Multi-model decomposition of nonlinear dynamics using a fuzzy-CART approach ☆

Jinendra K. Gugaliya [a], Ravindra D. Gudi [a,*], S. Lakshminarayanan [b]

[a] *Department of Chemical Engineering, Indian Institute of Technology Bombay, Powai, Mumbai 400 076, India*
[b] *Department of Chemical and Environmental Engineering, National University of Singapore, Singapore 117576, Singapore*

Abstract

In this work, we propose an extension of the CART (Classification and Regression Tree) based methodology proposed earlier [Ind. Eng. Chem. Res. 31(8) (1992) 1989; Comp. Chem. Eng. 16(4) (1992) 413], for modelling and identification of complex nonlinear systems. The suggested scheme employs the 'divide and rule' based strategy which decomposes the overall complex nonlinear dynamics into a set of linear or simple nonlinear models. The CART analysis picks up only the most representative model at any time. This model strategy involves discontinuous boundaries in the overall model structure. Therefore this structure is further refined here using a fuzzification procedure. The traditional backpropagation algorithm is used to incorporate the fuzzification. The fuzzification imposed over the CART skeleton replaces the crisp boundaries of the CART models by smooth boundaries thus enabling better prediction during transitions. This approach can deal with both steady state and dynamic data. The models built using the proposed fuzzy-CART methodology has been shown to give significant improvement in performance over that built using the CART alone. Validation results involving simulations of a nonlinear fermenter of Henson and Seborg [Chem. Eng. Sci. 47 (1992) 821] have demonstrated the practicality of the approach.

Keywords: Nonlinear systems; CART; Fuzzy-CART; Fzzification; Backpropagation; Multiple-models

1. Introduction

Mathematical models form the basis for the solution of most system analysis and design problems. Efficient identification of the structure, order and parameters of a model is very important for simulation, forecasting and controller design. Dynamic models can be developed by two alternative routes or a combination of them. The first route involves the use of conservation principles relevant to the system. An alternate route is based on experimentation where in the system is excited by a suitable input signal and the resulting input/output data are used to construct an empirical model (the system identification approach).

Chemical processes are usually nonlinear and have strong interaction among the process variables. The nonlinearity and the extent of interaction could vary with the range of independent variables. For chemical processes where model-based controllers are feasible, perturbed plant data involving cause and effect variables, have been used to generate the process models. Two approaches may be employed to tackle the problems arising out of the nonlinear nature of the cause

and effect relationships. The first is the development of an overall nonlinear model that performs sufficiently well over the entire operating range. The other option would be to devise a regime-wise modelling strategy employing simpler local models. As is well known, the former approach requires the resolution of issues related to choice of a suitable model structure and input design. Such models are also invariably complex. In the latter approach, namely a multiple-model-based strategy, issues related to: (i) division of the operating range into local regions, (ii) construction of local models and (iii) switching between models, need to be addressed. Takagi and Sugeno [1] proposed a multi-model-based nonlinear systems representation, that had good interpolation and extrapolation properties. Foss et al. [2] have proposed a strategy for decomposition of the nonlinear space into multiple linear structures using a prior process information. Johansen and Foss [3] proposed a strategy to identify local linear models and subsequent interpolation of these local linear models using smooth interpolation functions. Banerjee et al. [4] proposed the decomposition of nonlinear dynamics based on a bayesian representation of the local models. Azimzadeh et al. [5] present an approach based on the use of local linear models for on-line optimal trajectory control in a fermentation process. An overview of multiple model approaches for modelling and control of complex dynamic systems is presented in the Smith and Johansen [6]. Johansen and Foss [7] have also formulated an operating regime decomposition based on local models that employs a search strategy guided by heuristics and local model validity function. Kuipers and Astrom [8] have employed a multiple model approach that assumes homogeneity of all local models. An alternate methodology involving the deployment of disparate, regime dependent, model structures has been proposed by Box and Jenkins [9] and Ljung [10]. Kordon et al. [11] proposed a parallel control structure for process plants with multiple operating regimes. Schott and Bequette [12] proposed a weighing function-based multiple model adaptive control strategy to model and accommodate nonlinearities. In this context, structure identification for complex nonlinear system is a relatively difficult task. Venkat and Gudi [13] proposed a fuzzy segregation based nonlinear strategy to decompose the overall nonlinear behaviour into local linear models. Induction and classification methods, that have been traditionally used in the AI field, have also been proposed variously to address the task of multimodel building [14,15].

