# DESIGNING NEUROFUZZY SYSTEM BASED ON IMPROVED CART ALGORITHM

**Jia Li  Li Erguo and Yu Jinshou**

*Research Institute of Automation, East China University of Science and Technology, Meilong Road 130, Shanghai, 200237, People's Republic of China ,*
*Tel.: +86-021-64253399; Fax: +86-021-64253399; Email address: jli75@263.net*

Abstract: In this paper, a neuro-fuzzy system based on improved CART algorithm (ICART) is presented, in which the ICART algorithm is used to design neuro-fuzzy system. It is worth noting that ICART algorithm partitions the input space into tree structure adaptively, which avoids the curse of dimensionality (number of rules goes up exponentially with number of input variables). Moreover, it adopts density function to construct the local model for every node in order to overcome the discontinuous boundaries existed in CART algorithm. To illustrate the validity of the proposed method, a practical application are done. 

Keywords: decision trees; CART algorithm; ICART algorithm; neurofuzzy system; hydrocracking processing

## 1. INTRODUCTION

According to the published papers about neuro-fuzzy system, there are still important open problems in the neuro-fuzzy system. At first, most of the current neuro-fuzzy approaches address parametric identification or learning only. In general, the designer chooses membership functions shape and the respective parameters are adjusted Mauricio (1999). Secondly, for some neuro-fuzzy systems, e.g., the fuzzy inference network in Wang (1994), the self-organizing neural-network-based fuzzy system in Yin (1999), neuro-fuzzy networks in Mauricio (1999), fuzzy neural networks in Meng (2000), etc., the number of partitions or the cluster radius is determined by the user, which can't guaranteed a optimal fuzzy system. In addition, extracting significant input variables among all possible input candidates is another challenging problems in fuzzy structure identification.

Considering above disadvantages, decision tree is another useful tool to construct the neuro-fuzzy system and choose the input variables, which is currently the most highly developed technique for partition. It is generated from training data in a top-down, general-to-specific direction. The initial state of a decision tree is the root node that is assigned all the examples from the training set. If it is the case that all examples belong to the same class, then no further decisions need to be made to partition the examples, and the solution is complete. If examples at this node belong to two or more classes, then a test is made at the node that will result in a split. The process is recursively repeated for each of the new intermediate nodes until a completely discriminating tree is obtained. Obviously, the advantages are decision tree's understandable representation and adaptability to the inference Serge (2001). There are many methods have been used for modeling decision tree, such as ID3 and ID4 using entropy criteria for splitting nodes, SLIQ utilizing data structures and processing methods to build decision tree, CART utilizing the GINI for splitting nodes and so on. However, in these methods, classification and regression trees (CART) has been in extensive use, which was developed to analyze categorical and continuous data using exhaustive searches and computer intensive testing to select a decision tree by Breiman in 1984. Crawford (1989) states that in cases where data is "noisy" , CART is " a remarkably sophisticated tool for concept induction" . Jang et al. (1997), based on CART algorithm, propose a quick method to solve the problem of the fuzzy rule generation. This method generates a tree partition of the input space, which relieves the problem of curse of dimensionality (number of rules goes up exponentially with number of inputs) associated with grid partition. Moreover, the method combines CART with artificial neuro-fuzzy inference system (ANFIS) approach to complete the task of fuzzy modeling and provides a new approach for neuro-fuzzy designing. There are no similar articles appeared in recent years.

After deeply researching the CART algorithm, an improved CART algorithm, abbreviated as ICART algorithm, adopting density function to construct the local model for every node is proposed in this paper. It is worth noting that it decides every decision output value according to space distribution and thus smoothes the discontinuous boundaries existed in CART algorithm. This advantage is obvious especially when the decision tree is smaller. Then a neuro-fuzzy system based on ICART algorithm, which using ICART algorithm to design neuro-fuzzy system is proposed. In this method, ICART algorithm is used to elect relevant inputs and classify the input space into adaptive tree structure, which avoids the curse of dimensionality because the total number of fuzzy rules doesn't increase exponentially with the number of input variables and neuro-fuzzy system is utilized to refine the regression and make it smooth and continuous everywhere. It can be seen that ICART and neuro-fuzzy system are complementary and their combination makes a solid approach to fuzzy modeling.

