Generating Classification Rules with the Neuro–Fuzzy System NEFCLASS

Detlef Nauck, Ulrike Nauck, and Rudolf Kruse

Technical University of Braunschweig, Dept. of Computer Science,
Bueltenweg 74–75, D–38106 Braunschweig, Germany
Tel: +49.531.391.3155, Fax: +49.531.391.5936
E-Mail: D.Nauck@tu-bs.de, URL: www.cs.tu-bs.de/∼nauck

Abstract

Neuro–fuzzy systems have recently gained a lot of interest in research and application. In this paper we discuss NEFCLASS, a neuro–fuzzy approach for data analysis. We present new learning strategies to derive fuzzy classification rules from data, and show some results.

1 Introduction

In [10] we have presented the NEFCLASS model, a neuro–fuzzy model for data analysis. In this paper we present refined rule learning algorithms for our model. NEFCLASS is used to derive fuzzy rules from a set of data that can be separated in different crisp classes. The fuzzy rules describing the data are of the form:
R : **if** x_1 is μ_1 **and** x_2 is μ_2 **and** ... **and** x_n is μ_n **then** the pattern $(x_1, x_2, \ldots, x_n)$ belongs to class i, where $\mu_1, \ldots, \mu_n$ are fuzzy sets. The task of the NEFCLASS model (**NE**uro **F**uzzy **CLASS**ification) is to discover these rules and to learn the shape of the membership functions to determine the correct class or category of a given input pattern. The patterns are vectors $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ and a class C is a (crisp) subset of $\mathbb{R}^n$. We assume the intersection of two different classes to be empty. The pattern feature values are represented by fuzzy sets, and the classification is described by a set of linguistic rules. For each input feature x_i there are q_i fuzzy sets $\mu_1^{(i)}, \ldots, \mu_{q_i}^{(i)}$, and the rule base contains k fuzzy rules $R_1, \ldots, R_k$.

NEFCLASS does not induce fuzzy classification rules by searching for clusters, but by modifying the fuzzy partitionings defined on each single dimension. Other approaches for the generation of fuzzy rules from data usually search for hyper–ellipsoidal [6] or hyper–rectangular clusters [1, 2, 12]. Fuzzy sets are then constructed by projecting the clusters. In the case of hyper–ellipsoidal clusters the projection causes a loss of information which is avoided, when hyper–rectangular clusters are used. However, in both cases the induced fuzzy rules are sometimes not easy to interpret. NEFCLASS modifies the fuzzy sets directly, and can therefore use constraints to make sure that the induced fuzzy rules can be interpreted well.

This paper appears in: Proc. Biennial Conf. of the North American Fuzzy Information Processing Society (NAFIPS'96), Berkeley, 1996.

2 The NEFCLASS model

A NEFCLASS system has a 3–layer feedforward architecture that is derived from a generic fuzzy perceptron [11]. The first layer U_1 contains the input units representing the pattern features. The activation a_x of a unit $x \in U_1$ is usually equal to its external input. However, it can be different, if the input unit does some kind of preprocessing (normalization etc.). The hidden layer U_2 holds rule units representing the fuzzy rules, and the third layer U_3 consists of output units, one for each class. The activation of a rule unit $R \in U_2$, and the activation of an output unit $c \in U_3$ when a pattern $\mathbf{p}$ is propagated are computed by

$$\begin{aligned} a_R^{(p)} &= \min_{x \in U_1} \{W(x,R)(a_x^{(p)})\}, \\ a_c^{(p)} &= \sum_{R \in U_2} \frac{W(R,c) \cdot a_R^{(p)}}{W(R,c)}, \\ &\text{or alternatively: } a_c^{(p)} = \max_{R \in U_2} \{a_R^{(p)}\}, \end{aligned}$$

where $W(x, R)$ is the *fuzzy weight* on the connection from input unit x to rule unit R. $W(R, c)$ is the weight on the connection from rule unit R to output unit c. For semantical reasons all these weights are fixed at 1 [10]. Alternatively, the output activation can be computed by a maximum operation instead of a weighted sum. This can be selected in accordance to the application problem. Instead of min and max, other t–norms and t–conorms can be used.

