Available online at www.sciencedirect.com
SCIENCE DIRECT®
Fuzzy Sets and Systems 150 (2005) 179–197
FUZZY sets and systems
www.elsevier.com/locate/fss

Interpretability issues in data-based learning of fuzzy systems

Ralf Mikut*, Jens Jäkel, Lutz Gröll

Forschungszentrum Karlsruhe GmbH, Institute for Applied Computer Science, P.O. Box 3640, D-76021 Karlsruhe, Germany

Received 1 October 2003; received in revised form 19 April 2004; accepted 18 June 2004
Available online 8 July 2004

Abstract

This paper presents a method for an automatic and complete design of fuzzy systems from data. The main objective is to build fuzzy systems with a user-controllable trade-off between accuracy and interpretability. Whereas criteria for accuracy mostly follow straightforwardly from the application, definition of interpretability and its criteria are subject to controversial discussion. For this reason, a set of interpretability criteria is given which guide the design process. Consequently, interpretability is maintained by structural choices regarding the type of membership functions, rules, and inference mechanism, on the one hand, and by including interpretability criteria in the rule/rule base evaluation, on the other hand. An application in Instrumented Gait Analysis, to characterize a certain group of patients in comparison to healthy subjects, illustrates the proposed algorithm.

Keywords: Classification; Data mining; Decision tree; Linguistic fuzzy system; Fuzzy rule; Inductive learning; Interpretability; Machine learning

1. Introduction

Interpretability is considered to be the main advantage of fuzzy systems over alternatives like statistical models or neural networks. Interpretability means that human beings are able to understand the fuzzy system's behavior by inspecting the rule base. It is crucial in the field of data mining and knowledge discovery where knowledge should be extracted from data bases and represented in a comprehensible form or for decision support systems where the reasoning process should be transparent to the user. Fuzzy systems constructed from expert knowledge—the traditional approach—usually are well understandable.

* Corresponding author. Tel.: +49-7247-825726; fax: +49-7247-825786.
E-mail addresses: ralf.mikut@iai.fzk.de (R. Mikut), jens.jaekel@iai.fzk.de (J. Jäkel), lutz.groell@iai.fzk.de (L. Gröll).

0165-0114/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2004.06.006

At present, a vast number of algorithms exist for automatic data-based fuzzy modeling, which can be classified as clustering [2,16,54], neuro-fuzzy [8,26,30], machine learning [17,20,24,47,51] or evolutionary approaches [4,13,31]. However, fuzzy systems generated by these algorithms are not necessarily comprehensible, especially when the algorithm aims at reaching a maximum accuracy. In recent years, research has therefore started to focus on the trade-off between interpretability and accuracy (for an overview see [10]).

Whereas the definition of accuracy in a certain application is straightforward, the definition of interpretability is rather problematic. Most researchers and practitioners would agree on interpretability involving the following aspects [3,9,14,28,49]:

- The number of rules is small enough to be comprehensible. Moreover, rules should not contain degrees of plausibility or rule weights.
- The rule base is formed of rules describing (locally) relevant relationships. The rules are consistent (similar premises lead to similar conclusions).
- Rule premises should be easy in structure and contain only a few features (input variables) only.
- The fuzzy system should preferably use features and combinations of these, which are familiar to the user.
- Linguistic terms should be intuitively comprehensible. The form and parameters of the membership functions should correspond to the understanding of the linguistic expressions.
- The inference mechanism should produce technically and intuitively correct results.

Generally, interpretability can be maintained or enhanced during the fuzzy system's generation or obtained by post-processing of the resulting data-driven fuzzy system.

Examples for the first approach include constraints on membership functions and their parameters [3,19,42], a special syntax of fuzzy rules [14,24] or a special structure of the fuzzy system, e.g. a hierarchical structure [15,22,52]. The second approach comprises simplification by merging similar fuzzy sets or rules [11,27,48,53] or using linguistic hedges [18,50].

This paper proposes a modular data-driven algorithm for fuzzy system learning according to the first alternative. Different elements to improve interpretability are explicitly (in form of evaluation measures) or implicitly (in form of efficient heuristics in all steps of the learning algorithm) integrated in this algorithm:

- Feature selection finds the most relevant features. The relevance measure can incorporate a priori information on preferred features (user or technical preference).
- Automatic generation of membership functions and labels takes their interpretability into account (form of membership functions, reasonable rounded parameters, adaptation to the distribution of feature values).
- Generation of rule hypotheses by decision tree induction and their pruning favor simple premises and lead to derived linguistic terms.
- Textual presentation of rules provides additional information in natural language and is better readable than a formal fuzzy rule.

An implementation as MATLAB toolbox KAFKA enables the user to solve complex real-world problems interactively controlling the trade-off between interpretability and accuracy.

The paper is organized as follows: Section 2 introduces the data and a priori information as the input to the learning algorithm as well as basic settings of the fuzzy system. Criteria for evaluating features and

rules are discussed in Section 3. Section 4 presents the learning algorithm. In Section 5, the application to a diagnosis problem in Instrumented Gait Analysis is described.