This paper is organized as follows. In section 2 we introduce the designing neuro-fuzzy system based on ICART algorithm. It consists of ICART algorithm, neuro-fuzzy system based on ICART algorithm and optimization algorithm. In section 3 the method proposed in this paper is applied to quality prediction for hydrocracking processing. Finally, section 4 contains some conclusions.

## 2. DESIGNING NEURO-FUZZY SYSTEM BASED ON ICART ALGORITHM

### *2.1 Decision Tree*

Decision trees are generated from training data in a top-down, general-to-specific direction. The initial state of a decision tree is the root node that is assigned all the examples from the training set. If it is the case that all examples belong to the same class, then no further decisions need to be made to partition the examples, and the solution is complete. If examples at this node belong to two or more classes, then a test is made at the node that will result in a split. The process is recursively repeated for each of the new intermediate nodes until a completely discriminating tree is obtained.

A typical decision tree with three-dimensional input-vector and one-dimensional output-vector is showed as Fig.1. Where $x_1$ , $x_2$ and $x_3$ are respectively the three inputs and $y$ is the output. The decision tree is a tree structure that represents a subspace of all the possible rules. It consists of internal nodes (with two children) and terminal nodes (without children). Each internal node is associated with a decision function to indicate which node to visit next, while each terminal node shows the output of a given input vector that leads the visit to this node (Duan, 2001, Jang, 1997 and Serge, 2001). Obviously the decision tree in Fig.1 classifies the input space into five non-overlapping rectangular regions. Each is assigned a constant value $b_i$ as its decision output value, which is the output value of the given input vector. The main advantage of this decision tree is that it is a very easy-to-interpret representation of a nonlinear input-output mapping (Quinlan, 1986). They generate incomplete rules constrained to a given partitioning and offer a compact description of a given context by using only the locally most significant variables.

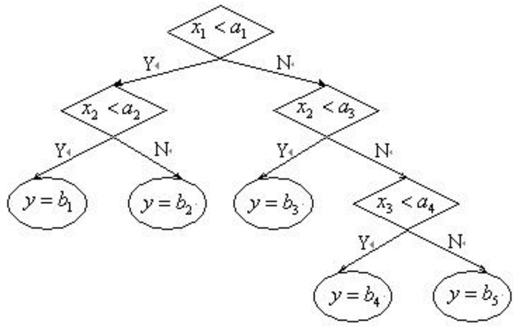



Fig. 1 The structure of decision tree

### *2.2 ICART algorithm*

In this section, we will describe the ICART algorithm. Before proceeding, the definition of CART algorithm must be introduced. The CART technique can be generalized as involving the partitioning of training data into terminal nodes by a sequence of binary splits, starting at a parent node. The procedure searches through all values of all the independent variables to obtain the variable and the value that provides the best split into child nodes. Once a best split is found, CART repeats the search process for each child node, continuing recursively until further splitting is impossible or stopped for some reason. Splitting is not possible if only one case remains at a particular node or if all the cases at that node are identical copies of each other. When all branches from the root reached terminal nodes, the tree was considered complete. CART produces more robust results by generating what is called a maximal tree and then examining smaller trees obtained by pruning away branches of the maximal tree. The important point is that CART trees are always grown larger than they need to be and are then selectively pruned back (Ina,1998). The final tree is picked up as the tree that performs best when the test data set is presented.

For terminal nodes with constant output values, CART can always construct an appropriate tree with a right size and, at the same time, find which inputs are irrelevant and thus not used in the tree. The

processing of determining the constant output values is stated as follows (Breiman, 1984).
For node t, the error function can be defined as:

$$E(t) = \min_{q} \sum_{i=1}^{N(t)} (y_i - d_t(x_i, q))^2 \qquad (1)$$

Where $\{x_i, y_i\}$ is a pair of input and output data, $d_t(x_i, q)$ is the local model of node t, $N(t)$ is the number of input and output data pairs belonging to node t and $q$ is variable parameters.

