Def Sci J, Vol 35, No. 3, July 1985, pp 361–366

Learning Systems : Stochastic Automata Models

M. A. L. THATHACHAR
Indian Institute of Science, Bangalore-560012

Abstract. Stochastic automata operating in an unknown random environment have been successfully used in modelling learning systems. For such learning automata a new algorithm which estimates the environmental characteristics is presented and is used in the solution of a pattern classification problem with unknown class-conditional densities. The algorithm selects the optimal threshold asymptotically with probability one.

1. Introduction

A learning system improves its behaviour with time, moving in some sense towards a goal. Such a system is needed, for instance, when basic uncertainties exist in the process being controlled. The concept of learning has been traditionally associated with intelligence, and the study of learning systems could be regarded as a part of the field of artificial intelligence.

A learning system has to experiment with the environment in which it is working, collect data and draw inferences regarding its performance and plan the future strategies. A mathematical model which fits well with this concept is that of the learning automaton[1].

A learning automaton is a stochastic automaton which has a finite number of output actions, one of which is selected at each instant. The selected action elicits a reward or penalty reaction from the environment and this in turn is used to update the probability distribution defined over the actions. Choise of proper algorithms for the updating leads to desirable asymptotic behaviour of the learning automaton.

In this paper the two-class pattern classification problem[2] with unknown class-conditional densities is posed as a learning problem. A new algorithm which estimates the characteristics of the environment is modified to suit the classification problem. The algorithm asymptotically selects the optimal threshold w.p.1.

2. Environment and the Learning Automaton

As mentioned earlier, a learning automaton is connected in a feedback fashion with an unknown random environment. Such an environment can be represented by a triple $\langle \alpha, \beta, c \rangle$ where $\alpha = \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ is the input action set, $\beta = \{0, 1\}$ is the output reaction set and $c = \{c_1, c_2, \ldots, c_r\}$ is the penalty probability set given by

$$c_i = P_r[\beta(k) = 1 \mid \alpha(k) = \alpha_i]\ (i = 1, \ldots r)\ (k = 0, 1, 2, \ldots) \tag{1}$$

Usually the reaction of the environment corresponding to $\beta(k) = 1$ is referred to as 'penalty' and $\beta(k) = 0$ as 'reward'. The $\{c_i\}$ are generally unknown.

The learning automaton is defined by the quadruple $\langle \underline{\alpha}, \underline{\beta}, p, T \rangle$ where α, β are the same as in case of the environment except that $\underline{\beta}$ is the input set and α is the output set for the automaton. p is the vector of action probabilities given by

$$p(k) = [p_1(k), p_2(k) \ldots p_r(k)]' \tag{2}$$

where $'$ is transpose and

$$p_i(k) = p_r[\alpha(k) = \alpha_i] \tag{3}$$

T is the transformation associated with the updating of $p(k)$

$$p(k+1) = T[p(k), \alpha(k), \beta(k)] \tag{4}$$

The vector $p(k)$ can be regarded as the state of the automaton at the instant k (Fig. 1).

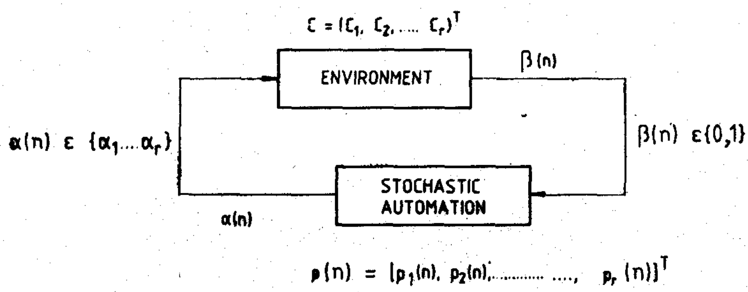

Figure 1. A learning automaton.

$$\text{Suppose } c_l = \min_i \{c_i\} \tag{5}$$

Then α_l is called the optimal action. An automaton is said to be optimal if $p_l(k) \to 1$ w.p.1 and ϵ-optimal if $p_l(k) \to 1$ with probability $(1 - \epsilon)$ where $\epsilon > 0$ can be made arbitrarily small by a suitable choice of the updating algorithm.

