A Classifier System-Based Learning Algorithm for Interpretable, Adaptive Nearest Neighbour Classifiers

Frank Klawonn, Katharina Tschumitschew
Department of Computer Science
University of Applied Sciences BS/WF
Salzdahlumer Str. 46/48
D-38302 Wolfenbuettel, Germany
E-mail: {f.klawonn,katharina.tschumitschew}@fh-wolfenbuettel.de

***Abstract*— Nearest neighbour classifiers provide intuitive classification decisions in the sense that they determine the class of an unknown object on the basis of the most similar cases in a sample database. However, in order to reach a good interpretability of a nearest neighbour classifier, two important aspects should be taken into account: The sample database should be kept reasonably small and the similarity or distance measure should be transparent. Especially, when objects with many attributes are considered, a distance measure like the Euclidean distance does not always lead to intuitive results.**

In this paper, we present an approach to construct a nearest neighbour classifier that tries to take care of these two aspects. The sample database is kept small by adopting ideas from classifier systems. We use an adaptive distance measure that enables us to explain the similarity measure and the classification decision in terms of simple fuzzy rules.

I. Introduction

Supervised classification is a learning task, where the class, i.e. the value of a nominal attribute of an object has to be predicted on the basis of some other given or measured attributes of this object. The classifier is constructed using a training set where in addition to the measured attributes also the correct classes of the objects are known.

A great variety of classifiers like (naive) Bayesian, logistic regression, decision trees, nearest neighbour or fuzzy rule-based classifiers exist. In this paper, we focus on nearest neighbour classifiers and demonstrate their relations to classifier systems used in evolutionary computation and to fuzzy rules.

Nearest neighbour classifiers solve classification tasks on the basis of a database of classified sample cases. A new data object is classified on the basis of the class(es) of its closest or most similar neighbour(s) in the sample database. Although a large database of sample cases might enhance the performance of a nearest neighbour classifier, it leads to higher computational costs, when classifying new data, and it does not contribute to the interpretability of the classifier. Simplicity and interpretability can only be achieved, when the database contains a reasonably small set of representative sample cases or prototypes. Therefore, in this paper we propose a heuristic method to keep the number of cases in the sample database reasonably small. This method is motivated by techniques applied in the context of classifier systems.

Another problem in the context of nearest neighbour classifiers is the question how to measure the distance or similarity between an object from the sample database and a new object to be classified. Here we propose an adaptive nearest neighbour classifier that uses a scaled distance similar as it is sometimes used in fuzzy systems. This enables us to specify fuzzy rules that explain the classification decision of our nearest neighbour classifier. In this way, the user obtains information about the similar cases in the sample database as well as on the underlying similarity or distance measure.

The paper is organized as follows. In section II we briefly review the concept of nearest neighbour classifiers. Section IV introduces some basic ideas from classifier systems and how they can be adopted in the context of nearest neighbour classifiers. Our algorithm to construct a simplified nearest neighbour classifier using the concepts from classifiers systems and an adaptive distance measure is explained in section V including some elementary examples. In section VI we outline the relation to fuzzy rules. The final conclusions contain perspectives for future work.

II. Nearest Neighbour Classifiers

A nearest neighbour classifier [1] performs a classification task, i.e. it defines a mapping from an input space $\mathcal{S}$ to a finite output space $\mathcal{C}$ of classes based on a database $\mathcal{B} \subseteq \mathcal{S} \times \mathcal{C}$ of samples and a distance measure $d : \mathcal{S}^2 \to [0, \infty)$. A k-nearest neighbour classifier (where k is a positive integer) determines the class of an object $s \in \mathcal{S}$ by computing the k closest objects in the sample database (nearest neighbours) and assigning the class to s which occurs most often among the k nearest neighbours. A 1-nearest neighbour classifier simply assigns the class of the closest sample in the database to an object s.

The definition of a suitable distance measure d is crucial for the performance of a nearest neighbour classifier. If the input space consists of a set of p continuous-valued attributes, i.e.

