The Annals of Mathematical Statistics
1971, Vol. 42, No. 3, 1075–1078

ON MEMORY SAVED BY RANDOMIZATION

BY MARTIN E. HELLMAN[1] AND THOMAS M. COVER[2]
Massachusetts Institute of Technology
and Stanford University

0. Summary. It is known that deterministic automata are generally not optimal in the problem of learning with finite memory. It is natural to ask how much memory is saved by randomization. In this note it is shown that the memory saving is arbitrarily large in the sense that for any memory size $m < \infty$, and $\delta > 0$, there exist problems such that all m-state deterministic algorithms have probability of error $P(e) \geqq \frac{1}{2} - \delta$, while the optimal two-state randomized algorithm has $P(e) \leqq \delta$.

1. Statement of problem and results. This note is concerned with finite memory learning algorithms for the two-hypothesis testing problems of the type discussed in Hellman and Cover [4]. The question of how much memory is saved by using randomized instead of deterministic algorithms has been raised by Chandrasekaran [1], [2] and Cover and Hellman [3] and is treated in this note.

We will find problems for which the probability of error for all m-state deterministic automata is arbitrarily close to $\frac{1}{2}$, while the optimal two-state randomized automaton has a probability of error arbitrarily close to 0.

For each $m < \infty$ a problem will be found for which the stationary distribution for any m–state deterministic machine is approximately the same under either hypothesis. Such an example would be a ten–state hypothesis test of a Bernoulli sequence with parameter $p_0 = 1 - 10^{-30}$ vs. a Bernoulli sequence with parameter $p_1 = 1 - 10^{-20}$. Intuitively speaking, the drift of the process into memory states entered by the observation $X = 1$ is overwhelming under either hypothesis, thus creating nearly identical stationary distributions. However, by introducing randomization a large difference in the stationary distributions can be achieved. For example, on all transitions with $X = 1$, stay in the original state with probability $1 - 10^{-25}$ and make the indicated transition with probability 10^{-25}.

From the foregoing discussion it would seem sufficient to consider Bernoulli distributions, and indeed this is so. Let $X_1, X_2, \cdots$ be independent identically distributed random variables drawn according to the distribution

$$X = \begin{matrix} \text{Heads, with probability} & p \\ \text{Tails, with probability} & 1 - p = q. \end{matrix} \tag{1}$$

Received August 31, 1970

[1] Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts. This work was supported by NSF Grant GK5800 at M.I.T.

[2] Departments of Statistics and Electrical Engineering, Stanford University, Stanford, California, and Bell Telephone Laboratories, Murray Hill, New Jersey. Preliminary support was provided under contract AF 49(638)1517. This work was performed at Bell Laboratories.

Consider the hypothesis testing problem

$$H_0 : p = p_0; \quad \text{vs.} \quad H_1 : p = p_1, \tag{2}$$

with $\frac{1}{2} < p_1 < p_0 < 1$, and equal prior probabilities $P\{H_0\} = P\{H_1\} = \frac{1}{2}$. It is assumed that p_0 and p_1 are known.

Now let us consider learning algorithms for the Bernoulli problem above. An algorithm of memory size m consists of a state space $S = \{1, 2, \cdots, m\}$, an initial state $j \varepsilon S$, two $m \times m$ stochastic matrices P_H and P_T which are the state transition matrices under Heads (H) and Tails (T) respectively, and a partition $\{S_0, S_1\}$ of S. The interpretation is that if, at time $n-1$, the automaton is in state i and $X_n =$ Heads (resp. Tails), then a transition to state k will be made with probability $(P_H)_{ik}$(resp. $(P_T)_{ik}$); decision H_t is made if the current state is contained in S_t, $t = 0, 1$. An algorithm is deterministic if P_H and P_T contain only zeros and ones.