2. Learning data, a priori information, and basic settings of the fuzzy system

As the main aim is to identify input–output relationships, e.g. between features and classes, the data set is organized as follows: Given are $k = 1, \ldots, N$ samples (often called examples)

$$(\mathbf{x}^{\mathrm{T}}[k], \boldsymbol{\mu}_y^{\mathrm{T}}[k]) = (x_1[k], \ldots, x_{\mathrm{l}}[k], \ldots, x_s[k], \mu_{B_1}[k], \ldots, \mu_{B_{m_y}}[k]),$$

where $x_{\mathrm{l}}[k]$ represent feature values and $\mu_{B_i}[k]$ a class assignment, such that $\sum_{i=1}^{m_y} \mu_{B_i}[k] = 1$. In case of $\boldsymbol{\mu}_y \in \{0, 1\}^{m_y}$, the problem is a "crisp", in case of $\boldsymbol{\mu}_y \in [0, 1]^{m_y}$ a fuzzy classification problem. The former may be regarded a special case of the latter one.

In approximation or regression problems, the output y is assumed to have a domain of $Y \subset \mathbb{R}$. By defining a fuzzy partition with fuzzy sets $\mu_{B_1}, \ldots, \mu_{B_{m_y}}$ on Y, this problem is transformed into a fuzzy classification problem. Therefore, only fuzzy classification will be considered in the following sections.

Input variables or features $x_{\mathrm{l}} \in X_{\mathrm{l}}$ are assumed to be ordinal, real or categorical. Ordinal or real numbers are transformed into membership values of m_{l} fuzzy sets $\mu_{A_{\mathrm{l},i}}$ defined on X_{l} resulting in vectors $\boldsymbol{\mu}_{x_{\mathrm{l}}} \in [0, 1]^{m_{\mathrm{l}}}$. For categorical features, values are taken from the finite set $\{A_{\mathrm{l},i} | i = 1, \ldots, m_{\mathrm{l}}\}$ and can, therefore, be coded by a vector of membership values $\boldsymbol{\mu}_{x_{\mathrm{l}}} \in \{0, 1\}^{m_{\mathrm{l}}}$.

The learning algorithm uses the following data matrices: The block matrix $\boldsymbol{\mu}_X$ with vector elements $\boldsymbol{\mu}_{x_{\mathrm{l}}}[k]$ and an overall dimension $(m_1 + \cdots + m_s, N)$, the matrix $\boldsymbol{\mu}_Y$ with dimension (m_y, N), consisting of vectors $\boldsymbol{\mu}_y[k]$, the matrix $\boldsymbol{\mu}_P$ of rule activations $\mu_{P_r}(\mathbf{x}[k])$ of dimension $(r_{\max} + 1, N)$ and the matrix $\hat{\boldsymbol{\mu}}_Y$ of estimates of membership values for the output classes (fuzzy sets) with dimension (m_y, N).

Additionally, the user may provide an a priori relevance weight $M_{\mathrm{l,ap}} \in [0, 1]$, by default $M_{\mathrm{l,ap}} = 1$, for each feature. These a priori relevance weights give preference to those features, which are familiar to the user or values of which can be obtained with greater confidence or less effort.

The fuzzy system to be generated contains rules ($r = 1, \ldots, r_{\max}$) with a general structure

$$R_r : \text{IF} \underbrace{\underbrace{x_1 = A_{1,Rr}}_{\text{partial premise } P_{r1}} \text{ AND } \cdots \text{ AND } \underbrace{x_s = A_{s,Rr}}_{\text{partial premise } P_{rs}}}_{\text{premise } P_r} \text{ THEN } \underbrace{y = B_{Rr}}_{\text{conclusion } C_r}$$

and a default rule $R_{r_{\max}+1}$: ELSE $y = B_{Rr_{\max}+1}$. The premise P_r consists of a conjunctive (AND, $\wedge$) combination of partial premises $P_{r1}, \ldots, P_{rs}$. The linguistic term $A_{\mathrm{l},Rr}$ can be a primary linguistic term $A_{\mathrm{l},i}$ of the feature x_{l} or a disjunctive (OR, $\vee$) combination of some neighboring or all linguistic terms of x_{l}, e.g. $A_{\mathrm{l},Rr} = A_{\mathrm{l},1}$ OR $A_{\mathrm{l},2}$ OR $A_{\mathrm{l},3}$, which is referred to as derived linguistic term [23]. In the latter case, this partial premise has no influence on the rule activation and in the presentation of the rule it is omitted.

Primary linguistic terms $A_{\mathrm{l},i}$ possess triangular (interior terms) or trapezoidal membership functions (first and last term). For each value of x_{l} they add up to one, i.e. they form a complete fuzzy partition. Thus, m_{l} parameters $a_{\mathrm{l},i}$, the x-coordinates of the maximum of the triangular or inflexion point of trapezoidal membership functions, are sufficient to determine the latter. Membership functions of derived terms result

from an operation with an appropriate co-t-norm on the membership functions of the primary terms. The chosen co-t-norm is the bounded sum $S_b(u, v) = \min\{u + v, 1\}$. This is the only co-t-norm for which firstly the resulting fuzzy sets of the derived terms are convex and secondly the membership function of disjunction of all primary terms of a variable x_l is identical to one on the whole universe X_l (for a detailed discussion see [24]). Hence, membership functions of derived terms in the form of disjunctive connections of neighboring primary terms are trapezoidal.

The conjunctive connection of partial premises is accomplished with the product as t-norm since features are assumed to be independent or non-interacting. Use of derived terms typically results in partially redundant rules. Conventional inference schemes like *max–min* or *sum–prod* often produce results that contradict the expectation from reading the rule base. Therefore, a special inference scheme [38], which takes the redundancy of rules into consideration, is applied.