$\mathcal{S} \subseteq \Re^p$, then commonly the (squared) Euclidean distance is chosen.

However, better performance can be achieved, when the distance measure is adapted, leading to an adaptive nearest neighbour classifier. Many techniques for adapting the distance have been proposed in the literature, like for instance [2] where the distance measure is adapted for each object in the sample database based on a (local) linear discriminant analysis. Although discriminant analysis is a powerful tool for classification, the resulting classifier (or the resulting distance measure in case of our nearest neighbour classifier) is not easy to interpret. Therefore, we propose the following strategy to obtain a simplified nearest neighbour classifier with an adaptive distance that can even be visualised for the user, providing better explanation of the resulting classification.

III. Fuzzy Classifiers and Nearest Neighbour Classifiers

Fuzzy classifiers are not based on prototypes like nearest neighbour classifiers, but they use fuzzy rules for the classification of an object. Many different types of fuzzy classifiers have been proposed in the literature. A thorough overview can be found in [3].

The type of fuzzy classifiers considered in this paper is based on a finite set $\mathcal{R}$ of rules of the form $R \in \mathcal{R}$:

R: If x_1 is $\mu_R^{(1)}$ and $\ldots$ and x_p is $\mu_R^{(p)}$ then class is C_R.

$C_R \in \mathcal{C}$ is one of the classes. The $\mu_R^{(i)}$ are assumed to be fuzzy sets on X_i, i.e. $\mu_R^{(i)} : X_i \to [0,1]$. In order to keep the notation simple, we incorporate the fuzzy sets $\mu_R^{(i)}$ directly in the rules. In real systems one would replace them by suitable linguistic values like *positive big*, *approximately zero*, etc. and associate the linguistic value with the corresponding fuzzy set.

Given an object $s = (s_1, \ldots, s_p)$ to be classified, the firing degree of a single rue R is determined by

$$\mu_R(s_1, \ldots, s_p) \;=\; \bigotimes_{i=1}^{p} \mu_R^{(i)}(s_i) \qquad (1)$$

where $\bigotimes$ is the extension of a suitable t-norm to p arguments. Very often, $\bigotimes$ is chosen to be the minimum, although this leads to very restricted class separation abilities of the fuzzy classifier [4], [5]. However, the Łukasiewicz t-norm, given by

$$\bigotimes_{i=1}^{p} \mu_R^{(i)}(s_i) \;=\; \max\left\{ \left(\sum_{i=1}^{p} \mu_R^{(i)}(s_i) \right) - (p-1), 0 \right\}, \quad (2)$$

leads not only to more flexible fuzzy classifiers, but enables us also to establish a connection between fuzzy classifiers and nearest neighbour classifiers later on.

After the firing degree (2) has been computed for each rule, the fuzzy classifier will assign the object s to the class $\mathcal{C}_R$ that occurs in the conclusion part of the rule R with the highest firing degree.

So far, we have not assumed any restrictions for the fuzzy sets $\mu_R^{(i)}$ in the rules. But in order to obtain interpretable rules, so that linguistic values like *positive big* or *approximately zero* make sense, it is necessary that the fuzzy sets are at least convex. This means that they are first increasing until they reach the membership degree one and afterwards they start to decrease. Triangular, trapezoidal and Gaussian membership functions are typical fuzzy sets with these properties. Consider such a fuzzy set μ and assume that $\mu(x_0) = 1$ holds. We consider as a prototypical representative of this fuzzy set. For triangular and Gaussian fuzzy sets x_0 is unique, for trapezoidal fuzzy sets we can choose for x_0 any of the points where the trapezoidal membership function assumes the value one, usually we would choose the midpoint of this interval. For such a fuzzy set the membership degree $\mu(x)$ can be interpreted as the degree of similarity of x to the prototype value x_0. When we consider the value $1 - \mu(x)$, this can be considered as the scaled and cut-off distance of x to the prototype value x_0. This distance can be understood as scaled and cut-off in the following sense. Let us first consider the most simple case of the triangular fuzzy set shown in figure 1.

