

G.Vagliasindi, A. Murari
and JET EFDA contributors

An Automatic Fuzzy Logic Classifier based on Classification and Regression Trees for Confinement Regime Identification on JET

"© – COPYRIGHT ECSC/EEC/EURATOM, LUXEMBOURG – 2008"

"Enquiries about Copyright and reproduction should be addressed to the
Publications Officer, EFDA, Culham Science Centre, Abingdon, Oxon, OX14 3DB, UK."

An Automatic Fuzzy Logic Classifier based on Classification and Regression Trees for Confinement Regime Identification on JET

G.Vagliasindi, A. Murari
and JET EFDA contributors*

¹*Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi-Università degli Studi di Catania,
95125 Catania, Italy*

²*Consorzio RFX-Associazione EURATOM ENEA per la Fusione, I-35127 Padova, Italy*

** See annex of M.L. Watkins et al, "Overview of JET Results ",
(Proc. 21 st IAEA Fusion Energy Conference, Chengdu, China (2006)).*

An Automatic Fuzzy Logic Classifier based on Classification and Regression Trees for Confinement Regime identification on JET.

G.Vagliasindi¹, A. Murari² and JET-EFDA Contributors*

JET-EFDA, Culham Science Centre, OX14 3DB, Abingdon, UK

1) Dipartimento di Ingegneria Elettrica Elettronica e dei Sistemi-Università degli Studi di Catania, 95125 Catania, Italy

2) Consorzio RFX-Associazione EURATOM ENEA per la Fusione, I-35127 Padova, Italy.

1 Introduction

The tokamak configuration can be operated in different confinement regimes, which can be significantly different not only in terms of performance but also of physics interpretation and control requirements. The High Confinement regime, the so called H mode [1], is a particularly relevant example. Its performance can be more than a factor of two better compared to the L-mode (Low confinement) but it is affected by edge instabilities, which require particular measures to avoid disruptions and could be very dangerous for the integrity of the entire device in the next generation of machines like ITER. Moreover the H-mode of confinement is an auto-organized state of the plasma, which develops spontaneously when certain conditions are met. The transition from one mode of confinement to the other presents some characteristics of phase transitions already studied in many other physical systems. On the other hand, the details of the plasma evolution from the L-mode to the H-mode state have not been fully understood yet neither from the point of view of the dynamics nor of the power requirements to trigger the transition. Even the nature of the control parameters remains unclear. In addition to defying physical interpretation, the H-mode of confinement presents also some important challenges from the point of view of control. The higher internal energy of the plasma, together with the increased plasma shaping used to improve performance, render the H-mode significantly more unstable and more prone to disruptions. On the other hand, up to now no reliable classifier of the confinement regime has been available for real time operation and therefore normally Tokamak devices are controlled in feed-forward, determining a priori in which type of regime the plasma will be at every point in time of the discharge. In case of unexpected transitions from state to another the control systems can therefore adopt a non optimal strategy which cannot only limit the performance but also compromise the equilibrium and contribute to trigger disruptions [2]. In the perspective of ITER, in which accessing the H-mode is essential but disruptions can have very harmful consequences, it is becoming urgent to develop a model to better interpret the H-mode physics and to identify the regimes in real time.

In this technical note, a Fuzzy Logic inference system is developed to both derive intuitive understanding of the L-H transition physics and to identify the confinement regime in real time. Given the fact that in JET GBytes of data can be acquired per discharge, the approach of Classification and Regression Trees (CART) [3] has been exploited for the feature selection, i.e. the identification of the most important signals to study the L-H transition (see section 2). It is worth pointing out that the CART method is fully non-linear and unbiased. It does not require any form of preliminary signal processing (not even normalization) and therefore is considered particularly suited to the exploratory phase of identifying the data with the highest explanatory power. The CART trees, trained to discriminate between L-mode and H-mode regimes using a large set of JET discharges, have also been used as the basis for the development of a Fuzzy Logic inference system [4]. The leaf nodes of the trees have been used first to determine the most appropriate membership functions for the various variables. Then an automatic method has been developed to derive the fuzzy rules from the CART topology. These rules have been determined by following the tree branches from each leaf node to the root and making appropriate decisions at each node (see section 3). The analysis has been particularized for time intervals far from the transitions, around the time of the L-H transition and around the time of the H-L transition.

The results of the Fuzzy Logic inference system are quite positive (see section 4). First of all the derived rules are very sound and quite intuitive; they formalize a body of common sense knowledge about the L-H transition that can contribute to knowledge acquisition and deeper studies of the underlying physics. The validity of these rules is proved by the classification capabilities of the system, which can exceed 90 % over the whole discharge. Further developments of the method and various refinements for real time applications are discussed in the final section.

2 CART for features selection

In order to build a system that identifies the confinement regime and, at the same time, provides intuitive understanding of the phenomenon under observations, the first step is the selection of the most relevant signals among the available ones in order to keep the system as simple as possible. To perform this feature extraction process Classification and Regression trees have been used.

Classification and Regression Tree is a non-parametric statistical method, which uses a decision tree to solve classification and regression problems using both categorical and continuous variables. The method was introduced by Breiman et al. [3] to build a decision tree, which describes one output variable as a function of different explanatory variables called also predictors. When the output is categorical, CART produces a classification tree, whereas if the output is continuous it will produce a regression tree. The method can be roughly summarized in three steps.

During the first stage, an overgrown tree is produced using a recursive partitioning technique to select variables and split points. Several criteria are available for determining the splits, described in detail in Breiman et al.. In the application described in this paper, the Gini criterion has been adopted. According to this method, to find the best variable for splitting a node, the algorithm checks all possible splitting variables (called splitters), as well as all possible values of the variable to be used to split the node, aiming at the maximization of the average “purity” of the two child nodes. For a binary target variable, for example mode L or H, the aim of the split is to group all the inputs into a group of L mode samples and another group of H mode ones. Since a complete separation is typically not achievable with one single variable, the procedure is repeated for the child nodes until pure terminal nodes are obtained.

The tree obtained at this stage is called maximal tree. It closely describes the training set and usually shows overfitting of the training data. A subsequent step is, then, required to converge on a better compromise between the tree complexity and its predictive power. This second step in CART algorithms is therefore the pruning of the maximal tree, which results in the creation of smaller subtrees obtained by successively cutting terminal branches. The pruning relies on a cost-complexity method, in which both the tree accuracy and complexity are considered. This method relies on a complexity parameter, called α , which is gradually increased during the pruning process. Beginning from the terminal nodes, the child nodes are pruned away if the resulting change in the predicted misclassification cost (a measure of the accuracy) is less than α times the change in the tree complexity. Thus, α is a measure of how much accuracy a split must add to the entire tree to counterpart the additional complexity. As α is increased, a series of trees with decreasing complexity is obtained.

The last stage is the selection of the optimal tree among the various pruned trees derived during the previous step. This selection is based on the evaluation of the predictive error also called misclassification or relative cost. The goal consists of selecting the optimal tree maximizing the relative cost and minimizing the tree complexity (i.e. the number of nodes) so that the information in the learning data set is fit but not overfit. In general this step would need an independent set of data to be accessed, but this requirement can be avoided using the technique of cross-validation (CV). It consists of dividing the entire sample randomly into N (usually 10) sub-samples, stratified by the response variable. One sub-sample is then used as the test sample and the other $N - 1$ (e.g., nine) are used to construct a large tree. The entire model-building procedure, i.e. tree growing and pruning, is repeated N times, with a different subset of the data reserved for use as the test dataset each time. Thus, N different models are produced, each one of which can be tested against an independent subset of the data. Trees within the sequences are matched up, based on their number of terminal nodes, to produce an estimate of the performance of the tree in predicting outcomes for a new independent dataset, as a function of the number of terminal nodes or complexity. Using this

method, a minimum cost is obtained when the tree is complex enough to fit the information in the learning dataset, but not so complex that noise in the data is fit as well. The final tree obtained has various nodes at which different variables are used for the splitting criteria.