3. Evaluation of features and rules

The evaluation of features and rules is based on a statistical approach since the relationship between values of the features and the output class in general is not deterministic. Possible reasons are missing features in deterministic relationships, measurement errors or erroneous class assignments.

Feature selection requires an appropriate definition of the concept of feature relevance and an adequate measure. Here, a feature is considered relevant for the solution to a classification problem if its presence in the set of inputs of a classifier improves the expected classification accuracy. A measure of the relevance of feature x_l, which is independent of the classifier and includes a priori preferences ($M_{l,ap}$, Section 2) is

$$M_l = M_{l,ap}^{\alpha} \underbrace{\frac{H(x_l; y)}{H(y)}}_{M_{l,ig}}, \quad \alpha \geqslant 0, \tag{1}$$

where $M_{l,ig}$ is the normalized mutual information or information gain. The parameter α controls the strength of preference for features with high a priori preference set by the user and, therefore, the accuracy/interpretability trade-off. Choosing $\alpha \gg 1$ strengthens user preferences, with α going to zero the influence of a priori preferences diminishes, setting $\alpha = 1$ is typically a good compromise.

$H(y)$ is the entropy of the output y in the learning data set

$$H(y) = -\sum_{j=1}^{m_y} p(B_j) \operatorname{ld} p(B_j), \tag{2}$$

a measure of the average information needed for identifying the output class of an example. $p(B_j)$ denotes the probability of the event "y is in class B_j". The mutual information $H(x_l; y)$

$$\begin{aligned} H(x_l; y) &= H(y) - H(y|x_l) \\ &= -\sum_{j=1}^{m_y} p(B_j) \operatorname{ld} p(B_j) + \sum_{i=1}^{m_l} p(A_{l,i}) \sum_{j=1}^{m_y} p(B_j|A_{l,i}) \operatorname{ld} p(B_j|A_{l,i}), \end{aligned} \tag{3}$$

is a measure of the average information provided by feature x_l about the class of y. Normalizing of (3) with $H(y)$ yields $0 \leqslant M_{l,ig} \leqslant 1$ with $M_{l,ig} = 0$ indicating an irrelevant feature and $M_{l,ig} = 1$ indicating a feature for which an unique relationship between its value and the output class exists.

Assessing the quality or relevance of a rule or rule base involves several aspects: Accuracy, statistical significance, and clearness. A single rule is considered as a rule base with $r_{\max} = 1$ (the investigated rule) and an ELSE rule with $\mu_{P,\mathrm{ELSE}} = 1 - \mu_P$.

The classification *accuracy* involves the minimum quadratic error E in terms of membership values of the output classes, which is obtained by the rule base according to

$$E = \min_{\mathbf{R}_{B|P}} \| \underbrace{\mathbf{R}_{B|P} \cdot \boldsymbol{\mu}_P}_{\hat{\boldsymbol{\mu}}_Y} - \boldsymbol{\mu}_Y \|_F^2 \quad \text{s.t. } \mathbf{R}_{B|P} \geqslant \mathbf{0}_{m_y \times r_{\max}+1},\ \mathbf{1}_{m_y}^{\mathrm{T}} \mathbf{R}_{B|P} = \mathbf{1}_{r_{\max}+1}^{\mathrm{T}}. \tag{4}$$

with

$$\mathbf{R}_{B|P} = \begin{pmatrix} \hat{p}(B_1|P_1) & \cdots & \hat{p}(B_1|P_{r_{\max}}) & \hat{p}(B_1|P_{\mathrm{ELSE}}) \\ \vdots & \ddots & \vdots & \vdots \\ \hat{p}(B_{m_y}|P_1) & \cdots & \hat{p}(B_{m_y}|P_{r_{\max}}) & \hat{p}(B_{m_y}|P_{\mathrm{ELSE}}) \end{pmatrix}.$$

The minimum quadratic error of the trivial model (a rule with an always true premise) is

$$E_0 = \min_{\mathbf{r}_B} \|\mathbf{r}_B \mathbf{1}_N^{\mathrm{T}} - \boldsymbol{\mu}_Y\|_F^2 \quad \text{with } \mathbf{r}_B = (\hat{p}(B_1), \ldots, \hat{p}(B_{m_y}))^{\mathrm{T}}.$$

Here, it is assumed that all possible rules exist for each premise and that $\mathbf{R}_{B|P}$ and $\mathbf{r}_B$, respectively, consist of rule weights. Each column of $\mathbf{R}_{B|P}$ corresponds to a premise in the rule base, each row to a possible conclusion (output class). The matrix of rule activations is assumed to have all column sums equaling one.[1] In this setting, the matrix $\mathbf{R}_{B|P}$ has another interesting interpretation as its elements $\hat{p}(B_j|P_r)$ may be regarded the probability of "y is in class B_j" given the event in the premise has occurred (the inputs assume the specified values), i. e. the posterior probability of class B_j. Hence, for $\hat{p}(B_j|P_r) = 1$ the output class B_j can be predicted unambiguously by a rule with the premise P_r. Therefore, $\hat{p}(B_j|P_r)$ is the precision or hit rate of the rule "IF P_r THEN B_j".