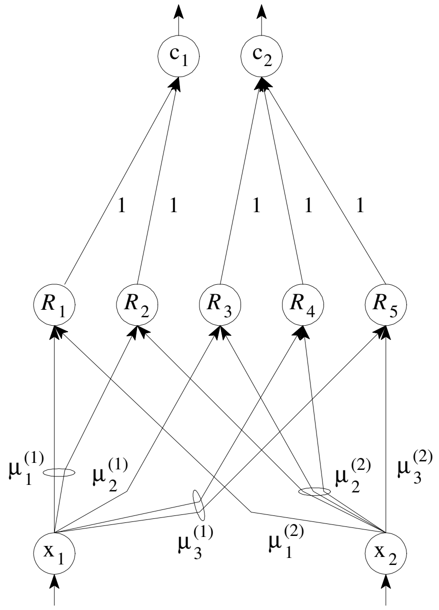

Figure 1: A NEFCLASS system with two inputs, five rules and two output classes

The rule base is an approximation of an (unknown) function $\varphi : \mathbb{R}^n \to \{0,1\}^m$ that represents the classification task where $\varphi(\mathbf{x}) = (c_1, \ldots, c_m)$ such that $c_i = 1$ and $c_j = 0$ ($j \in \{1, \ldots, m\}, j \neq i$), i.e. $\mathbf{x}$ belongs to class C_i. Because of the mathematics involved the rule base actually does not approximate φ but the function $\varphi' : \mathbb{R}^n \to [0,1]^m$. We will get $\varphi(\mathbf{x})$ by $\varphi(\mathbf{x}) = \psi(\varphi'(\mathbf{x}))$, where ψ reflects the interpretation of the classification result obtained from a NEFCLASS system. In our case we will map the highest component of each vector $\mathbf{c}$ to 1 and its other components to 0, respectively (winner takes all).

The fuzzy sets and the linguistic rules which perform this approximation, and define the resulting NEFCLASS system, will be obtained from a set of examples by learning. The Fig. 1 shows a NEFCLASS system that classifies input patterns with two features into two distinct classes by using five linguistic rules.

Its main feature are the *shared weights* on some of the connections. This way we make sure, that for each linguistic value (e.g. "x_1 is *positive big*") there is only one representation as a fuzzy set (e.g. $\mu_1^{(1)}$ in Fig. 1), i.e. the linguistic value has only one interpretation for all rule units (e.g. R_1 and R_2 in Fig. 1). It cannot happen that two fuzzy sets that are identical at the beginning of the learning process develop differently, and so the semantics of the rule base encoded in the structure of the network is not affected [9]. Connections that share a weight always come from the same input unit.

A NEFCLASS system can be build from partial knowledge about the patterns, and can be then refined by learning, or it can be created from scratch by learning. A user has to define a number of initial fuzzy sets partitioning the domains of the input features, and must specify a value for k, i.e. the maximum number of rule nodes that may be created in the hidden layer.

The learning algorithm for NEFCLASS is described in the following definition. We assume that triangular membership functions are used that are described by three parameters:

$$\mu : \mathbb{R} \to [0,1], \quad \mu(x) = \begin{cases} \frac{x-a}{b-a} & \text{if } x \in [a,b), \\ \frac{c-x}{c-b} & \text{if } x \in [b,c], \\ 0 & \text{otherwise.} \end{cases}$$

In addition, the leftmost and rightmost membership functions for each variable can be shouldered.

Definition 1 (NEFCLASS learning algorithm) *Consider a NEFCLASS system with n input units $x_1, \ldots, x_n$, $k \leq k_{\max}$ rule units $R_1, \ldots, R_k$, and m output units $c_1, \ldots, c_m$. Also given is a learning set $\tilde{\mathcal{L}} = \{(\mathbf{p}_1, \mathbf{t}_1), \ldots, (\mathbf{p}_s, \mathbf{t}_s)\}$ of s patterns, each consisting of an input pattern $\mathbf{p} \in \mathbb{R}^n$, and a target pattern $\mathbf{t} \in \{0,1\}^m$. The learning algorithm that is used to create the k rule units of the NEFCLASS system consists of the following steps* **(rule learning algorithm)***:*

(i) Select the next pattern $(\mathbf{p}, \mathbf{t})$ from $\tilde{\mathcal{L}}$

(ii) For each input unit $x_i \in U_1$ find the membership function $\mu_{j_i}^{(i)}$ such that

$$\mu_{j_i}^{(i)}(p_i) = \max_{j \in \{1, \ldots, q_i\}} \{\mu_j^{(i)}(p_i)\}$$

(iii) If there are still less than $k_{\max}$ rule nodes, and there is no rule node R with

$$W(x_1, R) = \mu_{j_1}^{(1)}, \ldots, W(x_n, R) = \mu_{j_n}^{(n)}$$

than create such a node, and connect it to the output node c_l if $t_l = 1$.