One of CART features is the evaluation of the importance of the different explanatory variables, i.e. the variable provided as input during the building of the tree, to describe the output through the so-called “variable ranking method”. A variable that is not selected in the final tree could be considered as less important in describing the dataset than the variables that appear in the tree. To estimate a variable importance, the sum across all nodes in the tree of the improvement scores that the variable induces when it acts as a primary splitter is performed.

The importance values so produced allow ranking the different input signals from high to low importance. In this way, CART can be used for feature selection, being able to identify the most important variables to describe the output.

Since the aim of the work is also to try to derive an intuitive understanding of the confinement mode transition in a tokamak plasma, the shots have been divided in three subsets and analyzed separately. The three subsets include data around the transition from L to H, around the transition back from H to L, and away from the transitions in steady state L and H mode phases. In more detail, the first two datasets include data acquired 300 ms before and after the L→H and the H→L transition. The last one is obtained using an interval of 500 ms in L mode, between 1200 ms and 700 ms before the L→H transition and interval of 500 ms in H mode, between 1200 ms and 700 ms before the H→L transition.

The three subsets have been provided to CART in order to build three different trees. According to the variable ranking provided by CART the following signals have been evaluated as most relevant:

Table 1: the four most relevant variable for the three different subset. The signals are sorted in descendent order of importance.

L→H		H→L		Steady-state	
Symbol	Description	Symbol	Description	Symbol	Description
W_{mhd}	Magneto-hydrodynamic energy	β_N	Beta normalized over diamagnetic energy	β_N	Beta normalized over diamagnetic energy
β_N	Beta normalized over diamagnetic energy	B_T	Toroidal magnetic field	B_T	Toroidal magnetic field
T_e	Electron temperature	$FDWDT$	Time derivative of diamagnetic energy	$Lid4$	Outer interferometry channel
B_{T80}	Axial toroidal magnetic field at $\psi=0.8$	q_{95}	Safety factor at $\psi=0.95$	$FDWDT$	Time derivative of diamagnetic energy

In addition to these parameters, it has been considered important to test also the influence of geometrical parameters that can account for the position/shape of the plasma inside the vacuum vessel. Among the various possible geometrical parameters, the choice has been limited to the plasma elongation and radial and vertical position of the lower X point, since these are thought to be the ones most influential on the L-H transition. Several tests have been performed using both the geometrical parameters and the ones which have produced a better results are the radial and vertical position of the X point. These parameters have appended to the others in all the subsets used to obtain the results described in the following.

3 From CART to the Fuzzy Inference System

Once the most relevant variables have been selected with the exploratory procedure described in the previous section, they have been used to build new trees. In particular it has been decided to build three different trees, one for each of the time intervals specified at the end of the previous section. The obtained trees have then been the starting point for the definition of three Mamdani Fuzzy Inference Systems (FIS) in a completely automatic way.

The first step in developing any Fuzzy System consists of the definition of the membership functions (mf) for each input and output variable. This choice comprises the selection of the number of membership functions for each I/O, the type of mf (e.g. triangular, trapezoidal, sigmoidal, ...) and the parameters that represent the shape of each membership functions. In the case of confinement regime identification, the

number of membership functions for each input variable has been fixed to three. This choice has been dictated by the restrictions imposed by the MATLAB Fuzzy Toolbox which is, at the moment, a fundamental tool. Indeed, the Toolbox does not allow for the selection of multiple memberships in the rules definition but just for a single membership or its negation. So, in order to permit the correct and automatic translation of the CART rules to Fuzzy rules it has been necessary to limit the choice of mf to three; in the case of three mf any rule can indeed always be expressed either as the assertion of one mf or as the negation of another. It is worth mentioning that, as shown later, this constraint on the number of membership function does not constitute a severe limitations for the case under study, neither in terms of final performance nor in terms of explanatory power of the method. The type selected for the mf has been the trapezoidal one since it provides enough flexibility to cover each variable domain by dividing it in three different regions.

The trapezoidal curve is a function of a vector x and depends on four scalar parameters a , b , c and d as given by

$$f(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x - a}{b - a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d - x}{d - c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (1)$$

The parameters a and d locate the “feet” of the trapezoid and the parameters b and c locate the “shoulders.” (see Figure 1)

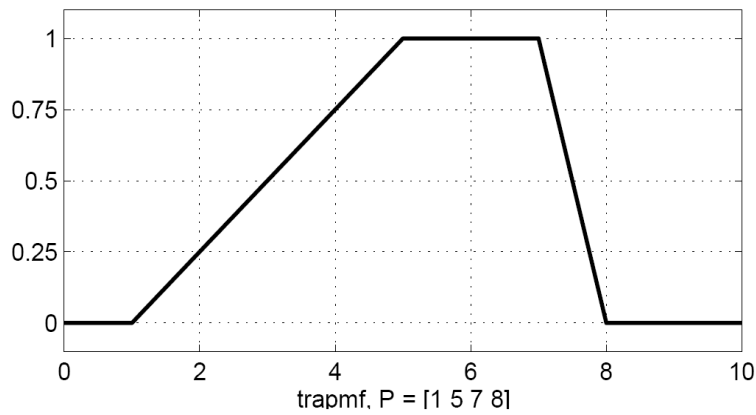


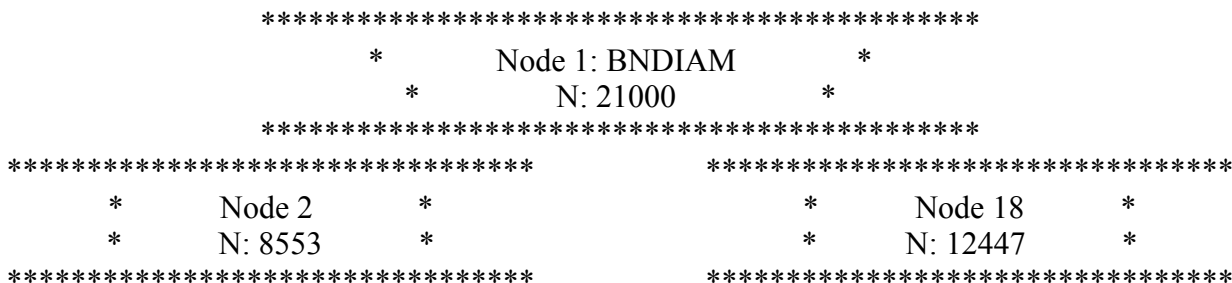
Figure 1: Schematic example of a trapezoidal membership function. 1 and 8 represent the feet of the trapezoid while 5 and 7 represent the shoulders.

As far as the output variable is concerned, being this just a categorical attribute, Low or High mode of confinement, it can only be divided into two fuzzy partitions.

Once defined the type of membership functions to be used, the subsequent step consists of retrieving from the tree the information required to formulate the FIS. This automatic formulation can be achieved in three main steps: a) extraction of the crisp rules from the CART tree; b) determination of the membership functions from the set of crisp rules obtained in the previous step; c) formulation of the fuzzy rules on the basis of the CART tree crisp rules and the mf. The algorithms necessary to perform these steps are described in detail in the next three subsections.

3.1 The extraction of the rules from the trees

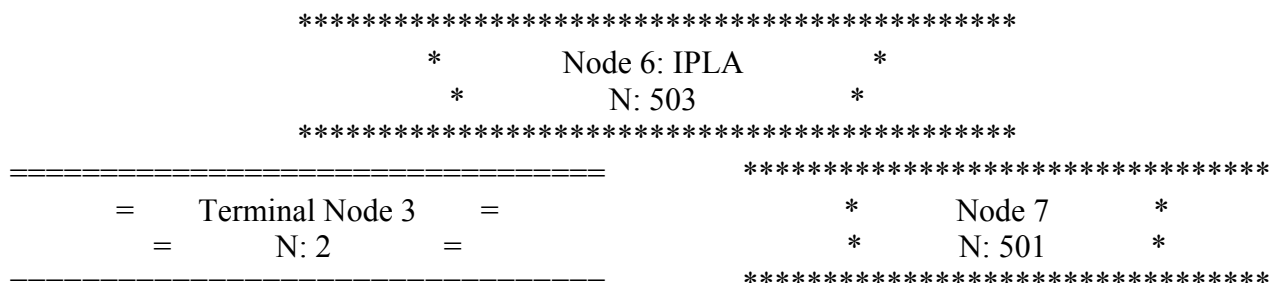
The CART provides, for every tree, a text file which contains the structure of the tree. In particular, for each node the splitting variable is reported together with the two child nodes, the splitting value and the number of input samples that are classified into the two child nodes. An example is shown in the following diagram 1.



