Decision tree search methods in fuzzy modeling and classification

L.F. Mendonça a,b, S.M. Vieira a,*, J.M.C. Sousa a

a Technical University of Lisbon, Instituto Superior Técnico, Department of Mechanical Engineering, GCAR/IDMEC, Pav. Eng. Mecânica III, Av. Rovisco Pais, 1049-001 Lisbon, Portugal
b Escola Náutica Infante D. Henrique, Department of Marine Engineering, Av. Eng. Bonneville Franco, Lisbon, Portugal

Received 12 September 2005; received in revised form 10 July 2006; accepted 18 July 2006
Available online 18 August 2006

Abstract

This paper proposes input selection methods for fuzzy modeling, which are based on decision tree search approaches. The branching decision at each node of the tree is made based on the accuracy of the model available at the node. We propose two different approaches of decision tree search algorithms: bottom-up and top-down and four different measures for selecting the most appropriate set of inputs at every branching node (or decision node). Both decision tree approaches are tested using real-world application examples. These methods are applied to fuzzy modeling of two different classification problems and to fuzzy modeling of two dynamic processes. The models accuracy of the four different examples are compared in terms of several performance measures. Moreover, the advantages and drawbacks of using bottom-up or top-down approaches are discussed.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Decision trees; Fuzzy modeling; Top-down approach; Bottom-up approach; Input selection

1. Introduction

In real-world data analysis, data mining, classification and modeling problems usually involve a large number of candidate inputs or features. Another area where a large number
of inputs can occur is in industrial processes, which are almost always nonlinear and are complex and difficult to model. Besides nonlinearity, sometimes it is difficult to choose the most relevant input variables.

Thus, input selection is a crucial step with the aim of reducing the model’s complexity and removing inputs that do not contribute to the model performance. Input selection methods for nonlinear systems have been studied in [9]. The methods found in the literature can generally be divided in two main groups: model-free methods and model-based methods [24]. The methods discussed in this paper belong to the group of model-based methods, where a particular model structure is used to find the significant inputs. Models with different sets of input variables are compared, and the model that minimizes a given performance criterion is selected. Usually these methods are straightforward and all the subsets of variables must be tested.

In this paper, two different approaches are proposed and compared: top-down tree search and bottom-up tree search. Both methods perform input selection on two types of applications: classification and nonlinear dynamic modeling. The top-down approach (TD) begins with all inputs and at each stage the worst ones are discarded. On the other hand, the bottom-up approach (BU) starts with a single input, and at each stage selects one input and discards the worst input(s). The models obtained at each iteration are constructed using fuzzy modeling, where the rules are identified using fuzzy clustering. The performance measure used in the algorithm is very important as it decides whether an input is selected or discarded. To tackle this problem several performance criteria were used and the results are compared. Real data is used for the design and validation of two classification databases and two dynamic processes used as examples.

The paper is organized as follows. Fuzzy modeling is briefly described in Section 2. Section 3 presents the proposed decision tree methods for input selection, and describes briefly four different performance measures used in the paper. Regression and classification examples of the proposed tree search methods are presented and discussed in Section 4. Finally, some conclusions are drawn in Section 5.

2. Fuzzy modeling

Fuzzy modeling using inputs of a system, as e.g. features in classification or measures of the process variables in dynamic fuzzy modeling, is a tool that allows an approximation of nonlinear systems when there is none or few knowledge of the system to be modeled [17]. The fuzzy modeling approach has several advantages when compared to other nonlinear modeling techniques. In general, fuzzy models can provide a more transparent model and can also give a linguistic interpretation in the form of rules. Fuzzy models use rules and logical connectives to establish relations between the variables defined to derive the model. This paper uses Takagi–Sugeno (TS) fuzzy models [20], which consist of fuzzy rules where each rule describes a local input–output relation, typically in an affine form. This affine form is given by

\[ R_i : \text{If } x_1 \text{ is } A_{i1} \text{ and } \ldots \text{ and } x_n \text{ is } A_{in} \text{ then } y_i = a_{i1}x_1 + \cdots + a_{in}x_n + b_i, \]  