The *statistical significance* of each rule is tested assuming $\hat{p}(B_j|P_r)$ and $\hat{p}(\bar{B}_j|P_r)$ for different r to be probabilities of binomially distributed events. They are examined for significant differences [32]. This test provides information on

- the significance of individual rules (test $\hat{p}(B_j|P_r)$ against $\hat{p}(B_j|\bar{P}_r)$) as well as,
- on the necessity of further generalization with a new premise $P_{r,\mathrm{gen}}$; $P_r \subset P_{r,\mathrm{gen}}$ (test $\hat{p}(B_j|P_r)$ against $\hat{p}(B_j|(\bar{P}_r \wedge P_{r,\mathrm{gen}}))$) (see Section 4.3).

Both tests take the number of examples covered by the respective premises into account. A rule is rejected if one of these tests is not fulfilled. They delete especially rules with premises only covering

[1] This means that the rule base is complete and the rules are mutually disjoint. If necessary, completeness and disjointness have to be ensured by appropriate procedures (Section 4.4 and [38]).

a few examples and rules with minor difference between premise and negated premise or premise and generalized premise. The latter finally leads to simpler and more interpretable rules.

To ensure an optimal interpretability, the final rule base should only contain unweighted rules. Consequently, the measure of *clearness* Q_{cl}

$$Q_{\mathrm{cl}} = \prod_{r=1}^{r_{\max}} \max_j(\hat{p}(B_j|P_r)) \tag{5}$$

assesses the deviation from the optimal situation.

Finally, the relevance is measured by

$$Q = \underbrace{\left(1 - \frac{E}{E_0}\right)}_{Q_{\mathrm{ac}}} Q_{\mathrm{cl}}^{\beta}, \quad \beta \geqslant 0 \tag{6}$$

a compromise between classification accuracy (Q_{ac}) and clearness of the rules (Q_{cl}) [23]. The compromise can be controlled using β. The larger β is, the more important is a conditional probability near one, at $\beta = 0$ the clearness is ignored. Rule bases with $Q = 1$ ($E = 0$, $Q_{\mathrm{ac}} = 1$, $Q_{\mathrm{cl}} = 1$) are optimal. For $Q = 0$ ($E = E_0$, $Q_{\mathrm{ac}} = 0$), they are not relevant because their performance is not better than that of the trivial model E_0. Rule bases with $Q < 0$ ($E > E_0$, $Q_{\mathrm{ac}} < 0$) contain false rules, thus decreasing the classification accuracy.

In (2)–(4), $p(A_{\mathrm{l},i})$, $p(B_j)$, $p(P_r)$, $p(B_j|A_{\mathrm{l},i})$ and $p(B_j|P_r)$ are probabilities of ordinary (in the case of an ordinary partition of the domain) or fuzzy events (in the case of a fuzzy partition). [2] For ordinary partitions [3] appropriate estimates are frequencies calculated from the data set. For fuzzy events the estimates

$$\hat{p}(A_{\mathrm{l},i}) = \sum_{k=1}^{N} \mu_{A_{\mathrm{l},i}}(x_{\mathrm{l}}[k])/N, \quad \hat{p}(B_j) = \sum_{k=1}^{N} \mu_{B_j}(y[k])/N, \quad \hat{p}(P_r) = \sum_{k=1}^{N} \mu_{P_r}(\mathbf{x}[k])/N$$

and (4) for $\hat{p}(B_j|P_r)$ and $\hat{p}(B_j|A_{\mathrm{l},i})$ with $P_r = A_{\mathrm{l},i}$, respectively, are used. [4]

From the conditional probabilities $\hat{p}(B_j|P_r)$ in $\mathbf{R}_{B|P}$ and the analogously estimated (conditional) probabilities or likelihoods, e.g. $\hat{p}(A_{\mathrm{l},i}|B_j)$ (frequency of the term $A_{\mathrm{l},i}$ when the class is B_j) or $\hat{p}(P_r|C_r)$ (sensitivity of the rule—fraction of examples with class as in C_r covered by its premise), additional information for explaining rules can be extracted (Section 5).

---

[2] Because of the specifications concerning membership functions the fuzzy partitions are complete and disjoint.

[3] If each term is assigned its α-cut with $\alpha = 0.5$ instead of the fuzzy set, then an ordinary partition of the domain of the respective feature results.

[4] In order to estimate joint probabilities of two fuzzy events—here, of the premise and the conclusion, some authors, e.g. [32,56], propose to use the product as t-norm. However, this choice, in a certain sense, assumes independence of both events and introduces a bias in rule evaluation and search [25].

4. Learning algorithm

4.1. Membership functions and labels

For the typically large number of features a manual specification of membership functions and labels is not feasible. Hence, a large number of data-based approaches has been conceived from simple heuristics (membership functions of equal width) to clustering and optimization-based methods [12,21]. However, it is a common problem of most of these methods that they often produce membership functions that are lacking interpretability and parameter values with an unreasonable precision.

The first problem can be tackled by incorporating restrictions on parameters (here: $a_{l,i}$ for the trapezoidal and triangular membership functions as described in Section 2), the second by favoring reasonable rounded values. Both methods can be included in an optimization-based method [37]. Its main objective is to maximize a compromise between the entropy of the partition of the feature and the mutual information between the feature and the output for a given number of membership functions. However, this method requires a high computational effort due to the nonlinear optimization problem. For an interactive design procedure this is unacceptable.