Node 1 was split on BNDIAM
A case goes left if BNDIAM <= 0.59149

Diagram 2: the first set of lines report the node label and the splitting variable, together with the total number of samples available at that node. The second set of lines report, instead, the two child node with their respective label and the number of samples that are separated into them. Finally, the last set of lines indicate again the splitting variable and the splitting values specifying also the rule governing the Node.

In the case a child node is a terminal one, this information is also reported, as shown in the following diagram 2.



Node 6 was split on IPLA
A case goes left if IPLA <= -.439215E+07

Diagram 2: the only difference with Diagram 1 is the labelling of the left child node as a Terminal one.

A schematic description of the terminal nodes is also reported by CART. For each terminal node, labeled with a specific number, the number of samples discriminated at that node is provided (the Learn column) together with the Class to which these samples belong.

Table 2: schematic description of terminal nodes. Column 1 is the node number, column 2 the number of samples provided to the node, columns 3 and 4 are the number of samples belonging to each class according to the node classification.

Terminal Node Target Class Counts			
Node	Learn	Class 1(L-mode)	Class 2 (H-mode)
1	174.00	0.00	174.00
2	17533.00	17533.00	0.00
3	8.00	8.00	0.00
4	2.00	0.00	2.00
5	1000.00	1000.00	0.00

6	258.00	0.00	258.00
7	678.00	678.00	0.00
8	261.00	0.00	261.00
9	305.00	0.00	305.00
10	120.00	120.00	0.00
11	1.00	1.00	0.00
12	14.00	0.00	14.00
13	17165.00	0.00	17165.00
14	45.00	45.00	0.00
15	93.00	0.00	93.00
16	294.00	294.00	0.00
17	49.00	0.00	49.00

In order to derive the FIS from the CART classification, the CART output file is scanned automatically to determine the splitting value for each variable used as splitter. At the same time for each terminal node a rule is determined. The definition of these rules is performed in the following way. First the leaves corresponding to the terminal nodes are searched in the document. For each terminal node, the corresponding branch is scanned up to the root and a specific rule is devised for each intermediate node, on the basis of the inequality defined at each node.

To explain the algorithm in a more intuitive form, let's suppose that a FIS has to be derived from the tree in Figure 3. The tree is built using three predictors and discriminates the samples in two classes, Class 0 and Class 1. If we want to determine the rule for Terminal Node 1, we start by finding its father node, in this case Node 2, and the splitting variable and value for the father node. At the level of node 2 the rule which permits to discriminate whether the samples in Node 2 belong to the Terminal Node 1 or not is *if $x_2 \leq a_{2,1}$* (in this notation x_j is the j-th input variable and a_{jk} is the k-th splitting value of the j-th variable). To complete the rule, it is necessary to reach the root node. In this case, since the father node for the Terminal Node 2 is Node 1, we have to find the node which has Node 2 as child node. Looking at the figure it is Node 1 and the rule for Node 2 is *if $x_1 \leq a_{1,1}$ then Node 2*. Since Node 1 is the root node the search is finished and the whole rule for Terminal Node 1 whose samples belong to Class 0 is: *if $x_1 \leq a_{1,1}$ AND $x_2 \leq a_{2,1}$ then Class is 0*.

The rule derived for the terminal node 1 is very simple but more complicated cases have to be considered. For instance for Terminal Node 4 the its rule is: *if $x_1 > a_{1,1}$ AND $x_3 > a_{3,1}$ and $x_2 \leq a_{2,2}$ then Class is 0*.

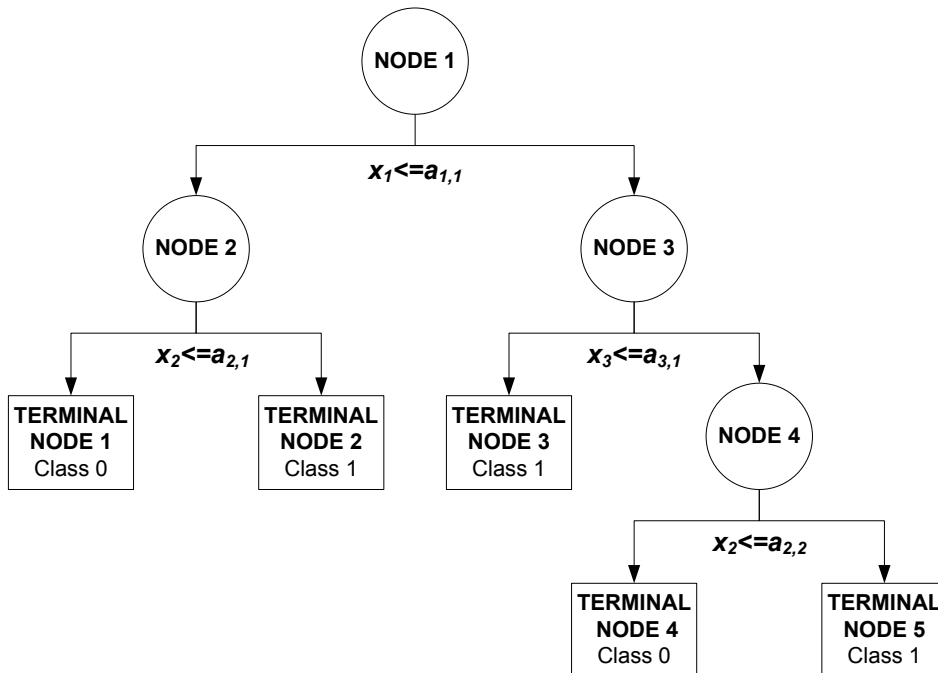


Figure 3: example of classification tree with three explanatory variables and five terminal nodes.

The output of the automated process just described is a matrix with as many rows as the total number of terminal nodes and as many columns as the total number of predictors plus a column reporting the class to which the terminal node belongs.

Table 3: tree rules matrix for the tree depicted in Figure 3.

	x_1	x_2	x_3	Class
Terminal Node 1	$x_1 \leq a_{1,1}$	$x_2 \leq a_{2,1}$	-	0
Terminal Node 2	$x_1 \leq a_{1,1}$	$x_2 > a_{2,1}$	-	1
Terminal Node 3	$x_1 > a_{1,1}$	-	$x_3 \leq a_{3,1}$	1
Terminal Node 4	$x_1 > a_{1,1}$	$x_2 \leq a_{2,2}$	$x_3 > a_{3,1}$	0
Terminal Node 5	$x_1 > a_{1,1}$	$x_2 > a_{2,2}$	$x_3 > a_{3,1}$	1

In the case of the simple pedagogical tree of Figure 3 each variable appears in each branch just once, so the determination of the final rule is simple. In the case of the slightly more complicated tree of Figure 4, some variables appear more than once while going up to the root node. There are four possibilities which are represented by terminal nodes 1, 2, 5 and 6.

In the case of terminal node 1, the final rule is *if $x_1 \leq a_{1,1}$ AND $x_2 \leq a_{2,1}$ AND $x_1 \leq a_{1,2}$ then Class is 0*. Since we are on the left side of the first splitting, therefore for values of $x_1 \leq a_{1,1}$ surely $a_{1,2}$ is smaller than $a_{1,1}$. The first inequality $x_1 \leq a_{1,1}$ is then overridden by the third $x_1 \leq a_{1,2}$ so the rule can be refined in this way: *if $x_1 \leq a_{1,2}$ AND $x_2 \leq a_{2,1}$ then Class is 0*.