(iv) If there are still unprocessed patterns in $\tilde{\mathcal{L}}$, and $k < k_{\max}$ then proceed with step (i), and stop otherwise.

(v) Determine the rule base by one of the following three procedures:

- **"Simple" rule learning***: Keep just the first k created rules (i.e. stop rule creation when $k_{\max} = k$ rules have been created).*

- **"Best" rule learning**: *Process $\tilde{\mathcal{L}}$ again, and accumulate the activations of each rule unit for each class of the propagated patterns. If a rule unit R displays a higher accumulated activation for a class C_j than for the class C_R specified by the rule conclusion, then change the conclusion of R to C_j, i.e. connect R to output unit c_j. Process $\tilde{\mathcal{L}}$ again, and compute for each rule unit*

$$V_R = \sum_{p \in \tilde{\mathcal{L}}} a_R^{(p)} \cdot e_p,$$
$$e_p = \begin{cases} 1, & \text{if } p \text{ is classified correctly,} \\ -1, & \text{otherwise.} \end{cases}$$

Keep those k rule units with the highest values for V_R, and delete the other rule units from the NEFCLASS system.

- **"Best per class" rule learning**: *Proceed like in "Best" rule learning, but keep for each class C_j those $\left\lfloor \frac{k}{m} \right\rfloor$ best rules whose conclusions represent the class C_j ($\lfloor x \rfloor$ is the integer part of x).*

The supervised learning algorithm of a NEFCLASS system to adapt its fuzzy sets runs cyclically through the learning set $\tilde{\mathcal{L}}$ by repeating the following steps until a given end criterion is met **(fuzzy set learning algorithm)***:*

(i) Select the next pattern $(\mathbf{p}, \mathbf{t})$ from $\tilde{\mathcal{L}}$, propagate it through the NEFCLASS system, and determine the output vector $\mathbf{c}$.

(ii) For each output unit c_i: Determine the delta value $\delta_{c_i} = t_i - a_{c_i}$

(iii) For each rule unit R with $a_R > 0$:

(a) Determine the delta value

$$\delta_R = a_R(1 - a_R) \sum_{c \in U_3} W(R, c)\delta_c.$$

(b) Find x' such that

$$W(x', R)(a_{x'}) = \min_{x \in U_1} \{W(x, R)(a_x)\}.$$

(c) For the fuzzy set $W(x', R)$ determine the delta values for its parameters a, b, c using the learning rate $\sigma > 0$:

$$\begin{aligned} \delta_b &= \sigma \cdot \delta_R \cdot (c - a) \cdot \operatorname{sgn}(a_{x'} - b), \\ \delta_a &= -\sigma \cdot \delta_R \cdot (c - a) + \delta_b, \\ \delta_c &= \sigma \cdot \delta_R \cdot (c - a) + \delta_b, \end{aligned}$$

and apply the changes to $W(x', R)$ if this does not violate against a given set of constraints Φ. (Note: the weight $W(x', R)$ might be shared by other connections, and in this case might be changed more than once)

(iv) If an epoch was completed, and the end criterion is met, then stop; otherwise proceed with step (i).

There are three ways to create a rule base for a NEFCLASS system. The "simple" procedure can only be successful, if the patterns are selected randomly from the learning set, and if the cardinalities of the classes are approximately equal. It works for simple problems like the Iris data. Usually a user will choose "best" or "best per class" rule learning. The latter procedure should be selected, when one supposes that the patterns are distributed in an equal number of clusters per class. "Best" rule learning is suitable, when there are classes, which have to be represented by a larger number of rules than other classes. Either way, rule learning is completed after three cycles through the data set.

The learning procedure for the fuzzy sets is a simple heuristic. It results in shifting the membership functions, and in making their supports larger or smaller. It is easy to define constraints Φ for the learning procedure, e.g. that fuzzy sets must not pass each other, or that they must intersect at 0.5, etc. The learning procedure cannot reach an error value of zero due to the mathematics involved. As a stop criterion usually the change in error is used. The sum in step (iii.a) is not really necessary, because each rule unit is connected to only one output unit. But it makes the model more flexible, because it is possible to also use adaptive rule weights. We refrain from it, because we want to keep the semantics of a NEFCLASS system [10]. We also found that rule weights are not necessary to obtain good classification results.