A computationally efficient approach to avoid nonlinear optimization was found to produce good results in many applications. It aims at obtaining a uniform distribution of examples between linguistic terms (related to maximum entropy of the partition) and interpretable parameters. For this purpose, the learning data for each feature $x_l[1], \ldots, x_l[N]$ are sorted in ascending order. From the sorted values m_l values $x_l^{\text{sort}}[j]$ are chosen such that each interval $x_l^{\text{sort}}[j] \leqslant x_l^{\text{sort}}[k] < x_l^{\text{sort}}[j+1]$ contains approximately $N/(m_l - 1)$ values. The number m_l of linguistic terms has to specified in advance and is a parameter of the algorithm. Regarding interpretability m_l should be in the range from 5 to 9 [41], however, the choice is not very critical as forming disjunctions of linguistic terms in the pruning process effectively reduces the number of terms.

The chosen $x_l^{\text{sort}}[j]$ will be rounded in order to improve interpretability (rd: rounding operator):

$$a_{l,i} = \frac{\text{rd}(10^{p_{l,i}} x_l^{\text{sort}}[j])}{10^{p_{l,i}}} \quad \text{with } j = \text{rd}\left(1 + (i-1)\frac{N-1}{m_l - 1}\right), \quad i = 1, \ldots, m_l. \tag{7}$$

Here, $p_{l,i}$ denotes an iteratively determined parameter specifying the number of significant decimal digits of $a_{l,i}$. The initial value of all i is chosen with respect to the range of x_l

$$p_{l,i} = -\text{rd}(\log_{10}(x_l^{\text{sort}}[N] - x_l^{\text{sort}}[1]) - 0.5).$$

If (7) results in identical values for some $a_{l,i}$, then the respective $p_{l,i}$ are increased by one until all $a_{l,i}$ have different values or a termination condition for $p_{l,i}$ is met.

If the number of different values for x_l^{sort} is smaller than m_l, then the feature is assumed to be a categorical one and the values of x_l^{sort} are used as parameters $a_{l,i}$.

The labels of the linguistic terms $A_{l,i}$ are chosen with respect to the values of the $a_{l,i}$ and the number of terms m_l. Terms with $a_{l,i} = 0$ are labeled ZE (zero). The label of terms with $a_{l,i} > 0$ depends on their number (1 term: POS, 2 terms: PS–PB, 3 terms PS–PM–PB, 4 terms PS–PM–PB–PVB, 5 terms PVS–PS–PM–PB–PVB). Here, POS stands for positive, PVS for positive very small, PS for positive small, PM for positive medium, PB for positive big and PVB for positive very big. For terms with $a_{l,i} < 0$ labeling is analog (NEG-negative, NVS, NS, NM, NB, NVB). Hence, the labels like small, medium or big are related to the distribution in the learning data set which parallels human usage. However, this method

just like all data-driven methods cannot guarantee to result in intuitively understandable membership functions and labels. In the presentation of rules the linguistic terms are therefore complemented by the α-cuts ($\alpha = 0.5$) (see Section 4.5). Section 5 gives an example illustrating the method.

4.2. Feature selection

Feature selection aims at finding a small subset of features with high-discriminating power and acceptance by the user (expressed as relevance weights). As an additional feature selection takes place implicitly in further stages of the algorithm, namely, in the induction of decision trees and rule pruning, there is no strong need for finding the optimal subset in this stage. Instead, it is the task to discard superfluous, strongly redundant or irrelevant features to speed up the search.

The relevance measure (1) only reflects the relevance and preference of individual features without accounting for the redundancy between them. It could be generalized to measure the relevance of feature combinations, but the estimate of normalized mutual information will be rather unreliable even in case of a moderate number of features.

The applied feature-selection algorithm performs step-wise forward selection. This means that starting from an empty set of selected features, the best feature according to (1) is selected in each step and removed from the set of potential features. In addition, all features that are redundant to the selected one are removed as well. Redundancy could be measured by mutual information or the magnitude of the (linear) correlation coefficient. If the value of the redundancy measure exceeds a specified threshold the feature is considered redundant. The algorithm is terminated, if the set of potential features is empty or a specified maximum number of selected features is reached.

As an alternative, wrapper approaches (see e.g. [29]) evaluate features and feature sets by the performance of the complete classifier. This may lead to better results—but the necessary computing time is much higher in comparison to the chosen filter approach due to the design effort for the classifier. This is not acceptable in interactive design for problems with some hundred or thousands of potential features in real-world problems.

4.3. Rule search

In the first step, rule hypotheses are generated by inducing a decision tree. In the second step, these rule hypotheses are generalized by different modifications of their premises. Finally, a subset is selected from the generalized rules to build the rule base.

4.3.1. Generating rules from decision trees

A decision tree represents a multi-step decision process for classifying an object (an example) based on the feature values. It consists of nodes (decision nodes or leaves) and branches. A decision node indicates a class B_j and contains a test on the value of a feature ($x_l = ?$). A node without a test is called a leaf. For each outcome of the test, a linguistic term $A_{l,i}$ of the tested feature, a branch starts from the decision node. Fig. 1 shows an example with $s = 2$ features and $m_y = 4$ classes.

