Tree-Oriented Hypothesis Generation for Interpretable Fuzzy Rules

Jens Jäkel, Lutz Gröll and Ralf Mikut
Forschungszentrum Karlsruhe GmbH, Institute of Applied Informatics (IAI)
P.O. Box 3640, D-76021 Karlsruhe, Germany
Phone: +49-7247/82-5736, Fax: +49-7247/82-5785
Email: {jaekel,groell,mikut}@iai.fzk.de

ABSTRACT: The paper presents a new approach to the automatic data-based generation of fuzzy rules. This is based on a tree-oriented rule induction algorithm and rule pruning. The hypothesis generation applies a set of measures for evaluation of fuzzy rules with respect to approximation quality, importance, clearness etc. In order to improve flexibility and interpretability linguistic hedges are used to create derived linguistic terms.

KEYWORDS: fuzzy system, fuzzy rule, rule induction, decision tree, linguistic modifier, interpretability

1 INTRODUCTION

Inductive learning strategies for the generation of rules from a set of examples have been studied for a long time initialized by the seminal work (Hunt et al., 1966). Tree-oriented approaches (Breiman et al., 1984; Quinlan, 1986) play an important role. However, their results can be insufficient if the examples contain vagueness, ambiguity or noise, as it is typical for many real-world applications. Among the theories to cope with uncertainty, fuzzy set theory (Zadeh, 1965) has become especially popular. A reason for this is that fuzzy set theory and fuzzy logic link this uncertainty to the uncertainty and ambiguity in human linguistic expressions.

The basic idea of inductive learning is to find a concept description which is able to split the example set into the set of positive and negative instances according to a given class. The concept description can be represented by a set of rules with the same conclusion. The rule premises contain tests on the values of variables, e. g. "height=tall". To decide which test to apply, i. e. which variables to include into the premises, often information theoretic measures are used.

Recently proposed methods can be distinguished by the way they handle the uncertainty present in the examples. A first group of methods uses gradual (fuzzy) class membership values and "fuzzified" versions of the information theoretic measures to induce fuzzy decision trees or fuzzy rules (Yuan and Shaw, 1995; Wang et al., 1999). The second group of methods starts from decision trees, generated e. g. with the ID3 algorithm (Quinlan, 1986), which is translated into fuzzy rules using fuzzy sets for describing values (linguistic terms) of variables (Maher and St. Clair, 1993; Chi and Yan, 1996; Otto and Malberg, 1998).

The paper presents a new method following the second approach. This method

- generates meaningful and comprehensible rules using the ID3 algorithm and rule pruning,
- selects cooperating rules to form a compact rule base,
- employs different measures evaluating accuracy, relevance, and clearness of rules,
- applies linguistic hedges to build derived linguistic terms in order to obtain comprehensible rule premises.

The paper is organized as follows: Section 2 briefly describes specific features of the fuzzy system, explains the generation of derived terms and their membership functions, and discusses a new inference scheme capable to process partially redundant and even contradictory rules resulting from the use of derived terms. Section 3 introduces measures for fuzzy rule evaluation and rating used in the rule generation algorithm which is topic of Section 4. Section 5 presents experimental results for three real-world data sets. Conclusions are given in Section 6.

2 PRELIMINARIES

This section describes the fuzzy system emphasizing the specific features. A more detailed presentation can be found in (Jäkel et al., 1998).

Notation: The following notation is used throughout the paper. A, B denote linguistic terms. μ_A is the membership function of a subset A, $\mu_A(x)$ the membership degree of x to A. $\boldsymbol{\mu}_x$ denotes the value of the linguistic variable for x, a vector of membership degrees. $\mathbf{1}_n$ is a n-dimensional vector of ones, $\mathbf{O}_{m\times n}$ a (m,n)-dimensional matrix of zeros. $\boldsymbol{M}_{\bullet j}$ denotes the j-th column of $\boldsymbol{M}$. The notation $\boldsymbol{M} \underset{\text{nat}}{\geq} \mathbf{O}_{m\times n}$ signifies that all elements of $\boldsymbol{M}$ are non-negative.

Rules take the form

$$\text{IF } (x_1 = A_{1,i} \text{ OR } \dots \text{ OR } A_{1,j}) \text{ AND } (x_2 = A_{2,i} \text{ OR } \dots) \text{ AND } \cdots \text{ AND } (x_m = A_{m,l} \text{ OR } \dots) \text{ THEN } y = B_k,$$

where the premises describing an input situation are not necessarily complete, i. e. not all input variables have to be specified. The $A_{i,j}$ are linguistic terms. The first index $i = 1, \dots, m$ refers to the number of the input variable, the second one $j = 1, \dots, m_j$ to the number of the linguistic term. In the conclusion the output is assigned a linguistic term or a class, respectively, depending on the modelling task. These are labelled with B_k, $k = 1, \dots, n$. The number of rules in a rule base is denoted with q.

