DETERMINATION OF QUANTIZATION INTERVALS IN RULE BASED MODEL FOR DYNAMIC SYSTEMS

Chien-Chung Chan
Department of Mathematical Sciences
University of Akron, Akron, OH 44325-4002
r1ccc@vax1.cc.uakron.edu

Celal Batur and Arvind Srinivasan
Department of Mechanical Engineering
University of Akron, Akron, OH 44325-3903

Abstract---This paper introduces two adaptive procedures for quantizing continuous data used by symbolic empirical learning programs to generate rule based models for dynamic systems. Empirical results show that refining quantization intervals based on performance of single rules can produce a smaller number of quantization intervals than using uniform quantization procedures.

I. INTRODUCTION

Recently, a linguistic model building approach for uncertain dynamic systems has been proposed in [3]. In this approach, a model is characterized by a set of production rules obtained by an automated knowledge acquisition methodology [4, 10, 11]. The technique does not require the specification of an explicit model structure, but a number of past process inputs and outputs has to be specified as a set of model attributes. This choice corresponds to the model order determination in other process modeling techniques. The advantage of this technique is that it is completely data driven and requires a minimum amount of assumptions for the model structure.

One of the requirements of building linguistic models is the quantization of continuous data. The process of quantization determines the performance of a model and the number of induced rules for representing the model. It was demonstrated in [3] that by increasing the number of quantization intervals the process-model error measured by the error variance between the process and the model output is reduced, but the number of rules also increases by a product factor depending on the training pattern and the number of quantization intervals used.

The contribution of this paper is to introduce an adaptive procedure for tuning the quantization intervals of continuous data such that performance of a rule based model is improved at the cost of increasing the number of quantization intervals as small as possible. The basic idea is to use a top-down iterative procedure for refining quantization intervals selectively. In each iteration, the quantization interval having maximum overall error rate is selected for refining. Each time a selected interval is divided into two new equal intervals. Based on the new quantization intervals, a new set of rules is generated and performance associated with each quantization interval is evaluated again. The refining procedure is applied repeatly until a user-specified performance index is reached.

The paper is organized as follows. The linguistic approach for modeling uncertain dynamic systems is reviewed in Section 2. The problem statement is given in Section 3. Two adaptive procedures for tuning quantization intervals of a linguistic model is presented in Section 4. In Section 5, the method is tested by two examples, one involving a simulated system, the other a real life gas furnace. Finally, concluding remarks are given in Section 6.

II. RULE BASED MODELS FOR UNCERTAIN DYNAMIC SYSTEMS

Modeling of non-deterministic dynamic systems is generally performed under three methodologies. Depending on the nature of assumptions involving uncertainties in the systems, these methodologies can be classified as: (a) parametric model building [1, 9], (b) fuzzy model building [2, 12, 14], and (c) neural network based model building [5 - 8].

Recently, decision trees and production rules have been used to model non-deterministic dynamic systems [3, 13]. The requirements and steps for generating a rule based model can be summarized in the following figure.

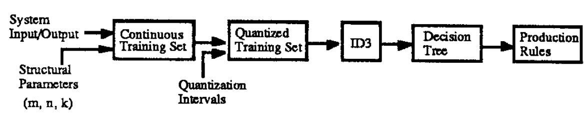

Figure 1. Requirements and steps for rule based model generation.

Figure 1 shows that a user must provide the triple (m, n, k) denoting the process order (m, n) and the dead time k, and quantization intervals to be used for generating a quantized training set. The set is used by an inductive learning program ID3 [10, 11] to generate a decision tree, which is then converted into a set of production rules.

III. PROBLEM STATEMENT

From Figure 1, we can see that one of the requirements of building rule based models is the quantization of continuous data. The process of quantization determines the performance of a model and the number of induced rules for representing the model. It was demonstrated in [3] that by increasing the

ISSN# 0-7803-0233-8/91 $1.00©1991 IEEE

number of quantization intervals uniformly the process-model error measured by the error variance between the process and the model output is reduced, but the number of rules also increases by a product factor depending on the training pattern and the number of quantization intervals used.

In this paper, we introduce a procedure for tuning the quantization intervals of continuous data such that performance of a rule based model is improved at the cost of increasing the number of quantization intervals as small as possible.

