A Visual Approach for Fuzzy Rule Induction

Sergio R. Cuesta, Ignacio Díaz, Abel A. Cuadrado and Alberto B. Diez
Área de Ingeniería de Sistemas y Automática
Universidad de Oviedo, Campus de Viesques. Ed. Dep. 2
33204 Gijón, Spain
Emails: srcuesta@hotmail.com, {idiaz,cuadrado,alberto}@isa.uniovi.es

***Abstract*—Models are descriptions of real facts that serve us to think and reason. In building models, a compromise always exists between accuracy, that is, how precisely the model describes reality, and simplicity, without which the model would be useless. So, a good model must be simple and intuitive while being accurate enough. In this paper we propose a novel approach based on visual techniques aiming to help the human in *fine-tuning* fuzzy decision trees to enhance its interpretability and insightfulness with a minimal loss of accuracy. By involving the human in the design process, these techniques allow to include prior knowledge in the selection of membership functions as well as to assess the significance of rules in the model to help in the pruning stage.**

I. INTRODUCTION

Today problems in many research areas such as bioinformatics, social sciences or industrial engineering involve the analysis of complex processes or phenomena for which we lack accurate models. Commonly, information is present in quite heterogeneous ways, while classical models, in general, are only able to deal with only one type of substrate for knowledge.

Particularly, in factory automation we often face increasing market requirements both in terms of product quality and productivity that demand a continuous improvement in the production process.

In complex processes such as, for instance, industrial processes, we lack precise models which describe the influence of all factors and process optimization has to be done on the basis of –lots of– historical data, coming from dozens of sensors and typically some non-structured prior knowledge, available in terms of more or less precise rules, correlations, cases, etc. We *do* have prior knowledge, but this knowledge is only available in non-structured, subtle ways [1].

Visual exploration of complex and heterogeneous information resources has become recently a major topic of research [2], [3], [4], [5]. The idea of the so called *visual data mining* paradigm [6] is to exploit the human's mind visual exploration and reasoning abilities. In using a visual representation for knowledge, data visualization techniques are specially helpful to the human for making decisions or understanding a given problem by combining in a visual manner quite different sources of knowledge he/she may have at hand, such as process data, rules –that may be vague, fuzzy– simple models explaining parts of the process, etc.

Another major substrate of knowledge is the linguistic one. The human being also thinks using words, and many of the models he makes to explain things are linguistic. Particularly, in a factory one can find that most technicians deal with linguistic models to describe process, such as

- *High armature voltage and low field current lead to very high speed*

The theory of fuzzy sets, developed by Zadeh [7] allows to formalize linguistic entities in a mathematical way suitable for use with computers. A vast amount of research has been done since then in methods for control, modeling and knowledge discovery using these ideas and particularly for building linguistic models from available data as, for example, decision trees (DT) [8], [9], [10].

However, these methods are *common-sense blind*, i.e., they provide us with a model which may be accurate, but complicated or with no sense for us. As long as a model aims to serve as a vehicle of knowledge that we will handle to operate with or to devise new knowledge, it must be essentially *simple* –but not trivial– in order to be useful.

This idea has suggested to place some constraints –pruning techniques– in the final model to reach a compromise between simplicity and accuracy. However, automatic algorithms to prune decision trees are not always provided with a measure of insightfulness of the final model and they still work blindly.

In this paper, we propose the use of the visual paradigm to improve the fuzzy rule induction process by integrating the human in this process, allowing him/her to interact on the basis of a visual representation of data, rules and knowledge.

This paper is organized as follows. In section II, a brief review of decision trees and fuzzy decision trees is given. Section III introduces some visual techniques aiming to help the human in *fine-tuning* the fuzzy decision tree to enhance its interpretability and insightfulness with a minimal loss of accuracy. Finally, in section IV, we apply the proposed ideas to analyze vibration and current data from rotating machinery under different fault conditions in order to illustrate the effectiveness of using this approach. Finally, section V concludes the paper.

II. DECISION TREES

A. Classic decision trees

Decision trees (DT) are tree structures geared to solving classification problems. DT define a bank of rules using a set of input attributes of the sample data set in the antecedent and a class label –output or decision attribute– in the consequent.

