AN OPTIMIZATION APPROACH TO THE ANALYSIS OF GENERALIZED LEARNING AUTOMATA ALGORITHMS

V.V. Phansalkar, P.S. Sastry and M.A.L. Thathachar,
Department of Electrical Engineering,
Indian Institute of Science,
Bangalore 560 012

ABSTRACT

Weak convergence methods are used to analyse generalized learning automata algorithms. The REINFORCE algorithm has been analysed. It is shown by an example that this algorithm can exhibit unbounded behaviour. A modification based on constrained optimization principles is proposed to overcome this problem. The relationship between the asymptotic behaviour of the modified algorithm and the Kuhn-Tucker points of the related constrained optimisation problem is brought out.

1. INTRODUCTION

Learning automata have originally been proposed to model the behaviour of biological systems [4]. They have also found applications in diverse areas such as pattern recognition and queueing systems [3].

A learning automaton interacts with an environment to choose the optimum action out of a finite set of actions. The response from the environment is used to update the selection probabilities. In this basic model, there is no scope for more than one optimal action. This is required when the environment can be in different states, each state having its own optimal action. This can occur in pattern recognition, where the environment state is characterized by the pattern vector to be classified. The state of the environment can be thought of as being represented by a vector known as the context vector.

The learning automaton also cannot be used as a unit in a network structure. This is again due to its inability to have any input other than the response from the environment.

There are two ways of overcoming this problem. One method is to use the learning automaton in a game [3]. Another method would be to generalize the concept of a learning automaton to include context vectors. This is known as the Generalized Learning Automaton (GLA) [3,5]. A GLA system consists of an environment and a learning system.

The GLA system functions as follows. At each instant, the environment presents a context vector to the learning system. Based on this vector and its internal state w, the learning system chooses an action. The environment the emits a Scalar Reinforcement Signal (SRS) based on the context vector-action pair. (The SRS is assumed to be stochastic). Using this information, the learning system updates its internal state w. w is assumed to be a real vector. A higher value of the SRS indicates better performance. The ideal goal of a GLA system is to learn the function which maps each context vector into its optimal action. This may not always be possible, since the structure chosen may not allow the function to be learnt. If the SRS is denoted by r, a natural criterion to maximize would be E[r/w]. If there exists a w* which implements the optimal action mapping, that w* also maximizes E[r/w].

GLA units can be connected together to model 'connectionist systems' or 'artificial neural networks'. To model a connectionist system as a GLA system, a basic unit for the GLA system is required. A method to connect these units has also to be specified.

Several algorithms have been proposed for GLA tasks. One is the A_{R-P} algorithm [1]. Under strong conditions it is shown that this algorithm learns the optimal action mapping. This result holds for a single unit and has not yet been generalized to a network of units. Another algorithm is REINFORCE algorithm [5]. This algorithm is a single step stochastic gradient algorithm, following the gradient of E[r/w]. This result holds for a network of units.

In this paper, asymptotic analysis of the REINFORCE algorithm is attempted using weak convergence techniques [2]. Unfortunately, it can be shown that the algorithm can exhibit unbounded behaviour. A modification based on constrained optimization techniques is suggested to overcome this problem.

Section 2 develops the formal notation required for analysis. The REINFORCE algorithm is analysed and an example is presented to show the unbounded behaviour of the algorithm in Section 3. A modification is suggested

and analysed in Section 4. Section 5 contains the simulation results and the conclusions from Section 6.

2. NOTATION

In this section, the formal notation for a GLA system is developed. The environment of a GLA system is defined by the tuple (C, A, R, D, P_c) where C is the set of context vectors, A is the set of actions and R the set of values the SRS can take. P_c is a probability distribution over C and the arrival of context vectors is governed by P_c. D is the set of reinforcement expectations,

$$D = \{d(a,c): a \in A, c \in C\}$$

where

$$d(a,c) = E[r/action = a, \ context = c]$$

A single GLA unit is defined by the tuple (X, Y, R,w, g, T) where X is the the set of outputs, R the set of SRS values and w the internal state of the unit. g is a function which generates the action probabilities given the context input and internal state. Thus, for any y in Y,

$$g(x,y,w) = Prob\{output=y \mid context=x, state=w\}$$

In a network of GLA units, the ith unit is defined by the tuple (X_i, Y_i, R, w_i, g_i, T_i) which corresponds to the tuple described for a single unit. The context vector to any unit can be composed of outputs of other units and components of the context vector from the environment.