Now suppose that the automation is started in state j. Define

$$P(p) = pP_H + qP_T. \tag{3}$$

Thus, if the probability of heads is p, the expected asymptotic proportion of visits to state i is given by

$$\mu_i(j, p, P_H, P_T) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} (P^n(p))_{ji}. \tag{4}$$

Since $P(p_0)$ and $P(p_1)$ are the state transition matrices under H_0 and H_1 respectively, the asymptotic probability of error, given initial state j, and decision scheme $\{S_0, S_1\}$, is given by

$$P_e(j, P_H, P_T, S_0, S_1) = \tfrac{1}{2}\textstyle\sum_{i\in S_1} \mu_i(j, p_0, P_H, P_T) + \tfrac{1}{2}\textstyle\sum_{i\in S_0} \mu_i(j, p_1, P_H, P_T). \tag{5}$$

Define $P^*(m)$ to be the greatest lower bound on the probability of error over all randomized m-state learning algorithms, and $P_d{}^*(m)$ to be the greatest lower bound over all m-state deterministic algorithms. (A deterministic algorithm is one for which the matrices P_H and P_T each have one and only one 1 in every row, the other elements all being zero. Note that there are thus m^{2m} deterministic (P_H, P_T) pairs.) In the context of this coin tossing problem, $P^*(m)$ and $P_d{}^*(m)$ are explicitly given by

$$P^*(m) = \inf_{j, P_H, P_T; S_0, S_1} P_e(j, P_H, P_T, S_0, S_1) \tag{6}$$

where the infimum over P_H and P_T is taken over all *stochastic* $m \times m$ matrices P_H and P_T; and

$$P_d{}^*(m) = \inf_{j, P_H, P_T; S_0, S_1} P_e(j, P_H, P_T, S_0, S_1) \tag{7}$$

where the infimum over P_H and P_T is taken over all $m \times m$ matrices corresponding to *deterministic* transition rules. Clearly,

$$P_d{}^*(m) \geqq P^*(m) \tag{8}$$

for any problem and memory size m. We shall prove the following:

THEOREM. *For any* $m = 2, 3, 4, \cdots,$ *and for any* $\delta > 0$, *there exist probabilities* p_0, p_1, *such that*

$$P^*(2) \leqq \delta, \quad P_d{}^*(m) \geqq \tfrac{1}{2} - \delta. \tag{9}$$

PROOF. From [4] we know that there exists a randomized two-state algorithm which achieves

$$P^*(2) = 1/(1+\gamma^{\frac{1}{2}}), \tag{10}$$

where

$$\gamma = p_0 q_1 / p_1 q_0 > 1. \tag{11}$$

The algorithm which achieves this bound is given by

$$P_H = \begin{bmatrix} 1-\Delta & \Delta \\ 0 & 1 \end{bmatrix}, \quad P_T = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \tag{12}$$

where (see [4], (56), (57), on page 776)

$$\Delta = (q_0 q_1 / p_0 p_1)^{\frac{1}{2}}, \tag{13}$$

and the decision regions are $S_0 = \{2\}$, $S_1 = \{1\}$.

We wish to find p_0, p_1 such that $\gamma = (p_0 q_1 / p_1 q_0) \gg 1$, and thus $P^*(2) \ll 1$; while at the same time

$$|\mu_i(j, p_0, P_H, P_T) - \mu_i(j, p_1, P_H, P_T)| \leqq \varepsilon \tag{14}$$

for all $i, j \in S$ and for all deterministic P_H, P_T. As we shall see (14) implies that $P_e(j, P_H, P_T, S_0, S_1) \geqq \frac{1}{2} - m\varepsilon$ for any partition (S_0, S_1). Thus the two hypotheses cannot be effectively discriminated by a deterministic algorithm.

It would not be difficult to find p_0, p_1 yielding (14) under the constraint $\gamma \gg 1$ if $\mu_i(j, p, P_H, P_T)$ were uniformly continuous in p, for $0 \leqq p \leqq 1$. However, this is not the case, and μ_i can be discontinuous at $p = 0$ or 1.