3 Deriving Fuzzy Rules From Data

In this section we will present some results of a NEFCLASS system applied to classification problems. The first task is to obtain classification rules to correctly classify the patterns of the well known Iris data set [3]. This data set contains 150 patterns belonging to three different classes (Iris Setosa, Iris Versicolour, and Iris Virginica) with 50 patterns each. The patterns have four input features (sepal and petal length and width of the iris flower). The first class is linearly separable from the other two classes, whereas the second and third class are not linearly separable from each other. This is a very simple domain, and it is often used as a benchmark.

For the NEFCLASS learning process the data set was split in half, and the patterns were ordered alternately within the training and test data sets. At first, we allowed the system to create a maximum of seven rules using the "best" rule learning algorithm. The domains of the four input variables were initially each partitioned by three equally distributed fuzzy sets. NEFCLASS created at first 19 rules (81 would be possible) and selected the best 7 rules (see Table 1). Fuzzy set learning stopped after 126 epochs, because the error was not decreased for 50 epochs. After learning, 3 out of 75 patterns from the training set were still classified wrongly (i.e. 96% correct). Testing the second data set, the NEFCLASS system classified only 2 out 75 patterns incorrectly (i.e. 97.3% correct). Considering all 150 patterns the system performed well with 96.67% correct classifications.

if	(s,	m,	s,	s)	**then**	*Setosa*
if	(s,	s,	s,	s)	**then**	*Setosa*
if	(m,	s,	l,	l)	**then**	*Virginica*
if	(l,	s,	l,	l)	**then**	*Virginica*
if	(l,	m,	l,	l)	**then**	*Virginica*
if	(m,	s,	m,	m)	**then**	*Versicolour*
if	(m,	s,	m,	s)	**then**	*Versicolour*

Table 1: Seven rules found by NEFCLASS to classify the Iris data (The antecedents are in a short form, with s = small, m = medium, l = large, and (s,m,s,s) meaning x_1 is small and x_2 is medium and x_3 is small and x_4 is small)

Because the Iris problem is extremely simple, it can also be solved by less rules using only two of the four input variables. We trained another NEFCLASS system, using only the third and fourth input, and allowing the system to create three rules. Using "best per class" rule learning, the system finds at first five rules (nine would be possible) and finally selects the three rules shown in Table 2. After training for 110 epochs, we get two errors on the training set, and three errors on the test set, i.e. the same performance like seven rules using all four features.

We have compared the learning result of NEFCLASS to the results obtained with the FuNe–I system presented in [5]. This neuro–fuzzy system reached a classification rate of 99% on the test set of the Iris data using 13 rules and four inputs, and a 96% classification rate using 7 rules and 3 inputs. FuNe I is offered in a limited test version by the authors of [5], so we could run our own test to compare it with NEFCLASS. We allowed the system to create 10 rules, and it came out with 5 classification errors on the test set after training. Therefore the two models are comparable in their performance, even though FuNe I has a much more complex structure and training procedure. FuNe I also uses a concept of weighted rules. We refrained from using this approach in NEFCLASS, because the semantics of weighted fuzzy rules is not clear [8].

Classification Rule		Score
if $(x_3 = s, x_4 = s)$	**then** *Setosa*	23.80
if $(x_3 = m, x_4 = m)$	**then** *Versicolour*	7.47
if $(x_3 = l, x_4 = l)$	**then** *Virginica*	15.53
if $(x_3 = l, x_4 = m)$	**then** *Virginica*	0.13
if $(x_3 = m, x_4 = s)$	**then** *Versicolour*	4.53

Table 2: Five classification rules found by NEFCLASS when only the last two features of the Iris data are used. The first three rules were selected by the rule learning procedure "best per class"

As a second test for the NEFCLASS learning algorithms we used the "Wisconsin Breast Cancer" data set [13]. This data set contains 699 cases distributed into two classes (benign and malign). We used only 683 cases, because 16 cases have missing values. Each pattern has nine features. To show how NEFCLASS performs when prior knowledge is supplied, we used a fuzzy clustering method to obtain fuzzy rules. We used a modification of the algorithm by Gustafson and Kessel [4] presented in [6]. Fuzzy clustering discovered three clusters that were interpreted as fuzzy rules by projecting the clusters to each dimension and finding trapezoidal membership functions that closely matched the projections. The membership functions were interpreted by *small, medium* and *large* resulting in three rules that caused 94 classification errors on the whole data set:

R_1: **if**	(s,s,s,s,s,s,s,s,s)	**then**	*benign*,
R_2: **if**	(m,m,m,m,m,l,m,m,s)	**then**	*malign*,
R_3: **if**	(m,l,l,l,m,l,m,l,s)	**then**	*malign*.