In the case of terminal node 2, instead, the final rule is *if $x_1 \leq a_{1,1}$ AND $x_2 \leq a_{2,1}$ AND $x_1 > a_{1,2}$ then Class is 1*. Since $a_{1,2} < a_{1,1}$ and the third inequality is $x_1 > a_{1,2}$ the refined rule is *if $a_{1,2} < x_1 \leq a_{1,1}$ AND $x_2 \leq a_{2,1}$ then Class is 1*.

For Terminal Node 5 the rule is *if $x_1 > a_{1,1}$ AND $x_3 > a_{3,1}$ AND $x_1 \leq a_{1,3}$ then Class is 0*. In this case $a_{1,3} > a_{1,1}$ therefore the refined rule is *if $a_{1,1} < x_1 \leq a_{1,3}$ AND $x_3 > a_{3,1}$ then Class is 0*.

For Terminal Node 6 the rule is *if $x_1 > a_{1,1}$ AND $x_3 > a_{3,1}$ AND $x_1 > a_{1,3}$ then Class is 1* and, like in the first case, the first inequality is overridden by the third one and the refined rule become *if $x_1 > a_{1,3}$ AND $x_3 > a_{3,1}$ then Class is 1*.

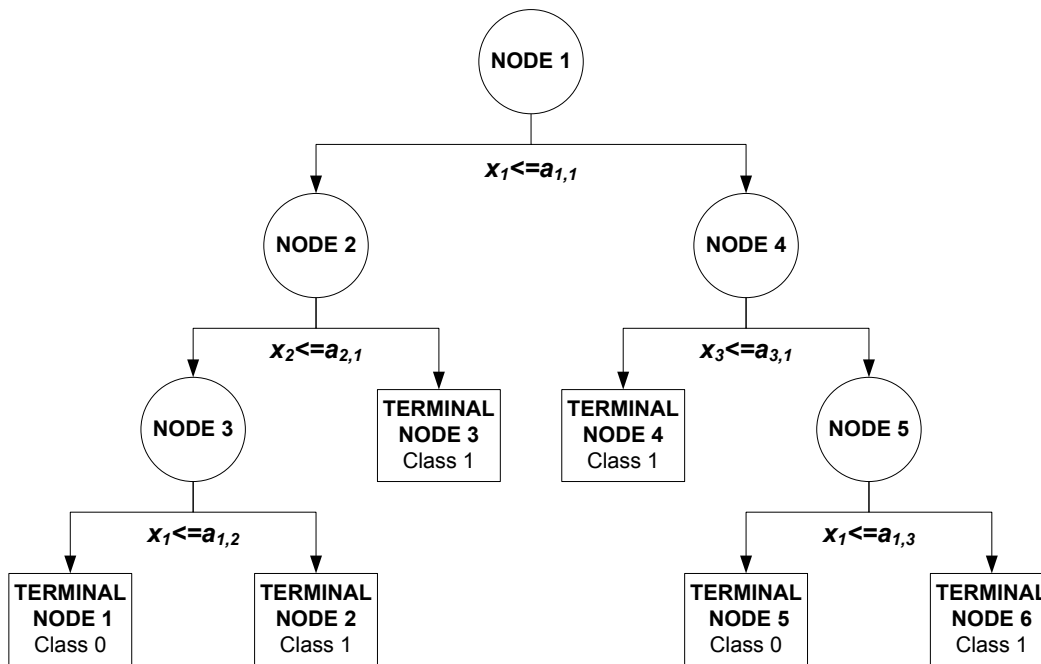


Figure 4: another example of classification tree with three explanatory variables and six terminal nodes.

Also in this case the output of the rules derived from the CART tree can be expressed as a matrix of the following form:

Table 4: tree rules matrix for the tree depicted in Figure 4.

	x_1	x_2	x_3	Class
Terminal Node 1	$x_1 \leq a_{1,2}$	$x_2 \leq a_{2,1}$	-	0
Terminal Node 2	$a_{1,2} < x_1 \leq a_{1,1}$	$x_2 \leq a_{2,1}$	-	1
Terminal Node 3	$x_1 \leq a_{1,1}$	$x_2 > a_{2,1}$	-	1
Terminal Node 4	$x_1 > a_{1,1}$	-	$x_3 \leq a_{3,1}$	1
Terminal Node 5	$a_{1,1} < x_1 \leq a_{1,3}$	-	$x_3 > a_{3,1}$	0
Terminal Node 6	$x_1 > a_{1,3}$	-	$x_3 > a_{3,1}$	1

The tree rules matrix so obtained is, then, used for both creating the membership functions for each input variable and for determining the fuzzy rules of the FIS.

3.2 From the tree rules to the fuzzy membership functions

According to Table 2, rules can be ranked depending on the number of total samples they are able to classify. Defining for a variable the membership functions is a sort of division of all the possible values a variable can assume, the so-called Universe of Discourse, in different classes of membership.

In order to create the proper membership functions a way to extrapolate the information contained in the tree rules has to be devised. To this end, all the portions of the rules involving the same variable, i.e. each column of the tree rules matrix, should be evaluated together.

The adopted procedure, aimed at coping with quite general cases, has been developed. On the basis of the rules which involve the same variable, i.e. a full column of the tree rules matrix, two functions (l_{ij} and h_{ij}) are created having the forms described in the following depending on the rule itself and the class to which the terminal node belongs to.

If the i -th terminal node belongs to class L, depending on the type of rule involving the variable j , the function l_{ij} is calculated as follows:

$l_{ij}(x_j) = \begin{cases} w_i : x_j \leq a_{jk} \\ 0 : x_j > a_{ij} \end{cases}$	$l_{ij}(x_j) = \begin{cases} w_i : x_j > a_{jk} \\ 0 : x_j \leq a_{jk} \end{cases}$	$l_{ij}(x_j) = \begin{cases} w_i : a < x_j \leq a_{jk} \\ 0 : x_j > a_{ij} \end{cases}$	$l_{ij}(x_j) = 0$ If no rule involves variable j
---	---	---	---

If the i -th terminal belongs to class H depending on the type of rule involving the variable j , the function h_{ij} is calculated as follows:

$h_{ij}(x_j) = \begin{cases} w_i : x_j \leq a_{jk} \\ 0 : x_j > a_{ij} \end{cases}$	$h_{ij}(x_j) = \begin{cases} w_i : x_j > a_{jk} \\ 0 : x_j \leq a_{jk} \end{cases}$	$h_{ij}(x_j) = \begin{cases} w_i : a < x_j \leq a_{jk} \\ 0 : x_j > a_{ij} \end{cases}$	$h_{ij}(x_j) = 0$ If no rule involves variable j
---	---	---	---

with $1 < i \leq N$ and $1 < j \leq M$ and $1 < k \leq P_j$

where

N is the total number of terminal nodes

M is the total number of predictors

P_j is the total number of different splitters for the j -th variable

x_j is the j -th variable

a_{jk} is the k -th splitting value of the j -th variable

w_i is a number comprised between 0 and 1 representing the ratio between the samples node i is able to classify and the total number of samples classified by all terminal nodes taken into consideration.

Once constructed the different $l_{ij}(x_j)$ and $h_{ij}(x_j)$ for the N terminal nodes, these functions are added up to obtain only two functions for each input variable:

$$L_j(x_j) = \sum_{i=1}^N l_{ij}(x_j) \text{ and } H_j(x_j) = \sum_{i=1}^N h_{ij}(x_j)$$

Then, fixed a threshold, the x_j values at which the functions $L_j(x_j)$ and $H_j(x_j)$ have an absolute variation above this threshold are determined. Two vectors **L** and **H** are, then, created to store the x_j values of the cross overs and the absolute values of the corresponding variations in L_j or in H_j .

Now a criterion to construct the membership, capable of coping with the very different configuration the L and H functions may assume, is needed. In this perspective, it is required to keep in mind that 4 parameters are required in order to define the trapezoidal membership function as described in equation 1 and in Figure 1.