0-7803-7937-3/03/$17.00 ©2003 IEEE

A DT is composed of *nodes*, which contain subsets of the initial data set, and *arrows* connecting the nodes. Each node –parent node– can be further split into several child nodes each of which fulfill an additional condition imposed on a given attribute. These conditions are represented by arrows connecting the parent node to each of its children. Thus, the leaves of the tree are associated to finite collections of the original data set fulfilling all the conditions defined by the path that connects them to the root.

A DT is grown starting from the most meaningful attribute and proceeding with other attributes at lower nodes of the tree. The objective is to produce nodes of the highest homogeneity. To achieve that, a recursive partitioning routine selects one attribute at a time, usually the one which maximizes some information measure. This attribute is used to split the node, using domain values of the attribute to form additional conditions leading to subtrees. Then, the same procedure is recursively repeated for each child node. A node is further split unless all attributes are exhausted or when all examples at the node have the same classification. Alternatively, other criteria can be imposed –for instance, requiring a minimum number of elements in a leaf– to stop the growth. This approach to decision tree construction corresponds to a top-down greedy algorithm that makes a locally optimal decision in each node.

Quinlan's ID3 [9], [8], and its extension to deal with continuous domains, C4.5 [10], are two of the most popular of such algorithms. ID3 aims at knowledge comprehensibility and it is based on symbolic domains. On the other hand, C4.5 does not require prior partitioning. The conditions in the tree are based on thresholds (for continuous domain), which are dynamically computed. Because of that, the conditions on a path can use a given attribute a number of times (with different thresholds), and the thresholds used on different paths are very likely to differ. Moreover, the number of possible thresholds equals the number of training examples. These ideas often increase the accuracy of the tree but they reduce its comprehensibility.

In the ID3 algorithm, the recursive tree building can be described as follows:

1) Compute the information content in a node N,

$$I_N = -\sum_{k=1}^{|C|} p_k^N \cdot \log(p_k^N) \tag{1}$$

where C is the set of all possible output classes, and p_k^N is the probability –estimated from data– that an example found present in the node has a given classification $k \in C$.

2) For each remaining input attribute a_i –previously unused on the path to N–, compute the information gain, G_i, based on using this attribute to split the node N as,

$$G_i = I_N - \sum_{j=1}^{|D_i|} w_j \cdot I_{N_j} \tag{2}$$

where D_i denotes the set of categories associated to a_i, I_{N_j} is the information content within the j^{th} child of N, and w_j is the proportion of examples in node N that satisfies the condition leading to N_j.

3) Expand the node using the attribute which maximizes the gain.

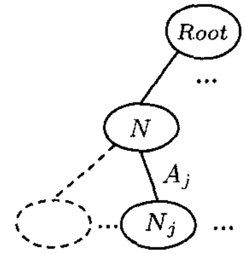

Fig. 1. Node expansion during the tree growing

The recursive partitioning method often leads to very complex tree structures that 'overfit the data', often showing a poor generalization (test) performance. One of the challenges in decision tree induction is to develop algorithms that produce decision trees of small size and depth, while being still accurate. To achieve that, pruning algorithms are used either during the construction (*prepruning*) –for instance establishing a set of stopping rules to avoid the growth of worthless branches with respect to prediction accuracy–, or after the tree has been constructed (*postpruning*). In both cases the idea is to remove branches with little statistical significance.

B. Fuzzy decision trees

Fuzzy decision trees (FDT) [11] aim at achieving high comprehensibility, normally attributed to ID3, with the ability to manage imprecise and vague information attributed to fuzzy systems.

While ID3 is based on classical crisp sets, so an example satisfies exactly one of the possible conditions out of a node, in FDT, in turn, an example can match more than one condition, for these conditions are now fuzzy restrictions based on fuzzy sets. Because of that, a given example can fall into many children nodes of a node and consequently into more than one leaf (inconsistency) with some degree of membership $\mu \in [0,1]$. Far from being a problem, this fact is actually advantageous, especially when dealing with noisy or incomplete information [12].

The tree-building routine follows that of ID3. In particular, the difference lies in the way the probabilities p_k^N are estimated.