We resort to the following simple approach. Let $\boldsymbol{\mu}(p) \varepsilon [0, 1]^{m^{2m+2}}$ be an ordered m^{2m+2} tuple corresponding to the values that $\mu_i(j, p, P_H, P_T)$ takes on for fixed p as (i, j) ranges over $\{1, 2, \cdots, m\} \times \{1, 2, \cdots, m\}$ and (P_H, P_T) ranges over the set of m^{2m} possible deterministic transition matrix pairs. Let us put a sup norm on $[0, 1]^{m^{2m+2}}$. Thus, in particular,

$$||\boldsymbol{\mu}(p) - \boldsymbol{\mu}(p')|| = \max_{i,j,P_H,P_T} |\mu_i(j, p, P_H, P_T) - \mu_i(j, p', P_H, P_T)|. \tag{15}$$

Consider the sequence $\boldsymbol{\mu}(1-\alpha^k)$, $k = 1, 2, \cdots$, for some fixed $0 < \alpha < 1$. Since $[0, 1]^{m^{2m+2}}$ is compact under the sup norm, then, by the Bolzano–Weierstrass Theorem this sequence must have a cluster point, $\boldsymbol{\mu}^*$, say. Hence, for any $\varepsilon > 0$,

$$||\boldsymbol{\mu}(1-\alpha^k) - \boldsymbol{\mu}^*|| < \varepsilon, \tag{16}$$

infinitely often. In particular, (16) implies

$$|\mu_i(j, 1-\alpha^k, P_H, P_T) - \mu_i{}^*(j, P_H, P_T)| < \varepsilon, \qquad \forall_{i,j,P_H,P_T} \tag{17}$$

for infinitely many values of k, where $\mu_i^*(j, P_H, P_T)$ may depend on the chosen value of α. For future reference note that if $\boldsymbol{\mu}^*$ is a cluster point then it necessarily follows that

$$\sum_{i=1}^{m} \mu_i^*(j, P_H, P_T) = 1, \tag{18}$$

for all j, P_H, P_T. This completes the preliminaries.

Let k_1, k_2, $k_1 > k_2$ be positive integers and set

$$\begin{aligned} p_0 &= 1-\alpha^{k_1} \\ p_1 &= 1-\alpha^{k_2}, \end{aligned} \tag{19}$$

where as before $0 < \alpha < 1$. Then, $p_0 > p_1$, and

$$\gamma = \frac{p_0(1-p_1)}{p_1(1-p_0)} > \frac{1-p_1}{1-p_0} = \frac{\alpha^{k_2}}{\alpha^{k_1}} = \frac{1}{\alpha^{(k_1-k_2)}} \geqq \frac{1}{\alpha}. \tag{20}$$

Now set $\alpha = \delta^2/(1-\delta)^2$, thus achieving

$$P^*(2) = 1/(1+\gamma^{\frac{1}{2}}) \leqq \delta. \tag{21}$$

Finally choose k_1, k_2 such that $k_1 > k_2$ and (16) holds with $\varepsilon = 2\delta/m$. Then by definition of $P_d^*(m)$ and (16) and (18), we find

$$\begin{aligned} P_d^*(m) &= \min_{j,P_H,P_T;S_0,S_1} \left[\tfrac{1}{2}\textstyle\sum_{i \in S_1} \mu_i(j, p_0, P_H, P_T) + \tfrac{1}{2}\sum_{i \in S_0} \mu_i(j, p_1, P_H, P_T)\right] \\ &\geqq \min_{j,P_H,P_T;S_0,S_1} \left[\tfrac{1}{2}\textstyle\sum_{i \in S} (\mu_i^*(j, P_H, P_T) - (2\delta/m))\right] \\ &= \tfrac{1}{2} - \delta. \end{aligned} \tag{22}$$

Thus for any m there exist coin flipping probabilities p_0 and $p_1(p_0 \approx 1, p_1 \approx 1$, in our example) such that any m-state deterministic automaton is arbitrarily bad (see (22)), while there exists a randomized 2-state automaton that is arbitrarily good (see (21)).

REFERENCES

[1] CHANDRASEKARAN, B. (1970). Finite memory hypothesis testing—a critique. *IEEE Trans. Information Theory* **IT-16** 494–496.

[2] CHANDRASEKARAN, B. (1971). Reply to finite memory hypothesis testing: comments on a critique. *IEEE Trans. Information Theory* **IT-17** 104–105.

[3] COVER, THOMAS M. and HELLMAN, MARTIN E. (1970). Finite memory hypothesis testing: comments on a critique. *IEEE Trans. Information Theory* **IT-16** 496–497.

[4] HELLMAN, MARTIN E. and COVER, THOMAS M. (1970). Learning with finite memory. *Ann. Math. Statist.* **41** 765–782.