The induction algorithm employed here is similar to the popular ID3 algorithm [44] and several methods for fuzzy decision tree induction [6,36,55]. In contrast to these methods, the feature relevance (1) is used to choose features for decision nodes taking interpretability and discriminative power into account. For the purpose of decision tree induction, the linguistic terms of each feature are assigned the α-cuts ($\alpha = 0.5$)

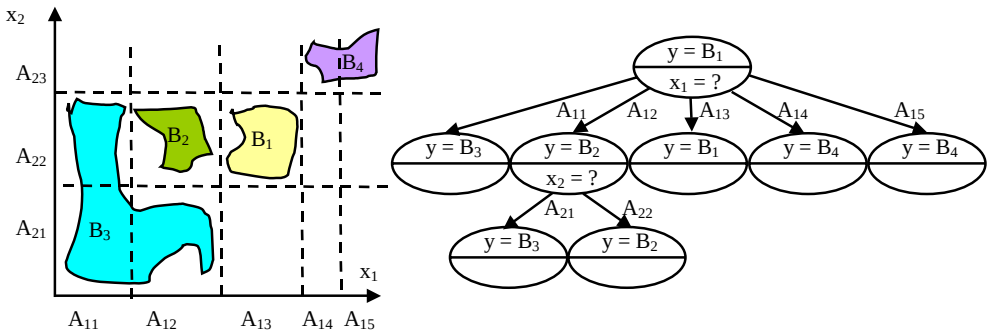

Fig. 1. Example (left) and decision tree (right).

of their fuzzy sets, thus resulting in an ordinary partition of its domain. If necessary, the same has to be done for the output.

Starting with the root node, the algorithm selects the most frequent class B_j in the N examples and the most relevant feature x_l according to (1) from the feature set which is then removed from it. It creates m_l new nodes and splits the examples up into m_l subsets according to the linguistic terms $A_{l,i}$. For each new node it repeats this procedure for the N_r examples contained in the node, i. e. the feature relevance (1) is calculated for the N_r examples. A node becomes a leaf, if there are no more features in the feature set with an estimated lower bound of $M_l > 0$ [40]. The algorithm terminates when all new nodes have become a decision node or a leaf.

For each leaf a rule R_r is extracted from the decision tree, with its conclusion C_r being obtained from this leaf. Traversing the tree from the leaf to the root node, a partial premise P_{rl} results from each node. It consists of the feature x_l and the linguistic term $A_{l,i}$ on the branch to the node. The decision tree in Fig. 1 possesses six leaves and, hence, yields six rules.

Assuming noise-free data from a deterministic relationship between classes and qualitative feature values, the algorithm produces a decision tree which classifies each example of the training set correctly. In the case of noisy data and a relatively small number of examples, the probability of misclassification of examples not contained in the training set may be substantial, i.e. the decision tree shows poor generalization ability. Therefore, pruning methods exist, which generalize the fully developed decision tree by taking back several splits. However, these methods cannot remedy non-optimal selections of features in the first stages, thus often leading to many identical subtrees at lower levels. For this reason, it is advantageous not to prune the decision tree, but the rules extracted from it (for a detailed discussion see [45]).

In order to obtain a comprehensive set of rule hypotheses, several decision trees are generated. Using this option, the user may define a compromise between a fast (only a single decision tree separating all classes) and a broad search for candidate rules by the number of additional trees. In the multi-class case with m_y classes, m_y trees are generated to separate the class B_j from the union of all other classes. Hence, the leaves contain the class B_j or its negation $\bar{B}_j$. B_j-leaves leading to rules according to Section 2. Using this strategy, the induction algorithm is not forced to find a compromise for separating *all* classes with a single tree. Consequently, extracted rule hypotheses tend to be simpler.

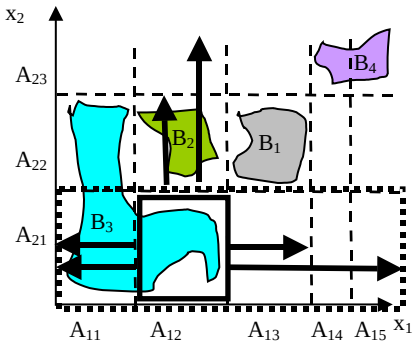

Fig. 2. Visualization of pruning possibilities for rule R_1 in Step 1.

In the two-class case, the trees are accomplished by the step-wise discarding of the most relevant feature in the feature set.

Both types of additional trees may contribute to a better performance and simpler rules with higher interpretability by finding additional candidate rules for the following steps.

4.3.2. Rule pruning

To improve the generalization ability, the rules are pruned using two kinds of modifications: (1) deleting a partial premise and removing the respective feature from the premise and (2) adding a linguistic term within a partial premise and forming a disjunctive combination. By restricting this extension to neighboring terms only interpretable derived terms result. The first modification may correct a non-optimal selection of a feature in upper nodes of the decision tree. The second one extends the scope of the rule with respect to one feature.

Pruning is performed for each rule in a hill-climbing procedure. In each pruning step all rules derivable using modifications (1) and (2) are generated, evaluated with the rule relevance measure (6) and compared to the original rule. For the calculation of E in (4) the matrix of premise activations $\boldsymbol{\mu}_P$ is formed for the rule premise P_r and its complement $\bar{P}_r$, the matrix $\boldsymbol{\mu}_Y$ for conclusion $C_r = B_j$, and its complement $\bar{C}_r = \bar{B}_j$. The best statistically significant rule will be accepted, if it is rated higher than the original one. The significance is measured against the negated premise and against all rules with a deleted partial premise $P_{r,\text{gen}}$ (test $\hat{p}(B_j|P_r)$ against $\hat{p}(B_j|(\bar{P}_r \wedge P_{r,\text{gen}}))$) (see Section 3). Pruning can lead to identical rules from which all but one will be deleted.