where \( i = 1, \ldots, K \) (\( K \) denotes the number of rules in the rule base), \( R_i \) is the \( i \)-th rule, \( x = [x_1, \ldots, x_n]^T \) is the antecedent vector, \( n \) is the number of inputs, \( A_{i1}, \ldots, A_{in} \) are fuzzy sets defined in the antecedent space, \( y_i \) is the output variable for rule \( i \), \( a_i = [a_{i1}, \ldots, a_{in}] \) is a parameter vector and \( b_i \) is a scalar offset. The consequents of the affine TS model are
hyperplanes in the product space of the inputs and the output. The model output, \( y \), can then be computed by aggregating the individual rules contribution: 
\[
y = \frac{\sum_{i=1}^{K} \beta_i y_i}{\sum_{i=1}^{K} \beta_i}
\]
where \( \beta_i \) is the degree of activation of the \( i \)th rule: 
\[
\beta_i = \prod_{j=1}^{n} \mu_{A_{ij}}(x_j), \quad \mu_{A_{ij}}(x_j) : \mathbb{R} \rightarrow [0, 1]
\]
is the membership function of the fuzzy set \( A_{ij} \) in the antecedent of \( R_i \).

The identification of fuzzy models is solved in two steps: structure identification and parameter estimation. First, the structure of the model must be determined. In other words, the significant inputs must be chosen. In this paper, this task is performed using the tree search algorithms proposed in Section 3. The number of variables must be small enough for the sake of simplicity, but with the sufficient number of variables to achieve the desired accuracy. To identify the model, the input matrix \( \mathbf{X} \) and an output vector \( \mathbf{y} \) are constructed from the available data:
\[
\mathbf{X}^T = [\mathbf{x}_1, \ldots, \mathbf{x}_N], \quad \mathbf{y}^T = [y_1, \ldots, y_N].
\]
Here \( N \gg n \) is the number of samples used for identification. The number of rules \( K \), the antecedent fuzzy sets \( A_{ij} \), and the consequent parameters \( a_i \) and \( b_i \) are determined by means of fuzzy clustering in the space of the input and output variables. Hence, the data set \( \mathbf{Z} \) to be clustered is composed from \( \mathbf{X} \) and \( \mathbf{y} \):
\[
\mathbf{Z} = [\mathbf{X}, \mathbf{y}]^T.
\]
Given the data \( \mathbf{Z} \) and the number of clusters \( K \), several fuzzy clustering algorithms can be used. This paper uses the fuzzy c-means (FCM) [1] and the Gustafson–Kessel (GK) [8] clustering algorithms to compute the fuzzy partition matrix \( \mathbf{U} \). The matrix \( \mathbf{Z} \) provides a description of the system in terms of its local characteristic behavior in regions of the data identified by the clustering algorithm, and each cluster defines a rule. Unlike the fuzzy c-means algorithm, which is usually more suitable for classification problems, the GK algorithm applies an adaptive distance measure, finding hyper-ellipsoid regions in the data that can be efficiently approximated by the hyper-planes described by the consequents in the TS model. This paper uses the FCM in the classification examples and the GK in the identification of dynamic systems.

The fuzzy sets in the antecedent of the rules are obtained from the partition matrix \( \mathbf{U} \), whose \( ik \)th element \( \mu_{ik} \in [0, 1] \) is the membership degree of the data object \( z_k \) in cluster \( i \). One-dimensional fuzzy sets \( A_{ij} \) are obtained from the multidimensional fuzzy sets defined point-wise in the \( i \)th row of the partition matrix by projections onto the space of the input variables \( x_j \):
\[
\mu_{A_{ij}}(x_{jk}) = \text{proj}^{\mathbb{R}_{n+1}}_{j}(\mu_{ik}),
\]
where \( \text{proj} \) is the point-wise projection operator [12]. The point-wise defined fuzzy sets \( A_{ij} \) are approximated by suitable parametric functions in order to compute \( \mu_{A_{ij}}(x_j) \) for any value of \( x_j \). The consequent parameters for each rule are obtained as a weighted ordinary least-square estimate. Let \( \theta_i^T = [a_i^T; b_i] \), let \( \mathbf{X}_e \) denote the matrix \([\mathbf{X}; 1]\) and let \( \mathbf{W}_i \) denote a diagonal matrix in having the degree of activation, \( \beta_i(x_k) \), as its \( k \)th diagonal element. Assuming that the columns of \( \mathbf{X}_e \) are linearly independent and \( \beta_i(x_k) > 0 \) for \( 1 \leq k \leq N \), the weighted least-squares solution of \( \mathbf{y} = \mathbf{X}_e \theta + \epsilon \) becomes
\[
\theta_i = [\mathbf{X}_e^T \mathbf{W}_i \mathbf{X}_e]^{-1} \mathbf{X}_e^T \mathbf{W}_i \mathbf{y}.
\]
Rule bases constructed from clusters can be redundant due to the fact that the rules defined in the multidimensional premise are overlapping in one or more dimensions. A possible approach to solve this problem is to reduce the number of inputs $n$ of the model. This paper proposes the use of decision trees to perform input selection, as explained in the following section.