The above approaches towards multi-model identification for control differ in terms of the extent to which the first principles or prior knowledge is used. Often times, depending on the application at hand, not much prior knowledge may be available. Also, the approaches above differ in the mechanisms of composition of the local models for the purposes of generating a prediction or for control. Finally, depending on the extent of complexity and interaction (for multivariable systems), there could be several challenges that could be posed at the decomposition (into local models) step. It may turn out that the resulting local linear models could have substantial overlaps in their regions of validity. This could imply that the classification space in which the segregation of dynamic behaviour is sought, is perhaps inappropriate. Alternately, the classification/decomposition algorithm itself may not be applicable. For example, the clustering method for segregation of nonlinear dynamics [13] could yield clusters that are close to each other for any set of classification spaces. In such a case, the clustering methodology is perhaps not appropriate and one may have to look at prior knowledge (if available) or other alternate techniques to segregate the dynamics and build local models. These are some of the crucial aspects that needs to be addressed in a multi-model decomposition framework.

This paper presents such an alternate approach towards decomposition of nonlinear dynamics, that is based on the induction methods cited above. Assuming minimal prior knowledge, we propose an extension of the CART (Classification and Regression Tree) algorithm, proposed earlier by Joseph et al. [15], which is an induction-based method, for decomposition of variable relationships. The extension to the basic CART algorithm that we propose are related to two of the shortcomings of the existing CART methodology, viz. decomposition metric and the crisp nature of the segregated models. The existing CART methodology relies on a decomposition metric called the information gain to select a variable over which the decomposition is done. Here we show that this could lead to a large tree structure implying an unnecessarily large set of local linear models. We propose a regression error-based metric for selection of the variable over which the decomposition needs to be done and show that this yields a relatively more compact set of local linear models. Further, the local relationships identified by the traditional CART methodology have crisp boundaries and so the predictions between transitions is hampered. This could result in large prediction errors during intermediate or transition regions. Therefore, we propose the fuzzification of the boundaries of the local models so as to enable accurate prediction during transition. By considering a representative nonlinear, multivariable fermentor application of Henson and Seborg [16], we demonstrate the practicality of the proposed fuzzy CART algorithm in modelling nonlinear dynamic relationships.

The paper is organised as follows. In Section 2, the nature of the problem considered is discussed and in Sections 3 and 4, an overview of the CART and the fuzzy-CART is provided. Section 5 contains the case

studies followed by results and discussion. Conclusions and scope for future work are outlined in Section 6.

2. Nature of the problem considered

Consider the simple relationship between the cause variable x and the effect variable y as shown in Fig. 1. Such a behaviour is observed in typical chemical process systems, for example the steady-state map between dilution rate and biomass concentration of the nonlinear fermentation problem [16] displays the above behaviour. It is evident that $x = a$ is the boundary between the regions of positive and negative slopes. There are two kinds of relationships in Fig. 1, the first has $x = a$ as the crisp boundary between the regions of changing slopes and the second has the boundary to be rather fuzzy and belongs in an interval $(a - \delta, a + \delta)$. In typical chemical modelling applications, the scenario represented by the latter is more likely because nonlinearities change gradually with the changing operating region. Given the data in y and x, the issues to be resolved in multi-model building are the identification of regions and models where the cause–effect relationships are similar and to evolve strategies to switch gradually between the models so as to maximise the accuracy of the prediction obtained from a single or composite models. In an earlier work, Joseph et al. [15] have proposed the CART methodology wherein the identification of local regions is based on a metric called the information gain. In their work, they have converted the numerical variable based representation into categorical representation. The resulting variant of the CART methodology which they termed as IPRT (Inductive Partitioning and Regression Tree) was shown to handle both, numerical as well as categorical variables. Thus the IPRT algorithm proposed by them overcame the drawback of the conventional CART algorithm which is applicable to categorical variables only. They have demonstrated this methodology on a representative regression problem as well as for classifying data from an autoclave reactor. The methodology proposed by them could be extended for the decomposition of complex nonlinear dynamics so as to yield simpler models. To facilitate this extension, the following issues need to be resolved:

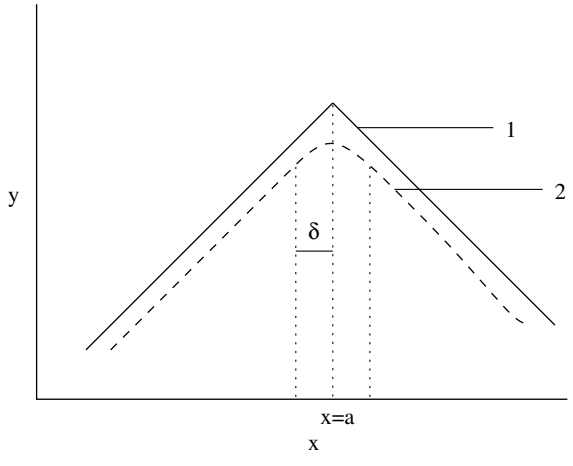

Fig. 1. Cause–effect relationship.