Let's denote $\mu_N(\mathrm{e})$ as the accumulated membership for example e at the node N; this accumulated membership is computed by applying a defined t-norm, $\wedge$, through all the fuzzy conditions leading to the node N. Let also $\mu_A(\mathrm{e})$ be the membership of example e to a fuzzy set A.

Clearly, $\mu_{Root}(\mathrm{e}) = 1$, as all the examples completely belong to the root node, and $\mu_{N_j}(\mathrm{e}) = \mu_N(\mathrm{e}) \wedge \mu_{A_j}(\mathrm{e})$, where N_j is the j^{th} child of N, and A_j is the fuzzy term associated with the fuzzy restriction leading to node N_j – see fig 1.

Then, at node N the cardinality of the fuzzy linguistic class k can be computed as $\sum_{\mathbf{e}\in E} \mu_N(\mathbf{e}) \wedge \mu_{C_k}(\mathbf{e})$. Given that, the probability p_k^N, which now is a fuzzy measure of the fraction of examples in the whole data set, E, that simultaneously belong to C_k and node N, can be estimated as

$$p_k^N = \frac{\sum_{\mathbf{e}\in E} \mu_N(\mathbf{e}) \wedge \mu_{C_k}(\mathbf{e})}{\sum_{k'=1}^{|C|} \sum_{\mathbf{e}\in E} \mu_N(\mathbf{e}) \wedge \mu_{C_{k'}}(\mathbf{e})} \quad (3)$$

In the inference routine, when a new example $\mathbf{e}_j = (e_j^1, ...e_j^n)^T$ is presented for classification, we have to find leaves whose restrictions are satisfied by the example, and combine their decision into a single crisp response δ. Trying to cope with inconsistency a defuzzification method based on the *Weighted-Fuzzy-Mean* [13] is applied,

$$\delta = \frac{\sum_{i=1}^{|L|} \mu_{L_i}(\mathbf{e}) \cdot S_i}{\sum_{i=1}^{|L|} \mu_{L_i}(\mathbf{e}) \cdot s_i} \quad (4)$$

where L denotes the set of all the leaves of the tree, and S_l and s_l reflect the information contained in a leaf

$$S_i = \sum_{k=1}^{|C|} p_k^{L_i} \cdot \alpha_k \cdot \zeta_k \quad (5)$$

$$s_i = \sum_{k=1}^{|C|} p_k^{L_i} \cdot \alpha_k \quad (6)$$

where $i = 1, \cdots, |L|$, α_k denotes the area and ζ_k the centroid of the fuzzy set C_k.

Considering all that, fuzzy decision trees have some advantages with respect to the crisp ones:

- the ability to deal with fuzziness, the same in the input attributes as in the output classes, which provides a richer information about the case of study.
- the possibility of including prior knowledge about the data set when designing the fuzzy partitions of the attribute's domains.
- the information is presented in a very easy way for human understanding.

III. Visual Tools and Methods

In this section, we describe some procedures that allow the interaction of a human expert in the generation of a meaningful fuzzy rule set.

A. The Self-Organizing Maps

The Self-Organizing Map (SOM) method is a powerful algorithm for the visualization of high-dimensional data [14]. The SOM may be described as a nonlinear, ordered, and smooth mapping of high-dimensional input data domains onto the elements of a regular, low-dimensional (typically 2D) rectangular grid (also called *visualization space*). The map is defined by a set of points (*codebook vectors*), $\mathbf{m}_i$, in the input space and a corresponding set of nodes in a rectangular grid $\mathbf{g}_i$. The SOM training algorithm arranges the codebook vectors $\mathbf{m}_i$ so that they acquire the same geometry of the input data in a smooth and ordered fashion. The mapping tends to preserve *topological relationships* of the input data.

