PFSA Modelling of Behavioural Sequences by Evolutionary Programming

Charlie H. Clelland and Douglas A. Newlands

School of Computing and Mathematics
Deakin University
Victoria 3217, Australia

E-mail: **fractal@deakin.edu.au and doug@deakin.edu.au**

Presented at: Complex '94 - Second Australian Conference on Complex Systems, 26-28 Sept 94, UCQ, Rockhampton, Queensland.

Citation: Clelland, C.H. & Newlands, D.A. (1994) "PFSA Modelling of Behavioural Sequences by Evolutionary Programming" in Stonier, R.J. & Yu, X.H. (eds) "Complex Systems: Mechanism for Adaptation", IOS Press, pp. 165-72.

Abstract

Behavioural observations can often be described as a sequence of symbols drawn from a finite alphabet. However the inductive inference of such strings by any automated technique to produce models of the data is a non-trivial task. This paper considers modelling of behavioural data using probabilistic finite state automata (PFSAs). There are a number of information-theoretic techniques for evaluating possible hypotheses. The measure used in this paper is the Minimum Message Length (MML) of Wallace.

Although attempts have been made to construct PFSA models by incremental addition of sub-strings using heuristic rules and the MML to give the lowest information cost, the resultant models cannot be shown to be globally optimal. Fogel's Evolutionary Programming can produce globally optimal PFSA models by evolving data structures of arbitrary complexity without the requirement to encode the PFSA into binary strings as in Genetic Algorithms. However, evaluation of PFSAs during the evolution process by the MML of the PFSA alone is not possible since there will be symbols which cannot be consumed by a partially correct solution. It is suggested that the addition of a "can't consume" symbol to the symbol alphabet obviates this difficulty. The addition of this null symbol to the alphabet also permits the evolution of explanatory models which need not explain all of the data, a useful property to avoid overfitting noisy data.

Results are given for a test set for which the optimal pfsa model is known and for a set of eye glance data derived from an instrument panel simulator.

1. Introduction

Inductive inference, the creation of models or hypotheses from data sets, has been accepted to be one of the primary formal operations of scientific inquiry since the 17th Century. However it has been known since then that a large and possibly infinite number of hypotheses may be inferred from a single data set. The selection of the best hypothesis inferred from a data set is guided by Occam's Razor. This may be interpreted as accepting the simplest hypothesis with the maximum explanatory capability. Ideally the process of inducing models should be capable of being automated by some computational procedure.

The problem of inductive inference may be considered as a search of possible models to find the optimal model which explains the data set. Some simplification of this search procedure

can be achieved by restricting the search to a single class of models. Even with this simplification it is known that inducing a single class of models (finite state machines) from a data set is NP-complete [Gold, 1978]. That is, the time requirement for an induction algorithm to find an optimal finite state machine is non-deterministic and increases as a polynomial function of the size of the data set. Given this limitation on the performance of automated induction procedures, the size of data sets which can be processed by a deterministic search algorithm is clearly limited [Gaines, 1976].

There are, however, a class of algorithmic techniques collectively known as Evolutionary Algorithms (EAs) which are non-deterministic and can give close to optimal solutions to many problems known to be NP-complete.

2. Problem domain

It is a long term goal in behavioural science to identify the underlying processes which mediate behaviour. As an initial step toward this goal the identification of regularities and patterns in observable behaviour is a necessary precursor activity. However this process is not trivial since most behavioural data is highly contaminated with noise and often the metrics of behaviour are not known. The simplest procedure to transform behavioural data for automated induction is to map the raw data into an finite alphabet of symbols [Solomonoff, 1964]. In one example used in the experimental work reported in this paper the data is eye-glance data obtained from an instrument panel simulator, the raw data is mapped to a 4-symbol alphabet where each symbol corresponds to an instrument on the simulator panel. The task is to find the optimal model which explains the scanning sequence embedded in the data.

3. PFSA models

The choice of model used is arbitrary; for this work Probabilistic Finite State Machine (PFSA) models [Rabin, 1963] are induced from the data sets. PFSAs are rather general models which consist of a set of states and transitions between those states. When a PFSA receives a symbol it "consumes" it if a transition for that symbol is present and moves to the state defined by the transition. In addition PFSAs have a probability associated with each outgoing transition from a state and can be used to predict the most likely symbol to be consumed next. In this application the PFSA transitions have a count of the number of symbols consumed on each transition associated with them, the probabilities are not calculated although in principle they can be. Note that unlike other Finite State models there are no output symbols associated with either transitions or states, in this respect the PFSA models used are similar to the Finite State Automata models used in parsing formal languages.