Input design for plant perturbation. Issues related to appropriate input signal design for nonlinear system identification has been as active area of work. Often times, the input is designed to be commensurate with the chosen model form, for example if the model form chosen is a Volterra Kernel, the input is appropriately designed for accurate identification of each order in the Kernel. However for complex systems that exhibit gain sign changes and varying dynamics, the steady-state map could indicate regions of changing gain. Decomposition based on this steady-state map could isolate regions having approximately the same gain to enable better input design. Here, we propose the use of the steady-state decomposition for establishing local relationships. We show that an extension of the CART methodology proposed by Jang et al. [17] and subsequent fuzzification proposed by Suarez and Lutsko [18] could yield better results especially when there are gain sign changes and during transition regions.

Modified choice of decomposition metric. When regions of plant operation are segregated, the granularity or the fineness of separation is an important issue. For example, the behaviour depicted by the curve 2 of Fig. 1 could be segregated in terms of two or three regions depending on the nature of transitions. Thus a metric to decide on how many segregations need to be carried out is necessary. In the earlier work of Joseph et al. [15], where they proposed the IPRT algorithm as an extension to the CART, the decomposition metric used was the information gain that was calculated from the data. Here, we show that this metric could result in a large tree structure which could be difficult to maintain. Alternatively, if the prediction error is used as a decomposition metric, this could yield a relatively simpler and manageable tree structure. We further demonstrate that fuzzification of the tree yields better predictions.

In the following section, we present the modified methodology that makes the resulting algorithm particularly attractive for decomposing nonlinear dynamics.

3. Extension of the CART methodology

Consider the regression problem of relating the dependent variable y and the independent variables $X = [x_1, x_2, \ldots, x_n]$. The CART methodology requires that the variable over which the data are to be split be identified along with its threshold. Unlike the earlier method based on information gain [15] (not included

here for brevity), an alternate method to arrive at this variable split can be proposed as follows:

The first step involves calculation of the mean squared error associated with the overall data. For this, a linear or simple nonlinear model, whose order is fixed by the principle of parsimony, is selected and the error associated with this root node regression is denoted by E_{root}. Obviously, if E_{root} is sufficiently small as per some tolerance criteria, the model building step is terminated here, i.e., only a single model is adequate. If the root node error is large, the next step then involves identification of the variable along with its threshold value, for splitting the data. Consider that the data has to be segregated based on the threshold value of variable x_1 having N data points (arranged in increasing order) as $[p_1, p_2, p_3, \ldots, p_N]$. Then, the set L of candidate split values of x_1 are generated as $[\frac{p_1+p_2}{2}, \frac{p_2+p_3}{2}, \ldots, \frac{p_{N-1}+p_N}{2}] = [l_1, l_2, \ldots, l_{N-1}]$, where $l_i = \frac{p_i+p_{i+1}}{2}$. For each l_i, the data are split into regions where $x_1 < l_i$ and $x_1 \geqslant l_i$ and the mean squared error associated with each of the segregated data sets is calculated as

$$E(t) = \min \sum_{i=1}^{N(t)} (y_i - d_t(X_i, \theta))^2, \tag{1}$$

where $\{X_i, y_i\}$ is a typical data point, $N(t)$ is the number of data points in the data set t, $d_t(X,\theta)$ is a local model (with modifiable parameter θ) for data set t. In the CART methodology proposed by Jang et al. [17], the local model d_t, can be a constant model or a linear model. In the present work, we have proposed the use of local, linear models. $E(t)$ is the mean-squared error of fitting the local model d_t to the data set. For any split of data set t into data sets t_l and t_r, the change in the error measure is expressed as

$$\Delta E(s_1, t) = E(t) - E(t_l) - E(t_r), \tag{2}$$

where s_1 is the threshold value of variable x_1 over which the data are split. The best candidate split s_1^* for variable x_1 is the one that maximises the decrease in the error measure:

$$\Delta E(s_1^*, t) = \max(\Delta E(s_1, t)). \tag{3}$$

This procedure is repeated for rest of the independent variables $x_2, x_3, \ldots, x_n$ in the data set and the best split values s_i^* associated with each of the independent variables are calculated. Of the best split values for each variables $[s_1^*, s_2^*, \ldots, s_n^*]$, that variable which results in maximum reduction of error measure as given by Eq. (3), is chosen for splitting the data set.