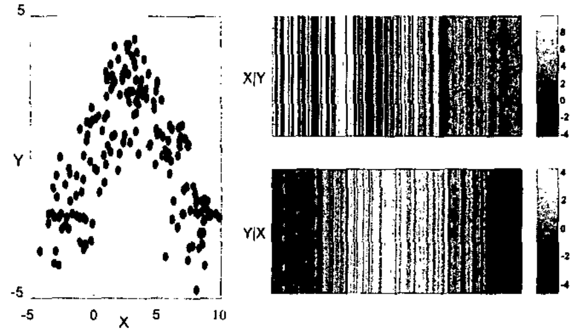

Fig. 2. Example of visual information

Particularly insightful are the so called *component planes*. The i^{th} component plane is a 2D image built on the grid by considering each node (neuron) as a pixel with a color level proportional to the value of the i^{th} coordinate of the corresponding codebook vector, $\mathbf{m}_i$. So, there are always n component planes available, each one corresponding to a component of the high dimensional input space. The visual information displayed in each plane is consistent with that of the rest of the planes. Thus, the simple visualization of each plane, provides the user with a big picture of the input values distribution.

B. Visual information

Another useful visualization technique is the so called *reorderable matrix* [15]. It is a suitable method to show the overall appearance of the data as a whole and not so much the individual quantitative values.

Here, a variation of this technique is proposed for visualizing the information utility of the attributes.

- First, an index vector is defined corresponding to the permutation generated by sorting one of the attributes in ascending order.
- Then, this index is applied to the attribute of interest; the one that is represented.

These two steps are repeated for each of the attributes. The objective is to observe if *order* in one attribute induces *order* in the attribute of interest. That reveals the meaningfulness of the attribute.

In the example of fig. 2, the structure of two variables X and Y is displayed in a scatter plot. In this case, if Y is the variable of interest, X carries useful information to describe Y and it

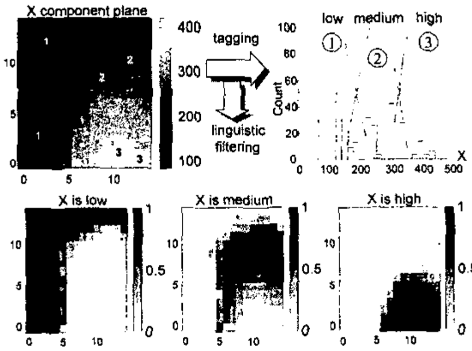

Fig. 3. Example of visual tagging: component plane and histogram (top) and fuzzy maps (bottom). In the fuzzy maps, dark gray represents high fulfillment of the fuzzy restriction

can be clearly seen in the bottom right image, where Y is ordered by X $(Y|X)$. Y can be described in a vague way like

- when X is *low*, Y is *low*
- when X is *intermediate*, Y is *high*
- when X is *high*, Y is *low*

By contrast, if the variable of interest is X, the upper image shows that Y is not useful to describe it.

This is a simple and effective way to reveal relations in the set of data that takes advantage of the human visual system.

C. Visual tagging

For building FDT, continuous attributes need to be partitioned into several fuzzy sets prior to the tree induction. It is a very important stage since the future structure of the tree depends on it. The use of the SOM is proposed as an insightful way to find collections of data that can be associated meaningfully. As explained before, the SOM preservation of topological relationships allows the visualization of clusters and relationships between the data. Thus, *component planes* constitute a powerful tool for discovering *categories* in the data and therefore they can be a useful backing to design the attribute's fuzzy partition.

By visual tagging, prior knowledge is used to label just the *interesting* clusters shown in the component planes in a rather simple way. The procedure consists of picking single elements (nodes of the map) that are representative of each of the fuzzy sets that set up the frame of cognition. The minimum and maximum values of the nodes labeled with a number i in the j^{th} component plane, delimit the flat zone of the i^{th} trapezoidal membership function associated to the j^{th} attribute. The overlapping width between these membership functions is the required for having a *fuzzy partition*[1] of the domain. The use of fuzzy sets helps to alleviate problems with

[1]Let $Q = \{A_1, \cdots A_k\}$ be a family of fuzzy sets on Ω. Q is a fuzzy partition of Ω when $\sum_{\gamma=1}^{k} A_\gamma(a) = 1,\ \forall a \in \Omega$

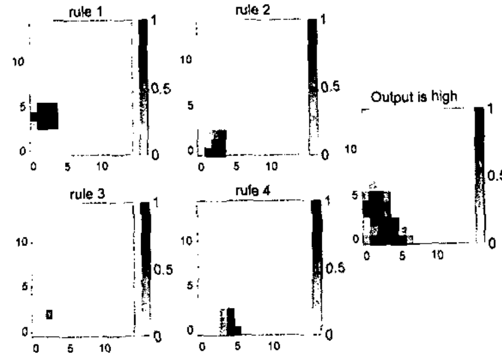

Fig. 4. Example of visual rule post-pruning: fuzzy maps of the rules (left) and consequent area covered by this subset of rules (right)

the classification of the elements in cluster's boundaries. Here, a balance between precision and simplicity has to be achieved by adjusting the level of specificity of the sets. High specificity usually yields accurate but meaningless set of rules.