The PFSA models used are constrained by rules of construction, these are:

- There must be at least 2 states or nodes in the PFSA if self-transitions are not allowed or 1 if they are.
- At least one out-going transition from any state.
- At least one incoming transition to any state.
- For an alphabet of n symbols the maximum number of outgoing transitions is n and there is only one occurrence of any symbol on outgoing transitions from a state.

Since the task is to find the optimal PFSA model, an evaluation procedure is required. Notionally, this could be done by some sort of quasi-objective scoring of nodes and transition counts in each model generated. A better method is that of estimating the length of the message required to communicate the structure of a PFSA model. The optimal model is that which has the shortest estimated message length [Wallace and Georgeff, 1984]. The minimum message length (MML) for a PFSA can be approximated by the expression:

$$\sum_{j=1}^{N}\left[\log_2\left(\frac{(t_j+V-1)!}{(V-1)!\prod_{i=1}^{V} n_{ij}!}\right)+m_j\log_2 N\right]-\log_2(N-1)!$$

.where:

t_j	number of times the j*th* state is visited
V	number of different symbols in the output alphabet
n_{ij}	no. times the i*th* symbol is produced from j*th* state
m_j	no. of different transition symbols from the j*th* state
N	total number of states in the pfsa.

Estimate of minimum message length for a pfsa [Wallace and Georgeff (1984)]

"The expression contains 3 terms corresponding to the message length costs of:

- The number of states in the PFSA.
- For each state, the transition probabilities for each output symbol.
- The destination state of each output symbol from each state" [Patrick and Chong, 1987].

The message length is expressed in nits, that is, the natural logarithm (bits). MML measures can be compared in a similar way to statistical measures. The difference between 2 MMLs may be taken as an odds ratio expressed as a power of 2. A difference of approximately 6 is reasonable evidence for accepting one model or hypothesis over another [Wallace and Freeman, 1987].

4. Evolutionary Programming (EP)

EP is a rather venerable algorithm which is attracting a resurgence of interest in recent years, the primary text dates from the 1960s [Fogel, Owens and Walsh, 1966]. The algorithm, in part, emulates biological evolution and operates on a population of candidate solutions to a problem. In this case the candidate solutions are PFSA models coded as dynamically sized two-dimensional arrays. Unlike Genetic Algorithms (GAs) EP places no restriction on the representations used which allows for simple and direct representation of complex data structures. GAs are usually restricted to bit string representations which do not permit changes in data structure size during a run [Zhou and Grefenstette, 1986]. Although versions of the basic GA algorithm exist which permit dynamic bit strings [Goldberg, Korb and Deb, 1989] the mapping of the bit string to the desired structure represents a significant computational overhead. EP may be expressed in pseudocode as:

Initialise a population of N candidate solutions
Evaluate the population
while (not terminate condition)
Copy each parent to create N children
Mutate each child
Evaluate the children
Stochastically select the best N individuals
Discard the poorest N individuals
end while

4.1 Initialisation

For the studies reported the initial PFSAs were simple ring structures of 2 to 20 nodes with a single transition between each node. It was assumed that there was no existing knowledge about the PFSA models which could be used to initialise the population.

4.2 Evaluation

The evaluation function used in EP supplies selection pressure on the population. In this case the primary evaluation measure is the MML of each PFSA. Procedurally each PFSA model was presented with the data set and counts maintained on each transition of the symbols consumed by that transition. There are 2 other conditions which must be selected against.

- PFSAs which do not consume the entire data set.
- PFSAs which contain redundant transitions.

In the first case a simple procedure was implemented. An additional "can't consume" symbol was added to the symbol alphabet which was always expressed as a self-transition on each node a symbol could not be consumed on. This had 2 benefits:

- Every PFSA contained the same number of transition occurrences so that the MMLs of all evolved PFSAs were comparable.
- The total number of unconsumed symbols could be used as an evaluation measure.

Similarly the total number of redundant transitions in a PFSA, that is, those transitions which were never used in consuming the data set was used in the evaluation function.The evaluation function used was:

$$\textbf{\textit{Score = MML + constant * (No. of unconsumed symbols) + constant * (No. redundant transitions)}}$$

The constants were empirically determined and were set to the same value, usually 5.

