ON THE FUTILITY OF ARBITRARILY INCREASING MEMORY CAPABILITIES OF STOCHASTIC LEARNING AUTOMATA

B. John Oommen

SCS-TR-66

November 1984
Revised May 1985

School of Computer Science
Carleton University
Ottawa, Ontario
K1S 5B6

Canada

This research was supported by the Natural Sciences and Engineering Research Council of Canada

ON THE FUTILITY OF ARBITRARILY INCREASING MEMORY CAPABILITIES OF STOCHASTIC LEARNING AUTOMATA

by

B. John Oommen[+]

ABSTRACT

By designing a new family of expedient learning machines, we prove the counter-intuitive fact, that arbitrarily increasing the memory of a learning automaton does not necessarily increase its learning capability. This complements the results of Hellman and Cover [1] who proved that it is impossible to obtain an arbitrarily accurate learning machine which possesses only finite memory.

---

* Partially supported by the Natural Sciences and Engineering Research Council of Canada.

+ School of Computer Science, Carleton University, Ottawa, Canada, K1S 5B6.

I. INTRODUCTION

Stochastic automata have been studied in the literature and have been used to model biological learning systems. The literature concerning such automata is extensive, but for the sake of brevity we merely cite the works of Narendra and Thathachar [3,4] and the book by Tsetlin [7]. The use of these automata to parameters optimization, path finding, telephone routing and game playing is also discussed in the survey paper by Narendra and Thathachar [4].

The learning process of the automaton can be described as follows. Consider Figure 1. The environment with which the automaton interacts offers the latter a finite set of actions. The automaton is constrained to choose one of these actions. Once the action is chosen, the automaton is penalized by the environment, the penalty probability being dependent on the action chosen. A learning automaton is one which learns the action with the minimum penalty probability and which ultimately chooses this with a higher probability compared to the other actions.

In this paper we are concerned with automata commonly known as variable structure stochastic automata. In the literature, all the automata of this type have the property that if the memory associated with the automaton is increased the capability of the machine choosing the most optimal action also increases correspondingly. In [1], Hellman and Cover proved an extremely powerful result, which, informally speaking, stated that a machine with finite memory can never possess an arbitrarily high learning capability. In other words, if the learning machine had N states, the maximum accuracy with which the machine could learn the optimal action, would be given by a probability ξ_N

where, $\xi_N < 1$ for <u>all</u> $N < \infty$. The converse question, which has interested information theoretic computer scientists is: Does there exist a family of non-degenerate learning automata whose K-state automaton ($K < \infty$), possesses an accuracy of learning ξ_K identically equal to ξ^*, the accuracy of the corresponding infinite state machine. In otherwords, does there exist a family of non-degenerate automata for which it is possible to attain the same accuracy with <u>finite</u> memory as with infinite? This paper proves that indeed such a family exists. By designing this family of expedient

automata, we shall indeed prove that arbitrarily increasing the memory of the machine <u>does not necessarily increase</u> the learning capability of a learning machine.

I.1 <u>Fundamentals</u>

An automaton is a quintuple $A = \{S, A, B, F(.,.), G(.)\}$,

where (1) $S = \{s_0, s_1, \ldots, s_N\}$ is its set of states; N is termed as the depth of memory of the machine.

(2) $A = \{a_1, a_2, \ldots, a_R\}$ is its set of actions.

(3) $B = \{0, 1\}$ is the set of possible inputs to the automaton. The input at time instant 'n' is b(n). b(n) = 1 indicates that the automaton has been penalized.

(4) F(.,.) is a map from S X B to S. It determines the next state of the automaton at time 'n+1' if its state at time 'n' is known. It is called the transition function (or matrix) and can be either deterministic or stochastic.

(5) The output function (or matrix), G(.), determines the output or the action chosen by the automaton at any time and is a function of the state in which the automaton is.

The automaton learns the optimal action in A by interacting with an environment. The latter is a triple {A, B, C}, where:

(1) A is the set of actions $\{a_1, a_2, \ldots, a_R\}$. One of these actions is the input to the environment.

