Exploiting Decision Trees in Product-based Fuzzy Neural Modeling to Generate Rules with Dynamically Reduced Dimensionality

Minas Pertselakis and Andreas Stafylopatis
National Technical University of Athens,
School of Electrical and Computer Engineering,
9 Iroon Polytechneioy Str., Zografou, 15780, Greece
mper@cslab.ntua.gr, andreas@cs.ntua.gr

Abstract. Decision trees are commonly employed as data classifiers in various research fields, but also in real-world application domains. In the fuzzy neural framework, decision trees can offer valuable assistance in determining a proper initial system structure, which means not only feature selection, but also rule extraction and organization. This paper proposes a synergistic model that combines the advantages of a subsethood-product neural fuzzy inference system and a CART algorithm, in order to create a novel architecture and generate fuzzy rules of the form "IF - THEN IF", where the first "IF" concerns the primary attributes and the second "IF" the secondary attributes of the given dataset as defined by our method. The resulted structure eliminates certain drawbacks of both techniques and produces a compact, comprehensible and efficient rulebase. Experiments in benchmark classification tasks prove that this method does not only reduce computational cost, but it also maintains performance at high levels, offering fast and accurate processing during real-time operations.

1 Introduction

Neuro-fuzzy modeling involves two stages: Parameter Identification and Structure Identification. The former is concerned with the adjustment of system parameters, such as the membership functions, the antecedent and consequent weights and so on, while the latter is related to finding a suitable number of rules, a feature selection scheme and a proper partition of the feature space, which usually is applied as an initialization technique in order to improve learning [1].

For structure identification the literature offers several heuristic but practical and systematic approaches, realized either by unsupervised learning and clustering (fuzzy

Please use the following format when citing this chapter:
Pertselakis, Minas, Stafylopatis, Andreas, 2006, in IFIP International Federation for Information Processing, Volume 204, Artificial Intelligence Applications and Innovations, eds. Maglogiannis, I., Karpouzis, K., Bramer, M., (Boston: Springer), pp. 19–26

c-means, k-means, etc) or supervised techniques, such as the CART-ANFIS combination [2]. However, the problem of structure identification in fuzzy modeling is by no means solved; there are many issues in practice remain to be addressed; the "curse of dimensionality" and finding the optimum number of rules being the most significant.

Decision trees as classification tools have a solid statistic foundation and offer advantages that make them suitable for high-dimensional data [3]. They are considered to be a nonparametric method and are capable of handling datasets with missing values or possible errors. Decision trees are also characterized by their ability to identify features with the most information, which may lead to dimensionality reduction, if exploited. However, decision tree algorithms suffer from certain drawbacks. They use the "divide and conquer" method and partition the feature space into mutually exclusive regions, thus they tend to perform well if few highly relevant attributes exist, but less so if many complex interactions are present. Moreover, decision trees show over-sensitivity to the training set due to their greedy behavior and their "generalization" capabilities are low. Noise and irrelevant attributes may very well affect their performance.

The combination of decision trees and neuro-fuzzy systems constitutes a solid approach to fuzzy modeling. This derives from the fact that decision trees offer a fast and efficient way to produce distinct rules and select relevant inputs, thus producing a roughly correct and compact structure. On the other hand, neuro-fuzzy networks demonstrate the ability to adapt and operate in both numeric and linguistic environments, they can refine the rules by learning from data and increase the "generalization" abilities of the system. In the special case where the neuro-fuzzy model incorporates a fuzzy product operator to aggregate activities at a rule node, the associations between attributes can be estimated and exploited.

Our approach suggests the use of a CART algorithm [4] to induce a top-down tree structure, where each path to a terminal node represents a rule. To "fuzzify" these rules we convert the resulted partitions of the *primary* feature's space into clusters. The clusters (rules) that show low reliability, as indicated by the decision tree, undergo a novel process, where the fuzzy c-means (FCM) clustering method is applied on the related patterns using only the *secondary* features, if any, so as to create two new rules on the same feature-space region. The resulted rulebase is implemented in a subsethood product neuro-fuzzy architecture and a gradient descent learning procedure is employed in order for the rules to be fine-tuned. Experiments in benchmark classification tasks show that the rules extracted by this method, not only reduce computational cost, while keeping performance at high levels, but offer also high interpretability due to the fact that both decision trees and neural fuzzy systems generate easily comprehensible rules.