In general, membership functions which we need to reproduce are of the following three forms, illustrated in Figure 5:

$$\begin{cases} \mu_{Fj}^1 = [MIN & MIN & \beta - \delta & \beta + \delta] \\ \mu_{Fj}^2 = [\beta - \delta & \beta + \delta & \alpha - \delta & \alpha + \delta] \\ \mu_{Fj}^3 = [\alpha - \delta & \alpha + \delta & MAX & MAX] \end{cases}$$

where MIN and MAX are the minimum and maximum value of the j-th variable respectively. These three functions represent the general three forms that the mf can assume in the present version of our algorithm for the automatic formulation of a FIS from a CART tree.

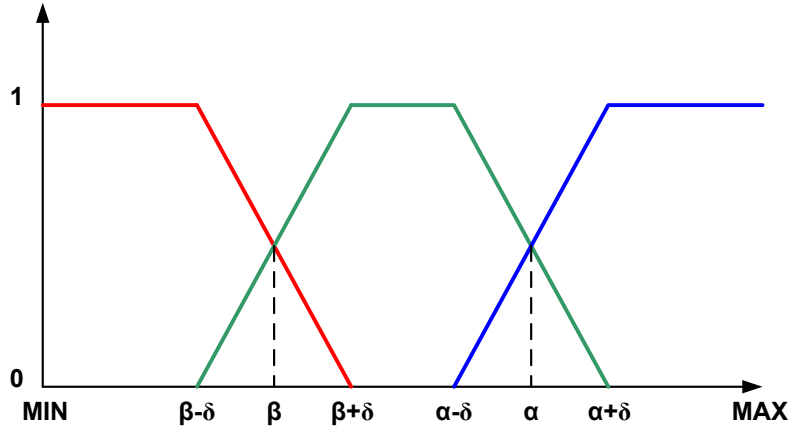


Figure 5

Several possibilities can arise:

1. The two **L** and **H** vectors are empty.
This could happen in two cases: whether the j-th variable does not appear in the tree or it appears only in the least important rules which discriminate very few samples. In this case, in order to maintain a similarity with the full construction of the FIS, three fake membership functions are created with the following form:

$$\begin{cases} \mu_{Fj}^1 = [0 & 0 & 0.9 & 1.1] \\ \mu_{Fj}^2 = [0.9 & 1.1 & 1.9 & 2.1] \\ \mu_{Fj}^3 = [1.9 & 2.1 & 3 & 3] \end{cases}$$

2. One of the two vectors **L** and **H** is empty.

There are two sub cases:

- a. The non empty vector has just one element. In this case, two memberships would be enough but, in order to maintain a standard in all the membership definition we devise three membership starting from the single element. Calling α this element, an interval δ is calculated considering the values 5% below the maximum the variable can assume and then three memberships are defined in the following way:

$$\begin{cases} \mu_{Fj}^1 = [MIN \quad MIN \quad \alpha - \delta \quad \alpha + \delta] \\ \mu_{Fj}^2 = [\alpha - \delta \quad \alpha \quad \alpha \quad \alpha + \delta] \\ \mu_{Fj}^3 = [\alpha - \delta \quad \alpha + \delta \quad MAX \quad MAX] \end{cases}$$

- b. The non empty vector has two or more elements. If this vector has just two elements, called α and β , with $\alpha > \beta$, δ is calculated as in the previous case and the membership are defined in the following way:

$$\begin{cases} \mu_{Fj}^1 = [MIN \quad MIN \quad \beta - \delta \quad \beta + \delta] \\ \mu_{Fj}^2 = [\beta - \delta \quad \beta + \delta \quad \alpha - \delta \quad \alpha + \delta] \text{ if } \beta + \delta < \alpha - \delta, [\beta - \delta \quad \beta \quad \alpha \quad \alpha + \delta] \text{ otherwise} \\ \mu_{Fj}^3 = [\alpha - \delta \quad \alpha + \delta \quad MAX \quad MAX] \end{cases}$$

The case with more than 2 elements can be reduced to the case with two elements just considering the two values which have the highest absolute value of the variation in L_j or in H_j .

3. The two vectors have one element each. Calling the two elements α_L and α_H and assuming without lack of generality that $\alpha_L < \alpha_H$, the membership are defined in the following way:

$$\begin{cases} \mu_{Fj}^1 = [MIN \quad MIN \quad \alpha_L - \delta \quad \alpha_L + \delta] \\ \mu_{Fj}^2 = [\alpha_L - \delta \quad \alpha_L + \delta \quad \alpha_H - \delta \quad \alpha_H + \delta] \text{ if } \alpha_L + \delta < \alpha_H - \delta, [\alpha_L - \delta \quad \alpha_L \quad \alpha_H \quad \alpha_H + \delta] \text{ otherwise} \\ \mu_{Fj}^3 = [\alpha_H - \delta \quad \alpha_H + \delta \quad MAX \quad MAX] \end{cases}$$

4. One of the two vectors has one element while the other two. Let's assume, without lack of generality, that the vector calculated from L_j has one element, α_L , while the other has two elements, α_H and β_H with $\alpha_H > \beta_H$. In this case, the two values for devising the membership are selected in the following way:

$$\begin{cases} \beta = \min(\beta_L, \beta_H) \\ \alpha = \max(\alpha_L, \alpha_H) \end{cases}$$

The membership functions are then calculated like in case 2b.

5. Both vectors have two or more elements. Considering the case both vectors have two elements and lets call them α_L and β_L with $\alpha_L > \beta_L$ and α_H and β_H with $\alpha_H > \beta_H$ respectively. The values for devising the membership are selected in the following way

$$\begin{cases} \beta = \min(\beta_L, \beta_H) \\ \alpha = \max(\alpha_L, \alpha_H) \end{cases}$$

and the membership are calculated as in case 2b.

The cases in which the vectors \mathbf{L} and \mathbf{H} have more than two elements can be reduced to the case with two elements just considering the two elements which have the highest absolute value of the variation in L_j and in H_j .

Figure 6 and Figure 7 report two examples of derivation of the membership function through this technique. In particular Figure 6 is an example of case 3, with just one element for each vector and $\alpha_s < \alpha_L$. Figure 7, instead, depicts an example of case 5 with two elements for each vector and $\beta = \beta_L$ and $\alpha = \alpha_H$.