(2) B = {0, 1} is its set of outputs. The output at time instant 'n', b(n), is 1 if it penalizes the automaton.

(3) C = $\{c_1, c_2, \ldots, c_R\}$ is the set of penalty probabilities characterizing the environment with:

$$c_j(n) = \Pr[b(n) = 1 | a(n) = a_j]$$

We assume that $c_j(n)$ is independent of 'n'. In other wrods the environment is stationary.*

The automaton-environment interaction can be described by Figure 1. Suppose the automaton is in state s(n) at time 'n'. Based on G(.) the action selected by it is G(s(n)). This serves as the input to the environment which immediately responds to the action by either a 0 or 1. Depending on the feedback it receives, b(n), and F(.,.), the automaton goes into a new state s(n+1) at the next time instant to decide on a new action.

---

* For a non-stationary environment obviously (see [6,7]) indefinitely increasing the memory is not so beneficial. This is because informally speaking, it is useless to remember an arbitrary long history of the environment if the information content of the history is meaningless. The latter will be true if the parameters which characterize the environment keeps changing.

We shall use the notation that $p_i^{(N)}(n)$ is the probability that the automaton which possesses a memory depth of N actually chooses the action A_i (i=1,...R) at the nth time instant. $Q^{(N)}(n) = \sum_{i=1}^{R} p_i^{(N)}(n) \cdot c_i$ is the expected loss of the machine at this time.

I.2 Learning Criteria

With no apriori information, the automaton chooses the actions with equal probability. The expected penalty is thus initially, Q_0, where, for all K,

$$Q_0 = \sum_{i=1}^{R} p_i^{(K)}(0)\, c_i = \frac{1}{R} \sum_{i=1}^{R} c_i, \quad (\text{since } p_i^{(K)}(0) = 1/R). \tag{2}$$

An automaton is said to learn expediently if, as time tends towards infinity, the expected penalty is less than Q_0. We denote the expected penalty at time 'n' as E[Q(n)]. Other norms of learning are described in [4,6,7].

For the rest of the paper we will be dealing with the two action case, i.e., $A = \{a_1, a_2\}$. With no loss of generality we shall assume that $c_1 < c_2$. In other words, a_1 is assumed to be the better action.

II. THE DISCRETIZED LINEAR INACTION-PENALTY (DL_{IP}) AUTOMATON

The Discretized Linear Inaction-Penalty (DL_{IP}) Automaton is defined as a quintuple (S, A, B, F(.,.), G(.)), where

$S = \{s_0, s_1, \ldots, s_N\}$, N being an even integer.

$A = \{a_1, a_2\}$, with A_1 being the superior action.

$B = \{0,1\}$, the set of inputs to the automaton 1 being the penalty input.

F is a map from SxB to S and is defined by (3) below for all i.

$$
\begin{aligned}
s(n+1) &= s_{i+1} && \text{if } a(n) = a_2 \text{ and } b(n) = 1, \\
&= s_{i-1} && \text{if } a(n) = a_1 \text{ and } b(n) = 1, \\
&= s_i && \text{if } a(n) = a_1 \text{ or } a_2 \text{ and } b(n) = 0. \qquad (3)
\end{aligned}
$$

Observe that the automaton is of an inaction penalty flavour. G is a map specifying the probability of choosing action A_j if the automaton is in state s_i and is defined by (2) below.

$$
\begin{aligned}
G(i,j) &= \frac{i}{N} && \text{if } j = 1 \\
&= 1 - \frac{i}{N} && \text{if } j = 2. \qquad (4)
\end{aligned}
$$

We can alternatively view the F(.,.) and G(.) maps as follows. Associated with the state s_i is the probability $\frac{i}{N}$. This represents $p_1^{(N)}(n)$, the probability of the automaton choosing the action a_1. Observe that in this state the automaton chooses the action a_2 with the probability $(1 - \frac{i}{N})$. Since any one of the action probabilities completely defines the vector of action probabilities, we shall, with no loss of generality, consider $p_1^{(N)}(n)$.

