Published in the volume "Information, Uncertainty and Fusion"
edited by B. Bouchon-Meunier, R.R. Yager and L.A. Zadeh, 2000, pages 131-145

On neurofuzzy and fuzzy decision tree approaches

Cristina OLARU, Louis WEHENKEL
University of Liège - Institute Montefiore
Sart-Tilman B28 B-4000 Liège BELGIUM
olaru@montefiore.ulg.ac.be, lwh@montefiore.ulg.ac.be

Abstract: In this paper two supervised learning approaches in the context of fuzzy information are presented, compared and combined : neurofuzzy systems based on multilayer feedforward neural networks and fuzzy decision trees based on top down induction. Neurofuzzy systems take advantage of fuzzy logic and of the learning and approximation capabilities of neural networks to build fuzzy rule bases, while preserving the massively parallel computing character. Fuzzy decision trees provide a way to manipulate fuzzy information and continuous input/output models while maintaining the interpretability and effectiveness of classical decision trees. A combination between the two ideas is discussed, which performs a global gradient-descent type optimization of the fuzzy rules corresponding to a tree, yielding improved accuracy while maintaining interpretability.

1 Introduction

From some real-world applications, precise data are difficult or expensive, if not impossible, to obtain. Usually, the information observed from a system is available either in the form of numerical (precise) information from measurements or computations and/or in the form of linguistic (imprecise) information from human experts. The theory of fuzzy logic was designed to exploit both kinds of information, providing a natural tool to model and process uncertainty and imprecision, being an interesting platform to cope with ambiguity present in human activity, especially decision processes.

Fuzzy models seem particularly well suited for a smooth transition from numerical (quantitative) to semantic (qualitative) information. Fuzzy rules have the additional advantage over classical production rules of allowing a suitable management of vague and uncertain knowledge. Using linguistic labels, they are more understandable for human experts and may be easily interpreted. The main functionality of linguistic variables is to provide a systematic means for an approximate characterization of complex or ill-defined phenomena [26].

The elasticity of fuzzy sets simplifies the task of translating between human reasoning, which is inherently elastic, and the rigid operation of digital computers. They also allow computers to use the type of human knowledge called common sense (knowledge in the form of statements that are usually, but not always, true) [27].

Fuzzy theory holds that all things are matters of degree [13]. Mathematically, fuzziness means multivaluedness or multivalence. Three-valued fuzziness corresponds to truth, falsity, and indetermination, or to presence, absence and ambiguity. A fuzzy subset A of a universe of discourse U is characterized by a *membership function* $\mu_A : U \rightarrow [0, 1]$ which associates with each element y of U a number $\mu_A(y)$ in the interval $[0, 1]$ which represents the grade of membership of y in A.

Fuzzy systems "reason" with parallel associative inference. When asked a question or given an input, a fuzzy system fires each rule in parallel, but to different degree, to infer a conclusion or output.

To train a fuzzy system, three learning methods are distinguished: supervised learning [2], [3],[4], [7], [9], [10], [11], [16], [17], [20], [21], [28], reinforcement learning [5], [14], [15] and unsupervised learning [12], [18], [24]. In supervised learning ("learning with a teacher"), for each input, the system receives the desired objective (output). In reinforcement learning ("learning with a critic"), the system gets a response, but it is not as direct, immediate and informative as in supervised learning (it is only an evaluative feedback from the environment, not an instructive signal). In unsupervised learning ("learning by observation") the correct response is not provided at all and the objects with similarities are grouped in clusters. We restrict our interest in this paper to two supervised learning approaches in the context of fuzzy information: neurofuzzy based on multilayer feedforward neural networks and fuzzy decision tree based on top down induction.

The term neurofuzzy (NF) stems from the use of heuristic learning strategies derived from the domain of neural network (NN) theory to support the development of a fuzzy system. A neurofuzzy system takes over the capabilities of neural nets to learn and generalize, as well as their massively parallel computing character; it inherits also their "universal approximation capabilities". On the other hand, it may take advantage of fuzzy logic to represent, manipulate and utilize information that possesses non statistical uncertainty. More importantly, NF systems aim at expressing a function in terms of interpretable linguistic rules as an alternative to pure neural network "black boxes", in which the function learned can only be observed through the input/output relationship.

The fuzzy decision tree (FDT) approach is an evolution of the inductive learning strategies towards the development of fuzzy decision rules. In the context of Machine Learning, decision tree induction has been shown to be quit powerful : a decision tree may easily extract understandable and explainable rules, or accurately classify some knowledge in a transparent form, or determine the most significant attributes for a studied phenomena. Further, these features are obtained in a computationally efficient manner, scaling up to large scale applications and real-time decision making. A fuzzy decision tree is a combination of fuzzy logic with decision tree induction, turned up from the necessity of manipulating fuzzy information or continuous input attributes while maintaining the interpretability and effectiveness of decision trees.

A fuzzy rule base represents human knowledge qualitatively, whereas membership functions change qualitative human reasoning into quantitative computations. For some problems, a fuzzy rule base can be clearly obtained by using human knowledge. In this case, a key problem is how to tune membership functions to fit knowledge efficiently. Since this tuning task can be viewed as an optimization problem, neural networks offer a possibility to solve this problem. For some other problems, no fuzzy rule base is available and a decision tree approach may be an effective way to determine it, extracting the fuzzy rules directly from quantitative data available about the problem.

