Evolving Prototype Rules and Genetic Algorithm in a Combustion Control

Runhe Huang
Comp. Sci. and Eng. Lab., The University of Aizu, 965-80, Japan.
email: r-huang@u-aizu.ac.jp

Abstract-*All approaches to rule-based control have relied upon a pre-classification of the state-space for their success. Given a classification of the state-space, either an individual control action or a whole set of control actions can be learnt. But quality of control is limited by the quality of the state-space classification. When no classification is specified, a more challenging problem is how to arrive at the most suitable partitioning of the state-space with its associated control actions. The approach presented in this paper is to generate prototype rules. Instead of learning a control action for every point encountered, a genetic algorithm is used to learn control actions for a set of limited number of prototype states and the nearest neighbour matching is employed to decide which of the rules to fire in any particular situation when controlling a system. The example of a simulated multiple burner combustion optimization is used to demonstrate this approach.*

1 Introduction

All approaches to rule based control have depended on a pre-classification of the state-space for their achievements. Michie [Mic68] adjusted the probability, based upon the results of experience, of applying a force either left or right in each of 256 preclassified situations in the cart-pole balancing problem. Leitch's qualitative control [Lei88] depends completely upon the pre-identification of the qualitative boundaries defining the qualitative inputs as the basis for the so-called "deep knowledge" about the domain. Fogarty's genetic algorithm [Fog89] was developed and tested, in which the state-space of carbon monoxide and oxygen readings was pre-classified into 25 "boxes" and genetic algorithm was used to learn rules in each of 16 of the "boxes". So given a suitable classification of the state-space, various ways have been devised of optimising the action to take, in each of resulting partitions. However, when no classification of the state-space has been specified, the more challenging task is how to arrive at the most suitable partition of the state-space together with their associated control actions. Work has been done on the automation of the process of classification both in statistics [Bre84] and in machine learning [Qui86] but this has not been extended to or applied in the area of real-time control and industrial applications. Thus is it possible to automate half of the process of rule-construction; the decision process of what action to take, although the best action to take will be a consequence of, and largely depend upon, the choice of partitioning. The standard approach is to generate a condition part for each of rules that specify a hyper-rectangle over the state-space. For this approach, the problems of conflict resolution and non-matching of rules are met. To solve these problems, Moore [Moo90] takes a memory based approach: each control rule is a point that has been encountered in the state-space with an associated action to be optimised. This avoid all rule-matching problems but is an extreme solution in that nothing is forgotten. Given that it is not possible to learn a control action for every point encountered, the problem is to decide which points to learn a control action and how to use this limited number of point based rules to control a system. The problem can be solved by generating prototype control rules and a genetic algorithm is used to learn actions for these prototype points. Thus, a genetic algorithm is employed to learn actions for only a limited number of points rather than to learn actions for every point encountered. Therefore, our approach can overcome both problems in a standard approach and Moore's a memory based approach.

This paper is organized as follows: the concept of prototype control rules is given in the following section; In section 3, the process of evolving prototype control rules is described. Evolving a set of prototype control rules for a combustion control problem is ad-

0-7803-2081-6/95 $1.00 © 1995 IEEE

IA&C - 95 - 39

dressed in section 4. In section 5, the implementation and results of applying our approach to the combustion control problem are shown; Finally, the conclusions are drawn in section 6.

2 Prototype Control rules

A. Prototype theory in concepts cluster

Prototype theory has emerged from the last ten years of psychological research into concepts, such as "apple" and "fish". The term "prototype" has sometimes been used to mean a representation of the best example for a given concept. The distinguishing doctrine of prototype theory is that entities fall neither sharply in nor sharply out of a concept's extension [Osh81]. It construes membership in a concept's extension as graded, determined by similarity to the prototype of the concept, i.e. the concept's best exemplar or the central tendency of the concept; the boundary between membership and non-membership in a concept's extension is thus fuzzy. Specially, the instances of any concept vary in how typical they are rated, and such ratings predict how quickly and accurately an instance can be categorized. The rating function employed is usually a distance metric function. The function assigns smaller numbers to pairs of dissimilar ones, thereby reflects relative psychological similarity among elements of a set A. In particular, if subjects give similarity rating for pairs, and then these ratings are converted into a multidimensional Metric space. As for a concept's prototype, it is supposed to be the best example for the concept in a set A and the value of the prototype on each dimension is roughly equal to the average value of all scaled instances on this dimension in the multidimensional Metric space. This provides some criteria for identifying the prototype with average. However, as for a member of the set A, its membership is construed by mapping Metric distance from the prototype concept into a value of the characteristic function. Small distances are more likely identical to be members of the concept. To sum up, prototype theory in the issue of identifying concepts and construing membership of instances has the following main features:

- using Metric space to represent objects;
- defining a prototype concept as a point in this space;
- construing membership by mapping Metric distances from the prototype concept into values of the characteristic function.

B. Extension of prototype theory to process control

The prototype theory in the issue of concept cluster can be extended to states classification in a state-space for a process control. This work has not been done in the area of process control engineering. In this paper, the new concepts of prototype states and prototype control rules are introduced and defined below.

<u>Definition 2.2.1</u>

A system state $s \in S$ is called as a **prototype state** (**prototype point**) sp_i if it is the best examplar state (point) of its class $c_i \in C(c_1, c_2, ..., c_i, ..., c_m)$ or its geometrical value on each dimension roughly equal to the average value of all members of its class on this dimension.

The definition of a prototype state provides a computation method to obtain prototype states in a state-space. It is easy to envisage that a prototype state is in the centre of all members in its class and all members of the class are around it within some threshold distance. On other hand, the smaller Metric distance between two states or two prototype states or one state and one prototype state, the higher similarity rating they have and the more likely they belong to the same class.

<u>Definition 2.2.2</u>

A system state $s \in S$ is classified as a **member** of the class $c_i \in C(c_1, c_2, ..., c_i, ..., c_m)$ if and only if the Metric distance from the prototype state sp_i is less than the Metric distances from the rest other prototype states sp_j, where j=1, 2, ..., m but $j \neq i$.

Thus, for any system state s, its membership depends upon distances between it and prototype states. Metric distance provides a measurement of similarity rating in such a way that the smaller distance reflects closeness and greater similarity to the prototype state. However, for a system state which has more than one prototype states from which the Metric distances are the same, the membership of the state is then fuzzy. In such a case, the simplest way but effective solution is to randomly assign a class to the state among those classes from whose prototype states the state has the same distance. Thus, each state has only unique membership, there is no overlapping problem. Classified in this way, a state-space is not partitioned into a hyper-rectangular boxes. The boundary of each class could be any shape as long as satisfying above membership definition.

Definition 2.2.3

A prototype state together with its associated action or actions forms a **prototype rule.** When controlling a system, such prototype rule is called a **prototype control rule.** The solution to the problem of controlling a system is a set of such prototype control rules.

3 Evolving prototype control rules

A solution to the problem of controlling a system is a set of prototype control rules and whether the control system can succeed depends upon the quality of this set of rules. Since a set of prototype states represents a partitioning of the state-space, the quality of a set of prototype rules in this sense does mean both how accurately the set of prototype states represents a partitioning of the state-space is and how good the associated actions are in each of resulting partitions. Therefore both the process of refining partitioning of the state-space and the process of learning actions for the prototype states become critical. The combination of these two processes is defined as the process of evolving prototype control rules.

In order to obtain a set of better quality prototype control rules, a genetic algorithm is used to learn actions for prototype states. The genetic algorithm is based on the process of evolution [Hol75]. A population of strings is generated and evaluated using a fitness function. New strings are generated by selecting old ones in the basis of fitness based selection scheme and genetic operators such as mutation and crossover are applied to them. The resulting strings are evaluated and those with higher fitness join to the population with replacement of those with lower fitness from this population. Therefore the strings with a high strength survive and those with a low strength die out. The process is iterated until some criteria are met at which point it terminates. For combustion control system, binary strings are used and a string represents a prototype control rule. For each prototype state, its associated action or actions is learnt during the process.