Only feedforward networks are considered here. Thus, there does not exist a set of indices $\{j_1,\ldots,j_k\}$ such that the context vector of j_i has the output of j_{i-1} as a component (for $2 \leq i \leq k$) and the context vector of j_1 has the output of j_k as a component.

3. ANALYSIS OF THE REINFORCE ALGORITHM

In this section, the REINFORCE algorithm is described and analysed. An example showing that unboundedness problems can exist is also presented.

At instant k, the updating as given by the REINFORCE algorithm is [5],

$$w_{ij}(k+1) = w_{ij}(k) + b\, r(k)[\partial \ln g_i / \partial w_{ij}],$$

where the partial derivative is evaluated at $(x_i(k), y_i(k), w_i(k))$, $b > 0$ is the learning parameter and w_{ij} is the jth component of w_i. w is the vector composed of the internal states of all units. It has been shown that REINFORCE algorithm has the following property [5]

$$E[w(k+1)-w(k) \mid w(k) = w] = b \nabla_w E[r/w]$$

where ∇_w denotes gradient with respect to w.

Define piecewise constant continuous time interpolations $w^b(.)$ of w(.) for a specific $b > 0$ as follows

$$w^b(t) = w(k) \text{ for } t \in [bk, b(k+1))$$

Then, using weak convergence results [2], the following theorem can be proved.

Theorem I: The sequence $w^b(.)$ converges weakly, as $b \to 0$, to z(.), where z(.) satisfies the ODE

$$\dot{z} = \nabla_z E[r \mid z], \ z(0) = w(0)$$

under the conditions that, for all i,

a) g_i is continuous with continuous first partial derivatives.
b) g_i is bounded away from zero on every compact set.

Theorem I can be used to study the behaviour of the algorithm by studying the associated ODE. Define

$$f(w) = E[r \mid w].$$

It is seen that the ODE performs a gradient ascent of the function f(.). It is known that the only stable points of a gradient ascent ODE are local maxima. But if there are no local maxima of f(.), the ODE can exhibit unbounded behaviour. This can be illustrated by the following example.

Consider an environment which consists of two context vectors c_1 and c_2. Let $c_1 = (0,1)^t$ and $c_2 = (1,0)^t$. The set of actions is $A = \{a_1,a_2\}$. At each instant, vectors are presented to the learning system with equal probability. Also

$$d(a_1,c_1) = d(a_2,c_2) = 0.9$$

and

$$d(a_1,c_2) = d(a_2,c_1) = 0.1$$

It is assumed that the SRS r can take the values 0 and 1 only.

The learning system consists of a single unit. Its internal state is $w = (w_1, w_2)^t$. Let

$$h(v) = 1/(1+\exp(-v))$$

and then define the probability generating function g as

$$g(c, a_1, w) = h(w^t c)$$
$$g(c, a_2, w) = 1 - h(w^t c)$$

Then,

$$f(w) = E[r \mid w]$$
$$= 1/2\{E[r \mid w,c_1]+E[r \mid w,c_2]\}$$
$$= 1/2\{E(r \mid w,c_1,a_1)h(w^t c_1) + E(r \mid w,c_1,a_2)(1-h(w^t c_1)) + E(r \mid w,c_2,a_1)h(w^t c_2)$$

$$+E(r \mid w,c_2,a_2)(1-h(w^t c_2))\}$$

$$= 1/2\{.9h(w_2) + .1(1-h(w_2)) +$$

$$.1h(w_1) + .9(1-h(w_1))\}$$

$$= .4h(w_2) - .4h(w_1) + .5$$

Thus,

$$(\delta f/\delta w_1) = -0.4(dh(w_1)/dw_1)$$

$$= -0.4\exp(-w_1)/(1+\exp(-w_1))^2$$

$$(\delta f/\delta w_2) = .4\ (\exp(-w_2)/(1+\exp(-w_2))^2$$

According to theorem I, the ODE is

$$\dot{w}_1 = -0.4\exp(-w_1)/(1+\exp(-w_1))^2$$

$$\dot{w}_2 = 0.4\exp(-w_2)/(1+\exp(-w_2))^2$$

This is a pair of decoupled ODEs and it is easily seen that w_1 decreases without bound and w_2 increases without bound. Thus $w_1(.)$ diverges to - and $w_2(.)$ diverges to + . The algorithm can be seen to exhibit unbounded behaviour, which is undesirable.

One way to overcome this difficulty is to show that in the particular problem to which the algorithm is applied, the weights are bounded. As this would require a proof for every new problem, another method would be to repose the problem as a constrained optimization problem with the constraint region being a compact set. This is done in the next section.