A simple and popular updating algorithm is the Linear Reward-Inaction (L_{R-I}) algorithm described as follows.

$$\left.\begin{aligned} &\text{If } \alpha(k) = \alpha_i \\ p_i(k+1) &= p_i(k) + \lambda(1 - p_i(k)) && \text{if } \beta(k) = 0 \\ &= p_i(k) && \text{if } \beta(k) = 1 \\ p_j(k+1) &= p_j(k) - \lambda p_j(k),\ (j \neq i) && \text{if } \beta(k) = 0 \\ &= p_j(k) && \text{if } \beta(k) = 1 \end{aligned}\right\} \tag{6}$$

$0 < \lambda < 1$ is a parameter of the algorithm. It has been shown that the L_{R-I} algorithm leads to ϵ-optimality.

3. The Estimator Algorithm

While the L_{R-I} and related algorithms have well-established convergence properties, they have generally been found to be slow in applications. In this section, a more sophisticated algorithm which makes use of estimates of penalty probabilities is given. The algorithm is called the 'Estimator Algorithm' and has been found to possess a high rate of convergence[3].

$$\textit{If}\ \alpha(k) = \alpha_i,\ \beta(k) = 0$$

$$p_i(k+1) = p_i(k) + \sum_{j \neq i} \lambda(\hat{c}_j(k) - \hat{c}_i(k)) \left(s_{ij} p_j(k) + s_{ji} \frac{p_i(k)}{r-1} (1 - p_j(k)) \right)$$

$$p_j(k+1) = p_j(k) - \lambda(\hat{c}_j(k) - \hat{c}_i(k)) \left(s_{ij} p_j(k) + s_{ji} \frac{p_i(k)(1 - p_j(k))}{r-1} \right) \quad (j \neq i) \qquad (7)$$

If $\beta(k) = 1$, the action probabilities are unchanged. It may be noted that $\hat{c}_i(k)$, $i = 1, 2, \ldots, r$, are the estimates of the penalty probabilities at the instant 'k' and are recursively computed as follows.

$$\hat{c}_i(k+1) = [(r_i - 1)\,\hat{c}_i(k) + \beta(k)]\,(1/r_i)$$

$$\hat{c}_j(k+1) = \hat{c}_j(k) \text{ for } (j \neq i) \qquad (8)$$

r_i is the number of times action α_i is chosen in k instants and

$$s_{ij} = 0 \text{ if } \hat{c}_i(k) > \hat{c}_j(k)$$

$$= 1 \text{ otherwise} \qquad (9)$$

It may be noticed that Eqn. (8) represents a binomial estimator of the penalty probabilities c_i. An important property of the estimator algorithm is that $p_i(k)$ converges to 1 in probability. Thus an automaton using the algorithm may be said to be 'optimal in probability'.

4. Pattern Classification

In this section an attempt is made to pose the pattern classification problem as a learning problem. Consider samples belonging to 2 categories ω_1 and ω_2. It is desired to classify a given sample as belonging to ω_1 or ω_2 in such a way that some average cost is minimized.

Suppose the classification is done on the basis of measurement of a feature x of the samples. Normally the class conditional densities $p(x|\omega_i)$ and *a priori* probabilities

$P(\omega_i)$ for $i = 1, 2$ are given and from these a threshold 't' is computed. A given sample is classified as belonging to ω_1 or ω_2 according as $x > t$ or $x < t$.

More realistically, the class conditional densities are unknown and only some samples of known classification are available. In such an event, the problem of classification may be posed as one of learning the optimal threshold 't' which minimizes a given cost criterion.

The problem can now be posed in terms of the automata model as follows. Let $t_1, t_2, \ldots, t_r$ be the possible values of the optimal threshold. These are made the actions of a learning automaton. When a labelled sample is presented, classification is done on the basis of $x \gtrless t_i$, where t_i is the threshold selected by the automaton. If the classification coincides with the true classification, the result is taken as a reward reaction from the environment. Otherwise it is a penalty. It can thus be seen that using the estimator algorithm, the probability of selecting the optimal threshold tends to unity in probability. The threshold thus selected minimizes the probability of error in classification. It may also be noted that the penalty probability c_i is the probability of error in classification when threshold t_i is selected.

5. Modified Algorithm

The Estimator Algorithm can be modified to give improved performance in the pattern classification problem. The main facility with the problem is that given a labelled sample, the consequences of using each threshold can be observed. This corresponds to choosing all the actions at each instant in the learning automaton model. The advantage is that estimates of all the penalty probabilities can be updated at each instant. It can be shown that the speed of convergence is improved and also that convergence of the optimal threshold takes place with prabability one.