In the tree development step, the root node of the tree contains all the data points. The variable and its threshold value is calculated to split the data in the root node as explained earlier. Assuming that the split variable is x_1 with threshold value a, then the data in the root node gets split into two subsets, one subset with $x_1 < a$ and the other subset with $x_1 \geqslant a$. In the next step, each of the subset is taken as a root node and the data are partitioned recursively using the same procedure. The decision tree so formed is a tree structure consisting of internal and external nodes connected by branches. An *internal node* is a decision making unit that evaluates a test to determine which child node to visit next. In contrast, *an external node*, also known as *terminal node*, has no child node and is associated with a value or an equation. Generally, as could be expected, the MSE decreases with an increase in the tree size. However representing a large tree is cumbersome while discriminating the data at each node. It is therefore necessary to determine a criterion that evaluates an optimal tree size. In the present work, an AIC (Akaike Information Criterion) like criterion is employed – the tree size above which there is no significant improvement in the tree performance (reduction in error) is taken as the optimal tree size.

For prediction using the CART model, a new data point $X^* = [x_1^*, x_1^*, \ldots, x_n^*]$ is subjected to an enumeration/check at each node to identify the region to which it belongs. The appropriate local model is then used to obtain a prediction for the data point X^*.

As is evident in the existing CART methodology, the representative data point gets classified to only one node and a single model at the node is used for the prediction. The models at each terminal node of the tree depict local behavioural relationships, i.e., the relationships in the region depicted by the path traversed to the terminal node.

While such a regime-wise modelling strategy segregates complex nonlinear relationships into simpler (linear/nonlinear) relationships, there is a trade off between the tree size (depth) and the accuracy of predictions. A large tree size involving a number of models may yield better predictions but may be difficult to maintain. It may therefore be interesting to explore the possibility of model composition so as to obtain the same prediction accuracy through the use of fewer local models. Here, we explore the possibility of fuzzy composition of individual models with a view of enhancing the quality of predictions during transitions, with the use of fewer model structures. The fuzzification aspects of the CART tree are discussed next.

4. Fuzzy-CART Analysis

It is evident from the earlier section that the CART modelling follows a crisp segregation as shown in Fig. 2. Assuming that a split occurs for a variable x at a value a, the relationship can be depicted by

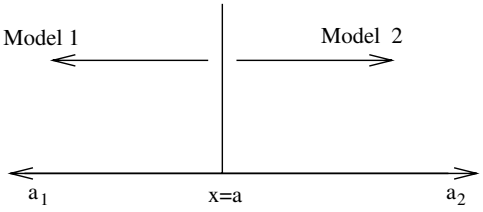

Fig. 2. Region of validity for crisp decomposition.

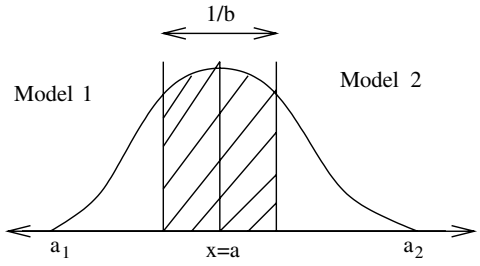

Fig. 3. Region of validity for fuzzy decomposition.

If $x < a$, then model 1 is valid,

If $x > a$, then model 2 is valid.

Fig. 3 shows an alternate, fuzzified view of the boundary. In the region $a_1 < x < a$, the membership value/applicability of model 2 increases from 0 (at $x = a_1$) to a value of 1 (at $x = a$). Similarly in region 2, the membership value/applicability of model 1 decreases with an increase in x from the value of a to a_2. Thus it is possible to construct a fuzzified boundary at $x = a$ whose width is characterised by a single parameter b (refer Fig. 3). Parameter b can be evaluated so as to minimise regression error objective as follows.