In this paper, neurofuzzy and fuzzy decision tree approaches are presented, compared and combined. Our main objective is to identify main similarities and relevant differences among these two approaches, so as to highlight their potential applications and complementarities. The paper is organized as follows. Next section is devoted to describing neurofuzzy systems in terms of basic architecture, learning algorithm based on gradient-descent, advantages and limitations. Section 3 focuses on fuzzy decision trees, discussing their structure, growing and pruning algorithms, advantages and limitations. Section 4 provides a comparison of the two approaches from a theoretical point of view and discusses a combination between the two. Finally, conclusions are pointed out in Section 5.

2 Neurofuzzy systems

Many recent studies have focussed on developing systems capable of operating in complex uncertain environments by merging fuzzy and neural networks techniques (see [19]). Such systems can be classified into three main classes:

(i) neural networks that incorporate the imprecision and linguistic data handling abilities of fuzzy systems [11], [12], [17], [21]; in such approaches a conventional NN architecture is fuzzified in order to be able to process fuzzy inputs; fuzzy techniques are used to create or enhance neural networks; we call these approaches *fuzzy (arithmetic based) neural networks*.

(ii) approaches where NNs are used to provide inputs or to change outputs of a fuzzy system and where parameters of this fuzzy system are not modified by a learning process; we prefer to call them *neural/fuzzy combinations*.

(ii) fuzzy inference systems that incorporate the learning and generalization abilities of neural networks [5], [7], [9], [10], [14], [15], [18], [20], [24]; NNs are used as tools in fuzzy systems; a fuzzy system is derived from data or enhanced by learning from examples; we consider these *neurofuzzy* systems (NF).

Some approaches learn only the rules, requiring the membership functions to be set up by hand [13], others start with a fixed number of rules and membership functions and optimize them until they fit the data [7], [10],[11], [18], others are capable of identifying a minimal set of rules and tuning membership functions automatically from the example data [9].

Neurofuzzy systems are usually represented as a multilayer feedforward neural network [2], [7], [10], [17], [21], but other NNs architectures are considered, as radial basis function (RBF) networks [12] or adaptive resonance theory (ART) networks [14], [24].

The learning methods used for NF networks are derived from the gradient-descent-based learning algorithm [2], [7], [10], [11], [13], [14], [15], [20], [21], or are based on clustering [24] or hybrid learning schemes [12]. The learning phase of a NF system can consist in structural optimization, fuzzy membership functions optimization and/or fuzzy reasoning optimization.

2.1 A basic neurofuzzy architecture

A linguistic rule-based fuzzy system is composed of a set of rules of the following form

$$\begin{aligned} Rule\ j: \quad & IF\ (x_1\ is\ A_{1j})\ and\ \ldots\ and\ (x_N\ is\ A_{Nj}) \\ & THEN\ (y\ is\ B_j) \end{aligned} \tag{1}$$

where $j = 1, \ldots, R, R$ is the number of IF-THEN rules, x_i $(i = 1, \ldots, N)$ are the inputs of the fuzzy system, y is the output, A_{ij} and B_i are linguistic labels. For simplicity, we only consider single-output systems $f : U \subset \Re^N \to \Re$, where U is the universe of discourse, because the case of m-output systems can be separated into m single-output systems.

A conventional fuzzy model consists of three basic steps: fuzzification, inference process and defuzzification. Fuzzification is a mapping from the observed input to the membership degrees of the fuzzy sets, i.e. from $U \subset \Re^N$ to the unit hypercube

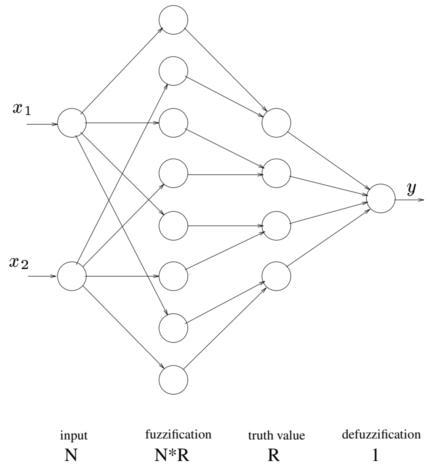

Figure 1: An example of neurofuzzy system. N=2, R=4.

$[0,1]^{N*R}$. Inference process is a decision making logic which determines fuzzy outputs corresponding to fuzzified inputs, with respect to the fuzzy rules. Defuzzification produces a non fuzzy output from the inferred fuzzy output. There are many different choices within each of these steps, leading to many different kinds of fuzzy models.

Translating the fuzzy rule-based system into a feedforward multilayer neural network architecture and considering a simplified "inference" process which performs only fuzzy AND operation, generally, a four layered structure can be identified [7], [10]. If N is the number of system inputs and R is the number of rules, there are R neurons in the third layer and $N * R$ neurons in the fuzzification layer computing the membership degrees of the inputs in the corresponding fuzzy sets. An example of such a structure is shown in Figure 1 for 2 inputs and 4 rules. Once we determine the number of inputs N and the number of rules R, we know the initial structure of the network.