Linguistic terms $A_{i,j}$ are characterized by membership functions $\mu_{A_{i,j}}$ which take values from $\{0,1\}$ in the case of ordinary subsets (crisp rules) and $[0,1]$ in the case of fuzzy subsets (fuzzy rules). Linguistic terms by themselves can be regarded as ordinary sets. The use of disjunctions like "$x_1 = A_{1,i}$ OR $A_{1,j}$" gives rise to the introduction of *derived* linguistic terms labelling unions of ordinary or fuzzy subsets. Using linguistic hedges like *at least, rather* or *not*, derived terms lead to compact and transparent rules premises. Note that in contrast to (Zadeh, 1972), here the application of linguistic hedges is concerned with labelling unions of linguistic terms and not the modification of single terms. For distinction, the a priori given terms are called *primary* terms. It is assumed that the membership functions of all primary terms of a variable x_i for each value of x_i sum up to one. Then, in the case of crisp rules the subsets form a partition of the input domain of x_i, X_i. In the case of fuzzy rules they form a so-called constrained or standard fuzzy partition.

As operators like *Maximum* or *Sum* fail in the construction of membership functions of derived terms (Jäkel et al., 1998; Jäkel et al., 1999) a pair of "switching" operators for intersection and union is defined as follows:

$$\mu_{A\cap B}(x) = \begin{cases} 0 & \text{if } A \cap B = \emptyset \\ \min\{\mu_A(x), \mu_B(x)\} & \text{else,} \end{cases} \tag{1}$$

$$\mu_{A\cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_{A\cap B}(x) = \begin{cases} \mu_A(x) + \mu_B(x) & \text{if } A \cap B = \emptyset \\ \max\{\mu_A, \mu_B\} & \text{else.} \end{cases} \tag{2}$$

Here, $A \cap B$ means the intersection of two linguistic terms regarded as ordinary sets. For example, the terms *at least large* and *large* have an intersection, namely the term *large* which is a subset of *at least large*. Whereas the primary terms *medium* and *large* do not intersect. These operators consider the subset relations of linguistic terms when forming the membership function of a derived term. To test subset relations or $A \cap B = \emptyset$, resp., there are two alternatives. The first one assumes that a hierarchy of the linguistic terms is defined, e. g. the term *positive* includes *rather small*, *medium* and *rather large*, *rather small* includes *very small* and *small* etc. Figure 1 gives an example for a hierarchy of linguistic terms and their membership functions. The second one uses the membership functions: $A \cap B \neq \emptyset$ if there is a x for which $\mu_A(x) = \mu_B(x) = 1$ holds, otherwise $A \cap B = \emptyset$.

Each rule premise P_r is characterized by a multi-dimensional membership function μ_{P_r}. μ_{P_r} results from combination of the one-dimensional membership functions using the algebraic product. $\mu_{P_r}(\boldsymbol{x})$ denotes the value of μ_{P_r} for given input vector $\boldsymbol{x}$, further called rule activation. As a consequence of incomplete premises and the use of derived terms even in case of ordinary subsets (crisp rules), premises can be partially redundant, i. e. the subsets specified by the premises overlap. Presenting an input vector generally the activation of more than one rule is non-zero. To resolves this, different inference strategies can be employed, like "best expert" or (weighted) averaging. A "best expert" strategy, where the conclusion of the rule with highest activation is chosen, is especially appropriate for crisp rules. A weighted average strategy (most popular the sum-prod inference) can produce unexpected results in the case of highly redundant rules or incomplete rule bases, i. e. if not the whole input space is covered by rule premises. The reason is that in the first case (high redundancy) the sum of rule activations is greater than one and in second case (no applicable rule) zero. Therefore, a modified inference strategy has been introduced in (Jäkel et al., 1998).

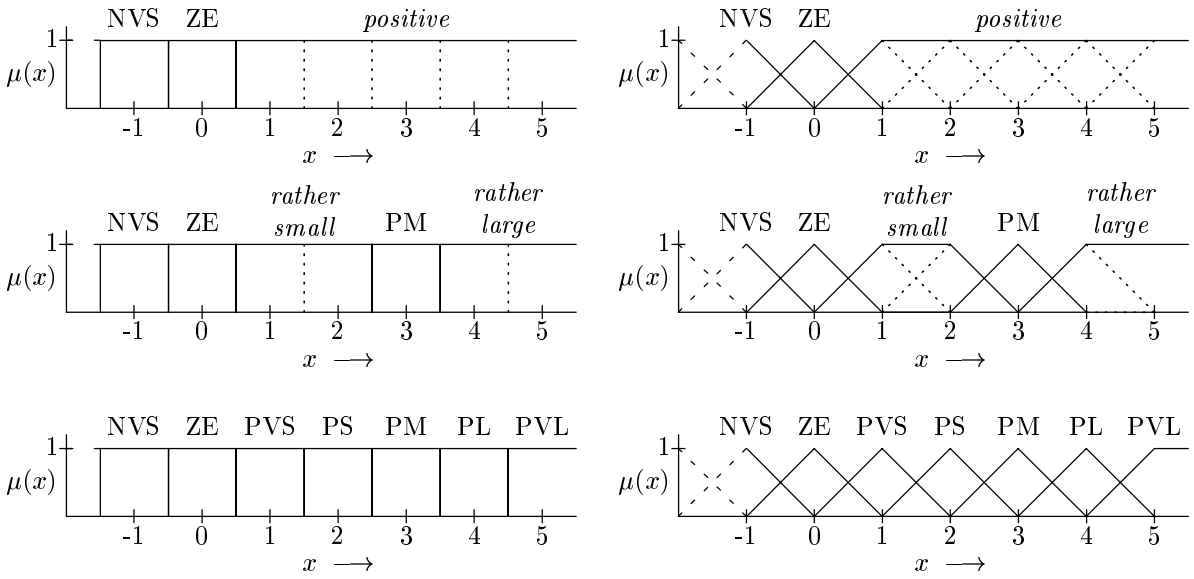

Figure 1: Primary and derived linguistic terms and their membership functions, ordinary (crisp) subsets (left) and fuzzy subsets (right); N *negative*, P *positive*, V *very*, S *small*, M *medium*, L *large*
, ZE *zero*