A set of prototype control rules, used in conjunction with the nearest neighbour matching, are employed to control a system. Each action is learnt by using a genetic algorithm, so each action is the relatively best action corresponding to its associated prototype state. Thus, the quality of control system depends on largely the quality of the partitioning of the state-space which is represented by this set of prototype states. If the quality of the control system is not good enough to achieve the control task, then it is necessary to modify the partitioning of the state-space by means of lumping and splitting existent partitions. The lumping process is accomplished by using the nearest neighbour matching to select which rules is to fire when controlling a system and the splitting process can be achieved by generating more prototype states. When more prototype states are generated, the genetic algorithm is applied to learn actions for these prototype states and meanwhile some current partitions are split into finer partitions. The process is repeated until the set of prototype control rules is good enough to achieve task.

4 Prototype rules for combustion control

The problem of combustion control is to automatically and continually optimise combustion in a ten burner furnace or boiler plant by altering the air inlet valve to each of the burners depending upon the carbon monoxide and oxygen readings taken from their common flue. The optimum air and fuel mixture will be different for each burner because of their varying type, age and condition. Fogarty [Fog88] developed a learning system, in which the state-space of carbon monoxide and oxygen readings was pre-defined into 25 "boxes" and rules in each of 16 of the "boxes" was learnt. The problem of this approach is that it requires expert knowledge of how to divide state-space and an expert has to distinguish one partition from another. To avoid this problem, the approach described above is applied to the combustion control problem.

The simulation of the effect, on overall carbon monoxide and oxygen concentration in the waste gases, of alternations to the air inlet valve of each of a number of burners sharing a common flue is built using chemical equations based on theoretical models and the results of practical experiments. Each burner in the installation is represented by two perfect burners at each firing level. Each perfect burner is defined by the minimum excess air when it is 100% close and the maximum excess air when it is 100% open. The concentrations of carbon monoxide and oxygen to be expected are dependent on the amount of excess air. If assuming the excess air for each perfect burner can be obtained by an exponential function of the closure of the associated air inlet valve [Wak86], the oxygen and carbon monoxide concentration can then be calculated [Fog89].

For the ten burners installation, its average carbon monoxide and oxygen concentration are obtained by averaging their concentrations of ten burners. According to its average carbon monoxide and oxygen concentration, the energy loss of the system at a time is then calculated by using tables or approximating equations [Fog89]. The control object is thus to optimize combustion, that is, to minimize the energy loss of the system by altering the air inlet valve to each of the burners. The six kinds of actions are as follows:

- lean burner correction routine;
- rich burner correction routine;
- rich or lean burner correction;
- reducing air to all burners;
- increasing air to all burners;
- no action.

For a rule based control system, each of the rules for controlling a system has the form of production rule as follows:

IF system states THEN action or actions

In this context, the system state is composed of in term average oxygen concentration and average carbon monoxide concentration. The action is to adjust inlet air valves by using one of the six different types action described above. A system state can be regarded as a point in a two dimensional space with a specified range of [maximum average carbon monoxide concentration, maximum average oxygen concentration]. Therefore, each prototype rule has the following form:

IF prototype state(co.con.average, o2.con.average)
THEN action

Each action for a given prototype state can be coded into a 7-bit binary string, in which 3 bits depict six kinds control action type and 4 bits represent increment ranging from 0% to 20%. A genetic algorithm is used to optimize control action for each of given prototype states.

The parameters used in the genetic algorithm are as follows:

- population size is 40 and offspring size is 10;
- mutation probability is 0.03 and crossover probability is 0.96;
- binary string length is 7 bits.

For a genetic algorithm, **fitness function** is a key for its success. Without s suitable evaluation function, the genetic algorithm can hardly show a good performance. For this combustion control problem, fitness is a measurement of how good an action corresponding to a particular situation. Assuming the system state moves from the current state **S** to the new state **S'** after taking an action **A**, the energy loss (is also called as stackloss) E_l corresponding to the current state and the energy loss E_l' corresponding to the new state can be calculated [Fog89]. The fitness of the action **A** is expressed as follows:

IF $E_l < E_l'$
THEN fitness(A) = 0;
IF $E_l >= E_l'$
THEN fitness(A) = $(E_l - E_l')/(E_l - E_{l_{min}})$.