If a complete inference process is considered, the network has a five layered structure, containing one more hidden layer for performing a fuzzy OR, to integrate the fired rules which have the same consequent [5], [15], [20].

Layer 1 (input): Each neuron in this layer is an input variable $x_i, i = 1, \ldots, N$ feeding the network. It only transmits input numbers to the next layer and makes no computations.

Layer 2 (fuzzification): This is a first hidden layer with $N * R$ neurons. Each node in this layer is connected with exactly one input node variable from the previous layer and feeds exactly one rule in the next layer. The first N neurons in this layer correspond to the condition part of the first rule, the second N neurons to the second rule, and so on. The layer computes the fuzzy membership degrees $\mu_{ij}(x_i) = \mu_{A_{ij}}(x_i)$ with respect to the fuzzy subsets $A_{ij}(j = 1, \ldots, R)$ defined in the universe of discourse of each input $x_i(i = 1, \ldots, N)$. Figure 2 shows three familiar possibilities of membership functions, each one characterized by two parameters: location α and shape β. For example, the output of the fuzzification layer can be computed as

$$\mu_{ij}(x_i) = e^{-\frac{(x_i - \alpha_{ij})^2}{\beta_{ij}}} \tag{2}$$

where μ_{ij} is the value of fuzzy membership function of the *i*th input variable corre-

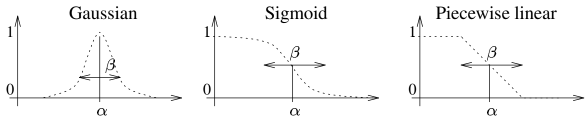

Figure 2: Example of fuzzy membership functions.

sponding to the jth rule.

Layer 3 (truth value): This is a second hidden layer with R neurons, one for each fuzzy rule. The truth values $\mu_j(x_1, \ldots, x_N), j = 1, \ldots, R$, of the fuzzy rules are computed. The rules' nodes should perform fuzzy AND operation. The most commonly used fuzzy AND operations (*t-norms*) are intersection (min), algebraic product, logical product, bounded product. For example, in the case of algebraic multiplicative operation, the output of this layer is

$$\mu_j(x_1, \ldots, x_N) = \prod_{i=1}^{N} \mu_{ij}(x_i) \tag{3}$$

where μ_{ij} are computed in the previous layer and μ_j is the truth value of the jth rule.

Layer 4 (defuzzification): This layer gives the system output by performing one of the defuzzification techniques: maximum-membership defuzzification scheme, fuzzy centroid defuzzification scheme, mean of maximum method. Considering the centroidal defuzzification method and assuming a discrete output universe of discourse, the real-valued output is computed as a fuzzy centroid

$$y = \frac{\sum_{j=1}^{R} \mu_j(x_1, \ldots, x_N) v_j}{\sum_{j=1}^{R} \mu_j(x_1, \ldots, x_N)} \tag{4}$$

where μ_j is the truth value of the jth rule for a given AND operation and v_j is the "center" of B_j, i.e. the average of the points in output space for which the membership function μ_{B_j} of B_j achieves its maximum value. From (4) we see that the "center" v_j of the consequence part of the *j*th rule is the synaptic weight of the connection between the *j*th neuron in the true value layer and the output neuron. Note that the shape of the output membership functions is not used in output computation, only the (average) peaks.

2.2 Learning algorithm

The connection weights of the resulting network, corresponding to the parameters of the fuzzy model $\alpha_{ij}, \beta_{ij}, v_j (i = 1, \ldots, N; j = 1, \ldots, R)$ are updated via a gradient-descent-based learning algorithm. The neurofuzzy system performs a tuning of the membership functions by the learning algorithm and thus it identifies the fuzzy rules.

For a given collection of input/output pairs of data, the learning algorithm modifies the network parameters minimizing the given performance function E. The general scheme of learning can be concisely expressed as

$$\Delta_{parameter} = -\eta \frac{\partial E}{\partial parameter} \tag{5}$$

where η denotes the learning rate. The parameters of the network are adjusted by

$$new_{parameter} = actual_{parameter} + \Delta_{parameter}. \tag{6}$$

For a performance function defined as

$$E = \frac{1}{2}(t-y)^2 \tag{7}$$

where y and t denote output of the neurofuzzy network and the desired output, respectively, and considering the relations [2] - [4], the gradients $\frac{\partial E}{\partial parameter}$ for the concerning parameters are:

$$\frac{\partial E}{\partial v_j} = -(t-y)\frac{\mu_j}{\sum_{j=1}^{R}\mu_j} \tag{8}$$

$$\begin{aligned} \frac{\partial E}{\partial \alpha_{ij}} &= -2(t-y)\frac{x_i-\alpha_{ij}}{\beta_{ij}}\frac{\mu_j}{(\sum_{j=1}^{R}\mu_j)^2} \\ &\quad (v_j\sum_{j=1}^{R}\mu_j - \sum_{j=1}^{R}\mu_j v_j) \end{aligned} \tag{9}$$

$$\begin{aligned} \frac{\partial E}{\partial \beta_{ij}} &= -(t-y)\frac{(x_i-\alpha_{ij})^2}{\beta_{ij}^2}\frac{\mu_j}{(\sum_{j=1}^{R}\mu_j)^2} \\ &\quad (v_j\sum_{j=1}^{R}\mu_j - \sum_{j=1}^{R}\mu_j v_j). \end{aligned} \tag{10}$$

The learning algorithm starts with a set of predefined membership functions for the input variables and applying the neurofuzzy scheme the optimal membership functions are derived.