The idea of this new inference strategy is to consider the redundancy of rule premises. The composition of the results of the activated rules uses the multi-dimensional and multi-variable extension of the union operator (2):

$$\mu_{P_1\cup\cdots\cup P_q}(\boldsymbol{x}) = \sum_{i=1}^{q}\mu_{P_i}(\boldsymbol{x}) + (-1)^1\sum_{i<j}^{q}\mu_{P_i\cap P_j}(\boldsymbol{x}) + \cdots + (-1)^{q-1}\mu_{P_1\cap\cdots\cap P_q}(\boldsymbol{x}). \tag{3}$$

The membership value of the intersections $\mu_{P_i\cap P_j}(\boldsymbol{x})$, assuming P_i is $(x_1 = A_{1,i})$ AND $(x_2 = A_{2,i})$ AND ... AND $(x_m = A_{m,i})$ and P_j, respectively, is calculated as follows:

$$\mu_{P_i\cap P_j}(\boldsymbol{x}) = \mu_{A_{1,i}\cap A_{1,j}}(x_1)\mu_{A_{2,i}\cap A_{2,j}}(x_2)\cdots\mu_{A_{m,i}\cap A_{m,j}}(x_m)$$

and $\mu_{A_{l,i}\cap A_{l,j}}(x_l)$ according to (1), analogically for intersections of more than two premises.

In matrix notation the inference can be written as

$$\hat{\boldsymbol{\mu}}_y = \boldsymbol{C}\left[\mu_{P_1}(\boldsymbol{x})\ \ldots\ \mu_{P_q}(\boldsymbol{x})\ \mu_{P_1\cap P_2}(\boldsymbol{x})\ \ldots\ \mu_{P_1\cap\cdots\cap P_q}(\boldsymbol{x})\right]^T = \boldsymbol{C}\boldsymbol{\mu}_{P'}(\boldsymbol{x}) = \boldsymbol{C}_1\boldsymbol{C}_2\boldsymbol{\mu}_{P'}(\boldsymbol{x}), \tag{4}$$

where $\hat{\boldsymbol{\mu}}_y$ is the vector of membership values of the output fuzzy sets and $\boldsymbol{\mu}'_P(\boldsymbol{x})$ comprises the vector of rule activations $\boldsymbol{\mu}_P(\boldsymbol{x})$ and the membership values of all non-empty intersections of premises. $\boldsymbol{C}_1$ with $c_{1,ij}\in\{0,1\}$ codes the conclusions of the rules and assigns them to the premises. $\boldsymbol{C}_2$ considers the redundancy of the premises according to (3) and modifies the vector of rule activations accordingly. As a result of this inference

$$\mathbf{1}_n^T\hat{\boldsymbol{\mu}}_y \le 1$$

holds with equality in case of a complete rule base.

For rule generation and evaluation a compact notation will be used:

$$\underbrace{\left(\hat{\boldsymbol{\mu}}_y[1]\ \hat{\boldsymbol{\mu}}_y[2]\ \ldots\ \hat{\boldsymbol{\mu}}_y[N]\right)}_{\hat{\boldsymbol{Y}}} = \boldsymbol{C}_1\underbrace{\boldsymbol{C}_2\left(\boldsymbol{\mu}_{P'}(\boldsymbol{x}[1])\ \boldsymbol{\mu}_{P'}(\boldsymbol{x}[2])\ \ldots\boldsymbol{\mu}_{P'}(\boldsymbol{x}[N])\right)}_{\boldsymbol{P}}, \tag{5}$$

where $\hat{\boldsymbol{Y}}$ is the matrix of predicted output membership values, $\boldsymbol{P}$ the matrix of modified rule activations and N the number of data tuples. Furthermore, the completeness of the input and output partition has to be assured. In the case of incompleteness this has to be accomplished by adding the complementary premise or conclusion, respectively, i. e.

$$\boldsymbol{P} := \begin{pmatrix}\boldsymbol{P}\\ \mathbf{1}_N^T - \mathbf{1}_q^T\boldsymbol{P}\end{pmatrix} \quad \text{or} \quad \boldsymbol{Y} := \begin{pmatrix}\boldsymbol{Y}\\ \mathbf{1}_N^T - \mathbf{1}_n^T\boldsymbol{Y}\end{pmatrix}, \tag{6}$$

where $\boldsymbol{Y} = (\boldsymbol{\mu}_y[1]\boldsymbol{\mu}_y[2]\ldots\boldsymbol{\mu}_y[N])$.