Since we shall be consistently speaking of the DL_{IP} automaton with a memory depth of N, for the sake of simplifying the notation, unless explicitly stated, $p_i(n)$ and $Q(n)$ shall be used to refer to $p_i^{(N)}(n)$ and $Q^{(N)}(n)$ respectively.

By virtue of the above G(.) function, observe that the transition map is well defined even at the end states s_0 and s_N. This is due to the fact that if $s(n) = s_N$, $a(n)$ cannot be a_2; conversely, if $s(n) = s_0$, $a(n)$ <u>cannot</u> be a_1. The transition map is pictorially given in Figure II in terms of the action probabilities.

By virtue of the probabilities associated with the states, the updating can alternatively be described in terms of the action probabilities as follows:

$$\begin{aligned} p_1(n+1) &= p_1(n) + \frac{1}{N} && \text{if } a(n) = a_2,\ b(n) = 1, \\ &= p_1(n) - \frac{1}{N} && \text{if } a(n) = a_1,\ b(n) = 1, \\ &= p_1(n) && \text{if } a(n) = a_1 \text{ or } a_2,\ b(n) = 0. \end{aligned} \tag{5}$$

Since $p_2(n) = 1-p_1(n)$, (5) fully defines the changes made on both the action probabilities.

The probability changes are indeed made in discrete jumps. Further, the automaton updates the action probabilities only when the environment responds with a penalty and hence it is called an inaction-penalty automaton. It is called linear because the discretized probability values vary as a linear function of the indices of the states. This warrants the automaton being called a Discretized Linear Inaction-Penalty (DL_{IP}) automaton.

II.1 The Asymptotic Properties of the DL_{IP} Automaton

Let us consider the Markovian properties of the DL_{IP} automaton. $\{p_1(n)\}$ behaves as a homogeneous Markov chain defined by a stochastic

matrix M. $M_{i,j}$, the arbitrary element of M is defined as:

$$M_{i,j} = \Pr[s(n) = s_j | s(n-1) = s_i].$$

From (5), the elements of M can be written down as below:

$$\begin{aligned} M_{i,i-1} &= g_i c_1 && \text{for } 1 \leq i \leq N, \\ M_{i,i+1} &= \bar{g}_i c_2 && \text{for } 0 \leq i \leq N-1, \\ M_{i,i} &= 1 - g_i c_1 - \bar{g}_i c_2 && \text{for } 0 \leq i \leq N, \end{aligned} \tag{6}$$

where $g_i = \frac{i}{N}$ and $\bar{g}_i = 1 - \frac{i}{N}$. All the other elements of M are identically zero.

The Markov chain consists of exactly one closed communicating class. Further, since it is aperiodic the chain is ergodic and the limiting distribution is independent of the initial distribution [2]. Let $\underline{\pi}(n)$ be the state probability vector, where, for all n,

$$\underline{\pi}(n) = [\pi_0(n), \pi_1(n), \ldots, \pi_N(n)]^T$$

and

$$\pi_i(n) = \Pr[s(n) = s_i] \text{ with } \sum_{i=0}^{N} \pi_i(n) = 1.$$

Then the limiting value of $\underline{\pi}$ is given by $\underline{\pi}^*$ which satisfies,

$$M^T \underline{\pi}^* = \underline{\pi}^* \tag{7}$$

Using (7) we now derive the asymptotic properties of the DL_{IP} automaton.

Theorem I

For the DL_{IP} Automaton whose memory depth is N, the limiting value of $\pi_k(n)$ is given by π_k^*, where,

$$\pi_k^* = \frac{\binom{N}{k} c_i^{N-k} c_2^{k}}{(c_1+c_2)^N} .$$

Proof: We intend to solve $M^T\underline{\pi}^* = \underline{\pi}^*$. We shall first obtain a solution $\underline{\pi}'$ where $M^T\underline{\pi}' = \underline{\pi}'$. Subsequently, we shall normalize $\underline{\pi}'$ to render it a probability vector. This normalized vector will indeed be $\underline{\pi}^*$. With no loss of generality let $\pi_0' = c_1^N$. We inductively prove that $\pi_k' = \binom{N}{k} c_1^{N-k} c_2^{k}$.