The paper is divided in the following sections: Section 2 describes the methodology of rulebase generation based on the CART algorithm, whereas in Section 3 we present the architecture and functionality of the subsethood-product neuro-fuzzy model. Experimental results can be found in Section 4, while useful deductions and plans for future research conclude the paper in Section 5.

2 Rule Generation using Decision Trees

It has been shown in various comparative studies that most decision tree algorithms fit into a simple algorithmic framework, whereas the differences concentrate on the tree growing criteria and the ways trees are pruned [3].

To construct an appropriate decision tree for our purposes, we utilize the CART algorithm which first grows the tree extensively based on a training data set, and then prunes the tree back based on a minimum cost-complexity principle [4]. To grow the classification tree, we employ the *Twoing* impurity function, which, at each node t selects a split that minimizes the following expression:

$$\frac{p_l p_r}{4}[\sum_j p_j(t_l)p_j(t_r)]^2 \ , \tag{1}$$

where p_l and p_r are the percentages of cases in the splitting node t that branch left and right, $p_j(t_l)$ and $p_j(t_r)$ are the probabilities of a data point in class j, given that the data set comes from the left and right children, respectively. The justification of the twoing rule can be found in the CART monograph [4]. The cost-complexity measure, as shown in several studies, tends to over-pruning, which creates smaller trees, but less accurate [5], [6]. This is desirable in our methodology since the resulted rules will be fine-tuned later on.

Our approach in structure identification involves two successive stages; the initial generation of the rulebase and the propagation of the less capable rules into a subsequent layer for further evaluation.

2.1 Phase 1: Generation of Initial Rulebase

Once the decision tree is constructed, we consider each path that leads to a leaf to represent a Rule. However, the rules created by a decision tree, as noted above, are crisp and partition the data space into mutually exclusive regions. Therefore, we need to convert these rules into a form that conforms to the multivariate concept.

Instead of partitioning, we introduce a cluster analysis approach. We define a cluster C by a center k, which is the mean value of the patterns that belong to the resulted partition (crisp rule) of the feature-space, and a spread σ, which represents the respective standard deviation. Since the cluster analysis approach is multivariate by definition, whereas the decision tree is univariate at each split, we assign random values, within the feature limits, to those attributes that do not take part in the specific crisp rule as defined by the tree. At this point, we should add some definitions that will be used extensively throughout the paper.

The attributes that participate in this procedure, that is the features the decision tree *as a whole* has selected to use, will be called *primary* attributes for the purposes of this paper. Those attributes that do not take part in the constructed decision tree, are called *secondary* attributes. If the total number of attributes of the dataset is N, then the number of the primary attributes is always $M \leq N$. In a high dimension problem it is usually M<<N.

Consequently, the rules that involve only the primary attributes compose the Primary Rule set, while the rules that are formed by the secondary attributes form the Secondary Rule set. The following step of our method deals with the latter type of rules.

2.2 Phase 2: Rule Bisection and Propagation

This second phase evaluates the resulted rules and performs certain actions on them, if required. More explicitly, the clusters that include very few patterns (1-2) are ignored, since those patterns are regarded as outliers, while the rules produced by phase 1 that present low classification reliability according to a reliability criterion subject to further processing. The proposed reliability measure R is an extension of the method suggested in [7], where the reliability criterion is the difference between the two greatest output values. In our approach, the difference value is normalized to enhance the consistency of the result. This extended measure can be formulated for a classifier as follows:

$$R = (Y_{win1} - Y_{win2}) * Y_{win1} \tag{2}$$

where Y_{win1} is the classification output of the leaf node for the class with the greatest value and Y_{win2} is the classification output for the class with the second greatest value.