4.3 Mutation

The mutations used were structural alterations of the PFSA consistent with the rules of construction of a PFSA. Each mutation was only accepted if the resulting PFSA could consume the data set. In part this restriction prunes the search space since the space which contains all "broken" and viable PFSAs is much larger than that for only viable PFSAs. The mutations used were:

- Add a transition
- Delete a transition
- Move the target of an outgoing transition.
- Add a node, this also requires 2 additional transitions, the incoming transition was either added to an existing node or an existing transition was moved to the new node.
- Delete a node.

The mutations were stochastically selected with each transition operation occurring 30% of total mutations and the node operations 5%. The norm in EP is to have equal probabilities for each mutation type but it was thought in this case the node operations would have gross and possibly disruptive effects on the PFSAs and were reduced in probability [Lutter and Huntsinger, 1969].

4.4 Selection

Selection occurred by "tournament" selection where each individual's evaluation measure in the population is compared to a randomly selected sample of the others in the population. A score of the number of "wins" determines the survival into the next generation. Tournaments of size 5 and 7 were used.

5. Experimental description

Two data sets were used:

5.1. "cba"

A 100 symbol set, which was known to be composed of the sequence "cba" interspersed with "d" [Gaines, 1976], a population size of 50 with 5-tournaments was used. Convergence was relatively rapid with the unconsumed symbol count falling to zero in approximately 20 generations with very little improvement noted after 100 generations. The mean evaluation score of 2 runs is shown in Fig. 1. with the 2 best PFSAs obtained in 20 runs shown in Fig. 2. The best PFSAs are 3-state with the best machine identifying the "cba" sequence. Gaines [Gaines, 1976] claimed the optimal PFSA was a 4-machine for this data set, this view is not supported by the present data.