3. Decision tree methods for input selection

The identification of fuzzy models is a quite complex task; when the system has a large number of variables, the most relevant ones must be chosen. Thus, it is necessary to select carefully the variables that are relevant for each output. As the relations between the variables are not well known, this paper proposes two automatic methods to determine which input variables influence the outputs: a top-down approach and a bottom-up approach. The first approach starts with all the inputs and eliminates successively the ones that are less relevant. The bottom-up approach starts with only one input, adds successively the most relevant ones and discards the worst. Both processes stop when a given criterion verifies that a better model is not possible to obtain.

The number of fuzzy rules (or clusters) that best suits the data must be determined for identification. The following criterion, as proposed in [19], is used to determine the number of clusters:

$$S(c) = \sum_{k=1}^{N} \sum_{i=1}^{c} (\mu_{ik})^{m} (\|x_k - v_i\|^2 - \|v_i - \bar{x}\|^2),$$

where $N$ is the number of data to be clustered, $c$ is the number of clusters ($c \geq 2$), $x_k$ is the $k$th data point (usually vector), $\bar{x}$ is the mean value for the inputs, $v_i$ is the center of the $i$th cluster, $\mu_{ik}$ is the grade of the $k$th data point belonging to $i$th cluster and $m$ is an adjustable weight. The parameter $m$ has a great importance in this criterion. The bigger the $m$ the bigger the optimum number of clusters is. Therefore, this value is normally around 2.

The number of clusters $c$ is increased from two up to the number that gives the minimum value for $S(c)$. Note that this minimum can be local. However, this procedure diminishes the number of rules and consequently the complexity of the fuzzy model. At each iteration, the number of clusters is determined using the fuzzy $c$-means to find the cluster centers $v_i$ and the process stops when $S(c)$ increases from one iteration to the next one. The first term of the right-hand side of (6) is the variance of the data in a cluster and the second term is the variance of the clusters themselves. The optimal clustering achieved is the one that minimizes the variance in each cluster and maximizes the variance between clusters.

3.1. Performance criteria

In order to select the proper model inputs, several performance criteria can be used. Previously, only the regularity criterion (RC) [19] was used both in bottom-up and top-down approaches [21, 22]. This paper extends the possible criteria to four possibilities, and tests all of them.

The simplest performance criterion is the root mean square (RMS) of the output error with respect to an independent set of checking data. Another possibility is to use the checking error criterion (CEC) [3]. This criterion divides the identification data into two
groups, A and B. The model is generated using the data from group A, and the validation is done in the group data B. The performance criterion CEC used to evaluate the models is defined as follows:

\[
\text{CEC} = \sqrt{\frac{1}{k_B} \sum_{i=1}^{k_B} (y_i^B - y_i^{AB})^2}
\]

(7)

where \(k_B\) is the number of data points of B, \(y_i^B\) is the output data of group B and \(y_i^{AB}\) is the model output for group A estimated using data from group B. The presented approach presents good results when the system has high prediction accuracy, and is sensitive to the choice of training data and simulation data.

Another criterion used in this paper is the unbiased criterion (UC), which is based on the regularity criterion. This criterion is derived from the group method of data handling (GMDH) [11]. Also in this criterion the data is divided into two groups of data, A and B. The performance criterion UC is defined in [18] as follows:

\[
\text{UC} = \sqrt{\left[ \sum_{i=1}^{k_A} (y_i^{BA} - y_i^{AA})^2 + \sum_{i=1}^{k_B} (y_i^{AB} - y_i^{BB})^2 \right]} ,
\]

(8)

where \(k_A\) and \(k_B\) are the number of data points of groups A and B, respectively, \(y_i^{BA}\) is the output for group B estimated using data from A, \(y_i^{AA}\) is the output of A estimated using data from A, and \(y_i^{BB}\) is the output of B estimated using data of B. When UC is used, the selected input variables are not the ones with the best prediction ability [18].