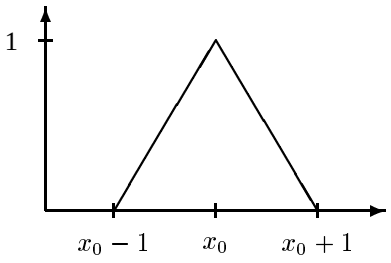

Fig. 1. A triangular fuzzy set

It is obvious that we have in this simple case

$$d_\mu(x_0, x) \;=\; 1 - \mu(x) \;=\; \min\{|x_0 - x|, 1\}. \qquad (3)$$

Thus $1 - \mu(x)$ is just the usual distance between x_0 and x, cut off at the value one. If the triangular membership function does not have a width of two as in figure 1, but another width, still being symmetric, equation (3) remains valid up to a scaling factor c that has to be introduced, i.e.

$$d_\mu(x_0, x) \;=\; 1 - \mu(x) \;=\; \min\{|x_0 - x| \cdot c, 1\}$$

where c is 2/(width of the triangle). For asymmetric triangular membership function different scaling factors are needed on the left- and right-hand side of x_0. This idea of scaling can even be generalised to trapezoidal, Gaussian or even more general membership functions [6], [7], [8]. However, in the context of this paper, it is sufficient to consider constant, but possibly different scalings for the left- and right-hand side of x_0.

With this interpretation of the membership degree as a dual concept of a scaled and cut-off distance to x_0, the firing degree (1) of a rule R can be considered in the following way. Let $s^{(R)} = s_1^{(R)}, \ldots, s_p^{(R)}$ be the prototypical values associated with the fuzzy sets $\mu_R^{(i)}$ occurring in the premise of the rule

R. This means, we have

$$d_{\mu_R^{(i)}}(s_i^{(R)}, s) \;=\; 1 - \mu_R^{(i)}(s_i).$$

for an object $s = (s_1, \ldots, s_p)$ to be classified. This means that

$$\mu_R^{(i)}(s_i) \;=\; 1 - \min\{d_{\mu_R^{(i)}}(s_i^{(R)}, s), 1\}$$

holds. Using the Łukasiewicz t-norm (2), the firing degree of rule R can be understood as the overall similarity of the object s to the prototypical object $s^{(R)}$:

$$\text{Sim}\left(s^{(R)}, s\right) \;=$$

$$\max\left\{\left(\sum_{i=1}^{p}\left(1 - \min\{d_i(s_i^{(R)}, s_i)\right), 1\}\right) \;-\; (p-1), 0\right\}. \tag{4}$$

Assuming for the moment that the d_i values are quite small, which means we can neglect the minimum and the maximum in (4), we obtain the overall similarity degree of the object s to the prototypical object $s^{(R)}$ by

$$\begin{aligned}\text{Sim}\left(s^{(R)}, s\right) \;&=\; \left(\sum_{i=1}^{p}(1 - d_i(s_i^{(R)}, s_i))\right) \;-\; (p-1)\\ &=\; 1 \;-\; \sum_{i=1}^{p} d_i(s_i^{(R)}, s_i).\end{aligned} \tag{5}$$

Finally, when we again understand similarity as the dual concept of distance in the sense of

$$\text{Dist} \;=\; 1 - \text{Sim}, \tag{6}$$

we derive from (4)

$$\text{Dist}\left(s^{(R)}, s\right) \;=\; \sum_{i=1}^{p} d_i(s_i^{(R)}, s_i). \tag{7}$$

This means that this overall distance is computed by aggregating the (scaled and cut-off) distances d_i by the 1-norm. In case of the Euclidean norm (the 2-norm), one would use

$$\text{Dist}_{\text{Euclidean}}\left(s^{(R)}, s\right) \;=\; \sqrt{\sum_{i=1}^{p} d_i(s_i^{(R)}, s_i)^2}.$$

However, in this paper, we will only consider the distance (7) associated with the 1-norm.