Fuzzy system parameter identification may be combined with structure optimization of the rule base. A structure learning scheme is used to delete or combine some nodes and links in the neural fuzzy system, by merging similar fuzzy terms in the fuzzification layer, or by combining rule nodes with the same consequents and some common preconditions (rule pruning) [15], [18]. In this way, a set of representative rules is found by identifying similarities and removing redundancy in the rule base. A way of pruning the connections is to approximate the NF network by its Boolean version by selecting optimal threshold levels [20]. Another way of simplifying the rules is to search the space of all possible rules and to select the best rules in terms of an information-theoretic measure of rule value with respect to a given discrete data set [9].

2.3 Advantages and limitations

Neurofuzzy is not a replacement, but an addition to other methods like statistics and neural networks. It is a tool for enhancing fuzzy systems by learning from data in an effective, highly parallel computational manner. The learning algorithm should take the semantics of the desired fuzzy systems into account and adhere to certain constraints. The learning result should also be interpreted, and the insights gained by this may be used to restart the learning procedure to obtain better results if necessary. If the NF is initially provided with some knowledge from the data set, the resulting search space is reduced, thereby leading to a more efficient learning.

Neurofuzzy systems based on gradient-descent learning inherit its drawbacks:

- slowness, time consuming training;
- convergence problems, possible local minima.

But, the tool is a good alternative to neural networks considering the following advantages :

- vague knowledge can be used;
- a neurofuzzy system is interpretable in form of linguistic rules;
- from an applicational view it is easy to implement, to use and to understand;
- complex systems presenting high nonlinearity can be represented using only a few rules;
- the network architecture may become simpler due to the inherent reduction of the redundancy among the connection weights;
- incorporating any initial domain knowledge in the network prior to its learning, the learning process may be speeded up;
- human experts accept more easily NFs because they find them less difficult to validate with respect to their prior knowledge than standard NNs;

An extremely controversial issue relevant to the basis of fuzzy theory is the subjectivity of the membership functions, one of the sources of objections to fuzzy theory. In the above defined neurofuzzy system, the rule learning needs an initial fuzzy partitioning for each variable, i.e. a fixed number of membership functions is a priori settled. Some other approaches have the ability to approximate any membership profile from the sample data [21], or the membership functions are generated automatically based on an error criterion which terminates in a small number of steps [9].

The price for the interpretation of neurofuzzy systems in form of simple fuzzy rules might be paid by a result in classification, control, decision making or data analyzes, that is not as good as it could be if other methods were used. For example, a good *and* in the same time interpretable fuzzy classifier cannot usually simply be found by an automatic learning process [18].

Conventional neurofuzzy systems use the local optimization technique rather then a global one and this because they can only optimize the fuzzy membership functions under *unchangeable* fuzzy operators such as Min & Max and Product & Algebraic-sum. But, the hardware implementation of the neural structure is facilitated because operations like Max and Min fit well requirements of digital implementation.

Another limitation for a neurofuzzy system is due to the fact that most of the learning methods of neural networks process only numerical data. For supervised learning problems, fuzzy neural networks have been proposed to process linguistic information by fuzzifying the weights, inputs and outputs of the network using fuzzy numbers represented by α-level sets, but in these cases the number of arithmetic operations becomes complex and time consuming.

3 Fuzzy decision trees

An approach for handling fuzzy information preserving at least the interpretability of a decision tree, if not the computational efficiency, was found to be the fuzzy decision tree approach. This is particularly suited for exploiting numerical input variables,

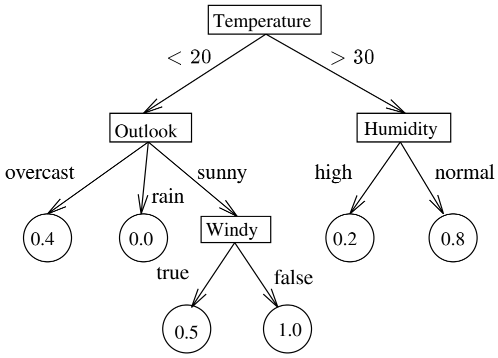

Figure 3: An example of fuzzy decision tree.

continuous-valued attributes, by providing smooth input/output mappings in the form of membership degrees, for creating fuzzy rule bases, or for giving a systematic correlation analyses among different attributes and identifying the most discriminating attributes among those proposed by an expert [25].

3.1 A fuzzy decision tree structure

Fuzzy decision trees are based on extensions of top down induction learning method : to infer the output information corresponding to given input attribute values, one traverses the tree, starting at the top-node, and applying sequentially the tests encountered to select the appropriate successor; when a terminal node is reached, the output information stored there is retrieved [25].