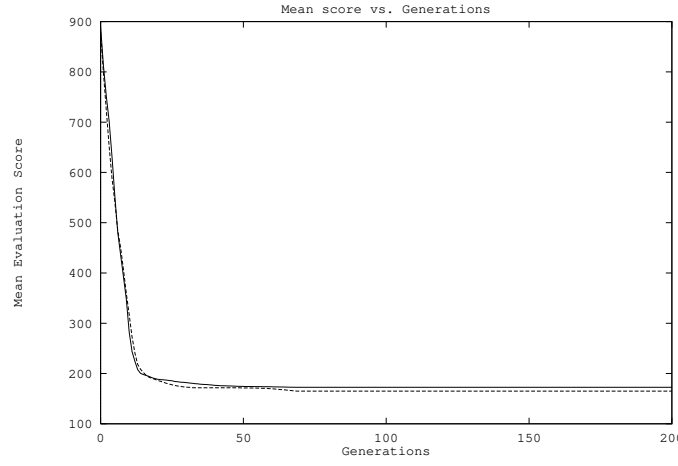

Figure. 1: Best runs for 100 symbol (cba) data set

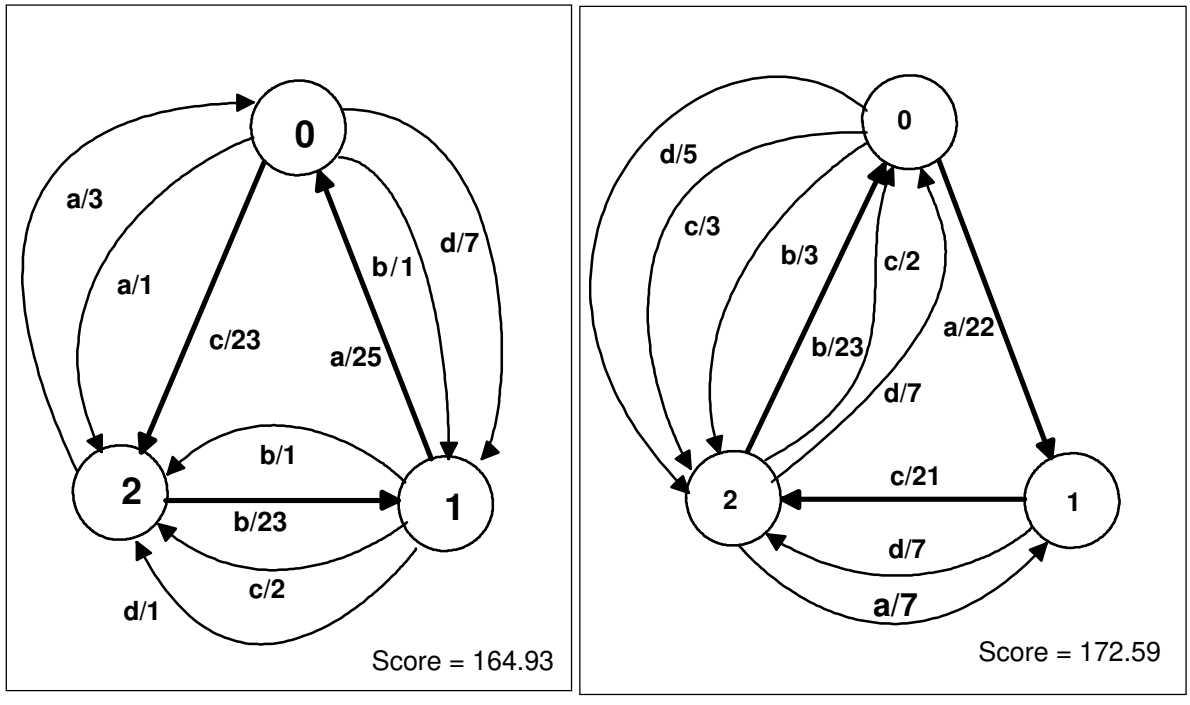

Figure 2: Best 2 PFSA models induced from 100 symbol (cba) data set.

5.2 Eye-glance

The second data set contained 2000 symbols and represented eye-glance data from an instrument panel simulator. In this case self-transitions were permitted in the PFSAs. A population size of 100 with 7-tournaments was used. The runs behaved in a similar fashion to the first data set although the convergence was somewhat slower. The best 2 PFSAs are shown in Figs. 3 and 4. Interpretation of these PFSAs is not obvious but there seems to be some sequences which occur with higher frequencies than others such as the sequence "bc" in both machines.