A fuzzy classifier will classify an object s based on the rule R for which s yields the highest firing degree. This firing degree can be interpreted as the similarity degree to the corresponding prototypical object $s^{(R)}$ associated with rule R. Changing from similarity to the dual concept of distance in the sense of (6), the fuzzy classifiers will classify object s to the class of the prototypical object $s^{(R)}$ (i.e. the class associated with rule R) to which s has the smallest distance.

With these ideas in mind, we can now establish the connection between fuzzy classifiers and 1-nearest neighbour classifiers. Assume a nearest neighbour classifier uses a distance function of the form

$$d(s^{(\text{sdb})}, s) \;=\; \sum_{i=1}^{p} d_i(s_i^{(\text{sdb})}, s_i)$$

where $s^{(\text{sdb})}$ is an object from the sample database and s is an object to be classified. Then each object $s^{(\text{sdb})}$ in the sample database induces a fuzzy rule with the fuzzy sets

$$\mu_i^{\left(s^{(\text{sdb})}\right)}(s_i) \;=\; 1 - \max\left\{d_i\left(s_i^{(\text{sdb})}, s_i\right), 1\right\}.$$

As long as the distances d_i are small at least to the closest object from the sample database, i.e. $d_i \leq 1$ for all i for this object, the corresponding fuzzy classifier will yield exactly the same classification as the nearest neighbour classifier.

The intention in this paper is not to construct a fuzzy classifier based on a nearest neighbour classifier. The reason is that a sample database of even more than one hundred objects might still be considered as reasonable. However, a fuzzy classifier with more than one hundred rules is very difficult to interpret. The purpose in this paper is to provide an explanation for the classification result of the nearest neighbour classifier based on fuzzy rules. If a user wants to know why the nearest neighbour classifier has decided to assign an object s to a certain class, he can have a look at the fuzzy rules associated with the closest or most similar objects to s. The fuzzy sets in these rules will also show the importance of the single attributes of s for its classification. In this way, we do not try to understand the rule base as a whole, but simply use it to explain single classification decisions.

The learning scheme for a nearest neighbour classifier which is suitable for this scheme is based on principles from classifier systems. Before we present the learning algorithm in detail in section V, we briefly introduce the necessary ideas from classifier systems in the following section.

IV. Nearest Neighbour Classifiers and Classifier Systems

Learning classifier systems were introduced by [9], [10]. Their learning scheme is based on evolutionary principles, especially genetic algorithm techniques. Classifier systems are collections of if-then rules where each single rule is called a classifier. In the simplest case, the premise of a rule consists of a fixed number of (binary coded) conditions. The conclusion part of the rule defines (binary coded) actions to be taken, when the premise of the rule is satisfied. In contrast to our view of classification, classifier systems are intended to act as sensor actor systems. Nevertheless, we can adapt some of their learning concepts.

A classifier systems acts as a rule base to determine, which action has to be taken in which situation. Each classifier of the classifier systems represents one rule. In a similar sense a nearest neighbour classifier act as "rule base" for classification where each case in the sample database corresponds to the rule

> If an object to be classified is very similar to me,
> then it should be assigned to the same class as me.

During the learning (or reinforcement) phase of a classifier system each rule is evaluated, depending on how often its premise is satisfied and how good the initiated action was. The evaluations of the rules are used as fitness values in a genetic algorithm for selecting, mutating and combining rules.

We adopt this idea, when we construct the sample database from the available classified cases. An object in the sample database is evaluated based on how many other objects it has as nearest neighbours and how many of them are classified correctly. The detailed algorithm will be described in the next section.

V. The Algorithm

We have motivated an evaluation scheme that can help us to construct a sample database for our nearest neighbour classifier. However, so far we have not touched the question of how to adapt the distance measure.