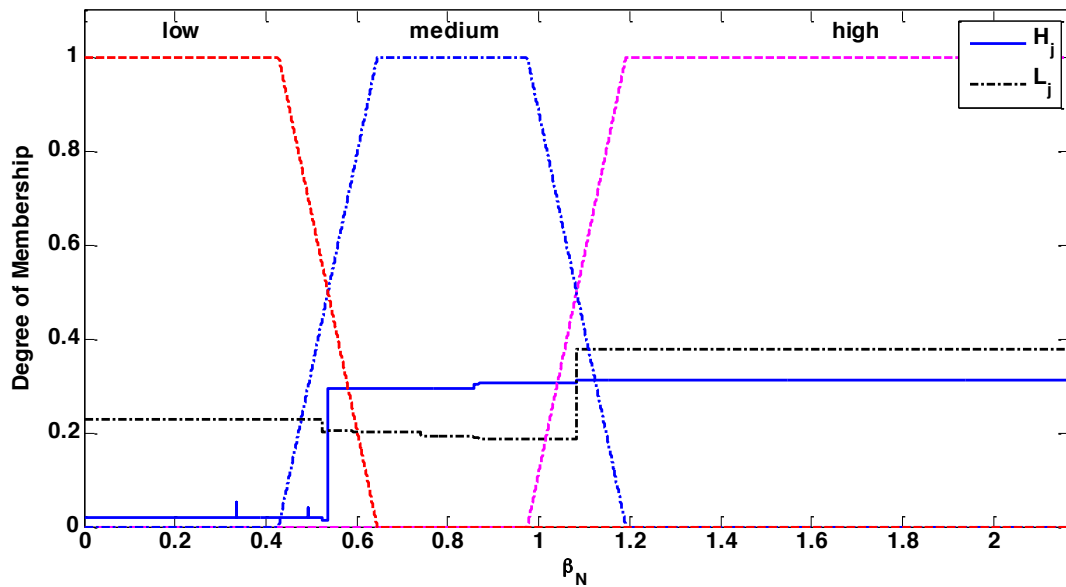


Figure 6: example of case 3 with $\alpha_s < \alpha_L$

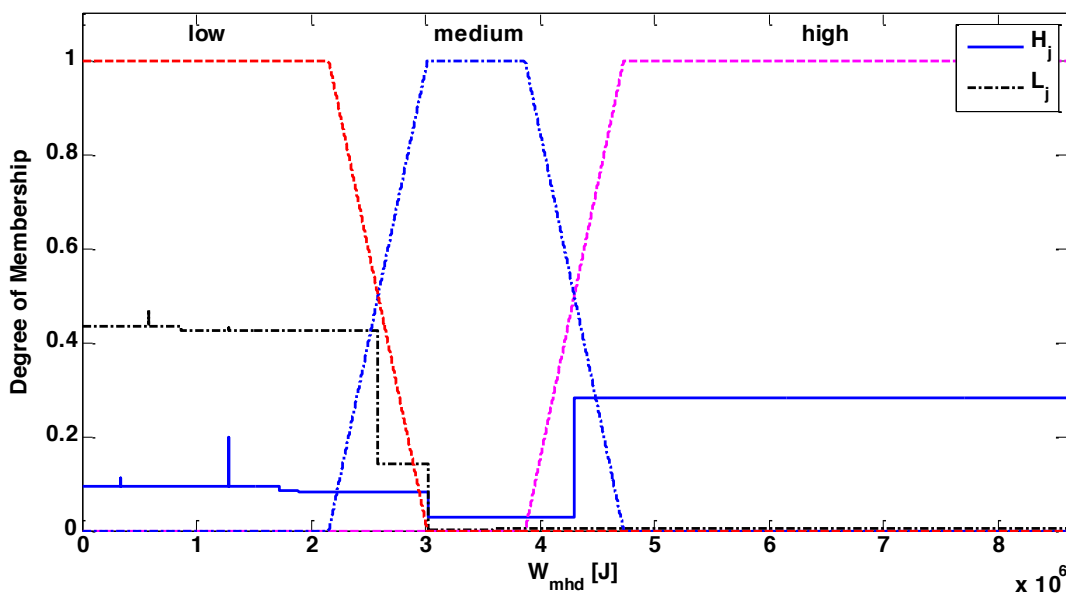


Figure 7: example of case 5. In this case the vectors have two elements.

3.3 From the tree crisp rules to the fuzzy rules

Once the membership functions are defined, to complete the FIS it is required to define the fuzzy rules. A fuzzy rule in the form *if...then...* is composed of an *antecedent*, the part between *if* and *then*, and the *consequent*, the part after the *then*.

Also in this case we start from the tree rules matrix. Each rule has already a form similar to a fuzzy rule apart from the fact that the antecedent is composed of inequalities using crisp values. We have, then, to translate the inequalities of the tree rules for the terminal nodes into inequalities based on the membership functions defined in the previous section.

To perform this step, two different cases have to be considered, depending on the form of the rule related to variable x_j :

1. the crisp inequality of the CART tree is in the form $x_j \leq a_{jk}$ or $x_j > a_{jk}$;
2. the crisp inequality of the CART tree is in the form $a_{j,k-1} < x_j \leq a_{jk}$

In case 1, we evaluate the membership which has the highest degree of activation for the splitting value a_{jk} and we can distinguish different subcases, depending on the membership function activated and the kind of inequality:

- a. if $\mu_{F_j}^1$ and \leq , the antecedent for variable x_j is *if x_j is $\mu_{F_j}^1$* ;
- b. if $\mu_{F_j}^1$ and $>$, the antecedent for variable x_j is *if x_j is not $\mu_{F_j}^1$* ;
- c. if $\mu_{F_j}^2$ and \leq , the antecedent for variable x_j is $\begin{cases} \text{if } x_j \text{ is } \mu_{F_j}^1 & \text{when } a_{jk} \leq (\alpha + \beta) / 2 \\ \text{if } x_j \text{ is not } \mu_{F_j}^3 & \text{when } a_{jk} > (\alpha + \beta) / 2 \end{cases}$;
- d. if $\mu_{F_j}^2$ and $>$, the antecedent for variable x_j is $\begin{cases} \text{if } x_j \text{ is not } \mu_{F_j}^1 & \text{when } a_{jk} \leq (\alpha + \beta) / 2 \\ \text{if } x_j \text{ is } \mu_{F_j}^3 & \text{when } a_{jk} > (\alpha + \beta) / 2 \end{cases}$;
- e. if $\mu_{F_j}^3$ and \leq , the antecedent for variable x_j is *if x_j is not $\mu_{F_j}^3$* ;
- f. if $\mu_{F_j}^3$ and $>$, the antecedent for variable x_j is *if x_j is $\mu_{F_j}^3$* .

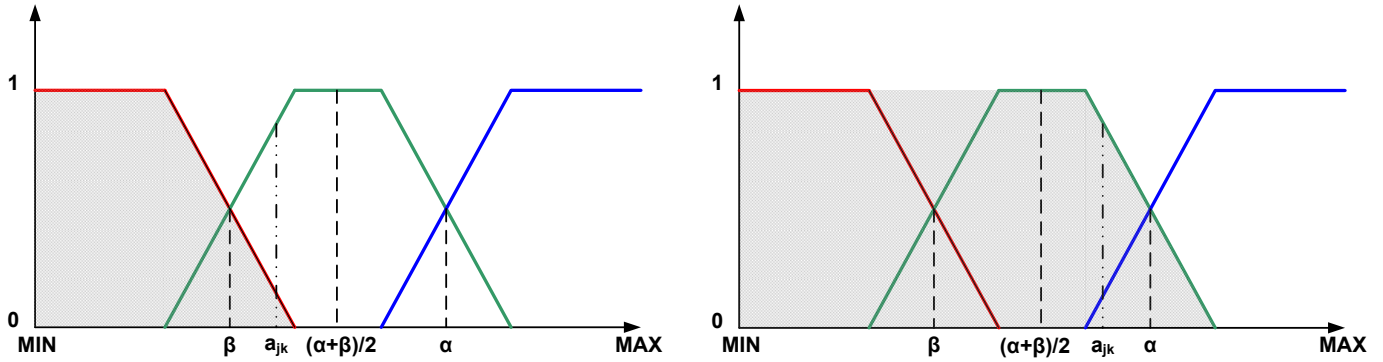


Figure 8: Case 1c. The shaded parts of the figure represents the antecedent chosen for variable x_j according to the value of a_{jk} .

In case 2 we evaluate the membership which has the highest degree of activation for both the splitting values $a_{j,k-1}$ and a_{jk} . We define $\mu_{F_j}^{k-1}$ and $\mu_{F_j}^k$ the membership functions which are activated by the two splitting values respectively and $m_{F_j}^{k-1}$ and $m_{F_j}^k$ the activations of each membership between $[0:1]$. Also in this case we can distinguish different alternatives:

- a. if $\mu_{F_j}^{k-1} = \mu_{F_j}^k$ the antecedent for variable x_j is *if x_j is $\mu_{F_j}^k$* ;
- b. if $\mu_{F_j}^{k-1} = \mu_{F_j}^1$, $m_{F_j}^{k-1} \neq 1$ and $\mu_{F_j}^k = \mu_{F_j}^2$ the antecedent for variable x_j is *if x_j is $\mu_{F_j}^2$* ;
- c. if $\mu_{F_j}^{k-1} = \mu_{F_j}^1$, $m_{F_j}^{k-1} \neq 1$, $\mu_{F_j}^k = \mu_{F_j}^3$ and $m_{F_j}^k \neq 1$ the antecedent for variable x_j is *if x_j is $\mu_{F_j}^2$* ;
- d. if $\mu_{F_j}^{k-1} = \mu_{F_j}^1$, $m_{F_j}^{k-1} \neq 1$, $\mu_{F_j}^k = \mu_{F_j}^3$ and $m_{F_j}^k = 1$ the antecedent for variable x_j is *if x_j is not $\mu_{F_j}^1$* ;
- e. if $\mu_{F_j}^{k-1} = \mu_{F_j}^1$ and $m_{F_j}^{k-1} = 1$ the antecedent for variable x_j is *if x_j is not $\mu_{F_j}^3$* ;
- f. if $\mu_{F_j}^{k-1} = \mu_{F_j}^2$ and $m_{F_j}^k \neq 1$ the antecedent for variable x_j is *if x_j is $\mu_{F_j}^2$* ;
- g. if $\mu_{F_j}^{k-1} = \mu_{F_j}^2$ and $m_{F_j}^k = 1$ the antecedent for variable x_j is *if x_j is not $\mu_{F_j}^1$* .

