sets, respectively; $H(n)$ and $T(n)$ are related by the output function e, and the state function f is defined by $T(n+1) = f[T(n), X(n+1)]$. Time flow is discrete. For our purposes, we say $\sum$ is deterministic if e and f are deterministic. If either involves randomization, we have a stochastic automaton.

Three interpretations of a finite-memory (FM) constraint are possible.

1) FM: If we ignore the distinction between read-only and read-and-write memory in computing memory requirements, then all randomizers consume memory whether they are data-dependent or not. In this case truly FM automata should have e and f purely deterministic (procedure 5), Section III).

2) FM: If we make the above-mentioned distinction and do not include data-independent memory in computing memory needs, then an FM automaton can be defined with the functions $H(n) = e'[T(n), y_e(n)]$ and $T(n+1) = f'[T(n), X(n+1), y_f(n)]$ where e' and f' must be deterministic functions and y_e and y_f are the outputs of suitable data-independent randomizers (procedure of [1], i.e., 4), Section III).

3) FM: If we identify as memory the number of states of the automaton alone, without regard to the memory needed for the randomizer, then e and f can be any randomized functions (procedure 1), Section III).

Procedures 2) and 3) cannot be optimal finite-memory devices under any of these definitions.

Finally, we take this occasion to present the following conjecture in regard to procedure 5). For any nonrandomized time-invariant transition rule applicable to a single simple-hypothesis testing problem defined in the second paragraph of Section I, the best P_e is given by

$$P_e = (1 + \gamma^{1/2m})^{-1}. \tag{5}$$

We emphasize that we are considering solution of just one such problem, and not a collection of them.

V. Concluding Remarks

In spite of our criticism of some of their important aspects, [1] and [2] should be viewed as presenting some contributions needed for the beginnings of a theory of finite-memory decision processes. Any future work in this area will have to follow their trail, as witness our attempts in Section IV. In addition, as we remarked in [7], Cover and Hellman present incisive solutions to some problems in learning automata.

In summary, we have attempted to demonstrate the inadequacies in the finite-memory solutions advanced in [1]. We point out that randomization requires memory, and this fact should be taken into account in the design. We show how some sort of degraded performance or hidden memory creeps into almost all the suggested procedures. Finally, we define finite memory in increasing order of permissiveness and indicate how some of the procedures can be fitted in this formalism.

Acknowledgment

The author is indebted to T. Harley, C. Gumacos, and K. Abend for many useful discussions regarding this subject.

B. Chandrasekaran
Dept. Computer and Inform. Sci.
Ohio State University
Columbus, Ohio 43210

References

[1] T. W. Cover and W. Hellman, "The two-armed bandit problem with time-invariant finite memory," *IEEE Trans. Information Theory*, vol. IT-16, pp. 185–195, March 1970.
[2] M. Hellman and T. M. Cover, "Learning with finite memory," *Ann. Math. Stat.* (to be published).
[3] V. I. Varshavskii and I. P. Vorontsova, "On the behavior of stochastic automata with a variable structure," *Automation Remote Control* (English transl. of Russian, *Auto. i Telemekh*), vol. 24, pp. 327–333. March 1963.
[4] B. Chandrasekaran and D. W. C. Shen, "On expediency and convergence in variable structure automata," *IEEE Trans. System Science and Cybernetics*, vol. SSC-4, pp. 52–60, March 1968.
[5] T. M. Cover, "Hypothesis testing with finite statistics," *Ann. Math. Stat*, vol. 40, pp. 828–835, 1969.
[6] B. Chandrasekaran, "Optimal strategies for automata games," *Proc. Internatl. Syst. Sci. Conf.* (Honolulu, Hawaii,) January 1970.
[7] ——, review of "Formulation of learning automata and automata games," by Fu and Li, *Computing Rev.* (to be published).

Finite-Memory Hypothesis Testing—Comments on a Critique

Several years of frustration with various definitions of finite memory have led us to write the papers [1], [2] referred to by Chandrasekaran [3]. In arriving at our formulation, we considered many of the same questions that have been raised in [3]. Most of the considerations that bother Chandrasekaran we feel to be of minor importance; and it is our difference in point of view that we hope to make clear in this correspondence.

Before proceeding, let us consider the basic hypothesis testing problem in which a sequence of independent identically distributed random variables $X_1, X_2, \cdots$ is drawn according to some unknown probability measure $\mathbf{P}$. To fix ideas, consider the following four hypothesis-testing problems.