This procedure is illustrated for the rule R_1 : IF $x_1 = A_{12}$ AND $x_2 = A_{21}$ THEN $y = B_3$ (Fig. 2, Table 1, Step 0: Rule from decision tree). In the first step, five rules are derived, of which the one marked with an asterisk with premise $x_2 = A_{21}$ obtains the best evaluation. This rule is accepted, as in Step 2 no rule can be found with a better evaluation.

4.4. Building rule bases

Pruning leads to more general rules which have mostly simpler premises and are better interpretable. However, the set of rules possibly contains very similar rules, i. e. rules with partially redundant premises. In order to obtain a compact rule base, a subset of individually good rules that complement each other is selected.

Table 1
Premises and conclusions of the rules derived by admissible modifications of rule R_1

Step	P_r	C_r
0*	$x_1 = A_{12} \wedge x_2 = A_{21}$	B_3
1	$x_1 = A_{12}$	B_3
1*	$x_2 = A_{21}$	B_3
1	$x_1 = (A_{11} \vee A_{12}) \wedge x_2 = A_{21}$	B_3
1	$x_1 = (A_{12} \vee A_{13}) \wedge x_2 = A_{21}$	B_3
1	$x_1 = A_{12} \vee x_2 = (A_{21} \vee A_{22})$	B_3
2	$x_2 = (A_{21} \vee A_{22})$	B_3
2	1	B_3

The rule selection algorithm performs a step-wise forward selection. It starts with an empty rule base ($r_{\max} = 0$) and an ELSE rule. The conclusion of the ELSE rule may be set by the user (to a pre-defined class or an additional rejection class as B_5 in Table 2) or determined from the data (the most frequent class in the examples not covered by other rules).

In subsequent steps, the best rule base with $r_{\max}$ rules is complemented by the candidate rule from pruning maximizing the relevance measure Q (6) for the rule base. The resulting rule base now contains $r_{\max} := r_{\max} + 1$ rules and the ELSE rule. For calculating E in (4), $\boldsymbol{\mu}_P$ is formed for the premises of $r_{\max}$ rules under consideration and their joint complement $\overline{\bigcup_{r=1}^{r_{\max}} P_r}$ and $\boldsymbol{\mu}_Y$ for all output classes $B_1, \ldots, B_{m_y}$.[5] The complement to the disjunction of all premises (corresponding to an ELSE rule) ensures the completeness of the rule base (the sum of rule activations for each example equals one). Furthermore, it has to be guaranteed that the premises are mutually disjoint [38]. The rule search terminates if no further rule significantly increases Q.

All approaches to choose the conclusion of the ELSE rule have some advantages and disadvantages. A rejection class as conclusion enforces the specification of all other classes by at least one rule. The use of a (user-defined or automatically found) class B_j may reduce the necessary number of rules. This may be reasonable e.g. for fault-detection where the rules for different faults should be found with a user-defined default rule "ELSE no fault". A further application is the search for compact regions of classes $B_i \neq B_j$ in a widely spread class B_j. Depending on this choice, different rule bases result (Table 2).

4.5. Presentation of results

The primary result of the algorithm described above are fuzzy rules in the form given in Section 2. In addition, the information e.g. on the precision of rules gained during the generation may be very valuable for the user when interpreting them. It is complemented by further information obtained ex post, for example, on the rule sensitivity (recall). This information and the rules themselves are presented in a

[5] Instead of estimating $\mathbf{R}_{B|P}$ in order to calculate Q, this matrix can also be fixed such that in each column corresponding to a rule premise the element corresponding to the conclusion is set to one, the remaining to zero.

Table 2
Two rule bases from the same data set as in Figs. 1 and 2: 5+1 rules with rejection class B_5 (left) and 3+1 rules with automatically chosen class B_3 for the ELSE rule (right)

IF	THEN	IF	THEN
$x_1 = A_{12} \wedge x_2 = A_{22}$	B_2	$x_1 = A_{12} \wedge x_2 = A_{22}$	B_2
$x_1 = A_{13}$	B_1	$x_1 = A_{13}$	B_1
$x_1 = (A_{14} \vee A_{15})$	B_4	$x_1 = (A_{14} \vee A_{15})$	B_4
$x_2 = A_{21}$	B_3		
$x_1 = A_{11}$	B_3		
ELSE	B_5	ELSE	B_3

Table 3
Linguistic terms expressing frequencies or fractions

$\hat{p}$	Linguistic term for all $\hat{p}$ but $\hat{p}(P_r\|C_r)$	Linguistic term for $\hat{p}(P_r\|C_r)$
[0, 0.025]	Never	None
(0.025, 0.20]	Rarely	Few
(0.20, 0.50]	Sometimes	Some
(0.50, 0.80]	Usually	Many
(0.80, 0.975]	Mostly	Most
(0.975, 1.00]	Always	All

textual form based on the Generalized Constraint Language [57,58]. This section only presents the textual frames and building blocks, for a detailed example the reader is referred to Section 5.

The additional information regarding rule precision and sensitivity is contained in the values of $\hat{p}(C_r|P_r)$ and $\hat{p}(P_r|C_r)$. These numeric values are converted into linguistic terms for the purpose of presentation using $\texttt{text}\ (\hat{p})$. Here, it is distinguished between terms expressing frequencies of occurrence and the fraction of examples for an output class covered by a premise $\hat{p}(P_r|C_r)$ (see Table 3 and [1,5,46]).