Stating the modified algorithm formally, if $\alpha(k) = \alpha_i = t_i$, Eqn. (8) gets replaced by

$$\hat{c}_j(k+1) = [(k-1)\,\hat{c}_j(k) + \beta_j(k)]\,(1/k) \quad j = 1, \ldots, r \tag{10}$$

and Eqn. (7) remains the same.

It can generally be observed that the algorithm works very fast when the lowest and the next higher penalty probabilities differ reasonably.

6. Simulation Results

To illustrate the performance of the learning algorithm, a two-class pattern classification problem was chosen as follows :

$$p(x|\omega_1) = N(2.0, 0.5)$$

$$p(x|\omega_2) = N(4.0, 0.5)$$

The class conditional densities are normal with only different means.

Ten values of thresholds from 2.2 to 4.0 with intervals of 0.2 were taken as the actions. Initial estimates $c_i(k)$ were obtained with 10 samples and then the modified estimator algorithm was applied. The action probability of the optimal threshold

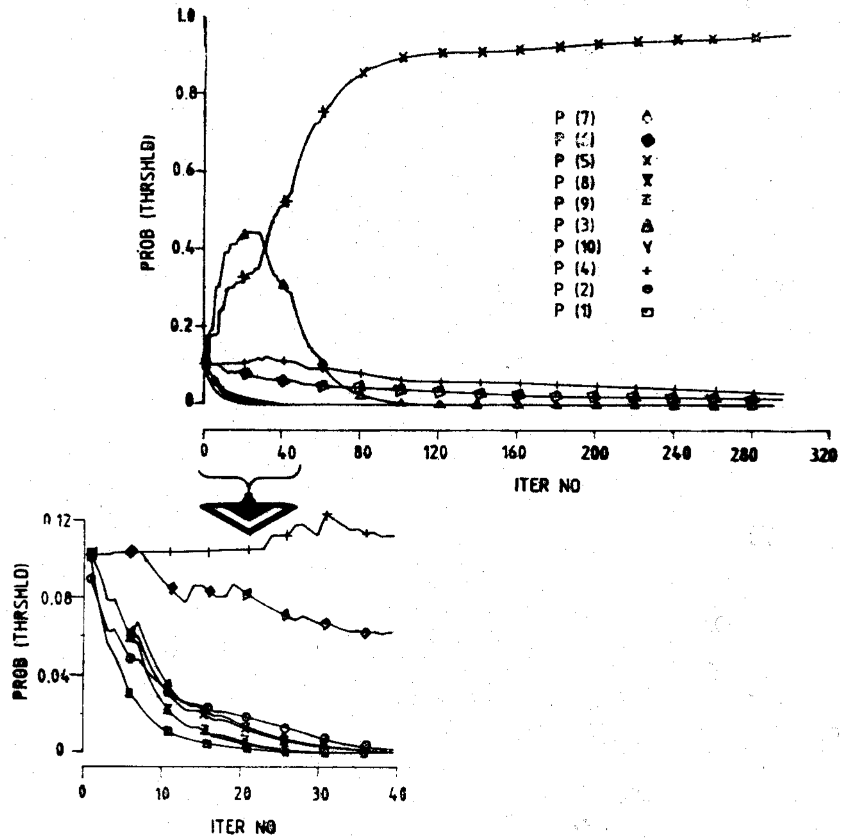

Figure 2. Probability of thresholds Vs. No. of iterations.

of 3.0 converged to about 0.93 in 200 iterations. The movement of this probability is shown in Fig. 2.

7. Conclusion

A model of learning systems using stochastic automata operating in an unknown random environment has been described. These automata have a finite number of output actions and update their action probabilities on the basis of the reactions received from the environment. Use of proper updating schemes results in convergence to the optimal action. The pattern classification problem has been posed as a learning problem and a modified estimator algorithm has been proposed for this problem. The results can be extended to multiple-feature classification problems.

References

1. Narendra, K. S. & Thathachar, M. A. L., *IEEE Transactions on Systems, Man and Cybernetics*, SMC-4, (1974), 323–334.
2. Duda, R. O. & Hart, P. E., 'Pattern Recognition and Scene Analysis', (John Wiley), (1973).
3. Thathachar, M. A. L. & Sastry, P. S., A class of Rapidly Converging Algorithms for Learning Automata, To be presented at the IEEE—SMC International Conference, Bombay, New Delhi, January (1984).
4. Ramesh, R., 'Learning Automata in Pattern Classification', M. E. Project Report, School of Automation, Indian Institute of Science, July (1983).