In the following discussions, a model for a given dynamic system is assumed to be in the following form

$$y(t) = f(Y_{t-1}, U_{t-1}, \theta) + n(t)$$

where the vectors Y_{t-1} and U_{t-1} indicate the past values of outputs y(t) and inputs u(t) as

$$Y_{t-1} = (y(t-1),\ y(t-2), ..., y(t-n))^T$$
$$U_{t-1} = (u(t-k-1), u(t-k-2), ..., u(t-k-m))$$

The process order is effectively defined by the (m, n) pair and k indicates the dead time. The term n(t) is the process noise and θ is a set of parameters describing the dynamics through function $f(\cdot)$.

Under this configuration the problem can be stated as follows. Given a set of observation as

$$\Phi(t) = \{Y_t, U_t;\ t = 1, 2, ..., N\}$$

and the triple (m, n, k), determine a set of quantization intervals as small as possible satisfying the user-specified performance index.

IV. ADAPTIVE QUANTIZERS

One fundamental issue in the application of decision tree based methodologies to dynamic systems is the number of quantization intervals that one must consider before the input/output data are quantized. [3] have shown that one practical way to resolve this problem is to reduce the quantization interval uniformly until the performance of decision tree, as a process model, is at an acceptable level. This procedure does not have any selectivity feature, apart from the overall performance index. What is desired however is an adaptive procedure that will decrease the quantization interval of data in regions where the performance needs to be improved. We propose two new procedures to implement an adaptive quantization on input/output data. The first procedure looks at the rule set which performed poorly and changes the quantization intervals associated with the attributes of this rule set. We will call this first procedure as adaptive quantizer based on rule set. Our second procedure looks at each individual rule and defines a Rule Performance Measure (RPM) associated with each rule. Now, the adaptive quantizer works only on the rule which has the poorest performance measure. We will refer to this procedure as adaptive quantizer based on rule performance measure. Details of these procedures are explained as follows.

A. Adaptive Quantizer Based on Rule Set

Consider Figure 2 where input/output data of a dynamic system is shown. Suppose that only two quantization intervals are used to quantify both input and output data. These intervals are shown in the same figure. For example, interval 1 is defined as

$$\text{Interval } 1 \triangleq \forall\ y(t) > y^*.$$

We expect two sets of rules under this configuration. These are defined below

$$\text{Set } 1 \triangleq \{\text{ all rules resulting in output in interval } 1\}$$
$$\text{Set } 2 \triangleq \{\text{ all rules resulting in output in interval } 2\}.$$

For example, rules in Set 1 might appear like the following.

Rule 1:

if $y(t-1) = q_{2,y}$ and $u(t-1) = q_{1,u}$ then $\hat{y}(t) = q_{1,y}$

Rule 2:

if $y(t-1) = q_{1,y}$ and $u(t-1) = q_{1,u}$ then $\hat{y}(t) = q_{1,y}$

where $q_{1,u}$ indicates that input $u(t-1)$ is at interval 1. Same definitions are also applied to $q_{1,y}$ and $q_{2,y}$.

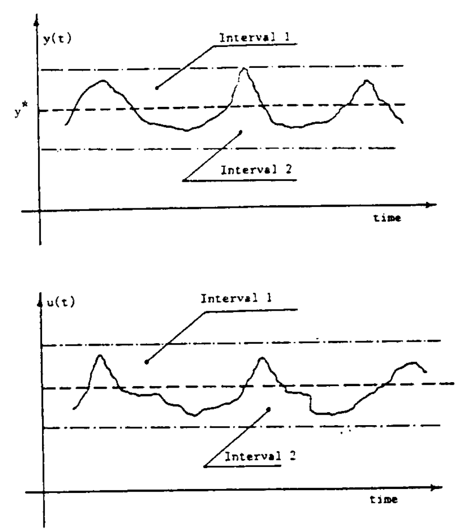

Figure 2. Initial quantization intervals.

To evaluate the validity of this initial quantization intervals, we compare Set 1 and Set 2 by evaluating their performance. The performance measure is based on the difference between the actual process output and the process output predicted by rules in the set. For instance, the performance measure of Set 1 is

$$J_1 = \sum_{i_1} \sum_j (y^{(j)}(t) - \hat{y}^{(i_1)}(t))^2$$

where the summation index (i_1) refers to all the rules in the first set and index j ranges over testing data points satisfying the (i_1)-rule in the set. The real process output associated with the j-th testing data point is denoted by $y^{(j)}(t)$. The predicted output of the (i_1)-rule in Set 1 is denoted by $\hat{y}^{(i_1)}(t)$, and this value is the same for all rules in the set.