If $d(x, q) = q$ is a constant function, the $q$ that can generate minimal error function $E(t)$ is

$$q^*_t = \frac{1}{N(t)} \sum_{i=1}^{N(t)} y_i \qquad (2)$$

Obviously, the CART algorithm only uses the average output of node t as its predictive output. This may cause discontinuous boundaries. In order to overcome this drawback, our improved CART (ICART) algorithm adopts the distributing density function. For node t, the predictive output is determined by the average output, the maximum output and the minimum output of this node, namely

$$q^*_t = (MAX\_Y_t - AVE\_Y_t)\exp\left(-x_{t1}\|MAX\_U_t - x\|^2\right) + (MIN\_Y_t - AVE\_Y_t)\exp\left(-x_{t2}\|MIN\_U_t - x\|^2\right) + AVE\_Y_t$$

$$AVG\_Y_t = \frac{1}{N(t)} \sum_{i=1}^{N(t)} y_i \qquad (3)$$

Where $MAX\_Y_t$ is the maximum output of node t, $MIN\_Y_t$ is the minimum output of node t, $AVG\_Y_t$ the average output of node t, $MAX\_U_t$ and $MIN\_U_t$ are, respectively the input data of $MAX\_Y_t$ and $MIN\_Y_t$. $x_{t1}$ and $x_{t2}$ are adjustable parameters. Obviously, when $x_{t1}$ and $x_{t2}$ are set to $x_{t1} \to \infty$ and $x_{t2} \to \infty$, Formulate (3) equate to Formulate (2). So Formulate (2) is a special case of Formulate (3).

Obviously, the main advantage of our proposed ICART algorithm is that it adopts density function to construct the local model for every crunodes and overcomes the discontinuity at the decision boundaries, which is unnatural and brings undesired effects to the overall regression and generalization.

*2.3 Designing neuro-fuzzy System Based on ICART Algorithm*

The decision tree in Fig.1 is equivalent to a set of crisp rules:

$$\begin{cases} If & x_1 < a_1 & And & x_2 < a_2 & Then & y = b_1 \\ If & x_1 < a_1 & And & x_2 \geq a_2 & Then & y = b_2 \\ If & x_1 \geq a_1 & And & x_2 < a_3 & Then & y = b_3 \\ If & x_1 \geq a_1 & And & x_2 \geq a_3 & And & x_3 < a_4 & Then & y = b_4 \\ If & x_1 \geq a_1 & And & x_2 \geq a_3 & And & x_3 \geq a_4 & Then & y = b_5 \end{cases} \qquad (4)$$

The CART procedure initially considers the data as belonging to a single group. This group is partitioned into two relatively homogeneous subgroups. More specifically, given any input vector (x; y), only one rule out of five will be fired at full strength while the other four rules are not activated at all and the output only is determined by the fired rule. Moreover, this crisp sets reduce the computation burden in constructing the tree using ICART and it also gives undesired discontinuous boundaries. Fuzzy inference, however, is the most basic human being's reasoning mechanism. The fuzzy set can smooth out the discontinuity at each split, so we use fuzzy sets to represent the premise parts of the rule set. The statement $x \geq a$ can be represented as a fuzzy set characterized by the sigmoid membership function:

$$m_{x \geq a} = sig(x; b, a) = \frac{1}{1 + \exp[-b(x - a)]} \qquad (5)$$

Obviously, when the premise parts of the rules set in decision tree are represented by fuzzy sets, the decision tree is equivalent to a fuzzy system. On basis of this fact, we use ICART algorithm to design neuro-fuzzy system. The proposed neuro-fuzzy system based on ICART algorithm is showed as Fig.2.

