m, \infty) \geq 1/2 - \delta$. Not surprisingly it suffices to consider Bernoulli distributions where $\mathcal{X} = \{0, 1\}$ since these are the antithesis of continuous distributions.

Let $p_0 = Pr(X = 1 \mid H_0)$ and $p_1 = Pr(X = 1 \mid H_1)$, and $q_0 = 1 - p_0$, $q_1 = 1 - p_1$. If $p_0 = 3/4$ and $p_1 = 1/4$ then $l(X = 1) = p_0/p_1 = 3$ and $l(X = 0) = 1/3$. Thus $\bar{l} = 3$, $\underline{l} = 1/3$ and $\gamma = 9$. If $\pi_0 = \pi_1 = 1/2$ we have, from (8), that $P^*(5, \infty) = 1/82$.

If $p_0 = 1 - 10^{-10}$ and $p_1 = 1 - 10^{-9}$ we have $\bar{l} = p_0/p_1 > 1$, $\underline{l} = q_0/q_1 = 1/10$ and $\gamma > 10$. Thus $P^*(5, \infty) < 1/101$. The optimal five state randomized machine actually has a lower probability of error for this problem than when $p_0 = 3/4$, $p_1 = 1/4$. It is fairly obvious that a five state deterministic machine cannot achieve an error probability much below $1/2$. Thus $P^*(5, \infty) \ll P_d^*(5, \infty)$. We can make the discrepancy even worse. If we make $p_0 = 1 - 10^{-19}$ and leave $p_1 = 1 - 10^{-9}$ then $\gamma \approx 10^{10}$ and $P^*(5, \infty) \approx 10^{-20}$, while $P_d^*(5, \infty)$ is still close to $1/2$. On the other hand if we keep the ratio q_0/q_1 fixed but make p_0 and p_1 even closer to 1 this does not affect γ and hence $P^*(m, \infty)$, but does cause $P_d^*(5, \infty)$ to be even closer to $1/2$ (e.g., $p_0 = 10^{-20}$, $p_1 = 1 - 10^{-19}$). By combining these two effects we can, for any $m < \infty$, make γ arbitrarily large (and thus $P^*(2, \infty)$ arbitrarily small), and yet keep $P_d^*(m, \infty)$ arbitrarily close to $1/2$.

Now let us turn to the other statement: For any problem, there exists a $b < \infty$ such that, for all m, $P_d^*(m2^b, \infty) \leq P^*(m, \infty)$. That is, deterministic rules lose at most b bits. The basic idea [7] is to demonstrate a class of deterministic machines which has error probability go to zero exponentially in m, the number of states. Say $P(m) \sim s^m$ for some $s < 1$. Since $P^*(m, \infty) \sim r^m$ where $r = \gamma^{-1/2} < 1$ we know that $s \geq r$. However there must exist $k < \infty$ such that $s^k < r$. Setting $b = [\log_2 k]^+$ yields the desired result.

Since the solution to the Bernoulli problem (i.e., $\mathcal{X} = \{0, 1\}$) is easily extended to the general problem we only consider it in this review. If $p_0 > 1/2 > p_1$ then the algorithm which moves up on state when $X = 1$ (unless in state m), down one state when $X = 1$ (unless in state 1), and decides H_0 in states $i > m/2$ and H_1 in states $i \leq m/2$ has probability of error $P(m)$ which goes to zero exponentially in m. That is $P(m) \sim s^m$ where $s = \max\{(q_0/p_0)^{1/2}, (p_1/q_1)^{1/2}\}$.

If $p_0 > p_1 > 1/2$ (or $1/2 \geq p_0 > p_1$) then the above machine does not have $P(m) \sim s^m$ for any $s < 1$. This is because under both hypotheses there is a drift to higher (respectively lower) numbered states. Since the problems $p_0 > p_1 \geq 1/2$ and $1/2 \geq p_0 > p_1$ are equivalent by interchanging the roles of $X = 0$ and $X = 1$, we consider only the former.

We can always find integers $N_1 > N_2$ such that $(p_0^{N_1}/q_0^{N_1}) > 1$ and $(p_1^{N_1}/q_0^{N_2}) < 1$. Thus a machine which moves up one state whenever a block of N_1 observations consists of all 1's and down one state whenever the first N_2 observations of the block are all 0's, has a drift toward higher numbered states under H_0 and a drift toward lower numbered states under H_1. If such a machine were to decide H_0 in states $i > m/2$ and H_1 in states $i \leq m/2$ it would have $P(m) \sim s^m$ where $s = \max\{(q_0^{N_1}/p_0^{N_1})^{1/2}, (p_1^{N_1}/q_1^{N_2})^{1/2}\} < 1$. Of course this machines requires a state transition function f which maps $\mathcal{S}\mathcal{X}^{N_1}$ to $\mathcal{S}$, violating our definition for an m-state, finite memory decision rule. However it is possible to implement such a rule using a finite memory rule with at most $(2^{N_1} - 1)\,m$ states. (Essentially construct a machine with m super-states each comprised of $2^{N_1} - 1$ regular states.) Thus $P(m) \sim [s^{1/(2^{N_1}-1)}]^m$ which still has the desired form.