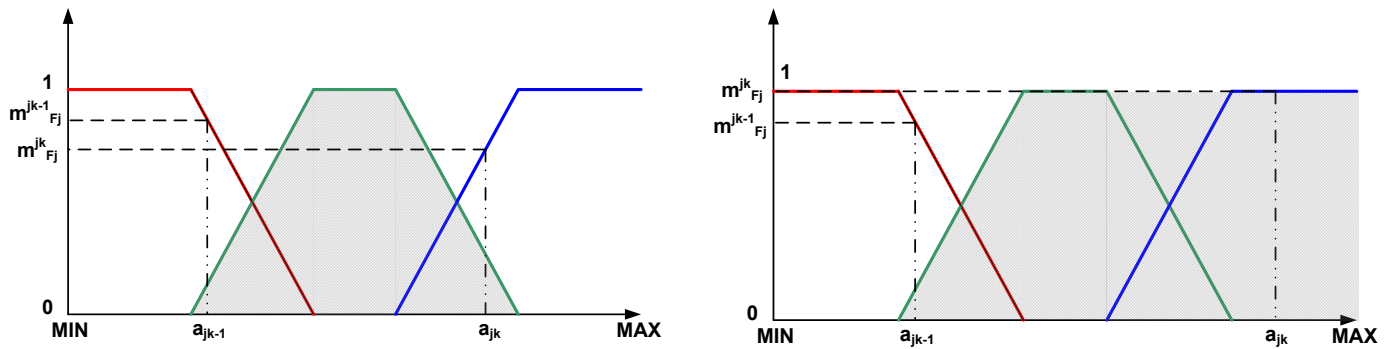


Figure 9: Case 2c and 2d. The shaded parts of the figure represents the antecedent chosen for variable x_j according to the value of a_{jk} .

4 Performance of the automatically generated Fuzzy Logic Inference system when applied to the L-H transition in JET

Following the steps described in the previous section, three fuzzy inference systems were built starting from the three different trees developed using CART and described in section 2.

Since the total number of rules is affected by the number of terminal nodes of the related tree, the complexity of the produced FIS depends on the number of nodes retained in the CART trees. Since, as shown in Table 2, some of the terminal nodes discriminate a very limited number of samples, the corresponding rules may introduce an excessive increase in complexity compared to the additional discrimination capability provided by them. Therefore, an investigation of the number of terminal nodes, and consequently number of rules, which provide the best results in term of correct classification of samples has been performed. At the same time, an investigation of the threshold value which maximise the output performance has also been performed.

Figure 10 reports the results of the above mentioned investigations executed on a test set of 17 pulses and a total of 234429 samples, represented by a surface plot of the performance as a function of the numbers of nodes and the threshold value.

Looking at Figure 10, it can be noticed that with a high number of terminal nodes and, consequently, an high number of rules in the corresponding FIS, the performance tends to be low and the best threshold value is 0.5. This suggests that too many rules may not necessarily increase the discrimination capability of the system. On the contrary, being some of them very specific and related to a small subset of the data, they may cause overfitting and move the output of the network to a small neighbourhood of 0.5. When moving towards a smaller number of nodes, the performance raises and tends to flatten both in the nodes and threshold direction. The flattening in the nodes direction may be caused by the data provided as test samples. Indeed, being the test data represented by whole pulses data, the majority of them are samples distant from a transition. Therefore, L and H samples present very different characteristics in terms of values in the space of parameters so that only a small number of rules is activated even if an higher number is present. The same reason can explain the flattening in the threshold direction. Being the data mainly represented by samples distant from a transition they are well separated by the systems.

To confirm these considerations, the three FIS were tested on a subset of the full test data. In particular they were tested on data taken from the specific interval that were used for training them. So the fuzzy inference system developed starting from the classification tree built using data around the L→H transition (from now on referred to as LH-FIS) was tested on data around the L→H transition, the FIS developed using data in the neighbourhood of H→L transition (HL-FIS) with data around the H→L transition and the FIS devised starting from data far from the transition (referred to as SS (steady-state)-FIS) with data far from the transition. The results of this tests are reported in Figure 11 a), b) and c) respectively. While the results achieved by the SS-FIS (Figure 11.c) are comparable with Figure 10.c although with an higher value of percentage of success, being the data similar but without the samples in the more uncertain region of L→H and H→L transition, the performance surface of LH-FIS and HL-FIS are slightly different. It is

possible to observe (Figure 11.a-b) that the flattening of the surface at small node numbers is no more present and the maximum performance are achieved with 7 and 11 nodes respectively (see Table 5 **Error! Reference source not found.**) indicating that more information is required to distinguish between L and H mode samples near to a transition.

Table 5 resumes the above mentioned results in a numerical way, showing the maximum percentage of success for the various developed FIS, together with the number of nodes taken into account to build the FIS and the threshold value which best discriminate between L and H mode samples.

Looking at the maximum percentages of success achievable by the various FIS, it can be noticed that while SS and LH are comparable, HL exhibits significantly lower performance. This can be due to a greater uncertainty in the H→L transition times contained in the database and, therefore, in a more uncertain classification of the samples in the neighbourhood of the H→L transition. It is, indeed, a well known fact that the H→L is less defined and more difficult to pin point with the measurements available. This uncertainty can lead to both mislearning during the training phase and a wrong estimation of the results during the testing phase.