The regularity criterion (RC) is another performance criterion that is used in this paper, and is defined as follows [6,19]:

\[
\text{RC} = \frac{\left[ \sum_{i=1}^{k_A} (y_i^A - y_i^{AB})^2 / k_A + \sum_{i=1}^{k_B} (y_i^B - y_i^{BA})^2 / k_B \right]}{2} ,
\]

(9)

where \(k_A\) and \(k_B\) are the number of data points of the groups A and B, respectively, \(y_i^A\) and \(y_i^B\) are the output data of the groups A and B, respectively, \(y_i^{AB}\) and \(y_i^{BA}\) are defined as in (8). This approach is the mean square error of the checking error criteria for the two models. The square root (SR) of RC is another performance criterion used to select the input variables, and is naturally defined as follows [3]:

\[
\text{SR} = \sqrt{\text{RC}}.
\]

(10)

All these performance criteria (PC) are used in this paper to check which one will be the most suitable to be chosen for input selection in fuzzy models.

3.2. Bottom-up approach

The bottom-up approach described in this paper follows the same principle of the RC approach [19], which is also a bottom-up approach. However, a more recent algorithm that minimizes the computational time with similar performance is used here [21]. This paper generalizes the model evaluation method by using the four different performance
criteria presented in Section 3.1 to test the fuzzy models. The bottom-up approach looks for single inputs that may influence the output and work to assemble the single input, in order to achieve a model with good performance.

Fig. 1 presents the bottom-up approach at each stage. By using two groups of data, A and B, two fuzzy models are built, one for each group, starting with only one input. At this stage, a fuzzy model is built for each of the $n$ inputs in consideration. The models are evaluated using the RC, SR, CEC and UC performance criteria. The chosen criterion is computed for each model at this stage, and the input that minimizes the performance criterion is selected as the best one. The one that maximizes the criterion is rejected and is not included in the next stage. At the next stage, the input already selected is fixed, i.e., it belongs to the model structure. The other input candidates, excluding the rejected input in the prior stage, are added to the previous fuzzy model one at a time. In Fig. 1, this stage 2 tests $n - 2$ models containing the inputs $(x_2, x_1), (x_2, x_3), (x_2, x_5), \ldots, (x_2, x_n)$. Note that in this example, the input $x_4$ was excluded at stage 1. When this second stage finishes, the fuzzy model has two inputs. The second input is chosen as the one that minimizes the value of the chosen performance criterion, and as before, the input that maximizes the value of the criterion is rejected. In Fig. 1, the input chosen is now $x_3$. This procedure is repeated until the value of the performance criterion increases. This situation is presented in Fig. 1 at stage 4. At this stage, one should have all the relevant input variables for the considered output. Then, the chosen inputs are, in this example, $x_1, x_2$ and $x_3$.

In a generic case, using the RC as proposed in [19], the maximum number of iterations is $n \times (n + 1)/2$, where $n$ is the number of possible state variables. The number of iterations using the bottom-up approach decreases. For an odd number of inputs the maximum
The number of iterations is \((n + 1)^2/4\) and for an even number of inputs the maximum number of iterations is \(n \times (n + 2)/4\). Thus, the number of iterations reduces significantly, and then the computational time is also reduced. Assuming that input and output data are collected from a given system, the selection of inputs using this methodology generally entails the algorithm described in Algorithm 1.

**Algorithm 1 (Bottom-up approach)**

- Cluster the data using fuzzy \(c\)-means with two initial clusters;
- Increase the number of clusters until \(S(c)\) in (6) reach its minimum;
- Divide the data set into two groups \(A\) and \(B\);
- For each input in the input vector that does not belong to the inputs of the model:
  - repeat
  - Build two models, one using data group \(A\) and other using data group \(B\);
  - Compute the chosen PC (RC, SR, CEC or UC, defined in Section Section 3.1);
  - Select the input with the lowest value of PC as a new input of the model;
  - Discard the input with the largest PC;
- until PC increases or the end of the input vector is reached.
- Select the final inputs;
- Using the number of clusters given from (6) and the inputs selected by the proposed approach, build a fuzzy model using a fuzzy clustering algorithm.

Summarizing, the bottom-up approach presented in Algorithm 1 differs from the RC algorithm proposed in [19] because it is possible to exclude one or more variables. This is an advantage, as it allows the reduction of the number of iterations per stage. In some cases, it allows even the reduction of the number of stages, reducing also the computational time.