Likewise J_2 is evaluated as

$$J_2 = \sum_{i_2} \sum_j (y^{(j)}(t) - \hat{y}^{(i_2)}(t))^2$$

Our rationale to change the quantization interval is based on the performance measure J. For example, if

$$J_2 < J_1$$

then we conclude that the quantization intervals associated with rules in Set 1 have to be improved. Since $q_{1,y}$, $q_{1,u}$, and $q_{2,y}$ are the intervals associated with rules in Set 1, we further decrease these intervals by a factor of two as shown in Figure 3.

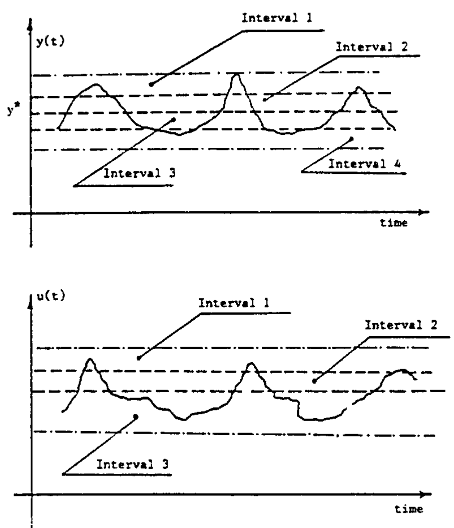

Figure 3. Refined quantization intervals.

Based on these different quantization intervals, a new decision tree is generated. From the new decision tree, we obtain a new family of rule sets. The performance of each rule set is evaluated, and quantization intervals associated with the poorest set of rules are refined again. The refining procedure is applied repeatedly until covergence is obtained.

Our decision of covergence is based on the following criterion.

$$J = \sum_{i=1}^{L} (J_i^{(k)}) \leq J_T$$

where L is the total number of rule sets in iteration (k), J is the overall performance index in iteration (k), and J_T is the user defined threshold.

B. Adaptive Quantizer Based on Rule Performance Measure

Referring to Figure 2, we assume that there are two quantization intervals initially. based on this quantization, a decision tree is generated and the performance of each rule is evaluated by a performance measure. We define a Rule Performance Measure (RPM) as

$$RPM_i = \sum_j (y^{(j)}(t) - \hat{y}^{(i)}(t))^2$$

where index j ranges over testing data points satisfying the i-th rule, $y^{(j)}(t)$ is the actual output of the process, and $\hat{y}^{(i)}(t)$ is the output predicted by the i-th rule.

The motivation behind evaluating performance of each rule is that if a rule is performing poorly then quantization intervals associated with its attributes should be improved. Note that this approach differs from the one suggested in Section A where instead of an individual rule, a set of rules is considered to measure the performance of current quantization intervals.

Each rule is evaluated by its RPM and the worst rule is determined as the one which creates the largest RPM. Condition and conclusion attributes of this rule are identified and the quantization intervals associated with the attributes are reduced by a factor of two. Based on this new quantization interval, a new decision tree is generated and the procedure continues until convergence.

The stopping criterion for convergence assessment is defined as

$$J = \sum_i RPM_i^{(k)} < J_T$$

where k is the iteration number which is incremented by one every time when a new decision tree is generated, the summation index i ranges over all the rules in the k-th iteration, J is the overall performance index, and J_T is a user defined threshold reflecting the user's assessment on how tight the process output should be predicted.

C. Effects of Heuristics

The following two parameters have to be assigned by the user:

(a) the threshold J_T used as stopping criterion for adaptive quantizers and

(b) a new quantization interval, i.e., the number of further division of region of data where the performance is found unsatisfactory.

The threshold J_T is defined as a function of prediction error $(y(t) - \hat{y}(t))$. At the expense of increasing the complexity of decision trees, one can lower the prediction error as much as possible. However, if the process output is noisy with a noise variance of σ^2 then the noise variance σ^2 defines the minimum achievable limit of the performance index J_T.

The proposed algorithms essentially identify regions of data where further quantization is required. Once a region is identified as the region that needs improvement, we increase the number of quantization intervals by a factor of two.