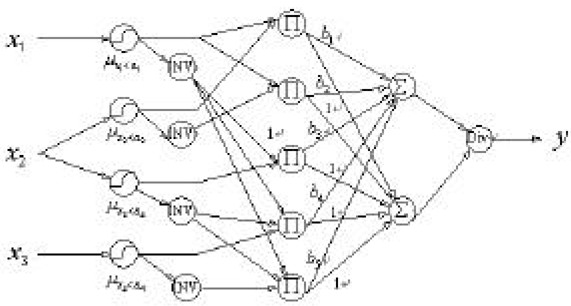



Fig. 2 The structure of neuro-fuzzy system based on ICART algorithm

The neuro-fuzzy system based on ICART algorithm consists of five layers. The first layer is input layer. Each node in this layer is called an input linguistic node and corresponds to one input variable. The node only transmits input values to the next layer directly.
Nodes in second layer are called input term nodes, each of which correspond to one linguistic label of an input variable. Each node in this layer calculates the membership value specifying the degree to which an input value belongs to a fuzzy set. INV nodes represent negation operator. A sigmoid membership function is used in this layer, which is described as:

$$m_{x \geq a} = sig(x; b, a) = \frac{1}{1 + \exp[-b(x - a)]}$$

$$m_{x < b} = 1 - sig(x; h, b) = 1 - \frac{1}{1 + \exp[-h(x - b)]} \qquad (6)$$

where $b$, $a$, $b$ and $h$ are the adjusted parameters in membership function.
The third layer consists of N neurons, which compute the fired strength of a rule. Multiplicative inference is used, so the output of this layer is:

$$y_j^{(3)} = \prod m_{x < a} \qquad (7)$$

There are two neurons in fourth layer. One of them connects with all neurons of the third layer through the weight $h_j$ representing the consequence of the $j$th rule and another one connects with all neurons of the third layer through unity weights.
The last layer has a single neuron to compute $y$. It is connected with two neurons of the fourth layer through unity weights. The integral function and activation function of the node can be expressed as:

$$y = \frac{\sum_{j=1}^{5} b_j y_j^{(3)}}{\sum_{j=1}^{5} y_j^{(3)}} \tag{8}$$

*2.3 Parameters optimization*

In parameters optimization learning phase input and output data are presented to adjust parameters and obtain better fuzzy model. Its goal is to minimize the error function:

$$E = \frac{1}{2}\left[y(t) - d(t)\right]^2 \tag{9}$$

where $y(t)$ is the current output and $d(t)$ is the desired output. For a training data pair, starting at the input nodes, a forward pass is used to compute the activity levels of all the nodes, a backward pass is used to compute $\partial E/\partial y$ for all parameters. The adjustable parameters can be adjusted as follows.

$$x_{ti}(k+1) = x_{ti}(k) - a\frac{\partial E}{\partial x_{ti}} + b\left(x_{ti}(k) - x_{ti}(k-1)\right), \quad i = 1,2 \tag{10}$$

$$\frac{\partial E}{\partial x_{t1}} = (y-d)b_t\left\|MAX\_U_t - x\right\|^2 (MAX\_Y_t - AVE\_Y_t)\exp\left(-x_{t1}\left\|MAX\_U_t - x\right\|^2\right) \tag{11}$$

$$\frac{\partial E}{\partial x_{t2}} = (y-d)b_t\left\|MIN\_U_t - x\right\|^2 (MIN\_Y_t - AVE\_Y_t)\exp\left(-x_{t2}\left\|MIN\_U_t - x\right\|^2\right) \tag{12}$$

And the membership parameters can also be adjusted as above supervised algorithm.

## 3. PRACTICAL APPLICATION FOR MODELING JET FUEL ENDPOINT OF HYDROCRACKING PROCESSING

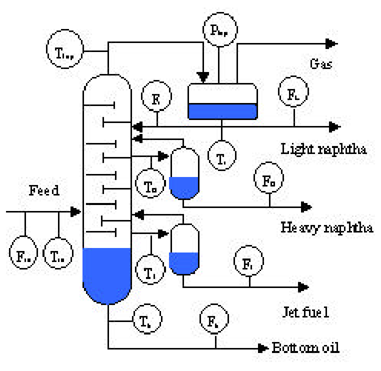



Fig.3 The schematic representation of the hydro-cracking fractionator

Hydrocracking is one of the most important processes in the petroleum industry. It upgrades heavy oil value by making high quality products, such as gasoline or kerosene. The purpose of the main fractionator of a hydrocracking process is to split a feed that produced from the former process into three product streams of different molecular weight, that is, light naphtha, heavy naphtha and jet fuel. Their endpoints are the key indicators to value the product quality. Figure 3 shows a schematic representation of the main hydrocracking fractionator.
According to the analysis of technological mechanisms, the endpoints of the three sides (i.e., light naphtha, heavy naphtha and jet fuel) are related with the above mentioned 13 variables, which can be measured and recorded on-line. In this section, the jet fuel endpoint will be studied. The relationship between it and above-mentioned 13 variables is described as equation (13):