An example of classification with a (non binary) FDT is given in Figure 3 for an illustrative problem of determining whether to play tennis based on weather conditions. The attributes are (pure) numerical as Temperature, symbolic as Outlook (rain, overcast, sunny), or numeric-symbolic which takes numerical values but the interpretation is either numerical or symbolic [16] as Humidity (high, normal) and Windy (false, true). The fuzzy decision tree built on the basis of a “data base” of hypothetical situations, allows one to infer the membership degree of the target class “Good for playing tennis” on the basis of the given input attributes. The test nodes are represented by attribute names within rectangles and the leaf nodes by memberships to the target class within circles.

The process of creating rules from the tree consists of performing a complete depth first search and every time a path reaches a leaf, a new rule is created with the antecedent being the **and** of all the tests encountered on the path, and the consequence given by the membership degree found at the leaf [8], [16].

We will focus on binary FDTs and numeric-symbolic attributes, while conforming to the method described in [4]. Fuzzy binary tree induction in the case of classification problems based on continuous numerical attributes consists in using as learning target a class-membership degree and in choosing at the tree growing step at each test node an optimal attribute together with *two* thresholds, defining a transition region between the left and right successors. Thus, an automatic fuzzy discretization of numerical attributes is performed. The objective is to represent a fuzzy class-membership degree $\mu_C(o)$ of an object o, on the basis of the available information about its attributes $a_i(o)$. In this approximation scheme, each node of the tree is interpreted as a fuzzy subset of the attribute space. These fuzzy subsets are built up in a recursive fashion using fuzzy dichotomies at test-nodes.

Test nodes. A test-node's fuzzy dichotomy corresponds to a "discriminator" function $\nu(a(o)) \to [0 \ldots 1]$ allowing one to compute the membership degree of an object o to the left (and right) successor subsets from the membership degree of the current node and the attribute values $a(o)$. Denoting by $S, S_{\text{left}}, S_{\text{right}}$ the sets associated with a test-node and its two successors, and by $X_a(o)$ the fuzzy set describing the value of attribute $a(\cdot)$ of o, the following partitioning rule may be used

$$\begin{aligned} \mu_{S_{\text{left}}}(o) &= \mu_S(o)\langle\nu\rangle_{X_a(o)} \\ \mu_{S_{\text{right}}}(o) &= \mu_S(o)(1 - \langle\nu\rangle_{X_a(o)}), \end{aligned} \tag{11}$$

using the algebraic product for fuzzy AND [3], [28], and the fuzzy average $\langle\nu\rangle_{X_a(o)}$ of the partition function, obtained by

$$\langle\nu\rangle_{X_a(o)} = \frac{\int_{dom(a)} \mu_{X_a(o)}(x)\nu(x)dx}{\int_{dom(a)} \mu_{X_a(o)}(x)dx}, \tag{12}$$

which, when $X_a(o)$ is a crisp singleton ($= \{a(o)\}$), degenerates to $\nu(a(o))$. $\langle\nu\rangle_{X_a(o)}$ is considered here as a satisfiability measure which measures the adequateness of a new object with the values presented in the test nodes.

Leaves. Leaves (terminal nodes) correspond to fuzzy subsets of the attribute space defined recursively by the fuzzy dichotomies installed at their parent nodes. Each leaf L is labeled by a number (denoted by $\tilde{\mu}_{C_L}$) representing the most typical membership to the target class in this fuzzy subset. Thus, more accurate information is provided about the affiliation to the target class comparatively with the class labeling in the case of crisp decision trees.

Approximation scheme. The value $\tilde{\mu}_C$ predicted by a fuzzy tree for an object of known attribute values (fuzzy or crisp) may be the average of the labels attached to the leaves, weighted by the degree of membership of the object to the corresponding fuzzy subsets. In other words

$$\tilde{\mu}_C(o) = \frac{\sum_{L \in leaves} \mu_{S_L}(o)\tilde{\mu}_{C_L}}{\sum_{L \in leaves} \mu_{S_L}(o)}, \tag{13}$$

where the $\mu_{S_L}(o)$ values are obtained by recursive application of eq. (11). Note that since the terminal subsets of a fuzzy tree are a *fuzzy partition* of the attribute space $\sum_{L \in leaves} \mu_{S_L}(o) = 1$, and thus [3], [28]

$$\tilde{\mu}_C(o) = \sum_{L \in leaves} \mu_{S_L}(o)\tilde{\mu}_{C_L}. \tag{14}$$

Clearly, if both $\nu(\cdot)$ and $a(\cdot)$ are crisp, this scheme degenerates into a crisp regression tree. If, moreover, the values attached at terminal nodes are crisp (0 or 1) it degenerates into a crisp decision tree. In the other cases, it provides an approximation scheme with smooth interpolation capabilities.

3.2 Fuzzy decision tree learning

Hypothesis space. A hypothesis space of fuzzy trees is defined indirectly by defining a set of candidate dichotomies to be installed at its test nodes. In tree induction, dichotomies generally only involve one attribute at the time, and the set of candidate dichotomies are defined separately for different types of attribute values (numerical, symbolic, numeric-symbolic). In binary fuzzy tree induction this amounts to define classes of mappings $\nu(\cdot)$ from attribute values into $[0 \ldots 1]$. The discriminator functions $\nu(\cdot)$ used may have the same shape like the membership functions used in neurofuzzy systems (see Figure 2): gaussian, sigmoid, piecewise linear.