To illustrate this, in fig. 3, a labeled component plane is shown with the family of membership functions generated by visual tagging and the histogram of the variable. The final domain partition can also be seen in the *fuzzy maps* [16] as a result of linguistic filtering [17]. The codebook vectors of SOM are *filtered* by each of the fuzzy sets obtaining the values of membership of each node of the SOM to the associated fuzzy term. As a result, color maps showing up these values are displayed in the same way as component planes.

D. Visual rule post-pruning

A decision tree can be rewritten to a collection of rules and there would be one rule for every leaf. The conditions leading to the leaf generate the conjunctive antecedent, and the classification of the examples of the leaf generates the consequent.

Once the fuzzy rule set is obtained, we must attempt to find irrelevant rules which can be deleted without affecting the relevance of the model constructed. Because of the inconsistency problem, rules has a disjunctive consequent so the set of all rules is divided into subsets that concern the same consequent category. Obviously, the same rule can belong to more than one subset with different degrees of membership. For each of these subsets, the related rules performance is summarized in fuzzy maps.

Fig. 4, shows the fuzzy maps of the four rules that make up the rule subset concerning one of the possible categories (high) of a consequent attribute (output). In the fuzzy maps of the rules, different gray intensities show different fulfillment degrees of the nodes to each rule antecedent (constituted by several fuzzy restrictions). The fuzzy map of the consequent is computed filtering the nodes of the consequent component plane by the fuzzy category covered by the rule subset under study. The worth of each rule in the subset is assessed just

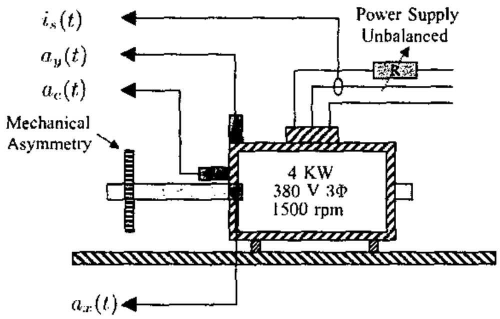

Fig. 5. Experimental test rig

looking into the maps:

- the *covering* of a rule, that is the number of cases (fuzzy count) that satisfies the rule's antecedent conditions. This is shown in the fuzzy map of the rule by the size of the colored zone.
- the *accuracy* of a rule is measured by the amount of gray pixels in the fuzzy map that are also colored in the consequent map filtered by the linguistic term covered by the rule subset.
- the *weight* of a rule within the subset is indicated by the intensity of the pixels. The more intensity, the more fulfillment of the colored nodes values to the rule. Then, the rule 3 in fig. 4 has a very little weight in the subset and does not make any new contribution to the subset as a whole so it can be removed.

All this makes up a powerful visual approach for fuzzy rule post-pruning.

IV. Results

The proposed approaches were tested through an experiment carried out on a 4 kW, 2 pole-pair asynchronous motor - see fig 5. Two kind of faults were induced to the motor: an *electrical fault* consisting of a power supply unbalance, provoked by the inclusion of a variable resistance on the phase R and a *mechanical fault* provoked by the presence of an asymmetric mass on the axis. Both types of faults produce different vibration patterns. Five sensors were installed in the motor: three vibration accelerometers and one current sensor

$$a_c(t), a_x(t), a_y(t), i_s(t).$$

Feature extraction was performed obtaining the harmonic content of these variables at 25Hz and 100Hz in the vibration accelerometers and at 50Hz in the current. A training set was composed of data from the four possible situations combining the electrical fault ($R = 0\Omega, R = \infty\Omega$) and the mechanical fault (with and without the asymmetric mass). The objective is the identification of the seven different states shown in table I using the signals provided by sensors.