We enable this in the following way. Here we assume that all attributes are continuous-valued and that we use as distance

$$d(s,s') \;=\; \sum_{i=1}^{p} d_i(s,s'). \tag{8}$$

where s' is the object to be classified, s is an object from the sample database, and s_i and s'_i is the ith attribute of s, respectively s'. In order to keep the distance adaptation interpretable, we allow for each object s' in the sample database and for each of its attributes an individual scaling in both directions. This means, for s and each of its attributes, we have two scaling factors $c_{s,i}^{(l)}$ and $c_{s,i}^{(r)}$. We define the scaled distance of an object s' to s with respect to the ith attribute by

$$d_i(s,s') \;=\; \begin{cases} c_{s,i}^{(l)} \cdot |s_i - s'_i| & \text{if } s'_i \le s_i \\ c_{s,i}^{(r)} \cdot |s_i - s'_i| & \text{if } s'_i \ge s_i \end{cases}$$

In the beginning all scaling factors are set to the value one. Later on, when we adapt the distances, we do this by adapting the scaling factors. When we try to adapt the distance measure, i.e. the individual scaling factors for an object, for each of its attributes we have to consider both the left and the right scaling factor. Both of them are adapted individually.

When we want adapt the scaling factors for an object in the sample database, for each of its attributes and both directions (associated with the scaling factors $c_{s,i}^{(l)}$ and $c_{s,i}^{(r)}$, respectively), we check whether

$$c_{s,i}^{(l,\text{new})} \;=\; \sigma \cdot c_{s,i}^{(l,\text{old})}$$

(and analogously for $c_{s,i}^{(r)}$) leads to a reduction of the misclassification rate. $0 < \sigma < 1$ is a learning rate. In case of a reduction of the misclassification rate, we will modify the corresponding scaling factor accordingly. In principle, we could also check, whether increasing a scaling factor leads to an improvement. However, as will be seen later on, preferring the decrease of scaling factors will help us in interpretung the nearest neighbour classifer, since attributes with (almost) zero scaling factors can be neglected.

The concrete steps of our algorithm to construct a sample database from our training set are the following ones:

First specify a tolerance bound for the misclassification rate on the training set. Then:

1) Initialise the sample database randomly (stratified, i.e. make sure that the classes are represented in the database with the same proportions as in the training set.
2) Apply the above described scaling scheme to the objects from the (random) sample database.
3) Evaluate the instances with the scaled distance in the database. The fitness of an object s in the sample database is defined by

   $$f(s) \;=\; \frac{\#\text{ correctly classified}}{1 + \#\text{ incorrectly classified}}$$

   where

   $$\#\text{ correctly classified} \;+\; \#\text{ incorrectly classified}$$

   is the number of data in the training set for which the corresponding object in the sample database is the nearest neighbour.
4) Remove all objects from the sample database with a fitness of zero.
5) Compute the confusion matrix, i.e. the matrix indicating for each class how many of its objects are classified to each other class, and determine which class is the one with the highest misclassifictaion rate. If class c_1 is wrongly classified in the majority of cases to class c_2, remove an instance from class c_2 from the sample database with a probability inversely proportional to its fitness.
6) Pick an object randomly from the training set that belongs to class c_1, but that is not contained in the sample database and add it to the sample database, unless this object has been removed from the sample database in a previous step.
7) Carry out scaling with this new object.
8) If the specified misclassification rate for the training set is still exceeded, repeat from step 3.

Note that we can also replace the misclassification rate by an arbitrary loss function

$$L : \mathcal{C} \times \mathcal{C} \to [0, \infty).$$

$L(c_1, c_2)$ is interpreted as the loss (of money) that will be caused, when we classify an object from class c_1 to class c_2. We assume that $L(c,c) = 0$ holds for all classes. The misclassification rate is based on the special loss function

$$L(c_1,c_2) \;=\; \begin{cases} 0 & \text{if } c_1 = c_2 \\ 1 & \text{otherwise.} \end{cases}$$

In many applications, this symmetric loss function is not realistic. Consider for example the case of an essential part of an aeroplane. The classification task during maintenance is to decide, whether this part will function until the next maintenance. If the part will function until the next maintenance, but the decision is made that we exchange the part for a new one already, the loss might be the costs to replace this part. However, when misclassify the part into the category that it will function, but it will fail in reality and its failure will cause the aeroplane to crash, the costs will be much higher, including

not only the loss of the aeroplane, but also the death of the passengers and the crew.