3 MEASURES FOR RULE EVALUATION

The quality of a fuzzy model has to be assessed considering its main purpose which can be predictive, descriptive or explanatory. Especially in the latter case the quality of individual rules plays an important role with regard to interpretability. Therefore, measures for rule evaluation have to take into account the quality of individual rules and groups of rules, the modelling objective and the level of abstraction.

In the context of automatic rule generation in (Jäkel et al., 1998; Jäkel et al., 1999) a set of measures is proposed comprising

- the approximation error,
- the clearness error, and
- the elementarity error.

The *approximation error* F_P represents a measure of the model accuracy. It allows to compare different rules or rule bases.

The approximation error is defined as the solution of the constrained least squares problem

$$F_P = \|\boldsymbol{R}\boldsymbol{P} - \boldsymbol{Y}\|_F \to \min_{\boldsymbol{R}}, \quad \text{subject to } \mathbf{1}_n^T \boldsymbol{R} = \mathbf{1}_q^T \text{ and } \boldsymbol{R} \underset{\text{nat}}{\geq} \mathbf{O}_{n\times q} \tag{7}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The constraints implied on $\boldsymbol{R}$, each column sum equals one and it has only non-negative elements, make it comparable to $\boldsymbol{C}_1$. But in contrast to $\boldsymbol{C}_1$, the elements of $\boldsymbol{R}$ can take values between zero and one. In the case of ordinary subsets assigned to the linguistic input terms each column vector of $\boldsymbol{R}$ represents a conditional probability distribution of the output classes for the given premise. In the case of fuzzy subsets results a conditional possibility distribution. This means, each premise is generally assigned more than one conclusion (output class) to a certain degree. The latter gives rise to an interpretation of $\boldsymbol{R}$ as rule weight matrix.

In the definition of $\boldsymbol{R}$ and the approximation error, resp., for the output not the real values $y[k]$ but the fuzzified data $\boldsymbol{\mu}_y[k]$ is used. In this way, the semantics of the output fuzzy sets is taken into account.

The *clearness error* is given with

$$F_K = 1 - \|\boldsymbol{R}_{\bullet j}\|_\infty. \tag{8}$$

This measure evaluates the deviation from a non-ambiguous assignment of a conclusion to a premise. In addition, it rates the fuzzy partition of the output space and gives hints for its improvement (e. g. if $r_{ij} \approx r_{i+1,j}$ then the definition of a new or shift of an existing fuzzy set could give improved results).

Further, the *elementarity error*

$$F_E = 1 - \min_{i\neq j} \|\boldsymbol{R}_{\bullet i} - \boldsymbol{R}_{\bullet j}\|_\infty \tag{9}$$

is introduced. This measure estimates the possibility of rule merging. Generalizing F_E as the distance between all column vectors of $\boldsymbol{R}$ (analysis of the singular values) the quality of the input partition is rated.

Using the approximation and the clearness errors, a new criteria for individual rules is defined as

$$Q = \left(1 - \frac{\min\{F_P, F_P^0\}}{F_P^0}\right)^\alpha (1 - F_K)^\beta \tag{10}$$

where the parameters α and β give the possibility to weight the objectives approximation quality and interpretability. F_P^0 is the approximation error for a rule whose premises covers the whole input space, i. e. $\mu_{P_0}(\boldsymbol{x}) = 1$ for every $\boldsymbol{x}$.

Alternative measures for the evaluation of individual rules are the Relevance Index (Kiendl et al., 1991; Krone and Kiendl, 1996) and related measures (Jessen and Slawinski, 1998).

The discussed measures mainly provide a means for the evaluation of existing rules. In this section, the generation of hypotheses and the use of the presented measures in this process will be explained.

The rule generation algorithm comprises four steps:

(i) induction of a decision tree

(ii) translation of the decision tree into fuzzy rules

(iii) pruning and logical reduction of fuzzy rules

(iv) selection of fuzzy rules

In the following these steps are described in more detail.

1. Induction of decision tree Applying the ID3 algorithm (Quinlan, 1986) a decision tree is induced using variables with linguistic terms characterized by ordinary subsets. This means, the ID3 algorithm treats all variables as categorical ones. The definition of only primary terms is presumed. The membership functions take only values from $\{0, 1\}$ (crisp subsets) and form a partition of the input domains.

A decision tree consists of nodes and branches. A node indicates a linguistic term or class of the output and contains a test on an input variable ($x_i = ?$) if it is a decision node. Otherwise it is called a leaf. For each outcome of a test, a linguistic term of the tested input, a branch starts from the decision node. Figure 2 shows an example for a decision tree.