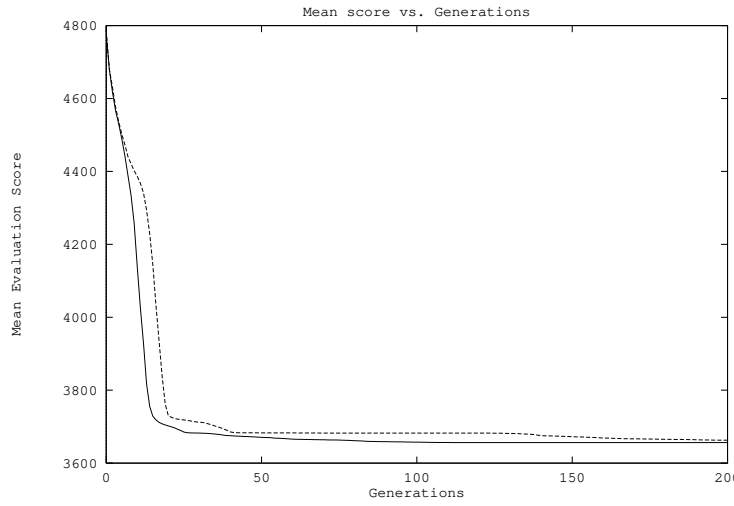

Figure 3: Best 2 runs for 2000 symbol (eye-glance) data set

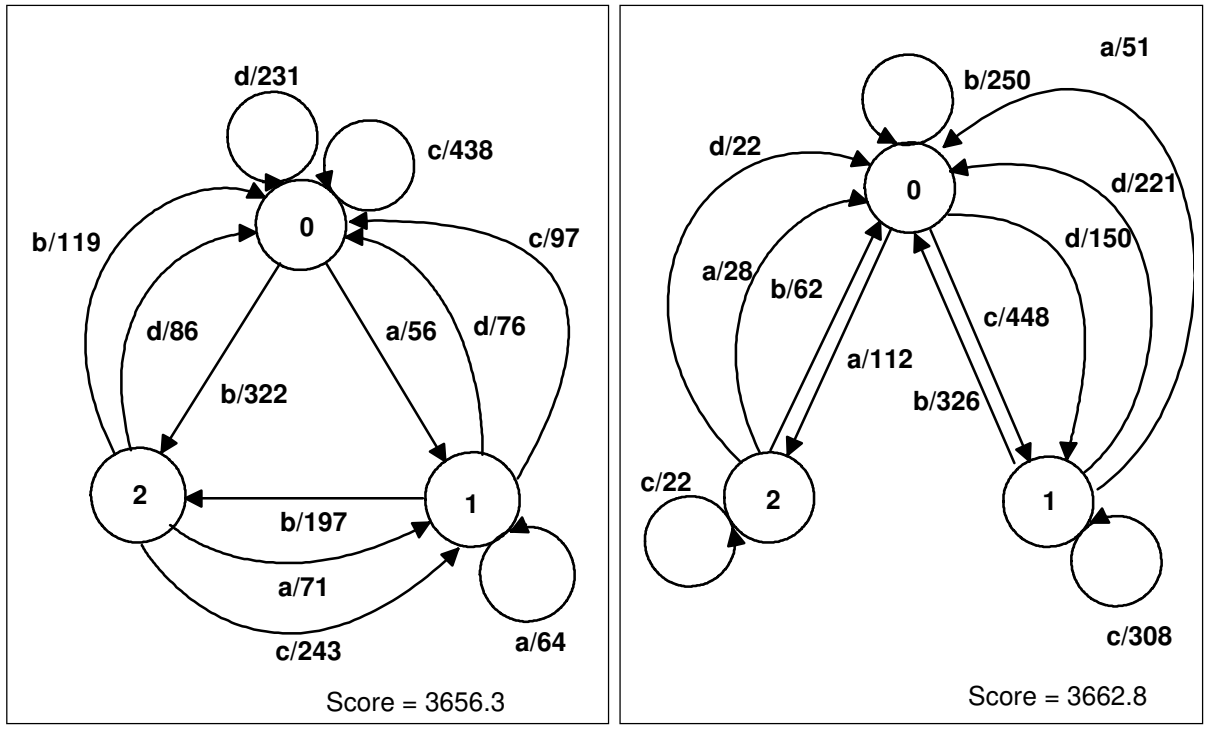

Figure 4: Best 2 PFSA nodels induced from 2000 symbol (eye-glance) data set.

6. Conclusions and Further Work

The combination of PFSA models and EP seems to show some possibilities for a robust rapid technique for inferring regularities and patterns in behavioural data. The technique appears to work on real data sets rather than on "toy" problems. However this conclusion must be considered tentative since the present method has no adequate way of avoiding "over fitting" noisy data. It is thought that by permitting the PFSA models to retain a proportion of "can't consume" symbols the evolutionary search may be able to retrieve regularities from noisy data sets. The present system underutilises the available information during the search, the MML

not only can give an assessment of PFSA structure but can also provide a measure of the effect of each mutation type on the PFSAs. This may allow for dynamic tuning of the mutation probabilities during a run in a similar fashion to evolving mutation schemes used in EP for continuous function optimisation [Fogel, 1993].

References

Fogel, L.J. (1993) "Evolving Mutation" Presented at Evolutionary Computation Workshop, Australian Joint Artificial Intelligence Conference, Melbourne, 1993.

Fogel, L.J., Owens, A.J. and Walsh, M.J. (1966) "Artificial Intelligence through Simulated Evolution" (pub) Wiley & Sons, New York.

Gaines, B.R. (1976) "Behaviour/structure transformations under uncertainty" Intl. J. Man-Machine Studies, 8, pp. 337-365

Gold, E.A. (1978) "Complexity of automaton identification from given data" Information and Control, 37, pp.302-320.

Goldberg, D.E., Korb, B. and Deb, K. (1989) "Messy Genetic Algorithms: Motivation, Analysis, and First Results" Complex Systems, 3, pp.493-530.

Lutter, B.E. and Huntsinger, R.C. (1969) "Engineering applications of finite automata" Simulation, July 1969, pp. 5 - 11.

Patrick, J.D. and Chong, K.E. (1987) "Real-time Inductive Inference for Analysing Human Behaviour" AI '87 - Proceedings Australian Joint Artificial Intelligence Conference, Sydney, pp. 305-322.

Rabin, M.O. (1963) "Probabilistic Automata" Information and Control, 6, pp.230-245

Solomonoff, R.J. (1964) "A Formal Theory of Inductive Inference. Part 1" Information and Control, 1, pp. 1-22

Wallace, C.S. and Freeman, P.R. "Estimation and Inference by Compact Coding" J. R. Statist. Soc. B, 49, 3, 240-265.

Wallace, C.S. and Georgeff, M.P. (1984) "A General Objective for Inductive Inference" Technical Report No. 32, Department of Computer Science, Monash University, Victoria, Australia.

Zhou, H. and Grefenstette, J.J. (1986) "Induction of Finite Automata by Genetic Algorithms" Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Atlanta, GA, pp.170-174.