The CART method based on the prediction error metric proposed in Section 3, fixes the architecture of the unknown function to be modelled, using the data rather than the heuristics. But, the CART tree is developed through the minimization of the local regression error rather than the minimization of the overall error. The global error function can be written in terms of crisp split functions of the training examples in the terminal nodes. But the crisp split function themselves are discontinuous in nature. So the parameters of the tree cannot be tuned by the analytical optimization routines which require continuous functions and calculation of gradients. The natural way to improve the performance is to replace the rigid decision functions by flexible decision functions. The crisp split associated with inner node t_i, which is given by equation $x < a_i$, can be replaced by a sigmoidal membership function of inverse width b_i,

$$\mu_{\mathrm{L}}^{i}(x) = \frac{1}{1 + \exp[b_i(x - a_i)]}, \tag{4}$$

$$\mu_{\mathrm{R}}^{i}(x) = \frac{1}{1 + \exp[-b_i(x - a_i)]} = 1 - \mu_{\mathrm{L}}^{i}(x). \tag{5}$$

The splitting threshold is thus broadened into a splitting band. Outside the band, the data points in the data are assigned to one of the child nodes with a degree of membership close to unity (i.e., they behave almost as if the split was crisp). Data points that fall within the band are assigned with significant degrees of membership to both child nodes. A decrease in the value of the parameter b_i increases the width of the fuzzy region/band. The rules given by Eqs. (4) and (5) approaches the crisp rules as $b_i \to \infty$. At the other extreme is a split where the value $b_i = 0$. Such a split assigns all examples in the parent node to both child nodes with equal degrees of memberships. The parent node is thus replicated into equivalent nodes and the global effect is almost to have no split at all. In the case of the CART tree, given a data point, the crisp tests are evaluated at each internal node and only one of the terminal node of the tree gets fired at full strength. The terminal node which gets activated gives the output value for the data point. But in the case of a fuzzy-CART tree, all the rules at the internal nodes of the tree get activated to a smaller or greater extent, and the data point gets membership from all the terminal nodes. One important point to note here is that the membership values of all the terminal nodes of the tree sum up to unity. So there is no need to normalise the membership values of the terminal nodes while applying the fuzzification procedure. The output of the complete decision tree is given as,

$$\bar{y}(x) = \sum_{1}^{l} \mu_l(x) d_l, \tag{6}$$

where $\mu_l(x)$ is the absolute membership of the point to terminal node t_l, which incorporates the model d_l, and l denotes the number of terminal nodes in the tree. The absolute membership of a point to a node i is given by,

$$\mu_{i\alpha}(x) = \mu_i(x)\mu_{\alpha}^{i}(x), \quad \alpha = \mathrm{L}, \mathrm{R}, \tag{7}$$

where α is a suffix to indicate the child node, L and R represent left and right node, respectively, $\mu_i(x)$ is the absolute degree of membership for the parent node t_i, which can be calculated by recursion of Eq. (7) until the root node is reached. All points belong to root node and, therefore

$$\mu_0(x) = 1.0 \quad \forall x. \tag{8}$$

The replacement of crisp splits by fuzzy splits improves the performance of the tree but it is necessary to determine the parameters (center and width) of the fuzzy

splits. If the fuzzy splits are centered exactly like the crisp splits, there is no significant improvement in the performance. In fact this may result in an inferior performance over the CART tree. The reason for this lies in the fact that the crisp splits are obtained using the criterion of the minimization of the local error. The fuzzification will prove effective only when the global error criterion is used. So it is necessary to develop an algorithm to calculate the parameters of the fuzzy-CART tree which takes into account the global error. The proposed algorithm is called backpropagation algorithm [18] which is explained in detail in Section 4.1.

4.1. Backpropagation algorithm

The objective of this section is to design an algorithm to estimate the parameters of the fuzzy regression tree. It is possible to have a constant value or a linear equation at each of the terminal nodes. The algorithm is first developed for the constant value terminal nodes and then it is extended for the linear equation terminal nodes. The data set is divided into two parts namely the training data and the test data. The tree parameters are first calculated using the training data and then the tree performance is validated on the test data. The criterion used for designing the algorithm is the minimization of the mean square error, as given by,

$$E = \frac{1}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} (y_n - \bar{y}(x_n))^2, \tag{9}$$

where y_n is actual output and $\bar{y}(x_n)$ is the predicted output, N_{train} is the number of data points in the training data. For the fuzzy regression tree, the problem can be solved by a global optimization algorithm in which the estimates at the terminal nodes are propagated upward in the tree to the root node.