If this confidence criterion is below a certain threshold, then the system considers the current rule insufficient and fires up the process of rule bisection. For this reason, we execute the FCM clustering algorithm requesting 2 clusters on the patterns that constitute the particular rule-cluster, but only on the attributes that do not take part in the tree structure; the *secondary* attributes. If there is none, then we perform the same clustering procedure with all the attributes of the problem, but this is not usually the case in tasks of high dimension.

In other words, we force the terminal node of the tree to "grow" two new multivariate leaves based on the less informative, according to the tree, features. These new rules aim to find a better solution (clustering) in the selected data space by exploiting all those attributes, $K = N - M$, that the tree have omitted. This leads to a novel system structure with enhanced abilities that will be discussed in detail in the next section.

2.3 Example of Rule Generation Procedure

As a simple example, we use the iris data set. The iris data set consists of 150 four-dimensional patterns, therefore $N=4$, which are categorized into three subspecies of the Iris flower, namely *Iris setosa*, *Iris versicolor*, and *Iris virginica*. The four features represent the sepal length, sepal width, petal length, and petal width measurements on the Iris flower. The generated tree is shown in figure 1.

It is apparent that the decision tree, as a whole, requires only two out of the four attributes to produce this result, namely petal length (PL) and petal width (PW), which implies that $M=2$. Therefore, we assume that all generated clusters are

described by these two attributes, even if some of the paths contain only one attribute (like that path leading to node 1), in which case we fill in a random value within the limits that characterize the missing feature (e.g. the petal width feature for node 1).

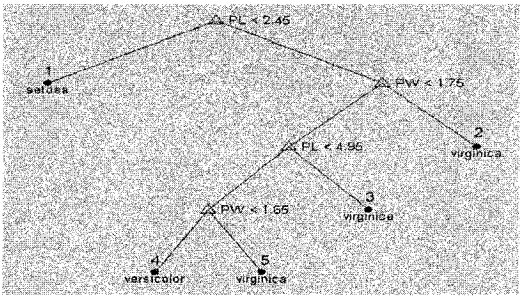

Fig. 1. Example of a decision tree for the Iris data using the CART algorithm

During the second phase, the leaf node 5 is removed due to its low population and leaf node 3 is split, since it presents low reliability, into two new rules defined by the secondary attributes of the dataset, which are the sepal length and width. The system results with 4 primary rules and 2 secondary rules as its initial structure.

3 System Architecture and Operational Details

The proposed system follows the paradigm of [8], as in the numeric inputs are fuzzified using feature-specific Gaussian fuzzy sets and linguistic inputs are presented as is. The antecedent and consequent weights are represented as Gaussian fuzzy connections of the network. The model employs a mutual subsethood-based activation spread and a product aggregation operator, bounded between zero and one, which works in conjunction with volume defuzzification in a gradient descent learning framework. In addition, human expert knowledge can be embedded directly in the form of fuzzy "IF-THEN" rules.

Using the product operator, instead of the more common fuzzy *min* operator for activity aggregation offers certain important advantages. It does not ignore information regarding the dimension of the input and provides a better estimate of the joint strength of the features involved. It is also capable of better discrimination, since it can clearly differentiate between inputs and weight vectors over a wide range of spreads. This approach leads to high performance and economy in parameters, but it also carries a significant drawback. In problems where the dimensionality of the feature space is large, the product operator produces values very close to zero, given that each factor assumes values between zero and one. Theoretically, if the dimension is set beyond a certain finite number, the value of which depends on the computing device, the product operator will underflow and equate to zero. Therefore,

there is an upper limit on the dimension a subsethood product based system can handle and this is one of the issues our approach addresses with its dimensionality reduction capabilities.