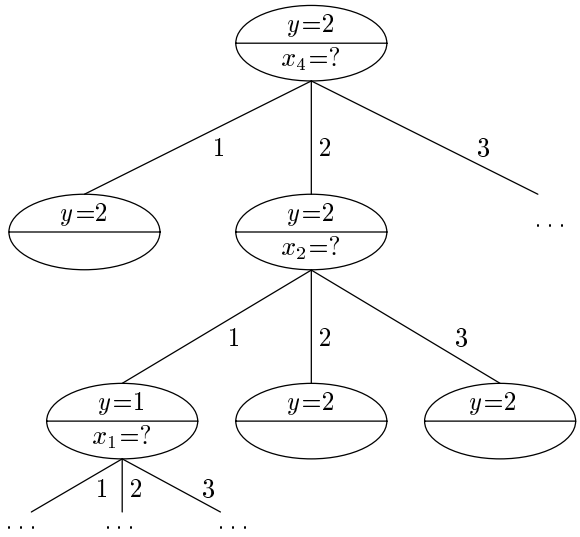

Figure 2: Section of a decision tree

The construction algorithm consists of step-wise splits of the set of training examples using a test $x_i = ?$ in each step. A decision node receives the most frequent output term or class in the respective subset of examples. Alternative tests (splits) are evaluated by an information theoretic measure, the mutual information between the test and the output term/class (called gain criterion in (Quinlan, 1993)), which is to be maximized:

$$H(y; x_i) = -\sum_{j=1}^{m_i} \sum_{k=1}^{n} p(A_{i,j} \wedge B_k) \operatorname{ld} \left(\frac{p(A_{i,j}) p(B_k)}{p(A_{i,j} \wedge B_k)} \right) \rightarrow \operatorname*{Max}_{i=1,\ldots,m_i} \quad \text{with} \tag{11a}$$

$$p(A_{i,j}) = \frac{n(x_i = A_{i,j})}{N},\ p(B_k) = \frac{n(y = B_k)}{N},\ p(A_{i,j} \wedge B_k) = \frac{n(x_i = A_{i,j} \wedge y = B_k)}{N}, \tag{11b}$$

where N is the number of training examples and $n(x_i = A_{i,j})$ the number of examples where $x_i = A_{i,j}$ etc. and m_i the number of terms of x_i.

A further development of the split criterion, a normalization with entropy of the test $x_i = ?$, $H(x_i)$, leads to the C4.5 algorithm (Quinlan, 1993). The argument for this modification, that the criterion (11) favours test with many outcomes, does not apply here as the number of linguistic terms is nearly the same for all variables.

Assuming noise-free data, the algorithm produces a decision tree, which can classify each example of the training set correctly. In the case of noisy training data and relatively small number of examples the probability of misclassification of unseen examples can be substantially, i. e. the decision tree shows a poor generalization ability.

2. Translation into fuzzy rules For each leaf a rule is established whose premise consists of a conjunction of all variables and the respective linguistic terms contained in the decision nodes on the path from the root node to the leaf. The premises of these rules are mutually exclusive. By assigning a fuzzy set instead of the ordinary set to linguistic terms as described in Section 2 a set of fuzzy rules are obtained. Figure 3 contains the rule extracted from the subtree in Figure 2.

$$
\begin{array}{ll}
1 & \text{IF } (x_4 = 1) \text{ THEN } y = 3 \\
2 & \text{IF } (x_4 = 2) \text{ AND } (x_2 = 2) \text{ THEN } y = 2 \\
3 & \text{IF } (x_4 = 2) \text{ AND } (x_2 = 3) \text{ THEN } y = 2 \\
4 & \text{IF } (x_4 = 2) \text{ AND } (x_2 = 1) \text{ AND } (x_1 = \dots) \dots \text{ THEN } y = \dots \\
\vdots & \qquad\qquad\qquad\qquad \vdots
\end{array}
$$

Figure 3: Rules extracted from the decision tree in Figure 2

3. Pruning and logical reduction To improve the generalization ability, the fuzzy rules are pruned using two kinds of modifications: 1) deleting a variable from the rule premise and 2) incorporating an additional linguistic term for a variable and forming a derived linguistic term. The first step is able to correct non-optimal decisions of upper nodes resulting in many identical subtrees. For all rules, in each pruning step all possible hypotheses using modifications 1 and 2 are generated, evaluated and compared to the original rule. The best hypothesis (modification) will be accepted if it is rated higher than the original rule. Otherwise the pruning of this rule will be stopped. There are different choices of measures to evaluate a rule as described in Section 3.

Figure 4 shows the rule base from Figure 3 after pruning. The second and third rule are merged. A derived term "2 OR 3" for variable x_2 is introduced.

$$
\begin{array}{ll}
1 & \text{IF } (x_4 = 1) \text{ THEN } y = 3 \\
2 & \text{IF } (x_4 = 2) \text{ AND } (x_2 = 2 \text{ OR } 3) \text{ THEN } y = 2 \\
3 & \text{IF } (x_4 = 2) \text{ AND } (x_2 = 1) \text{ AND } (x_1 = \dots) \dots \text{ THEN } y = \dots \\
\vdots & \qquad\qquad\qquad\qquad \vdots
\end{array}
$$

Figure 4: Rules from Figure 3 after pruning

After pruning rules cannot be translated backwards into a tree. Rule premises are no longer complete and mutually exclusive. Thus, rules are simpler and better interpretable but also have significant overlap. Moreover, there are rules whose premises are a subset of the premise of others rules with the same term in the conclusion. Then these rules can be deleted if they are lower rated. This step of logical reduction reduces the redundancy of the rule set.