V. EXPERIMENTAL RESULTS

Two systems are considered to test the adaptive quantizers introduced in previous section.

A. Simulated System

The system 1 is a linear process presented in its discrete form as

$$y(t) = 1.5\, y(t-1) - 0.7\, y(t-2) + u(t-1) + 0.5\, u(t-2)$$

This corresponds to a continuous second order system proceeding a zero order hold and the input/output data are sampled with one second time interval, i.e.,

$$\frac{Y(s)}{U(s)} = \frac{1 - e^{-s}}{s} \frac{2}{1.443\, s + 1}$$

Figure 4 shows the response of this system to a pseudo random binary input.

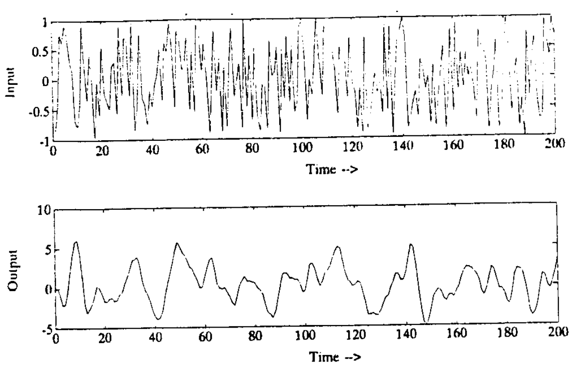

Figure 4. Response of simulated system.

Using a performance index threshold $J_T = 100$, the number of quantization intervals produced by the uniform quantizer and the adaptive quantizers introduced in previous section is shown in Table 1.1 – 1.3, where y(t – 1), y(t – 2), u(t – 1), u(t – 2), and y(t) are attributes used to characterize training data and J denotes the overall performance index. The number of rules generated by ID3 program is based on a quantized training set of 200 data points. Since the values of y(t – i), for i=0, 1, 2, come from the same signal source, the number of quantization intervals in these columns are the same. This also holds for u(t – i).

TABLE 1.1
Uniform Quantizer

# Intervals in attributes					J	# Rules
y(t – 1)	y(t – 2)	u(t – 1)	u(t – 2)	y(t)		
2	2	2	2	2	1077.5	6
3	3	3	3	3	1071.0	20
4	4	4	4	4	505.9	30
5	5	5	5	5	411.6	63
6	6	6	6	6	327.4	103
7	7	7	7	7	211.5	115
8	8	8	8	8	132.9	141
9	9	9	9	9	141.8	141
10	10	10	10	10	73.7	136

TABLE 1.2
Adaptive Quantizer Based on Rule Set Performance

# Intervals in attributes					J	# Rules
y(t – 1)	y(t – 2)	u(t – 1)	u(t – 2)	y(t)		
2	2	2	2	2	1077.5	6
4	4	4	4	4	505.9	30
7	7	8	8	7	33.7	141

TABLE 1.3
Adaptive Quantizer Based on Single Rule Performance

# Intervals in attributes					J	# Rules
y(t – 1)	y(t – 2)	u(t – 1)	u(t – 2)	y(t)		
2	2	2	2	2	1077.5	6
4	4	2	2	4	527.0	13
5	5	2	2	5	389.4	21
6	6	2	2	6	331.7	46
7	7	4	4	7	320.5	103
10	10	7	7	10	83.1	158

B. Gas Furnace

The system 2 of our second example involves real life gas furnace data reported by Box and Jenkins, 1970. The plot of system response is shown in Figure 5.

Using a performance index threshold $J_T = 90$, the number of quantization intervals produced by the three quantizers is shown in Table 2.1 –2.3. Since the number of quantization intervals for attributes y(t – 1), y(t – 2), u(t –1), and u(t – 2) are the same as y(t) and u(t) respectively, they are omitted from the tables. The number of rules generated by ID3 is based on a quantized training set of 190 data points.

VI. CONCLUDING REMARKS

This paper studies the relationship between quantizing continuous data and performance of rule based models for complex dynamic systems. Two adaptive quantizers, one based on performance of rule sets and the other based on performance of single rules, are compared to a uniform quantizing procedure. Two systems, one involving a simulated second order system and the other a real life gas furnace, are used to test the proposed adaptive quantizers. Given user-specified performance indices, the adaptive quantizer based on performance of single rules produces smaller number of quantization intervals than the uniform quantizer in both cases. The adaptive quantizer based on rule set performance generates the smallest number of intervals in the simulated system, but in the real life gas furnace case, it produces the largest number of quantization intervals.