### 3.3. Top-down approach

Another approach proposed to select the input variables is the top-down (TD) approach. This approach begins with all the input variables, and removes the one(s) with the worst performance at each stage. This approach is based on the algorithm, proposed in [22]. The identification data is divided into two groups, \(A\) and \(B\), as in the BU approach. The TD approach uses also the performance criteria RC, SR, CEC or UC presented in Section 3.1 to select the inputs.

Fig. 2 presents the top-down approach at each stage. Again, one model is built for each group \(A\) and \(B\) using all the inputs. The proposed approach begins, at stage 0, by using all the inputs. One of the proposed performance criteria (PC): RC, SR, CEC or UC is computed. This is considered as the value to decrease at the following stages. Then, at stage 1, \(n\) fuzzy models are obtained, where each one of them is identified without one of the inputs used at stage 0. The inputs that are not considered in the model are represented in Fig. 2 by the white circles. The values of the chosen PC, for each of the \(n\) models, are compared to the value obtained at stage 0. For each new value that is smaller, the corresponding input \(x_i\) is removed from the vector of inputs. This situation is represented in Fig. 2 by the dashed line between stage 1 and stage 2. At the next stage, a fuzzy model is identified using only the inputs that have not been discarded at stage 1. The value of the chosen perfor-
The performance criterion is computed, and is used as reference for the next stage. The fuzzy model obtained at stage 2 has \( n - p \) inputs, where \( n \) is the number of initial inputs and \( p \) is the number of inputs removed at stage 1. The presented procedure is repeated until the value of the performance criterion is not decreased by excluding any input. This situation is presented at stage 3, in Fig. 2. At this stage, all the obtained models have values of PC that are bigger than the PC value obtained at stage 2. Thus, the inputs considered at stage 2 are the ones that are used in the final model. The top-down approach proposed in this paper is described in Algorithm 2.

Algorithm 2 (Top–down approach)

Cluster the data using fuzzy \( c \)-means with two initial clusters;
Increase the number of clusters until \( S(c) \) in (6) reach its minimum;
Divide the data set into two groups \( A \) and \( B \);
\( i = 0 \), where \( i \) is the stage number;
repeat
  if stage is zero then
    Build a model using all the input variables;
    \( m = n \); where \( n \) is the number of initial inputs
  else
    \( i = i + 2 \);
    Build a model using the input variables not discarded at the previous stage;
  end if
Compute \( PC_i \), (RC, SR, CEC or UC, defined in Section 3.1);
for \( j = 1 \) to \( m \) do
Build two models, for groups A and B, using all the inputs except input j; Compute $PC_{i+1,j}$

if $PC_{i+1,j} < PC_j$ then
  Discard the input j not used in modeling;
end if
end for

$m = m - p$, where $p$ is the number of discarded inputs;

until (no input is discarded) OR (model has only one input) OR ($PC_i > PC_{i-2}$)

Using the number of clusters given from (6) and the inputs selected by the proposed approach, build a fuzzy model using a fuzzy clustering algorithm.

This algorithm differs from the bottom-up approach, as it obtains at each stage multivariable fuzzy models, begins with all the input vector, and discard one or more inputs at each stage. This is a clear advantage, which allows the reduction of the number of iterations per stage. Further, in some cases, it can even reduce the number of stages, and consequently the computational time can be reduced when compared to the BU approach. On the other hand, as the TD approach uses much more inputs to build each model from the beginning, the identification of each model can be computationally intensive. This is especially critical when the number of inputs is large.

4. Application examples

This section presents the application of the proposed approaches to four examples: the wine classification data [14], the ionosphere classification data [16], a column flotation process [23] and a container gantry crane process [13]. Note that dynamic processes are highly nonlinear systems. These examples can test the effectiveness of the proposed decision tree search in two different application domains: classification and dynamic modeling.

The classification data used in this paper is obtained from the repository of University of California Irvine [14]. The percentile misclassification (MSp) is used to measure the performance of the obtained classification models:

$$MSp = \frac{(n - MIS)}{n} \times 100\%, \quad (11)$$

where $n$ is the number of used samples and MIS is the number of misclassifications.