Let $\bar{y}_i(x_n)$ denotes the partial estimate of y at any node t_i for the point x_n. For a terminal node t_l, this quantity is same as the terminal node model value d_l. This is computed from the training data as,

$$\bar{y}_i = d_l = \frac{1}{N_i} \sum_{n=1}^{N_{\text{train}}} \mu_i(x_n) y_n, \tag{10}$$

where N_i is number of data points belonging to node i. For the fuzzy-CART tree, no a *prior* assumptions about d_l are made. As a result of global optimization procedure, d_l turns out to be a kind of average over training data assigned to the leaf. For an internal node, $\bar{y}_i(x_n)$ is defined as prediction of the subtree $T(t_i)$ (i.e., the subtree of T composed as t_i as the root node, and the descendent nodes of t_i). It can be computed from the recursive relation,

$$\bar{y}_i(x_n) = \mu_{\text{L}}^{(i)}(x_n)\bar{y}_{i\text{L}}(x_n) + \mu_{\text{R}}^{(i)}(x_n)\bar{y}_{i\text{R}}(x_n) \tag{11}$$

in terms of the partial estimates of its children nodes $\bar{y}_{i\text{L}}(x_n), \bar{y}_{i\text{R}}(x_n)$. The relative degrees of memberships are given by Eqs. (4) and (5).

The basis of the backpropagation algorithm is the observation that the value predicted by the full regression tree for the dependant variable given the set of attributes x_n can be obtained by iterating Eq. (11) from the predictions at the leaves $\bar{y}_l(x_n)$, upward, until the root node is reached as,

$$y(x_n) = y_{\text{root}}(x_n). \tag{12}$$

The optimization of the error given by Eq. (9) with respect to parameter α_j of node t_j yields the equation,

$$\frac{\delta E}{\delta \alpha_j} = -\frac{2}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} ((y_n - \bar{y}(x_n))) \frac{\delta \bar{y}(x_n)}{\delta \alpha_j} = 0. \tag{13}$$

For a leaf node t_l, $\alpha_l = d_l$, using Eq. (6)

$$\frac{\delta \bar{y}(x_n)}{\delta d_l} = \mu_l(x_n) \tag{14}$$

we have,

$$\frac{\delta E}{\delta d_l} = -\frac{2}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} (y_n - \bar{y}(x_n))\mu_l(x_n) = 0 \quad \forall \text{ Terminal nodes.} \tag{15}$$

The optimization equation for the parameters at the inner node t_i can be obtained using Eqs. (11) and (13) as,

$$\frac{\delta E}{\delta a_i} = -\frac{2}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} (y_n - \bar{y}(x_n))\mu_i(x_n)(\bar{y}_{i\text{L}}(x_n) - \bar{y}_{i\text{R}}(x_n)) \frac{\delta \mu_{\text{L}}^{(i)}(x_n)}{\delta a_i} = 0, \tag{16}$$

$$\frac{\delta E}{\delta b_i} = -\frac{2}{N_{\text{train}}} \sum_{n=1}^{N_{\text{train}}} (y_n - \bar{y}(x_n))\mu_i(x_n)(\bar{y}_{i\text{L}}(x_n) - \bar{y}_{i\text{R}}(x_n)) \frac{\delta \mu_{\text{L}}^{(i)}(x_n)}{\delta b_i} = 0, \tag{17}$$

with

$$\frac{\delta \mu_{\text{L}}^{(i)}(x_n)}{\delta a_i} = b_i \mu_{\text{L}}^{(i)}(x)\mu_{\text{R}}^{(i)}(x), \tag{18}$$

$$\frac{\delta \mu_{\text{L}}^{(i)}(x_n)}{\delta b_i} = -(x - a_i)\mu_{\text{L}}^{(i)}(x)\mu_{\text{R}}^{(i)}(x). \tag{19}$$

The solution of the system of equations (15)–(17) is the set $(a_i^*, b_i^*, \{d_l^*\})$ and its values are the parameters that characterise the globally optimal tree. For a regression tree with linear equation at terminal node l of the form,

$$\bar{y}_l = p_{l1}x_1 + p_{l2}x_2 + \cdots + p_{ln}x_n + p_{l(n+1)}, \tag{20}$$

$$\frac{\delta \bar{y}(x_n)}{\delta p_l} = \mu_l(x_n)X, \tag{21}$$

where $X = [x_1, x_2, \ldots, x_n, 1]$.