The advantage of pruning the rule set instead of the decision tree is discussed e. g. in (Quinlan, 1993). Pruning the decision tree itself leads only to the cut of over-specialized subtrees. It cannot correct non-optimal decisions in upper nodes as the rule pruning will do. Therefore, this third step often is able to reduce the rule set substantially. In addition, the rule set needs for classifying an example generally less tests than the respective decision tree.

4. Selection of rules In the set of rules obtained by pruning are possibly very similar rules, i. e. rules with partially redundant premises. In order to obtain a compact rule base, cooperating rules are selected. Cooperating rules are rules which give together a substantial decrease of the relative approximation error (see Section 3). In contrast to (7), where $\boldsymbol{C}_1$ is assumed to be unknown, here the term or class in the conclusion of every rule and, therefore, $\boldsymbol{C}_1$ is known. Consequently, the approximation error is calculated as $F_P = \|\boldsymbol{C}_1\boldsymbol{C}_2\boldsymbol{P} - \boldsymbol{Y}\|_F$ with $\boldsymbol{P}$

according to (6), i. e. the incomplete rule set is completed by the complementary premise. This complementary premise is the premise of the default rule for the evaluated rule set. The conclusion of the default rule is fixed to the term B_i resp. class i which is most frequent in the examples covered by the default rule with premise P_r, i. e. $i = \arg\max_i \sum_{k=1}^{N} \mu_{P_r}(\boldsymbol{x}[k])\mu_{B_i}(y[k]) / \sum_{k=1}^{N} \mu_{P_r}(\boldsymbol{x}[k])$. F_P^0 is computed for the global valid default rule whose conclusion contains the most frequent output term B_i resp. class i in all examples, i. e. $i = \arg\max_i \sum_{k=1}^{N} \mu_{B_i}(y[k])/N$.

The search starts with the rule giving the least relative approximation error. In each search step all hypothetical rule bases are generated which result from the so far collected rules adding one new rule. The rule base with the minimal error is selected. The algorithm terminates if a specified maximal number of rules is selected or the decrease in the error falls below a certain threshold.

As a consequence of the rule selection the rule base no longer covers the whole input space. Therefore, the default rule used in the calculation of the relative approximation error is incorporated into the rule base. The use of a default rule is especially advantageous if a frequent output class (e. g. the class "normal") is spread on the input space. Instead of many rules covering small hyperboxes in the input space, a default rule is created. This makes the rule base much more transparent.

5 EXPERIMENTAL RESULTS

This section describes experimental results from the application of the proposed method to three classification problems, the IRIS data set (Fisher, 1936), the diabetes and the Australian credit card assessment problem used in the STATLOG project (Michie et al., 1994). These well-known benchmarks stem from real-world applications.

The results are compared to those of other methods, mainly machine learning algorithms. Here, the focus is on classification accuracy. However, a comparison of different methods has to include further criteria, e. g. interpretability of the classifier system, time complexity, number of parameters in the algorithm. Comparing different methods, one has to bear in mind that the results are strongly problem dependent. This is to say, there is no best method for every problem. To evaluate a method, it has to be tested on a large set of different problems, a work which is under way for the proposed method.

5.1 IRIS DATA SET

The problem consists in the classification of three species of the iris flower, *Setosa, Versicolor* and *Virginica*, using four attributes, *sepal length*, *sepal width*, *petal length*, and *petal width*. All four attributes are numerical.

The data set contains 150 samples, 50 for each class. As the data set is small and in order to compare the results to those cited in (Wang et al., 1999) 10-fold cross validation (CV) is used. The data set is divided into ten subsets each containing 15 examples, five for each class. Nine of the then subsets form the training data set, the remaining subset is only used for testing. The test results are averaged over all ten runs.

To evaluate the effectiveness of the rule selection step, experiments with and without this step are performed. Further, the number of primary linguistic terms is varied from three to seven. The ordinary resp. fuzzy sets are evenly distributed on the input intervals.

The results are summarized in Table 1. It can be seen that with the selection of cooperating rules the results regarding the number of rules and the error rates for different numbers of linguistic input terms are nearly the same (except for four linguistic terms). Without the selection step, the size of the rule base depends on the number of terms. Also the error rates vary more. The selection step is even able to improve the classification accuracy in three of the five cases.

The rule base obtained for three linguistic terms for each input variable (*small, medium, large*) is:

Rule 1: IF (petal width=*small*) THEN *Iris Setosa*
Rule 2: IF (petal width=*medium*) THEN *Iris Versicolor*
Rule 3: IF (petal length=*large*) THEN *Iris Virginica*
Rule 4: ELSE *Iris Virginica*

no.	without selection step					with selection step				
terms	Setosa	Versicolor	Virginica	Average	$\bar{q}$	Setosa	Versicolor	Virginica	Average	$\bar{q}$
3	0.00	0.02	0.10	0.040	5.0	0.00	0.02	0.10	0.040	4.0
4	0.06	0.00	0.58	0.213	9.7	0.02	0.00	0.48	0.167	4.5
5	0.00	0.08	0.04	0.040	5.4	0.00	0.06	0.04	0.033	4.1
6	0.00	0.00	0.12	0.040	5.4	0.00	0.00	0.12	0.040	3.9
7	0.00	0.00	0.28	0.093	9.8	0.00	0.06	0.08	0.047	4.7

Table 1: Test error rates and mean number of rules $\bar{q}$ for the IRIS data set. (The number of rules includes the default rule in the case with selection step.)