The identification of dynamic systems is performed using measurement data. The experiments to obtain these data were design so that all system variables could take all possible values within their respective variable range. Note that the model designer must choose carefully the most relevant variables in order to keep the models as simple as possible. The performance indexes used for measuring the accuracy of the dynamic processes are the root mean square (RMS) and the variance accounted for (VAF). The root mean square is defined as

$$RMS = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}}, \quad (12)$$
where $y_i$ is a system output and $\hat{y}_i$ is the correspondent model output. The percentile VAF is defined as follows:

$$VAF = \frac{1 - \frac{\text{cov}(y_i - \hat{y}_i)}{\text{cov}(y_i)}}{\text{cov}(y_i)} \times 100\%$$  \hspace{2cm} (13)

where cov is the covariance of the respective vector.

4.1. Wine classification data

The wine data is a widely used classification data obtained from the University of California [14], and contains the chemical analysis of 178 wines grown in the same region in Italy, derived from three different cultivars. Thirteen continuous attributes are available for classification: alcohol, malic acid, ash, alkalinity of ash, magnesium, total phenols, flavanoids, non-flavanoids phenols, proanthocyanism, color intensity, hue, OD280/OD315 of diluted wines and proline.

The wine data classification was used in [4]. All the 178 samples were applied for learning 60 crisp (non-fuzzy) if–then rules in a real-coded genetic-based machine learning approach. The obtained results are: best classification rate 100%, average classification rate 99.5% and worst classification rate 98.3%, which corresponds to three misclassifications. Another approach was presented in [10]. In this paper, all the 178 samples are used to design a fuzzy classifier with 60 fuzzy rules using an integer coded GA and grid partitioning. The obtained results for 10 independent trials are: best classification rate 99.4% (one misclassification), average classification rate 98.5% and worst classification rate 97.8% (four misclassifications). In both approaches [4,10], the final rule base contains 60 rules. Similar results have been obtained in [15], but using only three fuzzy rules.

The TD and BU approaches proposed in this paper have been applied to the wine data. Ten runs were made, and the best, average and worst classification rates were computed. The obtained results are presented in Table 1.

The results are similar, but some differences can be noticed. When the BU approach is used, the best criterion is the CEC in terms of average and worst results. However this criterion misses always at least one classification, while all the others are sometimes able to classify correctly all the samples. On the other hand, the UC criterion is clearly the worst, in terms of average and worst classification results. All the criteria have only two clusters, i.e., two fuzzy rules, which is clearly remarkable when compared to [4,10], which used 60 rules. The BU approach uses a similar, but inferior, number of rules than the fuzzy model derived in [15], resulting in simpler models with similar or even better accuracy. The main drawback of the BU tree search is that the worst classification rates are slightly worse than

<table>
<thead>
<tr>
<th>Performance criteria</th>
<th>Bottom-up</th>
<th>Top-down</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Average</td>
</tr>
<tr>
<td>RC</td>
<td>100</td>
<td>96.7</td>
</tr>
<tr>
<td>SR</td>
<td>100</td>
<td>98.5</td>
</tr>
<tr>
<td>CEC</td>
<td>99.4</td>
<td>99.0</td>
</tr>
<tr>
<td>UC</td>
<td>100</td>
<td>92.2</td>
</tr>
</tbody>
</table>
the previous classification approaches, i.e., a very good model is not always obtained at the first identification.

For the TD approach, RC is clearly the criterion that presents the best results. Here there is no doubt that this should be the chosen criterion. The worst criterion is now clearly the SR one. Again, all the criteria used two clusters/rules. Note that the results obtained with the TD approach using RC are better than any other previous approaches presented in [4,10,15].

4.2. Ionosphere classification data

The ionosphere data is originally from the Space Physics Group; Applied Physics Laboratory; Johns Hopkins University, and was collected by a system in Goose Bay, Labrador. The system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kW. The targets were free electrons in the ionosphere [16]. “Good” radar returns are those showing evidence of some type of structure in the ionosphere. “Bad” returns are those that do not; their signals pass through the ionosphere.

The received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this database are described by two attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal. The number of instances of the ionosphere classification data is 351 and the number of attributes 35. The 35th attribute is either “good” or “bad” according to the definition summarized above. This is a binary classification task. The training set size was set to 200 points and the testing set size was set to 151 points. This division is commonly used to test this data set.

Previously, several classifiers have been tested in this data. A classifier based on resource limited artificial immune systems applied to the ionosphere data classification is presented in [25]. The ranking of the best results obtained using several classifiers can be found in [5]. Here, the best classification rate obtained is 98.7%, when a method using neural networks with the simplex method is applied.