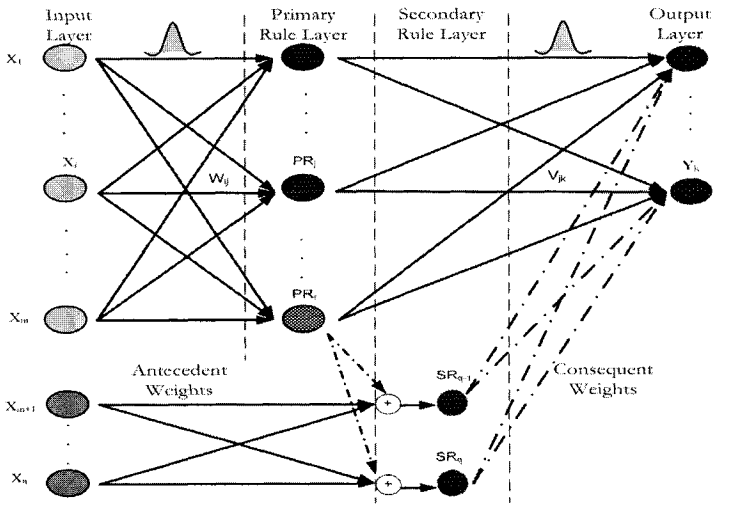

Fig. 2. System architecture. The less reliable primary rule PRr propagates its firing strength to the next layer when it is required, where the secondary attributes form two secondary rules.

The architecture of the proposed neural fuzzy model that constitutes the main difference from previous similar works is presented in figure 2. Fuzzy weights w_{ij} from input nodes i to rule nodes j are modeled by the center w_{ij}^c and spread w_{ij}^σ of a Gaussian fuzzy set and denoted by $w_{ij}=(w_{ij}^c, w_{ij}^\sigma)$. In a similar fashion, consequent fuzzy weights from rule nodes j to output nodes k are denoted by $v_{jk} = (v_{ij}^c, v_{ij}^\sigma)$. The spread of the i-th fuzzified input element is denoted as x_i^σ, while x_i^c is obtained as the crisp value of the i-th input feature element. The net value of the transmitted signal along the fuzzy connections is quantified by the extent of overlap between the two fuzzy sets, known as *mutual subsethood*. All mutual subsethood expressions, gradient descent learning and weight updating equations are omitted, due to lack of space, but can be found in [8].

In our approach, two rule layers exist; the Primary rule layer and the Secondary rule layer. Only one of the two is active at any given time and thus, only the weights that fan-in and out of the activated layer are updated. This behavior could also be depicted as two different networks.

During normal operation the Primary rule layer is responsible for the output. If an unreliable rule obtains the maximum value over all Primary rules then the secondary rule layer is activated. When that occurs, we sum up the product of all mutual subsethoods of the primary attributes that lead to the Primary rule in question

and the product of mutual subsethoods of the secondary attributes that are directed to the secondary rule. We are forced to follow this approach due to the nature of the product operator that requires the same number of factors in order to produce comparable values and the same order of magnitude for a balanced weight update. Thus, if z_{rq} is the firing strength of rule PR_r, then the activation strength z_q of rule SR_q is given by:

$$z_q = z_{rq} + \prod_{i=m+1}^{n} E_{iq} \ . \tag{3}$$

The output y_k over the k classes is determined using standard volume based centroid defuzzification and its expression for our system is:

$$y_k^p = \frac{\prod_{j=1}^{Q(p)} z_j v_{jk}^c v_{jk}^\sigma}{\prod_{j=1}^{Q(p)} z_j v_{jk}^\sigma} \ , \tag{4}$$

where Q(p) is the total number of rules that affect the result for the given pattern p. This robust and flexible structure scheme consolidates a new architecture in fuzzy modeling that enjoys the benefits of the dynamically reduced dimensionality.

4 Experimental Results

We test the performance of the system on 2 datasets, namely the Ionosphere data and the Pima Indians Diabetes data, both of which can be found at the UCI repository[1]. The Ionosphere data set includes 351 records of 34 attributes each that were collected by a radar system in Goose Bay, Labrador. The values are categorized in 2 classes labelled Good and Bad. The Pima Indians diabetes data set, on the other hand, consists of 768 patterns of 8 attributes taken from the National Institute of Diabetes and Digestive and Kidney Diseases. The classes are of binary form denoting positive and negative results in diabetes tests.