The extension of these results to non-Bernoulli problems is straightforward since we can always quantize the observation space more coarsely. The extension to more than two hypotheses is also not too difficult [7].

Horos [10] has also examined the differences between randomized and deterministic machines. He expands the decision space D to $\{H_0, H_1, \Phi\}$ where $d_n = \Phi$ indicates that no decision is made. If such decisions incur no cost, he finds that $P_d^*(m, \infty) = P^*(m, \infty)$ for symmetric problems (i.e., $k = 1$ is optimal).

4. FINITE SAMPLE SIZE

All analysis up to now has dealt with infinite sample sizes. The importance of the finite sample size problem is evident, but one can no longer use equilibrium conditions, making analysis rather difficult.

Flower [8] has investigated the symmetric Bernoulli problem (i.e., $p_0 = 1 - p_1$ and $\pi_0 = \pi_1 = 1/2$) for finite sample sizes. Using a computer search he has found that the optimal rule moves at most one state per transition. If $X = 1$ (or $X = 0$) the transition is to the next higher (or lower) numbered state, or to the same state. The probability of such transitions is 1 if the move is away from the middle of the machine, but is strictly less than one if the move is toward the middle of the machine and the sample size $n \gg m$. Experimentally it was found that $P^*(m, n) - P^*(m, \infty)$ approaches zero approximately as $1/n$. Analysis indicates that the actual form is $(\log n)/n$.

Freedman [9] has studied the special problem where $\mathcal{X}$ is the real line, $P_0 = \mathfrak{N}(+1,1)$, $P_1 = \mathfrak{N}(-1,1)$, $\pi_0 = \pi_1 = 1/2$ and $m = 2$. For this Gaussian problem $\gamma = \infty$ so that $P^*(2, \infty) = 0$. This would seem to indicate that a two state memory is not better than an infinite memory. However $P^*(2, n) \sim \exp[-2\sqrt{2 \ln n}]$, whereas $P^*(\infty, n) \sim \exp[-\alpha n]$ where $\alpha > 0$. Thus the difference between $m = 2$ and $m = \infty$ is quite marked for finite sample sizes.

5. DISCUSSION

It is seen that the theory is very general when applied to randomized algorithms and infinite sample sizes. Although deterministic algorithms are not as easily analyzed, their practical importance tempts us to exert additional effort. Similarly, finite sample size problems lack theoretical simplicity but not practical applications.

Future work will thus probably continue to pursue these two lines of research. In addition it may be possible to take account of the complexity of the functions f and d. As noted it is surprising that with no constraint on complexity the optimal f and d are rather simple for the infinite sample, randomized algorithm problem. It is therefore possible that even when a suitable such constraint is imposed, there will be little change in the optimal algorithm.

REFERENCES

1. Hellman, M. E. and Cover, T. M., Learning with finite memory. *Ann. of Math. Stat.* **41** (June 1970) 765–782.
2. Cover, T. M. and Hellman, M. E., The two armed bandit problem with time-invariant finite memory. *IEEE Trans. on Inf. Theory* **IT-16** (March 1970) 185–195.
3. Hellman, M. E. and Cover, T. M., Finite memory decision schemes. *Problemy Peredachi Informatsii* **6** (1970) 21–30, (in Russian).
4. Robbins, H., A sequential decision problem with a finite memory. *Proc. Nat. Acad. Sci.* **42** (1956) 920–923.
5. Tsetlin, M. L., On the Behavior of Finite Automata in Random Media. *Avtomat. Telemeh.* **22** (1961) 1345–1354.
6. Hellman, M. E. and Cover, T. M., On memory saved by randomization. *Ann. Math. Stat.* **42**, (1971) 1075–1078.
7. Hellman, M. E., The effects of randomization on finite memory decision schemes. *IEEE Trans. on Inf. Theory.*
8. Flower, R. A., *Hypothesis Testing with Finite Memory in Finite Time.* M. S. thesis, Dept. of Electrical Engineering, Massachusetts Institute of Technology 1971; Flower, R. A. and Hellman, M. E., Hypothesis testing with finite memory in finite time, to appear *IEEE Trans. on Inf. Theory*, 1972.
9. Freedman, M. A., *A Finite Memory, Finite Time Gaussian Hypothesis Testing Problem*, M. S. thesis, Dept. of Electrical Engineering, Massachusetts Institute of Technology, to be submitted 1971.
10. Horos, J. A., *Deterministic Finite Memory Learning Algorithms*, M. S. thesis, Dept. of Electrical Engineering, Massachusetts Institute of Technology, 1971.