TABLE I

Nº	State label	Resistance(Ω)	Mass
1	M-fault	0	Yes
2	ME-fault	∞	Yes
3	No-fault	0	No
4	E-fault5	5	No
5	E-fault10	10	No
6	E-fault15	15	No
7	E-fault20	20	No

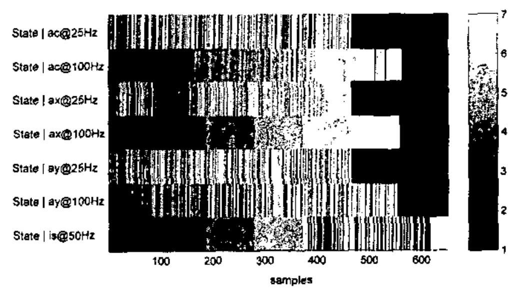

Fig. 6. Visual information about *state* provided by the attributes

1) Visual information: Fig. 6, represents the visual information about the attribute of interest, *state*, provided by the rest of the attributes. Thus, the *state* values have been ordered by each of the other attributes and displayed by rows in a gray scaled color matrix.
The attribute that best describes the *state* is *ax@100Hz*. It is clearly seen how the ordination of this attribute induces ordination in the attribute of interest. Therefore, *ax@100Hz* is the most meaningful attribute in the data set and it will be the one selected by the recursive tree-building algorithm to split the first node.

2) Visual tagging: Discretization in terms of fuzzy sets is easily done with the insightful data representation provided by the component planes shown in fig. 7.

3) Visual rule post-pruning: After training, we obtain a tree of 33 nodes. Then, we apply the C4.5 enhanced pessimistic