Problem 1—General Case

$$H_0 : \mathbf{P} = \mathbf{P}_0$$
$$H_1 : \mathbf{P} = \mathbf{P}_1$$

where $\mathbf{P}_0$ and $\mathbf{P}_1$ are arbitrary known probability measures.

Problem 2—Coin-Tossing Case

$$H_0 : p = p_0$$
$$H_1 : p = p_1$$

where $p = \Pr\{X = \text{heads}\}$ denotes the bias of a coin.

Problem 3—Symmetric-Coin-Tossing Case

$$H_0 : p = p_0$$
$$H_1 : p = p_1$$

with $p_0 = 1 - p_1$, where p is the bias of a coin.

Problem 4—Compound-Hypothesis-Testing Case

$$H_0 : p \geq \tfrac{1}{2}$$
$$H_1 : p \leq \tfrac{1}{2}.$$

Let the data be summarized after each new observation by an m-valued statistic $T \in \{1, 2, \cdots, m\}$, which is updated according to the rule $T_n = f(T_{n-1}, X_n)$. The goal is to minimize the asymptotic proportion of errors resulting from a decision rule $d(T_n)$. The comments raised about the complexity of computation of the state transition function f are well taken. The primary difference in emphasis is that we do not wish to count any of this complexity in memory, while Chandrasekaran wishes to count at least the randomized part.

The first formulation encompasses 2 and 3 and is the formulation we have solved in [1] and [2]. If the probability measures $\mathbf{P}_0$ and $\mathbf{P}_1$ have probability density functions (more precisely, if $\mathbf{P}_0$ and $\mathbf{P}_1$ have no atoms) then the ϵ-optimal learning algorithm is deterministic and requires no randomization. (See, for example, the case of two normal distributions as given in Section 4 of [1].) Quoting from Chandrasekaran [3], on the implementation of the memory-updating algorithm f, "When we examine how

Manuscript received February 3, 1970. This work was supported by Stanford University, Stanford, Calif., under Contract AF 49(638)1517, and by the Massachusetts Institute of Technology, Cambridge, under joint Services Electronics Program Contract DA 28-043-AMC-02536(E).

this might be implemented in a digital computer, no problems are encountered until we come to the randomizer." Thus, we are seemingly in agreement about the optimality of the procedure in [1] in the vast majority of cases, i.e., those in which X has a continuous distribution.

However, Chandrasekaran wishes to exclude the continuous case by remarking in Section 2, "... introduction of continuous variables in the model makes implementation one of infinite memory, by requiring infinite precision in the measurement." This makes no sense to us. What has memory to do with high-precision measurements? In any case, the only operations we require are the determination of whether the continuous observation variable X lies in certain sets H or T. The boundaries of H and T need only be roughly defined, and infinite precision is not necessary (see Example 3 in [1] in which H and T are half-spaces). Even if precision measurements were necessary, it is questionable whether memory is necessary to obtain them.

It is only when X has a discrete distribution that we must introduce the seemingly unnecessary artifice of randomization in order to bring these hypothesis-testing problems under the umbrella of the theory. Here, randomization plays much the same role in learning with finite memory as it does in the Neyman–Pearson decision theory, where the continuous and discrete cases are unified by the introduction of artificially randomized decision rules. To our minds, the increase in simplicity and generality of the theory in the randomized case helps establish the "rightness" of the formulation.

Let us now turn our attention to the conjecture made at the end of Section 4 of [3] that leads to the statement that "... the saving achieved by the proposed procedure over the usual intuitive one (equivalent to setting $\delta = 1$) is exactly one bit, whatever the order of magnitude of m." This conjecture is false. In fact, depending on the problem, randomization can yield an arbitrarily large saving in the number of states needed to achieve a given probability of error. Chandrasekaran appears to make this conjecture with respect to the difficult hypothesis-testing model 4 (the compound case, $p \geq 1/2$ versus $p \leq 1/2$). The conjecture is certainly false in this case. However, from the discussion, it is reasonably clear that the intended problem is Problem 3; p versus $1 - p$. Even in this simplest case, the conjecture is mildly wrong—the saving is not "exactly" one bit, but less than or equal to one bit (to see this, allow transitions to nonadjacent states in accordance with Robbins' [4] design for the two-armed bandit problem). The one-bit bound is very appealing, but is misleading because of the special symmetry of the problem. In the simple case of Problem 2, $p = p_o$ versus $p = p_1$, it can be shown that randomization yields arbitrarily large improvement. For example, let $p_o = 0.01$ and $p_1 = 0.001$. In this case, the intuitive deterministic procedure shown in Fig. 1 of [3] (where $\delta = 1$) has a probability of error of $P_e \approx 0.5$ regardless of m. (The drift due to p_o or p_1 causes the steady-state probability of the left-most state to be close to one under either p_o or p_1, and thus, no discrimination between hypotheses is achieved.) On the other hand, with randomization, a 2-state memory achieves $P_e \cong 0.01$. Randomization allows the drift to be cancelled by allowing a higher probability of exit from the left end state than from the right end state. Even the optimal m-state deterministic automaton will require $m \gg 2$ to achieve $P_e \approx 0.01$. Since this example may be made arbitrarily extreme, we hope that we have made our point that randomization results in a great practical saving in memory. Thus, the one-bit bound on memory improvement is either misleading or wrong.