Both TD and BU tree search approaches have been applied to the ionosphere data. Ten runs were made, and the best average and worst classification rates were computed. The obtained results are presented in Table 2. The best criterion is clearly the CEC for both TD and BU approaches. The best classification rate is 98.0%. The RC and SR criteria present very similar results. Further, the UC criterion is clearly worse in the BU approach. The results are not the best obtained (in [5] only one approach achieved a better classifi-

<table>
<thead>
<tr>
<th>Performance criteria</th>
<th>Bottom-up</th>
<th>Top-down</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best</td>
<td>Average</td>
</tr>
<tr>
<td>RC</td>
<td>97.4</td>
<td>96.7</td>
</tr>
<tr>
<td>SR</td>
<td>97.4</td>
<td>96.7</td>
</tr>
<tr>
<td>CEC</td>
<td>98.0</td>
<td>98.0</td>
</tr>
<tr>
<td>UC</td>
<td>88.0</td>
<td>87.4</td>
</tr>
</tbody>
</table>
cation rate of 98.7%) but are very close to the best ones. Again, only two fuzzy rules are used, and the models are quite simple.

4.3. Column flotation process

The flotation process separates fine solid particles based on physic and chemical properties of their surfaces [7]. Industrially, it is a continuous solid–solid separation process performed in a vessel where a three-phase system is present: solid particles, air bubbles and water. This pulp is previously conditioned with the controlled addition of small quantities of specific chemical reagents to promote the selective formation of aggregates between solid particles of a given composition and air bubbles. Air is continuously injected in the pulp, formatting air bubbles.

Hydrophobic particles adhere, after collision, to the air bubbles, which move upwards to the top of the vessel where they are recovered as the floated product. Hydrophilic particles settle in the pulp become the non-floated product or underflow. A flow of air is continuously injected in the medium of the vessel to transport the particles. There is also a shower of water in the top of the froth column, used to “wash” from the floated product hydrophilic particles that were dragged with the aggregates bubbles-hydrophobic particles. A simple representation of this process can be seen in Fig. 3. The flotation environment is influenced by variables, such as pulps pH, solids concentration, flotation intensity (speed of agitation and air flow rate), particles size and type of water medium. Besides mineral processing, it is used in some other fields, such as solvent extraction and recycling [2].

A previous study has been performed to obtain a fuzzy model of the process [23]. The inputs applied to column flotation process are: the air flow rate $Q_{\text{air}}$, the rejected flow rate $Q_{\text{reject}}$. 

![Fig. 3. Column flotation scheme, where M are flow meters and P are pressure sensors.](image)
The feed flow rate \( Q_{\text{feed}} \), and the washing water flow rate \( Q_{\text{ww}} \). The outputs to be estimated are the level \( H \), the bias flow rate \( Q_{\text{bias}} \), and the air holdup \( \epsilon \).

Top-down and bottom-up approaches have been used to identify the model structure, and it was found that the optimum number of clusters (fuzzy rules) is 5 for level \( H \), bias flow rate \( Q_{\text{bias}} \) and holdup \( \epsilon \). The four performance criteria (RC, UC, CEC and SR) were tested. The results in terms of VAF for validation data are presented in Table 3. The RMS errors are presented in Table 4.

In general, the decision tree method with best performance is the top-down approach. The different performance criteria used to select the proper inputs do not present large differences. In the TD approach, the CEC is slightly better than the other criteria. In the BU approach, the UC performance criterion is more regular than the others, as it always gives good results even if they are not the best.

In order to show that even a criterion that was not the best gives very good results, the validation of the final model using a test data set, when a TD approach with RC is applied, is depicted in Fig. 4. The difference between the model and the real output is very small, showing the good performance of the proposed tree search approaches.

### 4.4. Container gantry crane

A container gantry crane is normally used in ports, outdoors and near the shore, with many changes in weather conditions, for instance an abrupt change of the wind direction. A container gantry crane consists of a bridge girder on portal legs from which a trolley system is suspended, see Fig. 5. The trolley can travel along the bridge girder that stretches over the container ship and part of the quay for loading and unloading the ship. A hoisting mechanism consisting of a spreader suspended from the trolley by means of hoisting cables is used for grabbing and hoisting the container. The inputs are the voltage of the crane motors, \( U_1 \) and \( U_2 \). The outputs of the container gantry crane are the position of the trolley at a desired horizontal location \( x \), the rope length \( l \) and the swing of the load \( \theta \). Data