Feature x_9 has the same value in all rules, therefore we leave it out in the NEFCLASS system.

When we initialize a NEFCLASS system with these three rules, we at first get 240 classification errors. However, after 80 epochs of training, we obtained a result of only 50 errors (92.7% correct).

Trying to use NEFCLASS without prior knowledge, and a maximum of three rules, we do not get a result by using "best" rule learning, because only one class is covered by the best three rules. Using four rules and "best per class" rule learning results in a NEFCLASS system performing badly with 135 errors (80.4% correct). So in this case using prior knowledge

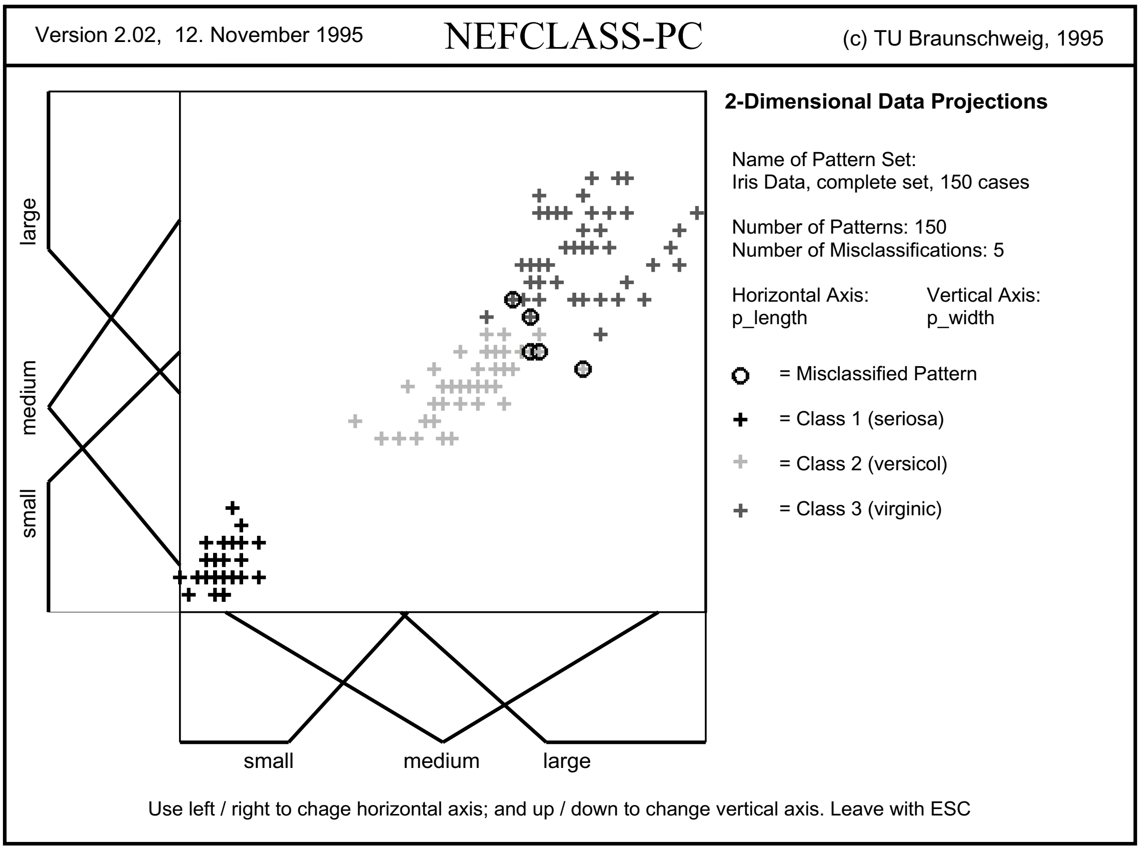

Figure 2: A two dimensional plot of the Iris data showing also the fuzzy sets learned by NEFCLASS. The system was initialized by three equally distributed fuzzy sets (shouldered or triangular) for each input feature. Only features x_3 and x_4 were used.

is a substantial advantage.