Greedy recursive tree growing. Fuzzy trees are grown in a top-down recursive partitioning approach, similarly to classical trees. Thus, growing starts with the top-node of the tree and the complete learning set and searches for a locally optimal dichotomy, using some ad hoc score function. It then partitions the learning set into two fuzzy subsets using the optimal dichotomy, and creates the corresponding sub-nodes, which are expanded in the same fashion. At each candidate node for splitting, the method first checks whether a stopping condition is met, in which case the node is declared to be terminal and is labeled.

Optimal splitting thus identifies the best attribute to use and determines the optimal parameters (α and β) of the used discriminator template, according to the local learning set. There may be three conditions for stopping to split [4] : (i) limitation of the depth of the tree; (ii) cardinality of the local growing set; (iii) look ahead : if the optimal one-level subtree at a node makes a good classification of the local growing set, then the node is developed and the two successors are declared terminal. Terminal node labeling merely computes a "sharpened average" value $\tilde{\mu}_{C_L}$ using the values of $\mu_C(o)$ of the local learning states.

Pruning. Like in crisp decision trees or neural networks, and in many other automatic learning methods, it is necessary to tradeoff data fit (how well the learning states are approximated by the fuzzy tree) with complexity (number of parameters) in order to provide good generalization capabilities. There are two possible approaches to reach such a tradeoff : (i) using hypothesis testing to stop splitting as soon as the improvements observed in the learning sample become statistically insignificant; (ii) build overly complex trees and use a pruning algorithm so as to derive a sequence of nested trees of decreasing complexity, together with cross-validation to select the appropriate complexity. At each step of such a pruning procedure, a test-node is selected, the subtree starting at this node is replaced by a terminal node and the resulting tree is stored. The process terminates upon reaching the trivial single-node tree. The generalization capabilities of all the intermediate trees are then evaluated on an independent cross-validation sample (not used for tree growing) and the best tree is chosen. Note that updating formulas may be used here in order to minimize the amount of computations.

Comments. The presented approach for fuzzy decision tree is suited for having a singular target class, but the result for the opposite class may be deduced. In the Figure 3 example, the result at the leaf with $\tilde{\mu}_{C_L} = 0.4$ for example, means that is good for playing tennis with the degree 0.4 and is not good for playing tennis with the degree of 0.6. When the objects from the training sample are associated with more then two classes, a forest of fuzzy decision trees is constructed, each tree being dedicated to the recognition of a single class against the union of all the other classes [16].

3.3 Advantages and limitations

The fuzzy decision tree approach has some advantages with respect to the crisp decision tree one:

- it has the ability to exploit fuzzy information about attributes and output classes;
- it produces an automatic fuzzy discretization of numerical attributes, at the tree growing stage; the profile of the discriminator function is not user defined, only the type of the discriminator (gaussian, sigmoid, piecewise linear) has to be settled a priori;
- using fuzzy output classes it provides richer and more accurate information about the problem comparatively with the YES/NO type of answer in the case of crisp decision tree;
- it offers a very simple and natural way to cope with unknown attribute values;
- by using continuous transition regions at test nodes, fuzzy tree induction imposes some regularity condition on the inferred input/output mapping which reduces both parameter and prediction variances[1]; the parameter variance reduction increases interpretability and the prediction variance reduction increases accuracy of the tree;
- fuzzy trees represent information similarly to the way human experts understand problems leading to more acceptable models;
- the accuracy of the tree may be globally increased due to a refitting process, a parametric model optimization, which is possible only thanks to the smooth nature of the FDT model;
- a fuzzy rule base may be automatically detected translating a FDT.

The presented fuzzy tree induction has unfortunately a drawback with respect to traditional trees. Any new instance has to be propagated in parallel through many branches of nodes, with a mechanism for storing memberships to each of them, instead of finding an unique decision path to classify like in a classical decision tree. An immediate consequence of this recursive fuzzification is the computational burden increase, which makes the method intrinsically slower than crisp tree induction. But, even so, it will be able to handle large scale system problems with small enough response times to allow interactive use.

A present limitation of the method described above is the fact that the tool is able to handle only numerical and numeric-symbolic attributes, the dichotomization of symbolic (pure qualitative) attributes has not yet been considered.

4 Comparison and combination of NFs and FDTs

4.1 Comparison

Two complementary methods of automatic learning able to exploit fuzzy information have been presented in this paper: neurofuzzy and fuzzy tree. The techniques have some common features:

[1] Variance denotes here the variability of the learned model with respect to the random nature of learning samples used.

- ability to manage fuzzy information;
- interpretability in the form of fuzzy rules;
- more acceptable models by human experts.