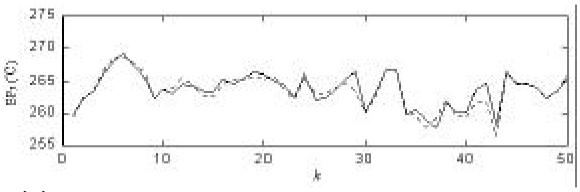
$$EP_J(k) = f(T_H(k), F_H(k), T_J(k), F_J(k), T_r(k), F_r(k), T_{top}(k), P_{top}(k), T_{in}(k), F_{in}(k), T_b(k), F_b(k), F_L(k)) + x(k) \tag{13}$$

where $EP_J(k)$ represent the endpoint (℃) of jet fuel, $f(\cdot)$ is the complex multivariable non-linear function, $x(k)$ represents the uncertain term. The task is to find the relationship between the endpoint of jet fuel and the selected 13 secondary variables, so we can estimate the product quality of jet fuel on-line.
From equation (13) and the views of technological mechanisms, though all the 13 variables have cause-and-effect relationships with the quality variable, selecting all thirteen variables as the input of self-organizing neuro-fuzzy system is totally unnecessary because the above-mentioned variables (i.e. $T_r$, $T_H$, $T_J$, $F_L$, $F_H$, $F_J$) are highly correlated each other. By statistical regression analyzing step by step, jet fuel endpoint is mainly affected by the following six measurable variables: $T_r$, $F_{in}$, $F_r$, $F_L$, $F_H$, $F_J$. So its model structure is represented as:

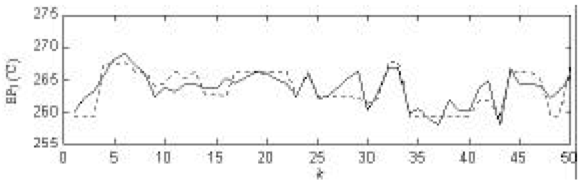
$$EP_J = g\left(Tr,\ Fin,\ Fr,\ FL,\ FH,\ FJ\right)$$

Then the proposed algorithm is used to establish the system. There are 223 sets of sample data of thirteen operating variables in different operating states. 173 pairs of them are used as off-line training data sets and another 50 pairs are used as on-line testing data sets, which verify the fuzzy inference power of the neuro-fuzzy system designed based on ICART algorithm. In the learning phase, all training data are scaled in the intervals [-1, +1].
After the learning, 173 sets training data are clustered into 40 categories, that is the number of IF-THEN rules of neuro-fuzzy system is 40 which is less than conventional grid partitioned neuro-fuzzy system's. In order to verify the generalization of the presented fuzzy model, another 50 sets are used to test it. The estimated values are shown in Fig.4 (a). In addition, the neuro-fuzzy system based on CART algorithm is used to build a soft sensing model shown as Fig4 (b). Table4 are about the comparison between CART algorithm and ICART algorithm. The results show that the proposed neuro-fuzzy system designed by ICART algorithm possesses better generalization ability and is smoother than CART algorithm.

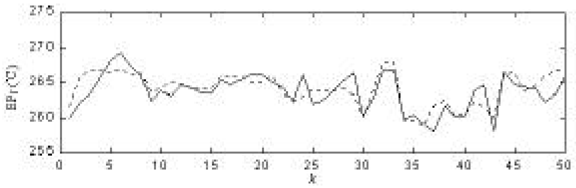
In order to verify the validity of the proposed neuro-fuzzy system designed by ICART algorithm, the method proposed in paper Jia (2001), which uses clustering algorithm to construct neuro-fuzzy system, is applied to build a soft sensing model shown as Fig4 (c). Comparisons between these two models are represented in Table.1. From Table.1 we learn that the method proposed in this paper possesses simple structure and better generalization ability than the method presented in paper Jia (2001).