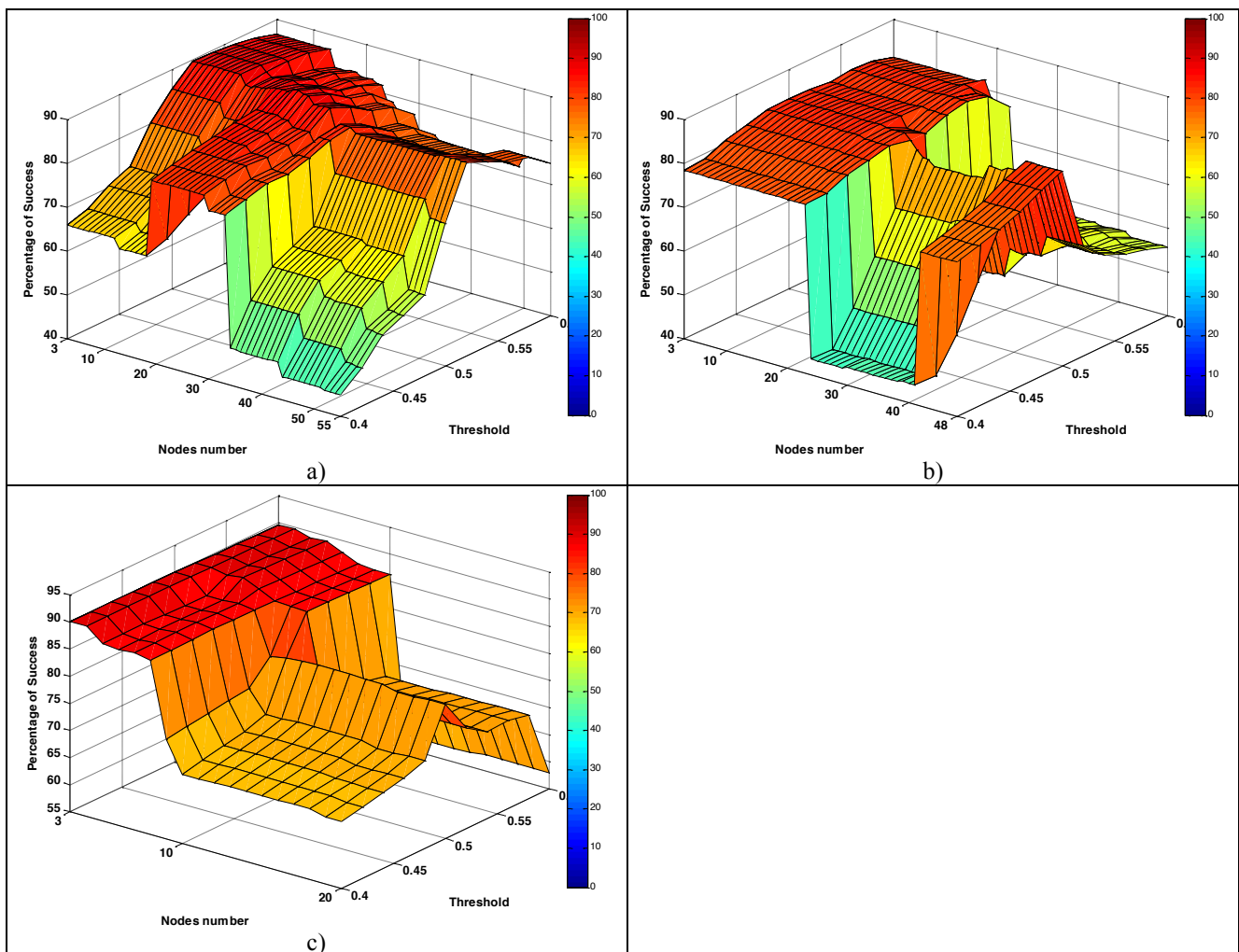


Figure 10: performance of the FIS generated from the CART data on full test data as a function of the number of terminal nodes taken into account and the threshold chosen to discriminate between the L and H mode. a), b) and c) are the results when the FIS is trained using data near the L→H transition, near the H→L transition and in steady-state condition respectively.

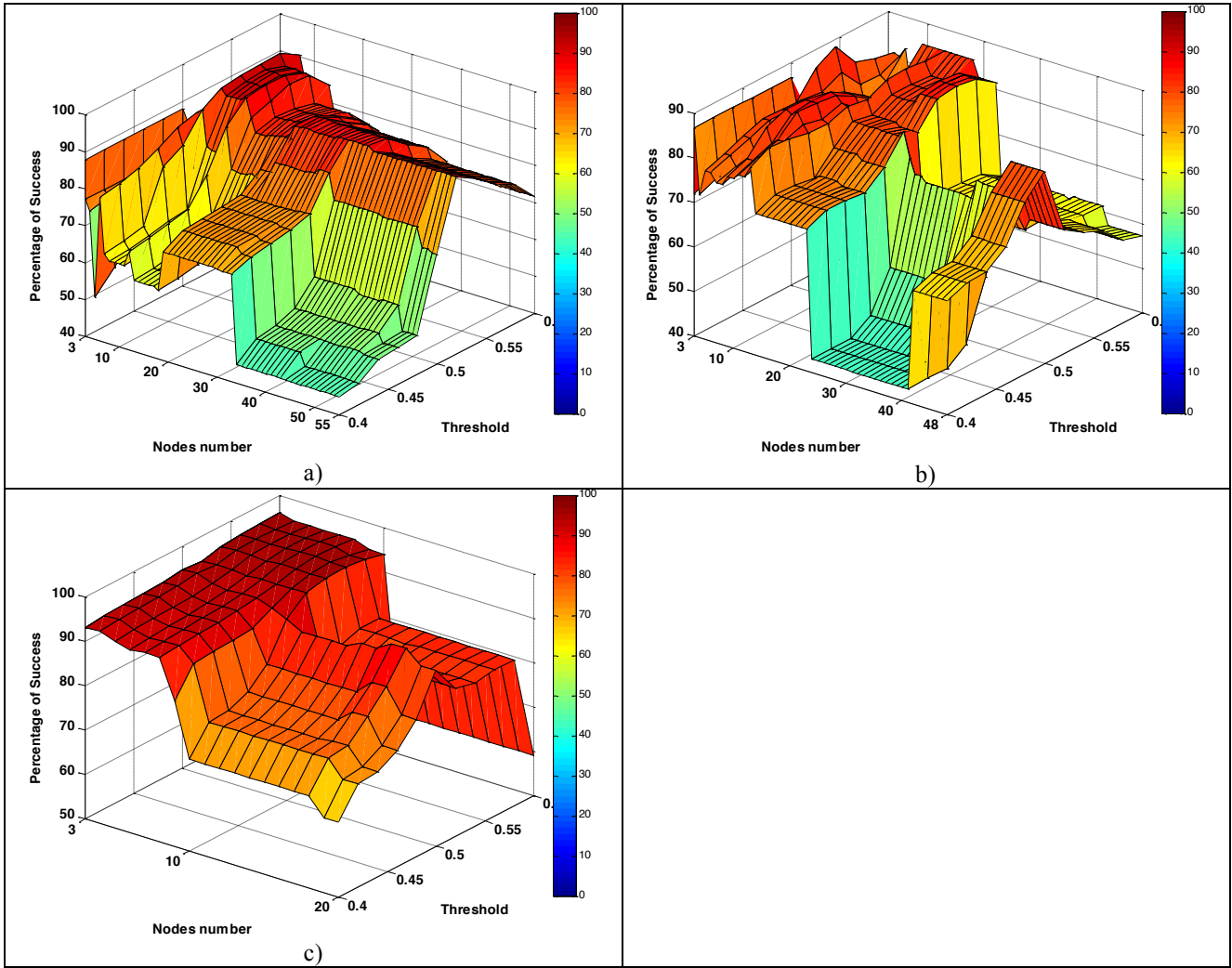


Figure 11: performance of the FIS generated from the CART data on specific test intervals as a function of the number of terminal nodes taken into account and the threshold chosen to discriminate between the L and H mode. a), b) and c) are the results when the FIS is trained and tested using data near the L→H transition, near the H→L transition and in steady-state condition respectively.

Table 5: performance on full test set and specific test sets

	Full Test (234429 samples)			Specific Test set (~17000)		
	%	thr	nodes	%	thr	nodes
L→H	89.66	0.52	3	93.58	0.58	7
H→L	84.31	0.5	42	87.45	0.48	11
Steady-state	90.13	0.44	4	96.14	0.56	3

5 Future developments

From the methodological point of view, the described method could be further developed by first of all increasing the number and types of membership functions. This would required also the determination of more sophisticated methods to derive the fuzzy rules.

In terms of applications, it would be very interesting to see how the approach performs when dealing with other classification problems like ELM discrimination and the prediction of disruptions.

References

- [1] Wagner, F. et al., "Regime of improved confinement and high beta in neutral beam heated divertor discharges of the ASDEX tokamak", *Physical Review Letters* 49, 1408 (1982).
- [2] P. Franzen et al., "On-line confinement regime identification for the discharge control system at ASDEX Upgrade", *Fusion Technol.*, vol. 33, Jan. 1998.
- [3] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, "Classification and Regression Trees" Chapman & Hall (Wadsworth, Inc.): New York, 1984.
- [4] L.A. Zadeh, "Fuzzy sets", *Infor. Control*, 8, pp.338-353, 1965.
- [5] B. Kosko, *Neural Network and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs, NJ(1992)
- [6] E. H. Mamdani and S. Assilian, "An experimenti in linguistic synthesis with a Fuzzy Logic controller", *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.