For six linguistic terms (for a compact notation: A_i, $i = 1, \ldots, 6$) the following rules are generated:

Rule 1: IF (petal width= A_1 OR A_2) THEN *Iris Setosa*
Rule 2: IF (petal width= A_3 OR A_4) THEN *Iris Versicolor*
Rule 3: IF (petal length= A_5 OR A_6) THEN *Iris Virginica*
Rule 4: ELSE *Iris Versicolor*

Labelling A_1 OR A_2 with *small*, A_3 OR A_4 with *medium* and A_5 OR A_6 with *large*, the rule bases generated for three and six primary terms as well as the membership functions of in the terms appearing in the rules are identical. Consequently, the classification accuracy of both fuzzy systems is the same. This example shows that forming derived terms in rule pruning leads to a reduced number of rules and compact premises, i. e. compact and comprehensible descriptions of a concept separating a class from others. Note that for other numbers of primary terms the rule bases are different from the above cited. This is due to other membership functions of the primary terms. However, the classification accuracy is nearly the same (except for four primary terms). Apparently, the algorithm takes advantage of the flexibility gained by the use of a default rule and the forming of derived terms.

Algorithm	Setosa	Versicolor	Virginica	Average
FIL	0.00	0.060	0.020	0.027
GVS	0.00	0.060	0.060	0.040
IVSM	0.00	0.060	0.067	0.042
NT growth	0.00	0.089	0.065	0.051
Dasaranthy	0.00	0.140	0.020	0.053
C4.5	0.00	0.094	0.089	0.061

Table 2: Test error rates of other learning algorithms for the IRIS data set (cited after (Wang et al., 1999)).

For comparison, results of other learning algorithms are given in Table 2 (cited after (Wang et al., 1999)). These are the Fuzzy Inductive Learning algorithm (Wang et al., 1999), the Generalized Version Space learning algorithm (Hong and Tseng, 1997), the Incremental Version Space Merging (Hirsh, 1994), the NT growth algorithm (Aha and Kibler, 1989), Dasaranthy's pattern recognition algorithm (Dasaranthy, 1980), and the C4.5 algorithm (Quinlan, 1993). Taking the best result obtained with the new method, an average error rate of 0.033, only the Fuzzy Inductive Learning algorithm shows a better accuracy. But it is to mention that the results for the FIL algorithm cited in (Wang et al., 1999) can not be reproduced. Using the rule base given in (Wang et al., 1999) together with the inference scheme proposed here, the test error rate is 0.047. Applying a max-min or max-prod inference with rule weighting, the test error rate is 0.04. However, this example shows that rule weights could improve the accuracy, but in the same time the processing of the fuzzy system becomes less transparent. Therefore, the decision to use or not to use rule weights has to be based on the requirements in the actual application. The integration of rule weighting into the proposed inference scheme is a topic of further research.

5.2 DIABETES AND AUSTRALIAN CREDIT DATA SETS

A description of both data sets is presented in Table 3.

Name	no. examples	no. classes (no. cl. 1/cl. 2)	no. inputs numerical	no. inputs categorical	test method
diabetes	768	2 (500/268)	8	–	12-fold CV
Austr. credit	690	2 (307/383)	6	8 (2-14 values)	10-fold CV

Table 3: Description of the diabetes and Australian credit data sets. (CV cross validation)

The method is tested for different settings concerning the number of primary linguistic terms as in the experiments described above. The rule selection step is performed in all experiments. Results for both data sets are given in Table 4.

no. terms	TER	$\bar{q}$
3	0.233	4.9
4	0.234	5.3
5	0.232	6.8
6	0.216	7.3
7	0.221	9.2

a) Diabetes problem

no. terms	TER	$\bar{q}$
3	0.147	5.9
4	0.138	5.8
5	0.146	5.1
6	0.151	6.2
7	0.151	6.0

b) Australian credit problem

Table 4: Test error rates TER and average number of rules $\bar{q}$ for diabetes and Australian credit problem.

Again, it can be noticed that the classification accuracy depends on the number of pre-defined primary terms. This observation gives rise to a further development of the algorithm to include an adaptation of the membership functions.

The number of rules in the final rule bases is in all cases comparatively small to the number of hypothetical rules forming the search space. Counting all different premises which can be formulated for three linguistic terms in the diabetes problem, the search space comprises $4^8 = 65,536$ hypothesis, for seven terms 5,764,801. Also the number of variables in the premises in all case not exceeds three. This shows that one of the main objectives, to obtain a transparent and interpretable rule base, is archived.

Algorithm	TER	Algorithm	TER
TRI	0.216	IndCART	0.271
ITrule	0.245	Bayes tree	0.271
Cal5	0.250	AC^2	0.276
CART	0.255	NewID	0.289
C4.5	0.270	CN2	0.289

a) Diabetes problem

Algorithm	TER	Algorithm	TER
Cal5	0.131	C4.5	0.155
ITrule	0.137	Bayes tree	0.171
TRI	0.138	AC^2	0.181
CART	0.145	NewID	0.181
IndCART	0.152	CN2	0.204

b) Australian credit problem

Table 5: Test error rates of the Tree-oriented fuzzy Rule Induction algorithm (TRI) and the machine learning algorithms investigated in the STATLOG project.