(a) ICART algorithm ( — Practical Output, ----Model Predictive Output)




(b) CART algorithm ( — Practical Output, ----Model Predictive Output)




(c) The algorithm proposed in paper Jia (2001) (—Practical Output, ----Model Predictive Output)

Fig.4 The comparison between CART algorithm, ICART algorithm and Method in paper Jia (2001)

Tab.1 the comparison between CART algorithm, ICART algorithm and Method in paper Jia (2001)

| Algorithm | Rule | RMSE | MAX |
|---|---|---|---|
| ICART | 40 | 0.8786 | 2.9812 |
| CART | 40 | 1.7772 | 4.1718 |
| Method in paper Jia (2001) | 45 | 1.6784 | 3.6864 |

In summary, the proposed neuro-fuzzy system designed by ICART algorithm possesses simple structure, better generalization ability and is smoother than CART algorithm. And it can be successfully applied to quality prediction for hydrocraking processing.

## 4. CONCLUSION

A neuro-fuzzy system based on ICART algorithm, which using ICART algorithm to design neuro-fuzzy system is proposed in this paper. It is worth noting that ICART algorithm classifies the input space into tree structure adaptively, which avoids the curse of dimensionality because the total number of fuzzy rules doesn't increase exponentially with the number of input variables. Moreover it adopts density function to construct the local model for every crunodes in order to overcome the discontinuous boundary existed in CART algorithm. The major advantage offered by this approach is that the user can now quickly determine the roughly correct structure of a fuzzy inference through ICART, and then refine the membership functions and output functions via efficient neuro-fuzzy system architecture. It can be seen that ICART and neuro-fuzzy system are complementary and their combination makes a solid approach to fuzzy modeling. In addition, a supervised scheme is used to adjust parameters to minimize the network output error and constructer optimal fuzzy model on the basis of ICART algorithm. Finally, to illustrate the validity of the proposed method, a practical application are done. The results show that the proposed method can provide optimal model structure and parameters for fuzzy modeling, possesses high learning efficiency and is smoother than CART algorithm. And it can be successfully applied to quality prediction for hydrocraking processing.

## REFERENCE

Breiman L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984). *Classification and regression trees*. Wadsworth, Inc., Belmont, Californiz

Duan J.C., Chung F.L. (2001). Cascaded fuzzy neural network model based on syllogistic fuzzy reasoning. *IEEE Trans. FS*. 9(2), 293-306

Ina S.M., Richard G.M. , Barry A.W.,(1998). A rule induction approach for determining the number of kanbans in a just-in-time production system. *Computers ind. Engn*g. 34(4), 717-727

Jang, J.S.R., Sun C.T., Mizutan E, (1997). *Neuro-fuzzy and soft computing: A computional approach to learning and mechine intelligent*. Prentice Hall

Jia, Li, Yu, Jinshou, (2001). A novel neural fuzzy network and application to modeling octane number. *Hydrocarbon Processing*. 12, 56-60

Quinlan J.R., (1986). *Induction of decision trees.* Mach. Learn., 1, 81-106

Mauricio, Figueiredo, Fernando, Gomide, (1999). Design of fuzzy systems using neurofuzzy networks. *IEEE Trans. Neural Networks.* 10(4), 815-824

Meng Joo Er, (2000). Fuzzy neural networks-based quality prediction system for sintering process, *IEEE Trans. F.S.*, 8(3), 314-323

Serge G, (2001). Designing fuzzy inference systems from data: an interpretability-oriented review. *IEEE Trans. FS*. 9(3), 426-443

Wang, L.X., (1994). *Adaptive fuzzy systems and control*, Prentice-Hall, Englewood Cliffs, NJ

Yin, Wang, Gang, Rong, (1999), A self-organizing neural-network-based fuzzy system, *Fuzzy sets and systems*, 103, 1-11