From our point of view, fuzzy decision trees differ from neurofuzzy techniques looking at the following aspects:

- A FDT carries out automatically the selection of the relevant attributes to a given problem. For a conventional NF the relevant attributes are a priori embedded in the network in the form of the inputs. This is an important issue which have to be considered when one of the techniques is chosen. The FDTs are better exploiting large amounts of input data and are recommended for large scale system problems. The NFs are better used in problems with reasonable number of inputs.
- In some problems human knowledge cannot be easily translated in rules and a fuzzy decision tree is a good tool for automatic extraction of a fuzzy rule base and thanks to the pruning process, a compromise is made between the rule base accuracy and simplicity. When a rule base is clearly available for a task, a neurofuzzy network is well suited for fuzzy rules translation into an optimization problem and for fuzzy rules adjustment for fitting the available data.
- In comparison to tree induction, multilayer feedforward network training is generally orders of magnitudes slower. Under the circumstances that a FDT is computational not so efficient like a traditional tree (due to the distribution which occur) and that a NF based on multilayer feedforward network is faster than the classical network (because the NF is not fully connected and because prior knowledge may be embedded in the network parameters), further investigation must be carried out to determine if a FDT training remains however faster than a NF one, or not. The expectation is that the larger the number of input features the more advantageous the FDT growing algorithm will be, since its complexity is linear in the number of candidate attributes.
- NF systems may depend on the first initialization of the parameters and on the order of introducing objects in the network, being possible to obtain different minima of the performance function. A FDT building is not influenced by these two aspects.
- For a NF, the output of the network is important only in the parameter optimization stage, because what we are looking for is the rule base identification. In the case of FDTs, the output of the tree is used per se.
- The learning process needs an initial fuzzy partitioning in the fuzzification stage. It is necessary to be a priori settled a fixed number of membership functions with fixed profiles in most of NF existing methods or discriminator functions with a priori shapes in FDTs approaches. In NFs the subjectivity of the membership functions may influence the parameters tuning, but this aspect is not crucial for FDT where discriminators shapes must be under some constrains but their choice is not critical.

4.2 Combination between NFs and FDTs

There are many hybrid approaches in the literature aiming at combining decision tree techniques with neural network optimization procedures combining the advantages while circumventing the weaknesses [1], [6], [23], [25]. Generally, a crisp decision tree is grown to determine the relevance of the attributes to the given problem, then a structure of four layers feedforward neural network is generated by translation and a tuning process is performed on the new structure. A hybrid method of fuzzy modeling was proposed in [22] as well, consisting on fuzzification of the decision rules extracted from a DT, their translation to a NN and final optimization of the synaptic weights.

In situations where expert examinations in the field of concern are available, neurofuzzy techniques may be applied directly to optimize the knowledge embedded in fuzzy rules. In situations where we do not know the important attributes for a task or how a minimal set of rules is looking like, a better start is by building a fuzzy decision tree who produce some rules or identify the relevant attributes. Next, a neurofuzzy technique may be used for tuning the parameters of the translated fuzzy rules.

Once the structure of a fuzzy decision tree was determined for a given problem, it can then be viewed as a parametric model. For example (restricting to numerical attributes), and denoting by α_T, β_T the location and shape parameters of the discriminators installed at some test-node T, $\tilde{\mu}_{C_L}$ the label at a leaf L, and a_i the attributes selected at the test nodes, this model can be written as

$$\tilde{\mu}_C(o) = f(\alpha_T, \beta_T, \tilde{\mu}_{C_L}, a_i(o)) \tag{15}$$

where $f(\cdot)$ is a continuous and differentiable function with respect to its arguments (parameters and attributes)[2] [4], depending in a linear fashion on the leaf parameters $\tilde{\mu}_{C_L}$ (see eq. (14)) and in a non-linear fashion on the parameters α_T, β_T and attributes a_i.

During tree growing and pruning, the parameters $\alpha_T, \beta_T, \tilde{\mu}_{C_L}$ are determined on the basis of local information only, hence, the idea of further tuning the FDT by adjusting its parameters aiming at a global optimum [4]. This refitting process would consist in translating the FDT into a NF structure and using a neural network training technique as for example a gradient-descent-based learning algorithm.

5 Conclusion

In this paper we have discussed salient features of neurofuzzy systems and top-down fuzzy decision tree induction in the context of supervised automatic learning, pointing out their main differences, but also their similarities.

Neurofuzzy systems appear as an alternative tool to classical neural networks with potentially better interpretability and the possibility of incorporating knowledge prior to its learning, simpler architecture and easier implementation. Fuzzy decision tree induction has demonstrated its ability to exploit fuzzy information, to automatically discretize numerical attributes, and construct fuzzy rule bases in a totally autonomous fashion.

Both tools are additions to existing methods rather than replacements and their effectiveness depends on the problem. In problems where human knowledge cannot be easily translated into rules, or where the relevant problem attributes are not known a priori, neurofuzzy networks are not well suited to solve the problem, but fuzzy decision trees are an appropriate tool for data mining, classification or approximation.

---

[2] everywhere, in the case of sigmoid discriminator templates; almost everywhere, in the case of piecewise linear ones

When a fuzzy rule base is available for a task, a neurofuzzy system is a good alternative to optimize the fuzzy rules until they fit the available data. The training process for a NF is faster than for a neural network from the computational point of view. Although the growing stage for a FDT is slower than for a classical tree, it is still much faster than neural network training.

The suggested combination between the two approaches performs a global optimization of the fuzzy rules or for the approximation scheme, which gives better accuracy with respect to pure fuzzy decision tree applying.

References

[1] R. Araya, P. Gigon. Segmentation Trees: A New Help for Building Expert Systems and Neural Networks. In *Proc. of Stats*, pages 119-124, 1992.