For our experiments we split the dataset into a training set and a testing set dividing the patterns in a rough 40% - 60% ratio, respectively. In each experiment, we extract an initial rule base using the CART algorithm as described in section 2 and train the fuzzy-neural network for 100 epochs with a fixed learning rate of 0.001. The results presented in Table 1 show the average of 10 such experiments.

Table 1. Experimental Results and testing accuracy comparison to other known methods[2]. The numbers in parenthesis in the Primary Rules field denote the number of unreliable rules.

Datasets	***Our Approach***	*Accuracy of other known methods*

[1] Available at http://www.ics.uci.edu/~mlearn/MLRepository.html

[2] Available at http://www.phys.uni.torun.pl/kmk/projects/datasets.html

	Attributes (Primary/Total)	Prim. Rules	Sec. Rules	*Accuracy (%)*	SVM	MLP+BP	CART	kNN, k=1
Pima Indians	**6/8**	**15 (6)**	**12**	**78.25**	77.6	76.4	72.8	71.9
Ionosphere	**6/34**	**7 (1)**	**2**	**94.4**	93.2	96	88.9	92.1

5 Conclusions - Future Work

This paper describes a novel methodology of rulebase generation and configuration by exploiting the decision trees' unique properties of feature selection and structure identification combined with a clustering technique. We first convert the crisp partitions created by the CART algorithm into clusters, each defined by a hyper-ellipsoid that takes into account only the primary, as selected by the tree, features. The rules-clusters that are considered insufficient according to a reliability measure are bisected by the FCM algorithm and propagate to a secondary network layer, where only the secondary attributes participate.

In other words, our system generates rules of the form "IF-THEN IF" where the first "IF" concerns the primary attributes and the more distinct regions of the dataset, while the second "IF" employs the secondary attributes in order to search for a better partitioning in the more "fuzzy" areas of the problem at hand. These rules are then fine-tuned by a fuzzy neural network that employs a product aggregating operator that estimates the joint strength of all inputs in contrast to the more common fuzzy *min* approach.

Further work on the same field includes the use of different algorithms to split the rules of low reliability, such as the execution of a new decision tree at each node that performs poorly and the exploitation of different aggregation operators, such as the harmonic mean, besides the inner product for rule activation. Various alternatives concerning the reliability measure is also an issue under consideration.

References

1. J.-S. R. Jang, C. -T. Sun, E. Mizutani, *Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence* (Prentice-Hall, Inc., 1997)
2. J.-S. R. Jang, Structure determination in fuzzy modeling: a fuzzy CART approach, *In proc. of IEEE Int. Conf. on Fuzzy Systems,* Orlando, Florida (1994)
3. L. Rokach, O. Maimon, Top-Down Induction of Decision Trees Classifiers – A survey, *IEEE trans. on Systems, Man and Cybernetics*, **35**(4), 476-487 (2005)
4. L. Breiman, J.H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees* (Wadsworth, Inc., Belmont, California, 1984)
5. J. R. Quinlan, Simplifying decision trees, *Int. J. Man-Machine Studies*, **27,** 221-234 (1987)
6. F. Esposito, D. Malerba, and G. Semerato, A comparative analysis of methods for pruning decision trees, *IEEE Trans. Pattern Anal. Mach. Intell.*, **19**(5), 476-492 (1997)
7. L. P. Cordella, P. Foggia, C. Sansone, F. Tortorella, M. Vento, Reliability Parameters to Improve Combination Strategies in Multi-Expert Systems, *Pattern Analysis and Application*, **2**, 205-214 (1999)
8. S. Paul, S. Kumar, Subsethood- Product Fuzzy Neural Inference System (SuPFuNIS), *IEEE Trans. Neural Networks*, **13**(3), 578–599 (2002)