By analyzing the fuzzy sets obtained by training the three rules used as prior knowledge we found that the fuzzy set *medium* substantially overlapped with either the fuzzy set *small* or *large* for almost all variables. This can be seen as evidence that *medium* is superfluous, and therefore we again trained a NEFCLASS system with "best per class" rule learning, allowing it to create four rules. This time we only used two fuzzy sets to partition the domains of each variable. After 100 epochs of training NEFCLASS made only 24 errors (96.5% correct). It has found the rules

R_1: **if** (s,s,s,s,s,s,s,s,s) **then** *benign*,
R_2: **if** (l,s,s,s,s,s,s,s,s) **then** *benign*,
R_3: **if** (l,l,l,l,l,l,l,l,s) **then** *malign*,
R_4: **if** (l,l,l,l,s,l,l,l,s) **then** *malign*.

The example with the Wisconsin Breast Cancer data shows that NEFCLASS can be used as an interactive data analysis method. It is useful to provide prior knowledge when it is possible. A combination with fuzzy clustering can help here. The learning result of NEFCLASS should be analyzed, and the obtained information can be used for another run that yields an even better result.

4 Conclusions

In this paper we have presented new rule learning procedures for the neuro–fuzzy classification model NEFCLASS. We tested it on the Iris data and the Wisconsin Breast Cancer data, and obtained satisfactory results. We showed that it is useful to initialize a NEFCLASS system with prior knowledge if available. A NEFCLASS learning result should be analyzed to obtain information that can be used to obtain an even better result by running NEFCLASS again with fewer rules or fuzzy sets. The goal of NEFCLASS is to provide an interpretable fuzzy classificator. Therefore the learning algorithm constrains the adaption of the fuzzy sets, and refrains from using adaptive weights.

The NEFCLASS software for MS–DOS PC that was used to obtain the results described in this paper can be freely obtained by anonymous ftp from ftp.ibr.cs.tu-bs.de in the directory /pub/local/nefclass, or from the World Wide Web (http://www.cs.tu-bs.de/~nauck).

References

[1] Shigeo Abe and Ming-Shong Lan. A method for fuzzy rules extraction directly from numerical data and its application to pattern classification. *IEEE Trans. Fuzzy Systems*, 3(1):18–28, February 1995.

[2] Shigeo Abe, Ming-Shong Lan, and Ruck Thawonmas. Tuning of a fuzzy classifier derived from data. *Int. J. Approximate Reasoning*, 14(1):1–24, January 1996.

[3] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7(Part II):179–188, 1936.

[4] D.E. Gustafson and W.C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC*, pages 761–766, San Diego, 1979.

[5] Saman K. Halgamuge and Manfred Glesner. Neural networks in designing fuzzy systems for real world applications. *Fuzzy Sets and Systems*, 65:1–12, 1994.

[6] Frank Klawonn and Rudolf Kruse. Constructing a fuzzy controller from data. *Fuzzy Sets and Systems*, 85(1), 1997.

[7] Rudolf Kruse, Jörg Gebhardt, and Rainer Palm, editors. *Fuzzy Systems in Computer Science.* Vieweg, Braunschweig, 1994.

[8] Detlef Nauck. Fuzzy neuro systems: An overview. In [7], pages 91–107. Vieweg, Braunschweig, 1994.

[9] Detlef Nauck and Rudolf Kruse. Choosing appropriate neuro-fuzzy models. In *Proc. Second European Congress on Fuzzy and Intelligent Technologies (EUFIT94)*, pages 552–557, Aachen, September 1994.

[10] Detlef Nauck and Rudolf Kruse. NEFCLASS – a neuro–fuzzy approach for the classification of data. In K.M. George, Janice H. Carrol, Ed Deaton, Dave Oppenheim, and Jim Hightower, editors, *Applied Computing 1995. Proc. of the 1995 ACM Symposium on Applied Computing, Nashville, Feb. 26–28*, pages 461–465. ACM Press, New York, February 1995.

[11] Detlef Nauck and Rudolf Kruse. Designing neuro–fuzzy systems through backpropagation. In Witold Pedrycz, editor, *Fuzzy Modelling: Paradigms and Practice*, pages 203–228. Kluwer, Boston, 1996.

[12] Nadine Tschichold-Gürman. Generation and improvement of fuzzy classifiers with incremental learning using fuzzy rulenet. In K.M. George, Janice H. Carrol, Ed Deaton, Dave Oppenheim, and Jim Hightower, editors, *Applied Computing 1995. Proc. of the 1995 ACM Symposium on Applied Computing, Nashville, Feb. 26–28*, pages 466–470. ACM Press, New York, February 1995.

[13] W.H. Wolberg and O.L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proc. National Academy of Sciences*, 87:9193–9196, December 1990.