Table 5 compares the error rates of the presented method (TRI tree-oriented fuzzy rule induction) to those of the machine learning algorithms evaluated in the STATLOG project which produce a decision tree or a rule base. This comparison seems adequate as the TRI algorithm as well as the machine learning algorithms aim at classifier systems, which are easy to interpret by humans. Regarding accuracy, apparently, the TRI algorithm proves advantageous when the variables are numerical (cf. diabetes problem). Whereas for problems where many variables are categorical and their values are described by crisp sets the advantages of fuzzy rules vanish (cf. Australian credit problem). Nevertheless, applying the TRI algorithm a fuzzy system with a good accuracy is obtained.

6 CONCLUSIONS

The paper presents a new method for the generation of fuzzy rules. This method is based on a decision tree induction algorithm, contains rule pruning and the selection of cooperating rules. Here, different measures for evaluating and rating rules are applied. These are information theoretic measures for the tree induction and certain newly introduced measures for pruning and selection. The latter allows the user to influence the compromise between accuracy and interpretability.

The method aims at a high degree of interpretability and transparency of the generated rules. For this reason, linguistic hedges are used to form derived linguistic terms, which help to make rule premises more compact. As a consequence, a new inference scheme has to be considered.

Experimental results show the effectiveness of the method. The generated rule bases contain a relatively small number of rules, which have a compact premise structure. At the same time, the fuzzy systems achieve a high accuracy.

Further developments include the automatic determination and adaptation of the membership functions and rule weighting.

References

Aha, D. W. and Kibler, D. (1989). Noise-tolerant instance-based learning algorithms. In *Proc. 11th Int. Conf. Artificial Intelligence*, pages 794–799, Detroit, MI.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984). *Classification and Regression Trees.* Wadsworth, Belmont, Ca.

Chi, Z. and Yan, H. (1996). ID3-derived fuzzy rules and optimal defuzzification for handwritten numeral recognition. *IEEE Trans. Fuzzy Systems*, 4(1):24–31.

Dasaranthy, B. V. (1980). Noising around the neighborhood: A new system structure and classification rule for recognition in partially exposed environments. *IEEE Trans. Pattern Analysis Machine Intell.*, 2:67–71.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7:179–188.

Hirsh, H. (1994). Generalizing version spaces. *Machine Learning*, 17:5–46.

Hong, T. P. and Tseng, S. S. (1997). A generaliszed version space learning algorithm for noisy and uncertain data. *IEEE Trans. Knowledge Data Engng.*, 9:336–340.

Hunt, E. B., Marin, J., and Stone, P. T. (1966). *Experiments in Induction.* Academic Press, New York.

Jäkel, J., Gröll, L., and Mikut, R. (1998). Bewertungsmaße zum Generieren von Fuzzy-Regeln unter Beachtung linguistisch motivierter Restriktionen. In *Berichtsband 8. Workshop* Fuzzy Control *d. GMA-FA 5.22*, pages 15–28, Dortmund.

Jäkel, J., Gröll, L., and Mikut, R. (1999). Automatic generation and evaluation of interpretable rule bases for fuzzy systems. In *Computational Intelligence for Modelling, Control and Automation CIMCA'99*, pages 192–197. IOS Press, Amsterdam.

Jessen, H. and Slawinski, T. (1998). Test and rating strategies for data based rule generation. Technical Report CI-39/98, Universität Dortmund, Dept. Computer Science.

Kiendl, H., Krabs, M., and Fritsch, M. (1991). Rule-based modeling of dynamical systems. In Popovic, D., editor, *Analysis and Control of Industrial Processes*, pages 217 – 231. Vieweg, Braunschweig.

Krone, A. and Kiendl, H. (1996). Rule-based decision analysis with fuzzy-rosa method. In Felix, R., editor, *Europ. Workshop Fuzzy Decision Analysis for Management, Planning and Optimization EFDAN*, pages 109–114. Fuzzy Demonstrations-Zentrum, Dortmund.

Maher, P. E. and St. Clair, D. (1993). Uncertain reasoning in an ID3 machine learning framework. In *Proc. 2nd IEEE Int. Conf. on Fuzzy Systems*, pages 7–12, San Francisco.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, London.

Otto, P. and Malberg, H. (1998). Fuzzy-Modellbildung zur Analyse von Wechselwirkungen bei Biosignalen des Herz-Kreislaufsystems. In *Berichtsband 8. Workshop* Fuzzy Control *d. GMA-FA 5.22*, pages 82–95, Dortmund.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

Quinlan, J. R. (1993). *C4.5:Programs for Machine Learning*. Morgan Kaufmann, San Mateo, Ca.

Wang, C.-H., Liu, J.-F., Hong, T.-P., and Tseng, S.-S. (1999). A fuzzy inductive learning strategy for modular rules. *Fuzzy Sets and Systems*, 103:91–105.

Yuan, Y. and Shaw, M. J. (1995). Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69(2):125–139.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8:338–353.

Zadeh, L. A. (1972). A fuzzy set theoretic interpretation of linguistic hedges. *Journal of Cybernetics*, 2:4–34.