Where, $E_{l_{min}}$ is 4.48% which is minimum energy loss in a perfect burner but 4.8% is considered as the criteria for real industrial burners.

Since each action is learnt by using the genetic algorithm for a given prototype state, all actions are the relative best actions corresponding to their associated prototype states. The set of prototype states together with actions learnt form a set of prototype control rules to control the combustion system. Whether it is a good controller or not depends mainly the quality of the partitioning of the state-space which is represented by the set of prototype states. In this context, as a good controller, it should be able to reduce the energy loss by adjusting the each of ten burners' air inlet valves step by step until below certain level 4.8% and can cope with changes on firing level, which means that the energy loss may increase dramatically when the firing level is changed, but the controller should be able to bring the energy loss down step by step again. If it is not good enough to achieve this task, then it is necessary to modify the partitioning of the state-space. This is done by lumping or spliting existent partitions. The system starts with generating 200 random states in the state-space, the genetic algorithm is used to learn actions for these 200 initial states to form a set of initial rules. This set of rules is applied to the simulated combustion system in each trial run. During each trial run, the activated states are recorded. Those rules who fire in the last trial run are considered as a set of prototype rules and their associated states are regarded as a set of prototype states which represents a partitioning of the state-space. If this set of rules is not good enough to achieve the control task, then a new set of states is necessary to be formed as follows:

- prototype states in last trial run are kept;
- new states are generated randomly;
- new states are generated on the basis of the activated states in last trial run by selecting points within the activated points and applying genetic algorithm to them.

Thus a new set of states is generated and the process is iterated unitl a good controller is obtained. As it can be seen, the lumping process is conducted by selecting activated states in a trial run as prototype states and the splitting process is done by generating more prototype states in the state-sapce.

5 Implementation and results

Both the algorithm and the simulation of the combustion system are run on the T800 transputer on the Meiko computer surface. The successful set of prototype control rules as an intelligent controller is applied the simulated combustion system in which the firing level was changed with the probability of 0.0625. The energy loss is recorded at each step for 60 steps as the performance of the controller. The units used below in the experiment are:

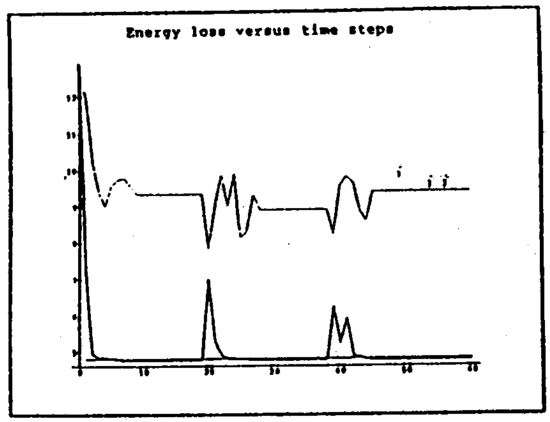
- for each burner state is %;
- for carbon monoxide reading is ppm;
- for oxygen reading is %.

<u>The results</u>

For the initial state of the system, its associated carbon monoxide and oxygen concentration readings are (7454, 0.7228) and the corresponding energy loss to the initial state of the system is $E_{l_{initial}} = 12.74564$.

For the final state of the system, that is, after the intelligent controller is applied to the combustion system, its associated carbon monoxide and oxygen concentration readings are (26, 1.1129) and the corresponding energy loss to the final state of the system is $E_{l_{final}} = 4.753204$.

The performance of the random controller which is a set of 200 randomlr generated rules is compared to the performance of the intelligent controller. Their performance versus time steps are shown in Fig 1. Where the line with 4.8% is the desired level. The upper plot line is the performance of the random controller and the below plot line is the performance of the elarning controller. The results show that the intelligent controller can optimise combustion by reducing energy loss down to below 4.8 (per cent) and keep it there when there are no firing level changes and no noise. The results also show that the controller can cope with both changes in the firing level and noise from carbon monoxide reading very well by bringing down energy loss from high caused by firing level change or noise to low.