In order to incorprate such a loss function, we have to make the following changes in the above algorithm.

- The specified tolerance bound refers to the expected loss, not the misclassification rate.
- Define the fitness of an onject s in step 3 by

$$f(s) = \frac{\#\text{ correctly classified}}{1 + \text{ sum of the losses of all objects misclassified by } s}$$

- Instead of computing the confusion matrix in step 5, we compute the loss matrix. This means, instead of simply counting for each entry (i, j) in the matrix, how many objects are classified from class c_i to class c_j, we add up the corresponding losses. Find the highest entry – the greatest loss – in this matrix. If the highest loss is found at the entry (i, j), then remove an instance from class c_i from the sample database with a probability inversely proportional to its fitness.
- In step 6, pick an object randomly from the training set that belongs to class c_j, but that is not contained in the sample database.
- Consider the sum of all losses divided by the number of objects in the training set, instead of the misclassification rate in step 8.

In order to demonstrate the performance of our nearest neighbour classifier, we have carried out experiments with three simple data sets from the UCI Machine Learning Repository (http://www.ics.uci.edu/∼mlearn/MLRepository.html), namely the Iris and the wine data set. We have applied 10-fold (stratified) cross validation. This means, the data set was partitioned into 10 equally sized subsets. The distribution of the classes in each subset is the same as in the whole data set. Then we take out one of the 10 subsets and construct our nearest neighbour classifier based one the remaining 90% of the data. After the classifier has been constructed, we test its performance on the 10% of the data that were left out. This procedure is applied to each of the 10 subsets. Table I shows the average performance as well as the average size of the sample database.

TABLE I
RESULTS

data set	average database size	average misclassification rate
iris	18.2	2.7%
glass	19.4	2.9%

VI. DERIVING FUZZY RULES

In addition to accuracy, interpretability is an important issue for any kind of classification system [11]. Taking a closer look to the distances, we can find the following. Most of the scaling factors have been decreased to almost zero. Looking at a typical sample database for the iris or the wine data, we find scaling factor configurations where most of the scaling factors have decayed to (almost) zero (here a scaling factor less than 0.1 is considered as almost zero). Table II shows an instance from the sample database of the wine data set.

TABLE II
AN INSTANCE FROM THE WINE SAMPLE DATABASE

attribute	left scaling factor	right scaling factor
Total phenols	0.3	
Nonflavanoid phenols		1.0
Proanthocyanins	1.0	0.3
Colour intensity		0.3
OD280/OD315		1.0
Proline		1.0

We have only entered those scaling factors in the table greater than 0.1. The wine data set uses 13 attributes for predicting the class. But this instance determines the distance/similarity on the basis of only 6 attributes. Note that the scaling factors refer to normalised attributes.

Although looking at the scaling factors provides already some interesting information, it might be desirable, to have a simpler representation for non-expert users. For this we use the very intuitive framework of fuzzy rules that is easily understandable for a non-expert user. We make use of the correspondence between nearest neighbour classifiers and fuzzy classifiers that we have established in section III.

The only problem that occurs here is that equation (4) applies only in the case, when the values $d_i(s_i^{(R)}, s_i)$ are small enough. This can be avoided, by applying an additional (suitably small) scaling factor to all objects and attributes to guarantee for overall small distances. With this additional scaling factor we obtain a one-to-one correspondence between fuzzy rules and nearest neighbour classifiers. However, the additional scaling factor would lead to very wide (triangular) fuzzy sets. Therefore, for the visualisation and user friendly interpretation of the nearest neighbour classifier, we propose not to use the corrected fuzzy sets (incorporating the additional small scaling factor), but to use the original ones just taking the individual scaling factors $c_{s,i}^{(l)}$ and $c_{s,i}^{(r)}$ into account. Note that this will show an effect only in the case of lower membership degrees, i.e. when even the closest object in the sample database is not very similar to the object to be classified.