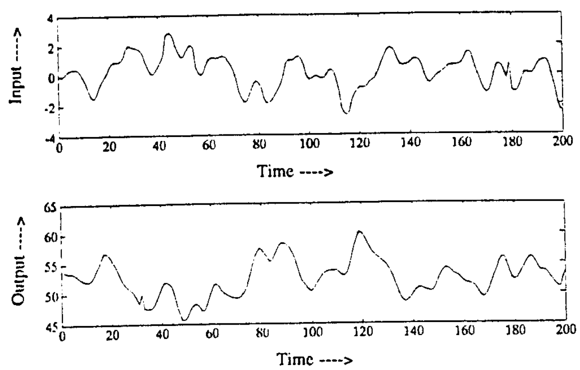

Figure 5. Gas furnace data.

TABLE 2.1
Uniform Quantizer

# Intervals in attributes		J	# Rules
y(t)	u(t)		
2	2	1805.6	7
3	3	752.0	6
4	4	481.6	12
5	5	314.8	23
6	6	266.7	37
7	7	189.6	39
8	8	143.1	55
9	9	120.6	66
10	10	101.2	76
11	11	83.8	86

TABLE 2.2
Adaptive Quantizer Based on Rule Set Performance

# Intervals in attributes		J	# Rules
y(t)	u(t)		
2	2	1805.6	7
3	4	1679.6	9
6	7	397.3	31
11	23	132.3	94
14	32	54.2	127

TABLE 2.3
Adaptive Quantizer Based on Single Rule Performance

# Intervals in attributes		J	# Rules
y(t)	u(t)		
2	2	1805.6	7
4	4	481.6	12
6	4	422.6	19
7	5	152.1	48
8	6	135.2	55
9	7	104.2	74
10	8	74.5	89

REFERENCES

[1] Astrom, K.J. and Wittenmark, B. (1989). *Adaptive Control*. Addison-Wesley.

[2] Batur, C. and Kasparian, V. (1991). Predictive fuzzy expert controllers. *Computers Ind. Engng.*, Vol. 20, No. 2, 199-209.

[3] Batur, C., Srinivasan, A., and Chan, C.-C. (1991). Automated rule based model generation for uncertain complex dynamic systems. *Proc. of 1991 IEEE Int. Symposium on Intelligent Control*. (To appear)

[4] Chan, C.-C. and Grzymala-Busse, J.W. (1990). On attribute redundancy and the learning programs ID3, PRISM, and LEM2. (Submitted for publication).

[5] Chen, S., Cowan, C.F.N., Billings, S.A., and Grant, P.M. (1990a). Parallel recursive prediction error algorithm for training layered neural networks. *Int. J. Control*, Vol. 51, No. 6, pp. 1215 - 1228.

[6] Chen, S., Billings, S.A., and Grant, P.M. (1990b). Non-linear system identification using neural networks. *Int. J. Control*, Vol. 51, No. 6, pp. 1191 - 1214.

[7] Kawato, M., Uno, Y., Isobe, M., and Suzuki, R. (1988). Hierarchical neural network model for voluntary movement with application to robotics. *IEEE Control Systems*, pp. 8 - 16, April.

[8] Levin, E., Gewirtzman, R., and Inbar, G.F. Neural network architecture for adaptive system modeling and control. *IEEE publications*, Catalog Number 89CH2765-6.

[9] Ljung, L. (1987). *System Identification*. Prentice-Hall.

[10] Quinlan, J.R. (1983). Learning efficient classification procedures and their application to chess end games. In R.S. Michalski J.G. Carbonell, and T.M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach*, pp.463-482. Tioga.

[11] Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning* 1, pp. 81-106.

[12] Tong, R.M. (1978). Synthesis of fuzzy models for industrial processes – some recent results. *Int. J. General Systems*, Vol. 4, pp. 143 - 162.

[13] Wehenkel, L. and Pavella, M. (1991). Decision trees and transient stability of electric power systems. *Automatica*, Vol. 27, No. 1, pp. 115-134.

[14] Xu, C.-W. (1989). Fuzzy system identification. *IEEE Proceeding*, Vol. 136, Pt.D, No. 4, July.