---

Table 3  
Column flotation process

<table>
<thead>
<tr>
<th>Output</th>
<th>Bottom-up</th>
<th>Top-down</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RC</td>
<td>UC</td>
</tr>
<tr>
<td>( H )</td>
<td>83.1</td>
<td>90.7</td>
</tr>
<tr>
<td>( Q_{\text{bias}} )</td>
<td>95.6</td>
<td>92.5</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>85.2</td>
<td>86.3</td>
</tr>
</tbody>
</table>

Table 4  
Column flotation process

<table>
<thead>
<tr>
<th>Output</th>
<th>Bottom-up</th>
<th>Top-down</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>RC</td>
<td>UC</td>
</tr>
<tr>
<td>( H )</td>
<td>3.4</td>
<td>2.4</td>
</tr>
<tr>
<td>( Q_{\text{bias}} )</td>
<td>0.8</td>
<td>0.5</td>
</tr>
<tr>
<td>( \epsilon )</td>
<td>0.6</td>
<td>0.6</td>
</tr>
</tbody>
</table>

VAF of the obtained fuzzy models using tree search approaches.

RMS of the obtained fuzzy models using tree search approaches.
from the container gantry crane is used to identify the model in normal operation. The container gantry crane presented in [13] is used to obtain the data.

Again, top-down and bottom-up approaches were both used, as well as the four performance criteria (RC, UC, CEC and SR). When both TD and BU approaches are used to identify the model structure, the optimum number of clusters is 5 for the horizontal
location, 5 for the rope length and 8 for the swing angle output. The results in terms of VAF for validation data are presented in Table 5. The RMS errors are presented in Table 6.

In this system, it is clear that the top-down approach performs much better than the bottom-up one. The values obtained with the bottom-up tree search for the swing angle are meaningless in terms of VAF (negative values), and as so they are not presented in Table 5. Again, the different performance criteria used in the decision tree search do not present large differences in terms of results.

The validation of the final model using a different data set is presented in Fig. 6, when the TD approach with RC is used. Also with the container gantry crane, all the outputs show a very good performance. The comparison between the model output (continuous line) and the system output (dashed line) is very difficult because the two lines are almost identical, and cannot be distinguished in the figure.

4.5. Discussion

The decision tree search methods proposed in this paper were applied to derive fuzzy models for two classification problems and for two dynamic systems. In all the systems the approach presents a good performance. The use of top-down or bottom-up approaches with any of the proposed performance criteria is not too relevant to select correctly the inputs of the models. However, the TD approach can use more inputs, because it starts with all of them, and can stop when a large number of inputs are still being used. This can lead in general to more accurate models, as the container gantry crane showed clearly. The BU approach starts with only one input, and can stop before a very accurate model is derived. Therefore, it can end up with simpler but less accurate models. Both proposed tree search approaches are limited in terms of time, when a model has more than about 50 inputs; note that the computational time increases exponentially with the number of inputs. In these cases (huge data sets), these methods need further research.
5. Conclusions

This paper proposes two different decision tree search methods, top-down and bottom-up, for the selection of the model inputs and compares the performance of the obtained models using four different performance criteria. A fuzzy model identification technique is used, where the rules are estimated using FCM or GK clustering algorithms. The approach is applied to four examples: wine classification data, ionosphere classification data, column flotation process and a gantry crane process. The obtained results show that the structure of the model is well determined using the decision tree methods. For the classification examples, the use of different performance criteria is not significantly relevant. The structure of dynamic examples is also well determined especially when a top-down approach is used. Minor limitations in terms of performance can be obtained when the bottom-up approach is applied. On the other hand, the top-down approach can leave some unnecessary inputs in the model.

Future research will try to combine synergetically both approaches to avoid the disadvantages when they are applied separately. Further, both decision tree search methods are difficult to apply when the system has more than about 50 inputs. Thus, methods to alleviate the computational burden in huge databases are also under study.

Fig. 6. Validation results of the gantry crane process using TD with RC (continuous line: model output; dashed line: system output).
Acknowledgements

This work is partially supported by the project POCTI/ECM/38296/2001-APTDEC, Fundação para a Ciência e Tecnologia, Ministério do Ensino Superior e da Ciência, Portugal, co-sponsored by FEDER, Programa Operacional Ciência e Inovação 2010, FCT, Portugal and by the “Programa do FSE-UE, PRODEP III, acção 5.3, no âmbito do III Quadro Comunitário de apoio”.

References