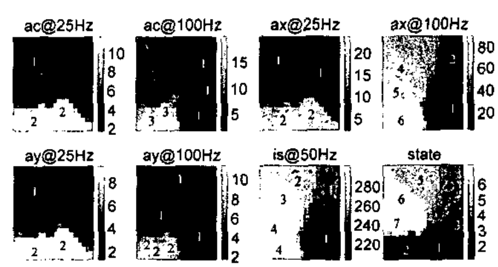

Fig. 7. Labeled component planes

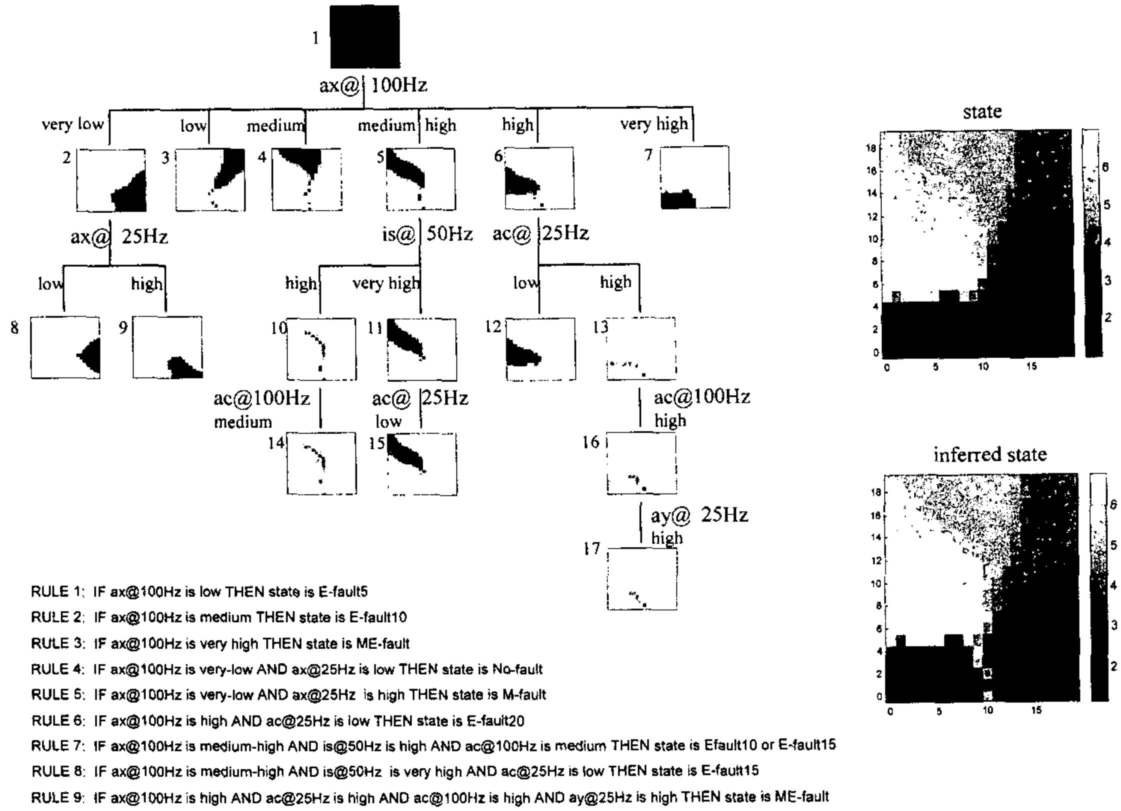

Fig. 8. Before visual post-pruning

post-pruning process [10] to obtain a pruned tree of 17 nodes without any reduction of the predicted error rate.

The final results are presented in fig. 8. A tree structure using fuzzy maps to represent the nodes shows the evolution of the partitioning of the original data set in terms of fuzzy *regions* through the branches of the tree. The decision-making inference procedure matches the new data values with the conditions associated to these regions and classifies the new data element. The result of this process is shown in the inferred map of the variable of interest where it is seen how the objective of identifying the different working states is achieved by the induced tree. We also show the set of fuzzy rules extracted from the tree in a textual form.

Despite its good inference performance, this tree structure has several nodes in which the covered area describes boundaries between the clusters and it has very little specific weight, shown by their low gray intensity. Using this visualization we can easily prune these nodes that have little significance for process understanding. As it can be seen in fig. 9, the result is a smaller tree describing just the interesting zones with an acceptable level of accuracy, as shown by the inferred map. All this leads to a simpler, more understandable and meaningful set of rules that describe the process.

V. Conclusion

In this paper we have proposed a novel approach to devise linguistic models of process from data using visual techniques to achieve a good compromise between accuracy and insightfulness. The key idea here is to involve the human in the design process of the linguistic model in two main fronts. First, through an interactive selection of the fuzzy sets, which are the "bricks" for building the model; and later, using SOM fuzzy maps to visualize the rules previously obtained in an automated way (FDT) helps in the postpruning process in order to simplify the model with a minimal loss of accuracy.

We have also shown the effectiveness of the proposed ideas through a real problem involving the analysis of vibration and current data from rotating machinery under different fault conditions.

In sum, we believe that visual approaches like that proposed in this paper reveal themselves as powerful ways to stress the role of prior knowledge as well as subjective aspects like *common sense*, *simplicity*, *insightfulness* or *comprehensibility* in the design process of a model.