Basis Step: Expanding the equation $M^T\underline{\pi}' = \underline{\pi}'$ for k=0 yields,

$$(1-c_2)\pi_0' + g_i\, c_1\, \pi_1' = \pi_0' .$$

Since $g_1 = \frac{1}{N}$, and $\pi_0' = c_1^N$, π_1' can be solved for. This yields,

$$\pi_1' = N \cdot \frac{c_2}{c_1} \cdot c_1^N = \binom{N}{1} c_1^{N-1} c_2.$$

Inductive Step: Assume that

$$\pi_{k-1}' = \binom{N}{k-1} c_1^{N-k+1} c_2^{k-1}$$

and

$$\pi_k' = \binom{N}{k} c_1^{N-k} c_2^{k} .$$

Expanding the equation in (7) which corresponds to π_k', we get,

$$\bar{g}_{k-1}\, c_2\, \pi'_{k-1} + (1 - g_k\, c_1 - \bar{g}_k\, c_2)\, \pi'_k + g_{k+1}\, c_1\, \pi'_{k+1} = \pi'_k \ .$$

Solving for π'_{k+1} yields,

$$\pi'_{k+1} = \frac{(g_k\, c_1 + \bar{g}_k\, c_2)\, \pi'_k - \bar{g}_{k-1}\, c_2\, \pi'_{k-1}}{g_{k+1}\, c_1} \ . \qquad (8)$$

Substituting the inductive hypothesis, and noting that for all i, $g_i = \frac{i}{N}$ and $\bar{g}_i = \frac{N-i}{N}$, (8) can be rewritten as,

$$\pi'_{k+1} = \frac{[\frac{k}{N} c_1 - \frac{N-k}{N} c_2] \binom{N}{k} c_1^{N-k} c_2^{k} - [\frac{N-k+1}{N} c_2] \binom{N}{k-1} c_1^{N-k+1} c_2^{k-1}}{\frac{k+1}{N} c_2}$$

which after considerable algebra simplies to

$$\pi'_{k+1} = \binom{N}{k+1} c_1^{N-k-1} c_2^{k+1} \ .$$

Thus for all i,

$$\pi'_i = \binom{N}{i} c_1^{N-i} c_2^{i} \ .$$

Normalizing to get π^*_i yields,

$$\pi^*_i = \frac{\pi'_i}{\sum_{i=0}^{N} \pi'_i} = \frac{\binom{N}{i} c_1^{N-i} c_2^{i}}{\sum_{j=0}^{N} \binom{N}{j} c_1^{N-j} c_2^{j}}$$

$$= \frac{\binom{N}{i} c_1^{N-i} c_2^{i}}{(c_1 + c_2)^N} \ . \qquad ***$$

We now derive the limiting values of $E[p_1(n)]$ and $Var\,[p_1(n)]$.

Theorem II

The limiting distribution of $p_1(n)$ has the following mean and variance:

$$E[p_1(\infty)] = \frac{c_2}{c_1+c_2}$$

$$\text{Var}\,[p_1(\infty)] = \frac{c_1\, c_2}{N(c_1+c_2)^2} .$$

Proof: From theorem I,

$$\pi_k^* = \binom{N}{k} \frac{c_1^{N-k}\, c_2^{k}}{(c_1+c_2)^N}$$

By regrouping the terms, we can equivalently write,

$$\pi_k^* = \binom{N}{k} \left(\frac{c_1}{c_1+c_2}\right)^{N-k} \left(\frac{c_2}{c_1+c_2}\right)^{k}$$

$$= \binom{N}{k} q^k (1-q)^{N-k}, \quad \text{where } q = \frac{c_2}{c_1+c_2} .$$