6 Conclusions

The evolving prototype rules and genetic algorithm based approach can build a controller for a control system automatically without being given preclassfication of the state-space elicited from an expert. In comparing the performance of Fogarty's rule-based expert system, our approach has following advantages:

- it does not require much prior knowledge and preclassification of a state-space, therefore it overcomes the problems so called "knowledge acquisition bottleneck" and "falling off the knowledge cliff" in an expect system and other rule learning methods.
- it is very efficient at dealing with noise and firing level changes, while Fogarty found in his experiments that the performance of the expert system for optimising a combustion may be superior when there is no noise but it deteriorates considerably and proportionally as the level of noise increases and the fixed rule-based expert system can not cope with the changes on firing level.

The approach presented in this paper can be useful and practical to many applications in real industrial control.

7 Acknowledgements

The author wishes to thank Dr Terence C Fogarty for his helpful discussions.

References

[Bre84] Breiman, L., Friedman, J., Olshen, R. and Stone, C., *Classification and regression trees*, Wadsworth

[Fei83] Feigenbaum, E.A. and McCorduck, P., *Fifth generation:Artificial intelligence and Japan's computer challenge to the world*, 1983.

[Fog88] Fogarty, T. C.,*An expert system for optimising and controlling combustion in multiple burner furnances and boiler plants'*, in the Proceedings of the 27th IEEE conference on decision and Control, 1988, pp2249-50.

[Fog89] Fogarty, T.C., *Learning new rules and adapting old ones with the genetic algorithm*, in 'Artificial intelligent in Manufacturing' G. Rzevski (ed.), Springer-Verlag, pp275-290.

[Fog89] Fogarty, T.C., *Simulating multiple burner combustion for rule-based control*, presented at System Science X: 10th International Conference on System Science, China.

[Gol89] Goldberg, D., *Genetic algorithms in search, optimisation and machine learning*, University of Alabama.

[Gre90] Grefenstette, J.J., Ramsey, C.L. and Schultz, A.C.,*Learning sequential decision rules using simulation models and competition*,Machine Learning, volume5,number 4, pp335-381, 1990.

[Han89] Hanson, S.J. and Bauer, M., *Conceptual clustering, categorization, and polymorphy*, in 'Machine Learning', volume 3, number 4 pp343-372.

[Hol75] Holland, J. H., *Adaptation in natural and artificial systems*, Ann Arbor: The Univ. of Michigan Press, 1975.

[Hol87] Holland, J.H., *Escaping bristtleness: The possibilities of general-purpose learning algorithms applied to parallel rule-based systems* in 'Machine Learning: An Artificial Intalligent Approach' volume 2, pp 593-623, 1987.

[Hua92] Huang, R., *Evolving prototype rule for process control*, Ph.D dissertation, Univ. of the west of England, 1992.

[Lei88] Leitch, R., *Qualitative modelling of physical systems for knowledge based control*, NATO ASISeries, volume F47 Advanced Computing Concepts and Techniques in Control engineering, Edited by M.J.Denham and A.J.Laub. 1988.

[Mic68] Michie, D. and Chambers, R.A., *BOXES:an experiment in adaptive control*, in Dale, E. and Michie, D. (eds) 'Machine Intelligence 2' pp137-152.

[Moo90] Moore, A. W., *Acquisition of dynamic control knowledge for a robotic manipulator*, in the Proceedings of the 7th International Conference on Machine Learning, 1990, pp137-152.

[Osh81] Osherson D.N. and Smith, E.E., *On the adequacy of prototype theory as a theory of concepts*, in the Journal of Cognition, 9, pp35-58.

[Qui86] Quinlan, J.R., *Induction of decision trees*, in 'Machine Learning 1', pp81-106.

[Smi81] Smith, E.E. and Medin, D.L., *Categories and concepts*, Cambridge, MA:Harvard University Press.

[Smi84] Smith, E.E. and Osherson D.N., *Conceptual combination with prototype concepts*, Cognitive Science, volume 8, pp337-361.

[Wak86] Wakefield, A.C., *An expert system for methane flame stoicio-metric control*, MSc dissertation, Bristol Polytechnic.

[Web73] Weber, T.W., *An introduction to processing dynamics and control*, Wiley-Interscience.