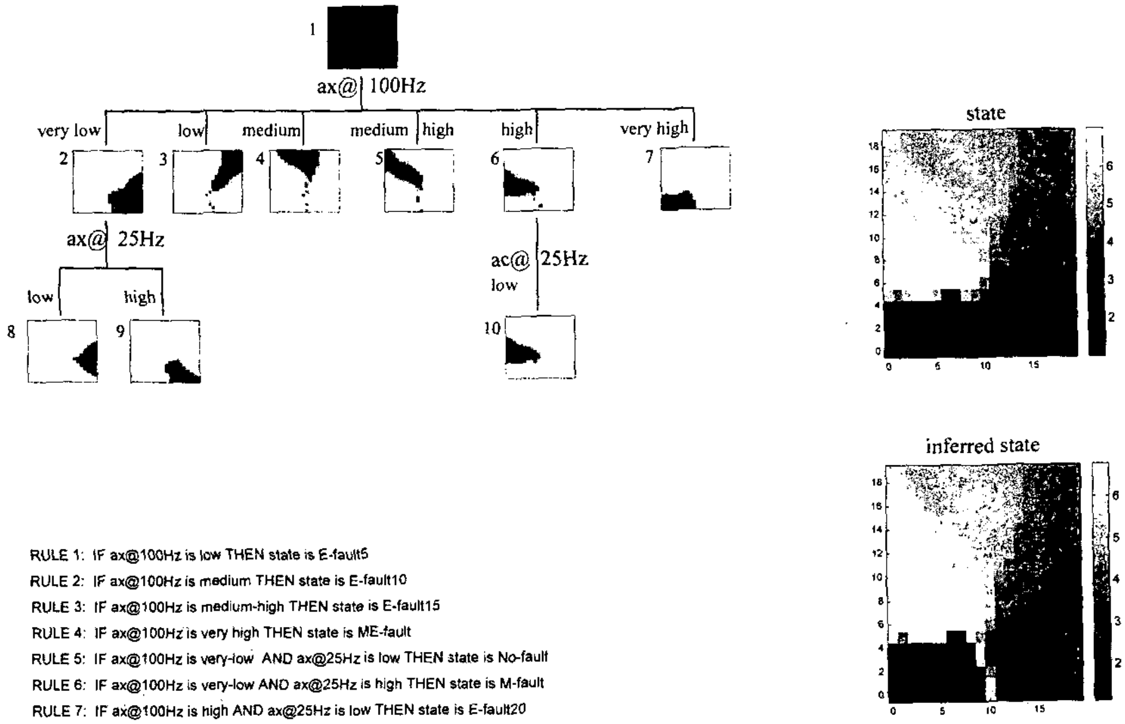

Fig. 9. After visual post-pruning

ACKNOWLEDGMENT

This work has been done within the framework of project DPI2002-01599 financed by the spanish *Ministerio de Ciencia y Tecnología* and FEDER funds (*Fondo Europeo de Desarrollo Regional*).

REFERENCES

[1] I. Díaz, A. B. Diez, A. A. Cuadrado, and M. Domínguez, "Prior knowledge integration in self organizing maps for complex process supervision," in *International Federation of Automatic Control* 15^{th} *IFAC World Congress*, (Barcelona, Spain), 2002.

[2] D. A. Keim, "Designing pixel-oriented visualization techniques: Theory and applications," *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, pp. 59–78, Jan.–Mar. 2000.

[3] D. A. Keim, "Visual exploration of large data sets," *Communications of the ACM*, vol. 44, pp. 39–44, Aug. 2001.

[4] D. A. Keim, "Information visualization and visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 1–8, Jan.–Mar. 2002.

[5] M. Kreuseler and H. Schumann, "A flexible approach for visual data mining," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 39–51, Jan.–Mar. 2002.

[6] P. C. Wong, "Visual data mining," *IEEE Computer Graphics and Applications*, September/October 1999.

[7] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[8] J. R. Quinlan, "Discovering rules by induction from large collections of examples," in *Expert Systems in the Micro Electronic Age* (D. Michie, ed.), pp. 168–201, Edinburgh University Press, 1979.

[9] J. R. Quinlan, "Induction of decision trees," in *Readings in Machine Learning* (J. W. Shavlik and T. G. Dietterich, eds.), Morgan Kaufmann, 1990. Originally published in *Machine Learning* 1:81–106, 1986.

[10] J. R. Quinlan, *C4. 5: Programs for Machine Learning*. San Mateo, CA: MK, 1993.

[11] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. Syst., Man, Cybern.*, vol. 28, pp. 1–14, 1998.

[12] L. Zadeh, "Fuzzy logic and approximate reasoning," tech. rep., Electronics Research Lab, UC-Berkeley, Nov. 1974.

[13] R. Jager, *Fuzzy Logic in Control*. PhD thesis, Department of Electrical Engineering, Delft University of Technology, 1995.

[14] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," *Proceedings of the IEEE*, vol. 84, pp. 1358–1384, Oct. 1996.

[15] J. Bertin, *Graphics and Graphic Information Processing*. Walter de Gruyter, 1981.

[16] A. A. Cuadrado, I. Díaz, A. B. Diez, F. Obeso, and J. A. González, "Fuzzy inference maps for condition monitoring with self-organizing maps," in *Second Conference of the European Society for Fuzzy Logic and Technology, EUSFLAT 2001*, (Leicester, UK), pp. 55–58, Sept. 2001.

[17] Hirota and Pedrycz, "Fuzzy computing for data mining," *PIEEE: Proceedings of the IEEE*, vol. 87, 1999.