Let X^* be the limiting index of the state of the automaton. From the above, clearly X^* is Binomially distributed, with parameters N and q. Hence,

$$E[X^*] = Nq$$

and

$$\text{Var}\,[X^*] = N\, q(1-q)$$

Thus, $E[p_1(\infty)] = \frac{1}{N} E[X^*] = q = \frac{c_2}{c_1+c_2}$ and $\text{Var}[p_1(\infty)] = \frac{1}{N^2} \text{Var}[X^*] = \frac{1}{N} q(1-q) = \frac{c_1\, c_2}{N(c_1+c_2)^2}$ and the theorem is proved.

Corollary I

Independent of the number of states is possesses the DL_{IP} automaton is expedient in all random environments.

Proof: We know that $E[Q^{(N)}(n)] = c_1 E[p_1(n)] + c_2 E[p_2(n)]$. Using the results of Theorem II,

$$\lim_{n\to\infty} E[Q^{(N)}(n)] = \frac{2 c_1 c_2}{(c_1+c_2)} .$$

The result follows since the RHS of the above equation is strictly less than $\frac{c_1+c_2}{2}$.

Theorem III

The family of Discretized Linear Inaction Penalty Automaton is a family of non-degenerate expedient fixed structure stochastic automata for which it is futile to arbitrarily increase the memory capability.

Proof: The theorem follows from the above corollary, since,

$$\lim_{n\to\infty} E[Q^{(N)}(n)] = \frac{2 c_1 c_2}{c_1 + c_2}$$

in dependent of N, the memory of the particular machine which is the member of the family of DL_{IP} automata.

The nondegeneracy of the members of this family is obvious.

We can summarize the results of the above theorems as follows. The family of DL_{IP} Automata is a family of non-degenerate machines which has an expedient learning behaviour. However, the best automaton in the family is the automaton which has the minimal memory associated with it. The Performance of this machine is equalled (but never excelled) by every memory of the same family, including the case when the memory is increased

indefinitely. We have thus proved the counter-intuitive fact that increasing the memory capability of a machine does not necessarily increase its learning capability.

III. CONCLUSIONS

Hellman and Cover [1] presented a powerful result concerning finite memory learning. They proved that one cannot design a machine that learns with an arbitrarily high accuracy if the memory capability of the machine is finite. In this paper we have proved a result that is rather counter-intuitive, and which complements their result. We have shown that there exists a family of non-degenerate learning automata for which the best performance is obtained with finite memory. Thus, there are expedient learning machines for which it is futile to increase the memory capability.

REFERENCES

1. Hellman, M.E., and Cover, T.M., "Learning With Finite Memory", Annals of Mathematical Statistics, 1970, Vol.41, pp.765-782.