Eq. (15) gets modified as,

$$\frac{\delta E}{\delta p_l} = -2\sum_{n=1}^{N_{\text{train}}}(y_n - \bar{y}(x_n))X\mu_l(x_n) = 0$$
$$\forall \text{ Terminal nodes.} \tag{22}$$

Thus the problem of optimization is equivalent to solving the set of nonlinear equations (15) or (22) and (16) and (17) with the number of nonlinear equations to be solved, being equal to the number of parameters of the fuzzy-CART tree to be obtained. The optimization problem is solved by a quasi-Newton method (the Broyden–Fletcher–Goldfarb–Shanno optimization algorithm). Backpropagation is used in order to obtain, from the estimations at the leaves, d_l, the values of $\bar{y}_i(x_n)$ which are needed in the computation of error function (Eq. (9)) and its derivatives with respect to the parameters characterizing the fuzzy-CART tree.

4.2. Selection of initial guess

The backpropagation algorithm formulates the optimization problem in terms of solving a set of nonlinear equations. The number of nonlinear equations is equal to the number of parameters to be calculated. Given the large dimensionality of optimization problem, it is possible that the algorithm may get trapped in some local minima. Therefore, the starting guess values for the parameters need to be selected carefully. The crisp CART tree generated provides guess values for the centers of the fuzzy splits. The crucial parameter which can affect the convergence of algorithm to global optima is initial width selection for fuzzy splits. The initial guesses for widths (b_i) are selected such that splits in the crisp tree leading to a larger improvement of the quality of regression are initialised crisper than those that appear less significant.

$$\frac{1}{b_i^{\text{init}}} = 2\min[\max_{\text{Train}}(x_n - a_i), \max_{\text{Train}}(a_i - x_n)]/f^{(i)}. \tag{23}$$

The factor in the numerator scales the width of the split to the range of the splitting variable in that node.

$$f^{(i)} = \sqrt{\frac{N_i E(t_i)}{N_{i\text{L}}E(t_{i\text{L}}) + N_{i\text{R}}E(t_{i\text{R}})}} - 1, \tag{24}$$

where the error rate for a node is

$$E(t_i) = \frac{1}{N_i}\sum_{n=1}^{N_{\text{train}}}\mu_i(x_n)(y_n - \bar{y}_i)^2. \tag{25}$$

If the crisp split on node t_i leads to a perfect regression, it remains crisp and if it leads to no improvement, it is made to be maximally fuzzy [18].

5. Case studies involving nonlinear fermenter

In this section, we present the validation of the methodology presented in the earlier sections. For the validation, we consider the nonlinear, multivariable continuous fermenter example considered in Henson and Seborg [16]. The process exhibits significant nonlinearities in terms of gain sign changes and time varying process behaviour. The model equations for the fermenter are presented as follows:

$$\dot{x}_1 = -u_1 x_1 + \mu x_1, \tag{26}$$

$$\dot{x}_2 = u_1(u_2 - x_2) - \mu x_1/Y_{xs}, \tag{27}$$

$$\dot{x}_3 = -u_1 x_3 + (\alpha\mu + \beta)x_1, \tag{28}$$

where

$$\mu = \frac{\mu_m\left(1 - \frac{x_3}{p_m}\right)x_2}{k_m + x_2 + \frac{x_2^2}{k_i}}, \tag{29}$$

u_1 and u_2 are the two inputs to the system, namely dilution rate and feed substrate concentration. x_1, x_2 and x_3 are the state variables of the system representing the biomass concentration, substrate concentration and product concentration, respectively. The constants and the steady-state values of the system are the same as given in Henson and Seborg [16]. The measurable outputs of the system are x_2 and x_3.

We consider the problem of building dynamic models relating the two input variables with an output variable in a MISO framework. As mentioned before, the design of the perturbation signal becomes quite difficult in the presence of gain sign changes. Hence we propose to first segregate the process relationships on considerations of the steady state gain. Appropriate signal design is done within each segregated region for generating the data required for dynamic local model identification. For both the steady state and dynamic modelling exercises presented below, white Gaussian noise having a variance of 10% was added at the outputs. The same segregated regions, with dynamic model identified in these regions, are used for the composite prediction.

5.1. SISO-steady state data

This case study considers SISO-steady state data of the fermenter process. The steady-state map with feed substrate concentration as the input, and biomass concentration as the output from the fermenter is shown in Fig. 4. The second input viz. the feed dilution rate is kept constant at 0.1636 h^{-1}. The input values ranges between 4 and 40 and the output value shows a peak at 23.40. So at peak value of 23.40, there is a change in the process gain. The data are first subjected to IPRT (Inductive Partitioning Regression Tree) algorithm