Summing up, Chandrasekaran makes some interesting points concerning "finite memory," "finite computation," and the implementation of our rules. We agree that one must always be wary of directly applying a mathematical theory to a real life problem. However, we feel that randomized rules are of importance because of their power in generalizing and simplifying the theory and in providing insight into the theory of deterministic rules. Finally, the question of whether or not randomization requires memory is one whose answer depends on the application involved.

Thomas M. Cover
Dept. Elec. Engrg. and Stat.
Stanford University
Stanford, Calif. 94305
Martin E. Hellman
Dept. Elec. Engrg.
M. I. T.
Cambridge, Mass. 02139

References

[1] M. E. Hellman and T. M. Cover, "Learning with finite memory," *Ann. Math. Stat.*, vol. 41, June 1970.
[2] T. M. Cover and M. E. Hellman, "The two-armed bandit problem with time-invariant finite memory," *IEEE Trans. Information Theory*, vol. IT-16, pp. 185–195, March 1970.
[3] B. Chandrasekaran, "Finite memory hypothesis testing—A critique," this issue, pp. 494–496.
[4] H. Robbins, "A sequential decision problem with a finite memory," *Proc. Natl. Acad. Sci.*, vol. 42, pp. 920–933, 1956.

On the Detection of a Sudden Change in System Parameters

***Abstract*—This correspondence deals with the problem of detecting a sudden change in system parameters by means of noisy observations made on the system. The solution given here is dependent on the classical Bayes criterion. However, by forming an auxiliary sequence of 0's and 1's as the observations are taken, it is shown, at most, three log-likelihood numbers need to be updated recursively at any stage. An example is given to illustrate the principle.**

I. Problem Formulation

The problem of detecting a sudden change in system parameters is important in such applications as fault detection in a gyronavigational system [1]. The precise formulation is as follows.

A1) $(x_1, x_2, \cdots, x_n)$ is a statistically independent finite sequence of observations (scalar or vector) made on a system at times $t_1, t_2, \cdots, t_n$ $(t_1 < t_2 < \cdots < t_n)$.

A2) The system undergoes a sudden change at some time $t = t_c$ that does not coincide with any of the sampling instants.

A3) The a priori probabilities that the change occurs in any of the $(n + 1)$ intervals $(-\infty, t_1)$, (t_1, t_2), (t_2, t_3), $\cdots$, (t_{n-1}, t_n), $(t_n, +\infty)$ are all equal. In other words, if H_i is the hypothesis that the change has occurred in the ith of these intervals

$$p(H_1) = p(H_2) = \cdots = p(H_n) = p(H_{0n}) = \frac{1}{n+1},$$

where $H_{0n} = H_{n+1}$ is the hypothesis that no change has occurred up to t_n.

A4) The probability density functions of the random observation x before and after the change takes place are completely known and denoted by $p_0(x)$ and $p_1(x)$, respectively.

It is required to formulate a sequential test to accept, for every i, one of the hypotheses $H_1, H_2, \cdots, H_i, H_{0i}$ after observing $(x_1, x_2, \cdots, x_i)$. Here H_{0i} is the hypothesis that no change has occurred up to t_i and, if accepted, will lead to the further observation x_{i+1}. The test will terminate at the first t_i at which any one hypothesis among $H_1, H_2, \cdots, H_i$ is accepted.

Manuscript received November 26, 1969; revised February 3, 1970. This research was supported by the National Aeronautics and Space Administration under Grant NGR 22-007-143.