In this way, we can construct suitable fuzzy sets for each object in the sample database and each of its attributes. A fuzzy set for an attribute of an instance is constructed by choosing the membership degree one at the corresponding value of the attribute and decreasing the membership degrees with a slope proportional to the right scaling factor to the right and with a slope proportional to the left scaling factor to the left.

Figure 2 shows the fuzzy sets associated with the attributes *Proanthocyanins* and *Colour intensity*, respectively, of the object described in table II. P_0 and C_0 stand for the value of the corresponding attribute of the considered object. We

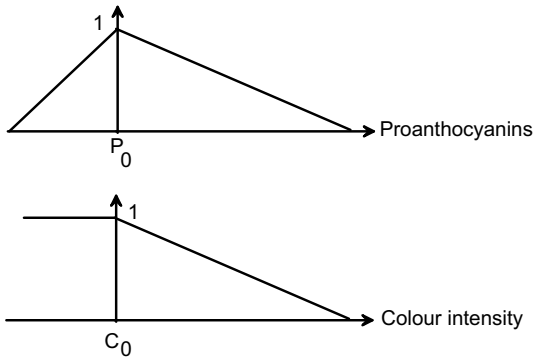

Fig. 2. Fuzzy sets induced by the scaling factors

can use these fuzzy sets and corresponding fuzzy sets for the remaining four attributes in table II to construct a fuzzy if-then rule for this object and all other instances in the sample database. The conclusion part is the class of the corresponding instance. In this way, we actually build a fuzzy rule base for the classifier.

VII. Conclusions

We have demonstrated, how an interpretable adaptive nearest neighbour classifier can be constructed by transforming it into an equivalent classifier using fuzzy rules. We have introduced a scheme for keeping the number of instances in the sample database (or, equivalently, the fuzzy rules) small as well as an adpation scheme for the distances of the nearest neighbour classifier that correspond to the similarity degrees and the shapes of the fuzzy sets.

Further research will concentrate on how we can enforce that most of the scaling factors will decrease to zero-values automatically, so that the corresponding fuzzy rules will use only a small number of attributes.

References

[1] B. Dasarathy, Ed., *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, California: IEEE Computer Society Press, 1991.

[2] T. Hastie and R. Tibshirani, “Discriminant nearest neighbour classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 607–616, 1996.

[3] L. Kuncheva, *Fuzzy Classifier Design*. Heidelberg: Springer, 2000.

[4] F. Klawonn and E.-P. Klement, “Mathematical analysis of fuzzy classifiers,” in *Advances in Intelligent Data Analysis*, X. Liu, P. Cohen, and M. Berthold, Eds. Berlin: Springer, 1997, pp. 359–370.

[5] B. von Schmidt and F. Klawonn, “Fuzzy max-min classifiers decide locally on the basis of two attributes,” *Mathware and Soft Computing*, vol. 6, pp. 91–108, 1999.

[6] F. Klawonn, “Fuzzy sets and vague environments,” *Fuzzy Sets and Systems*, vol. 66, pp. 207–221, 1994.

[7] F. Klawonn and R. Kruse, “Fuzzy partitions and transformations,” in *3rd IEEE Conference on Fuzzy Systems*. Piscataway: IEEE Press, 1994, pp. 1269–1273.

[8] F. Klawonn, “Reducing the number of parameters of a fuzzy system using scaling functions,” *Soft Computing*.

[9] J. Holland and J. Reitmann, “Cognitive systems based on adaptive algorithms,” in *Pattern-Directed Inference Systems*, D. Watermann and F. Hayes-Roth, Eds. New York: Academic Press, 1978, pp. 313–329.

[10] J. Holland, K. Holyoak, R. Nisbett, and P. Thagard, *Induction: Processes of Inference, Learning, and Discovery*. Cambridge, MA: MIT Press, 1986.

[11] Casillas, J. Cordón, O. Herrera, and L. F. Magdalena, Eds., *Interpretability Issues in Fuzzy Modelling*. Berlin: Springer, 2003.