4. MODIFICATION

In this section, a modification of the REINFORCE algorithm is proposed and analysed. Instead of maximizing E[r | w] without constraints, the problem is reposed as

maximize $E(r \mid w)$

subject to $s_{ij}(w) = 1/2(M_{ij}^2 2 - w_{ij}^2) \geq 0$

Each component is bounded separately. The algorithm is modified to

$$w_{ij}(k+1) = w_{ij}(k) + b\{r(k)\ [\delta \ln g_i/\delta w_{ij}]$$

$$+ K_{ij}\{h_{ij}(w_{ij}(k)) - w_{ij}(k)\}$$

where the partial derivative is evaluated at $(x_i(k), y_i(k), h_i(w_i(k)))$,

$$h_{ij}(x) = \begin{cases} M_{ij} & \text{for } x \geq M_{ij} \\ x & \text{for } -M_{ij} \leq x \leq M_{ij} \\ -M_{ij} & \text{for } x \leq -M_{ij} \end{cases}$$

and $K_{ij} > 0$ is a constant.

When the state is w_i, the value used for generating the action probabilities is $h_i(w_i)$ where

$$h_i(w_i) = (h_{i1}(w_{i1}),\ h_{i2}(w_{i2}), \ldots)$$

and

$$h(w) = (h_1(w_1),\ h_2(w_2),\ \ldots)$$

Again, considering continuous time interpolations $w^b(.)$ as before, the following theorem can be proved.

Theorem II

The sequence $\{w^b(.)\}$ converges weakly, as $b \to 0$, to $z(.)$ where $z(.)$ satisfies the ODE

$$\dot{z}_{ij} = (\delta f/\delta z_{ij})(h(z)) + K_{ij}(h_{ij}(z_{ij}) - z_{ij}),$$

$$z(0) = w(0)$$

under the conditions that g_i is bounded away from zero on $\prod_i [-M_{ij}, M_{ij}]$ and has continuous first partial derivatives.

The asymptotic behaviour of the algorithm is approximated by the ODE. The behaviour of the ODE can be related to the solutions of the constrained optimization problem by the following results.

Result 1: If w satisfies the first order necessary Kuhn-Tucker conditions for a local maximum [6] then there is an unique u such that $h(u) = w$ and u is a zero of the ODE.

Proof:

Let w satisfy the first order necessary Kuhn-Tucker conditions. Then [6],

$$(\delta f/\delta w_{ij})(w) + l_{ij}\ (\delta s_{ij}/\delta w_{ij})(w) = 0,$$

$$l_{ij} \geq 0,\quad s_{ij} \geq 0,\quad l_{ij}\, s_{ij} = 0.$$

Since, $s_{ij}(w) = (1/2)\ (M_{ij}^2 - w_{ij}^2)$,

$$(\delta f/\delta w_{ij})(w) - l_{ij}\, w_{ij} = 0.$$

Define $u_{ij} = w_{ij}\ (1 + (l_{ij}/K_{ij}))$
Then,

$$w = h(u)$$

and

$$(\delta f/\delta w_{ij})(h(u)) + K_{ij}(h_{ij}(u_{ij}) - u_{ij})$$

$$= (\delta f/\delta w_{ij})(w) + K_{ij}(w_{ij} - w_{ij} - (l_{ij}/K_{ij})w_{ij})$$

$$= (\delta f/\delta w_{ij})(w) - l_{ij}\, w_{ij} = 0$$

Thus, u is a zero of the ODE. To prove uniqueness, let v be a zero of the ODE and $h(v) = w$. Then,

$$v_{ij} = (1 + m_{ij})\ w_{ij},\quad m_{ij} \geq 0$$

Since v is a zero of the ODE,

$$(\delta f/\delta w_{ij})(h(v)) + K_{ij}(h_{ij}(v_{ij}) - v_{ij})$$

$$= (\delta f/\delta w_{ij})(w) + K_{ij}(w_{ij} - w_{ij} - m_{ij}\, w_{ij})$$

$$= (\delta f/\delta w_{ij})(w) - K_{ij}\, m_{ij}\, w_{ij} = 0$$

This is true iff $m_{ij} = l_{ij}/K_{ij}$, that is, $v = u$ and thus the zero is unique.

Result 2

If u is a zero of the ODE, h(u) satisfies the first order necessary Kuhn-Tucker conditions.