2. Isaacson, D.L., and Madson, R.W., "Markov Chains: Theory and Applications", Wiley, 1976.

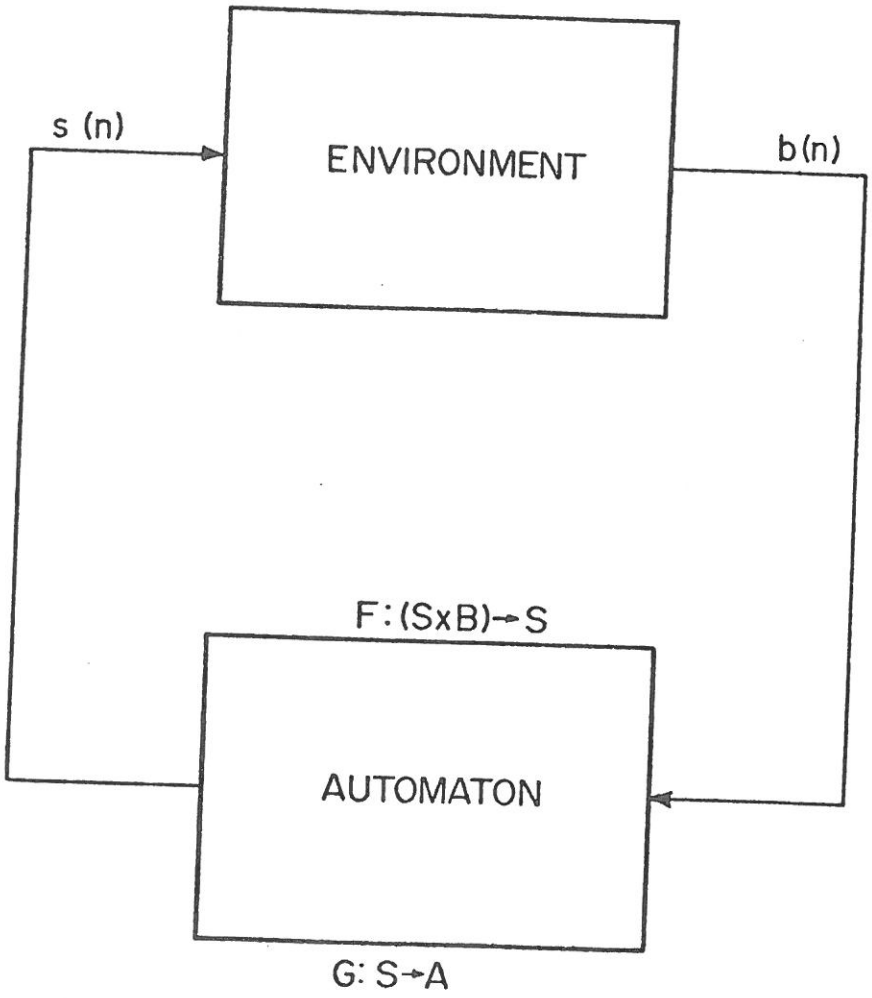
3. Narendra, K.S., and Thathachar, M.A.L., Forthcoming book on Learning Automata.

4. Narendra, K.S., and Thathachar, M.A.L., "Learning Automata -- A Survey", IEEE Trans. Syst. Man and Cybern., Vol.SMC-4, 1974, pp.323-334.

5. Paz, A., "Introduction to Probabilistic Automata", New York, Academic, 1971.

6. Tsetlin, M.L., "On the Behaviour of Finite Automata In Random Media", Automat. Telemekh., Vol.22, 1961, pp.1345-1354.

7. Tsetlin, M.L., "Automaton Theory and the Modelling of Biological Systems", New York and London, Academic, 1973.



$b(n) \in \{0, 1\} = B$

$s(n) \in \{s_1, s_2, \ldots s_N\} = S$

$a(n) \in \{a_1, a_2, \ldots a_R\} = A$

FIG. 1 : THE AUTOMATON-ENVIRONMENT INTERACTION

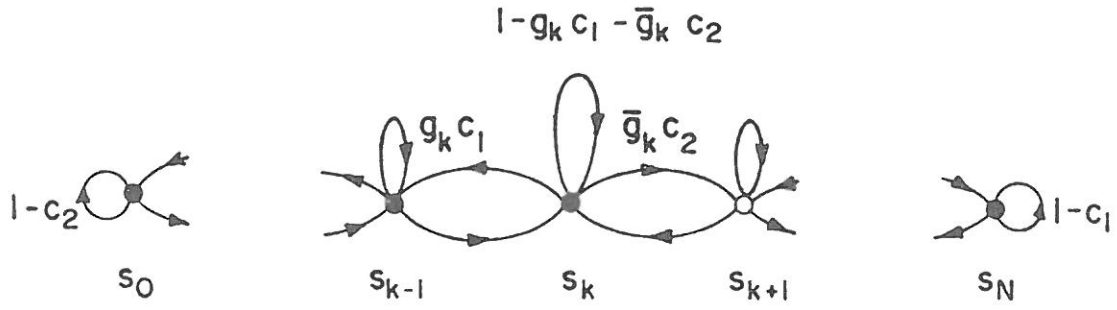


FIG.II : THE DL_{IP} AUTOMATON