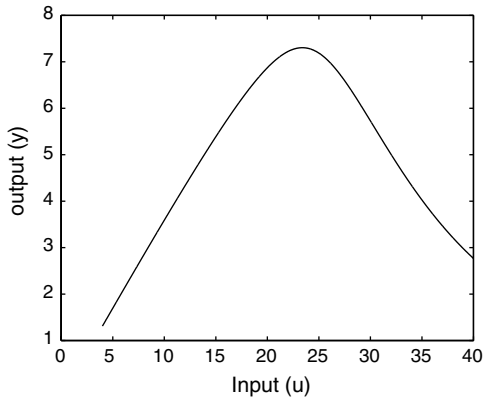

Fig. 4. Fermenter model data.

proposed by Joseph et al. [15] , for identifying linear models in the data. The model to be fitted is chosen of the form,

$$y = pu + c, \tag{30}$$

where y is the biomass concentration and u is the feed substrate concentration.

The linear regression applied on the original data shows that the model is not well fitted. So step two of the IPRT algorithm is carried on data to calculate maximum information gain associated with u. The plot of information gain versus value of u is shown in Fig. 5. From Fig. 5, it can be seen that the information gain shows peaks at $u = 13.7$ and $u = 31.5$. However, it is evident from the Fig. 4, that the peak point at $u = 23.40$ may perhaps be a more suitable point for the classification. When the algorithm was continued further to establish the individual nodes and the tree structure, by considering the information gain peak to be at 13.7 as suggested in Joseph et al. [15], it resulted in a structure with eight terminal nodes (i.e., 8 models were found to be necessary to describe the behaviour). A visual examination of the steady-state behaviour in Fig. 4, also suggests that these many models may perhaps not be necessary. We now consider the modelling of this

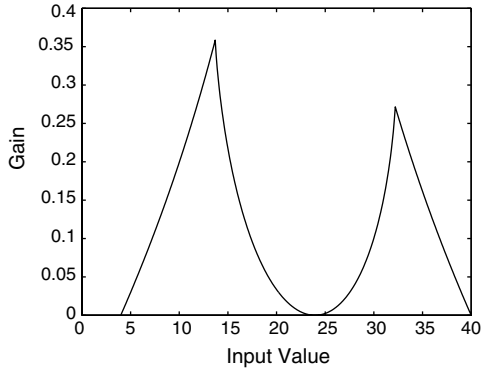

Fig. 5. Information fain vs input.

steady-state relationship using the CART algorithm proposed by Jang et al. [17] in which the partitioning is based on the regression error rather than the information gain. The criterion for goodness of fit is taken to be 'MSE'. The decision tree for this problem can be obtained by using either constant value predictors or linear equation predictors at the terminal nodes. For the constant node tree, the model chosen at the terminal node is of the form,

$$y = d, \tag{31}$$

where d is a constant while for linear predictor tree the model chosen at the terminal node is of the form given by Eq. (30). A data set consisting of 3600 data points was divided into two parts namely training set and test set. The training set consisted of 2500 data points and the test set consisted of 1100 data points. The CART algorithm was applied on the training set which generated a tree of size 3 for constant predictors and a tree of size 2 for linear predictors. The tree structures are shown in Figs. 6 and 7, respectively. In Figs. 6 and 7, the ellipse representing internal nodes also shows the number of the node and the variable over which the split is done. The ellipse representing the terminal node also contains the number of the node and the prediction equation (d_i). The actual data and the predicted output for the constant node tree are shown in Fig. 8. It is clear that for the constant node tree, the CART model only approximates the structure in the data in terms of three local models and therefore either fuzzification is further required to smoothen the structure or linear predictors

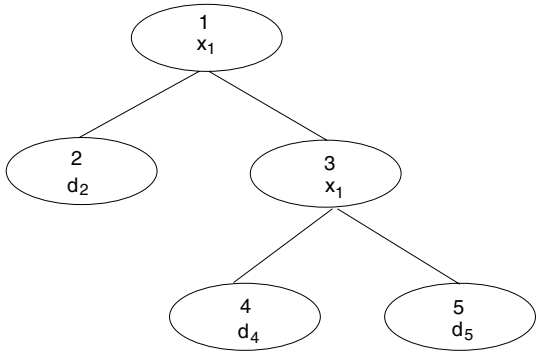

Fig. 6. Tree structure for SISO fermenter data (constant predictor).

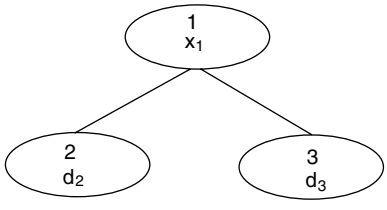

Fig. 7. Tree structure for SISO fermenter data (linear predictor).