Proof

As u is a zero of the ODE,

$$(\delta f/\delta w_{ij})(h(u)) + K_{ij}(h_{ij}(u_{ij}) - u_{ij}) = 0$$

Let $w_{ij} = h_{ij}(u_{ij})$. Then, $w = h(u)$ and

$$u_{ij} = a_{ij} w_{ij}, \quad a_{ij} \geq 1$$

Define

$$l_{ij} = (a_{ij} - 1) K_{ij}$$

Then

$$l_{ij} \geq 0 \quad \text{and}$$

$$\begin{aligned}
&(\delta f/\delta w_{ij})(h(u)) - l_{ij} h_{ij}(u_{ij}) \\
&= (\delta f/\delta w_{ij})(h(u)) - K_{ij}(a_{ij}-1)w_{ij} \\
&= (\delta f/\delta w_{ij})(h(u)) + K_{ij}(w_{ij} - a_{ij}w_{ij}) \\
&= (\delta f/\delta w_{ij})(h(u)) + K_{ij}(h_{ij}(u_{ij}) - u_{ij}) \\
&= 0
\end{aligned}$$

The other Kuhn-Tucker conditions are also satisfied at h(u). Thus, h(u) satisfies the first order necessary Kuhn-Tucker conditions.

The above two results show the equivalence between the zeros of the ODE and the first order necessary Kuhn-Tucker conditions of the optimization problem. In an unconstrained optimization problem, the zeros are the points where the gradient of the function being maximized is zero. Here, the gradient is replaced by the first order necessary Kuhn-Tucker conditions.

The first order conditions are necessary conditions, not sufficient. A zero might satisfy the first order conditions, but still not be a maximum.

For further analysis, the second order conditions are used. Using linearisation of the ODE around a zero, the following results can be proved.

Result 3

If w is a strict local maximum of the optimization problem, then its related zero u is locally stable under the condition that all active constraints at w are strictly active.

$s_{ij}(w) \geq 0$ is a constraint of the optimization problem. It is said to be active at w if $s_{ij}(w) = 0$. It is said to be strictly active at w if $l_{ij} > 0$. As $l_{ij}\, s_{ij}(w) = 0$ is part of the first order necessary conditions, this implies that $s_{ij}(w) = 0$.

It can be easily shown that the solutions of the ODE are bounded and that the constrained optimization problem has atleast one solution.

5. SIMULATIONS

The problem described in Section 3 was simulated for the REINFORCE algorithm and the modified algorithm. For the original algorithm the values kept on increasing in magnitude, albeit slowly. This is clear even from the analysis of the ODE, as it can be shown that after a finite time T, $w_2(t) \leq 2\ln t$ even though $w_2(t) \to +\infty$. For the modified algorithm, the bounds were chosen to be 5 (i.e., $M_{ij} = 5$ for all i,j) and K_{ij}s were identically set to 1. For both the algorithms, b was chosen to be 0.5.

In 10 runs, the modified algorithm always converged to values around -5 and +5 for w_1 and w_2 respectively. This is also what is expected, since the point (-5, +5) is the maximum of E(r/w) in the feasible region.

The values kept increasing in the REINFORCE algorithm. In 10^6 steps, the values had reached 12.2 in magnitude.For all runs, the initial conditions were $w_1(0) = w_2(0) = 0$.

6. CONCLUSIONS

The REINFORCE algorithm has been analysed using weak convergence techniques. An example is presented to show that the algorithm can exhibit unbounded behaviour. This is also indicated by the simulation results. A modification has been proposed based on constrained optimization principles. Analysis shows that the zeros of the related ODE have an one-one onto relationship with the Kuhn Tucker points of the constrained optimization problem.

7. REFERENCES

1. Barto, A.G., and P. Anandan, 'Pattern Recognizing Stochastic Learning Automata', IEEE T-SMC, SMC-15, 1985, pp. 360-375.
2. Kushner, H.J., Approximation and Weak Convergence Methods for Random Processes, MIT Press, Cambridge, 1984.
3. Narendra, K.S., and M.A.L. Thathachar, Learning Automata - An Introduction, Prentice Hall, Englewood Cliffs, 1989.
4. Tsetlin, M.L., Automata theory and Modelling of Biological Systems, Academic Press, New York, 1973.
5. Williams, R.J., 'Toward a Theory of Reinforcement Learning Connectionist Systems', Technical Report NU-CCS-88-3, Northeastern University, 1988.
6. Zangwill, W.I., Nonlinear Programming: An Unified Approach, Prentice-Hall, Englewood Cliffs, 1969.