[2] J. M. Benitez, J. L. Castro, I. Requena. Are Artificial Neural Networks Black Boxes?. *IEEE Trans. Neural Networks*, 8(5), pages 1156-1164, 1997.

[3] X. Boyen. Design of Fuzzy logic-based Decision Trees applied to Power System Transient Stability Assessment. Master's thesis, University of Liège, 1995.

[4] X. Boyen, L Wehenkel. Automatic induction of fuzzy decision trees and its application to power system security assessment. *Fuzzy Sets and Systems*, 102, pages 3-19, 1999.

[5] M. M. Chowdhury, Y. Li. Evolutionary Reinforcement Learning for Neurofuzzy Control. In *Proc. of Int. Fuzzy Systems Assoc. World Congress*, volume 2, pages 434-439, Prague, 1997.

[6] K. J. Cios, N. Liu. A Machine Learning method for Generation of a Neural Network Architecture: A Continuous ID3 Algorithm. *IEEE Trans. Neural Networks*, 3(2), pages 280-291, 1992.

[7] P. Y. Glorennec. Learning Algorithms for Neuro-Fuzzy Networks. In *Fuzzy Control Systems*, ed. by A. Kandel, G. Langholz, CRC Press, Inc., pages 4-18, 1994.

[8] L. O. Hall, P. Lande. Generating fuzzy rules from decision trees. In *Proc. of Int. Fuzzy Systems Assoc. World Congress*, volume 2, pages 418-423, Prague, 1997.

[9] C. M. Higgins, R. M. Goodman. Fuzzy Rule-Based Networks for Control. *IEEE Trans. Fuzzy Systems*, 2(1), pages 82-88, 1994.

[10] I. Hiraga, T. Furuhashi, Y. Uchikawa, S. Nakayama. An Acquisition of Operator's Rules for Collision Avoidance Using Fuzzy Neural Networks. *IEEE Trans. Fuzzy Systems*, 3(3), pages 280-287, 1995.

[11] H. Ishibuchi, R. Fujioka, H. Tanaka. Neural Networks That Learn from Fuzzy If-Then Rules. *IEEE Trans. Fuzzy Systems*, 1(2), pages 85-97, 1993.

[12] N. B. Karayiannis, G. W. Mi. Growing Radial Basis Neural Networks: Merging Supervised and Unsupervised Learning with Network Growth Techniques. *IEEE Trans. Neural Networks*, 8(6), pages 1492-1506, 1997.

[13] B. Kosko. *Neural networks and fuzzy systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall, 1992.

[14] C-J. Lin, C-T. Lin. Reinforcement Learning for An ART-Based Fuzzy Adaptive Learning Control Network. *IEEE Trans. Neural Networks*, 7(3), pages 709-731, 1996.

[15] C-T. Lin, Y-C. Lu. A Neural Fuzzy Systems with Linguistic Teaching Signals. *IEEE Trans. Fuzzy Systems*, 3(2), pages 169-189, 1995.

[16] C. Marsala. Apprentissage inductif en présence de données imprécises: construction et utilisation d'arbres de décision flous. Thèse de doctorat, Université Paris 6, 1998.

[17] S. Mitra, R. K. De, S. K. Pal. Knowledge-Based Fuzzy MLP for Classification and Rule Generation. *IEEE Trans. Neural Networks*, 8(6), pages 1338-1350, 1997.

[18] D. Nauck, R. Kruse. What are Neuro-Fuzzy Classifiers?. In *Proc. of Int. Fuzzy Systems Assoc. World Congress*, volume 2, pages 228-233, Prague, 1997.

[19] D. Nauck, F. Klawonn, R. Kruse. *Neuro-fuzzy Systems*. John Wiley and Sons, 1997.

[20] W. Pedrycz, A. F. Rocha. Fuzzy-Set Based Models of Neurons and Knowledge-Based Networks. *IEEE Trans. Fuzzy Systems*, 1(4), pages 254-266, 1993.

[21] G. Purushothaman, N. B Karayiannis. Quantum Neural Networks (QNN's): Inherently Fuzzy Feedforward neural Networks. *IEEE Trans. Neural Networks*, 8(3), pages 679-693, 1997.

[22] J.-S. Roger Jang. Structure determination in fuzzy modeling: a fuzzy CART approach. In *Proc. of the Third IEEE Int. Conference on Fuzzy Systems*, volume 1, pages 480-485, 1994.

[23] I. K. Sethi. Entropy Nets: From Decision Tress to Neural Networks. In *Proc. of the IEEE*, 78(10), pages 1605-1613, 1990.

[24] P. K. Simpson. Fuzzy Min-Max Neural Networks - Part 2: Clustering. *IEEE Trans. Fuzzy Systems*, 1(1), pages 32-45, 1993.

[25] L. Wehenkel. *Automatic Learning Techniques in Power Systems*. Kluwer Academic, Boston, 1998.

[26] L. A. Zadeh. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Trans. Syst., Man, Cybern.*, SMC-3, pages 28-44, 1973.

[27] L. A. Zadeh. Making computers think like people. *IEEE Spectrum*, pages 26-32, 1984.

[28] J. Zeidler, M. Schlosser. Continuous-Valued Attributes in Fuzzy Decision Trees. In *Proc. of Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 395-400, Granada, 1996.