For the presentation of a rule R_r, a first text frame lists the features appearing in the premise with their complete name $\texttt{text}\ (x_\mathrm{l})$, a statement on their relative values in comparison to examples of other classes $\texttt{text}\ (comparison)$, their linguistic terms $\texttt{text}\ (A_{\mathrm{l},i})$ and $\texttt{text}\ (B_{R_r})$, respectively, and the frequency of occurrence $\texttt{text}\ (\hat{p}(A_{\mathrm{l},i}|C_r))$:

> *The samples for* $\texttt{text}\ (B_{Rr})$ *are characterized by* $\texttt{text}\ (x_\mathrm{l})$. *This feature is* $\texttt{text}\ (\hat{p}(comparison))$ $\texttt{text}\ (comparison)$ *than otherwise* $(\texttt{text}\ (\hat{p}(A^{\mathrm{sort}}_{\mathrm{l},1}|C_r))\ \texttt{text}\ (A^{\mathrm{sort}}_{\mathrm{l},1})$ *and . . . and* $\texttt{text}\ (\hat{p}(A_{\mathrm{l},m^{\mathrm{sort}}_\mathrm{l}}|C_r))\ \texttt{text}\ (A^{\mathrm{sort}}_{\mathrm{l},m^{\mathrm{sort}}_\mathrm{l}}))$.

To improve readability the linguistic terms $A_{\mathrm{l},i}$ are sorted according to their frequencies $\hat{p}(A_{\mathrm{l},i}|C_r)$ in decreasing order $(A^{\mathrm{sort}}_{\mathrm{l},i})$ and terms $A^{\mathrm{sort}}_{\mathrm{l},m^{\mathrm{sort}}_\mathrm{l}+1} \ldots A^{\mathrm{sort}}_{\mathrm{l},m_\mathrm{l}}$ with $\hat{p}(A_{\mathrm{l},i}|C_r) < 0.2$ are discarded.

The building block for comparisons consists of `text` (*comparison*) with the terms of *greater*, *smaller*, and *different*. It provides statements on the values of x_1 for class $C_r = B_j$ in comparison to the other classes $\bar{B}_j$. The analysis is based on conditional probabilities of terms as described in [33]. The frequency of this comparison being true is expressed by $\hat{p}(comparison)$ and also presented as a term `text` ($\hat{p}(comparison)$).

Furthermore, the meaning of linguistic terms of features is explained by providing the α-cut ($\alpha = 0.5$) of their membership functions. For the first and the last term, the building blocks *smaller than* and *greater than*, for interior terms *between ... and ...* are used to express these intervals.

Text blocks for further features start with *In addition, the samples for ...*.

A second text frame presents the rule itself:

From this dependencies follows a rule to describe `text` ($\hat{p}(P_r|C_r)$) *cases of* `text` (B_{Rr}). *If* `text` (P_{r1}) *and* $\cdots$ *and* `text` (P_{rs}) *follows* `text` ($\hat{p}(C_r|P_r)$) `text` (B_{Rr}).

To shorten the partial premises terms combined by disjunction (*first term*, *second term*,... or *last term*) are replaced by *first term* to *last term*, the statements on α-cuts are summarized by the α-cut corresponding to the disjunction.

5. Application to diagnosis in Instrumented Gait Analysis

This section presents a complex example to demonstrate the potential of the methodology introduced. The example is a problem from Instrumented Gait Analysis [43], where the task consists in finding a characterization of patients with diplegic cerebral palsy (ICP) in comparison to healthy test persons in terms of their gait patterns. This task is solved as a classification problem with four classes: ICP patients—B_1 (86 examples), test persons with medium walking pace B_2, slow walking pace B_3 and fast walking pace B_4 (20 examples in each class) [34–39]. From the recordings of a 3D video system time series of several joint angles (pelvis, hip, knee, foot) in three different planes (sagittal, coronal, transverse) are computed. Feature extraction from the time series results in $s = 4620$ features which are assigned a priori relevance weights according to their category reflecting interpretability and measurement reliability. The mean a priori preference for all features is $\varnothing M_{\mathrm{l,ap}} = 0.43$. There are 16 features with $M_{\mathrm{l,ap}} = 1$.

The rule base generated by the algorithm described in Section 4 (see Complete Design in Table 4) with $\alpha = 1, \beta = 10$ contains six rules using eight features altogether. A tenfold cross-validation estimates a mean classification error (MCE) of approx. 3.9%. The most important rule

R_1: IF x_{298} = (PM $\vee$ PB $\vee$ PVB) $\wedge$ x_{540} = (PVS $\vee$ PS $\vee$ PM)
THEN Person = ICP patient

characterizes the gait pattern of ICP patients (Fig. 3).

The parameters of the membership functions are found by the algorithm from Section 4.1 with $m_i = 5$ for all features, illustrated for feature x_{540}

$$x_{540}^{\mathrm{sort}}[j_1, \ldots, j_5] = (\,0.66\ 1.58\ 2.61\ 3.46\ 5.28\,),$$
$$p = -\mathrm{rd}(\log_{10}(5.28 - 0.66) - 0.5) = 0,$$
$$a_{540,1\ldots5} = (\,1\ 2\ 3\ 3\ 5\,) \rightarrow a_{540,1\ldots5} = (\,1\ 2\ 2.6\ 3.5\ 5\,).$$

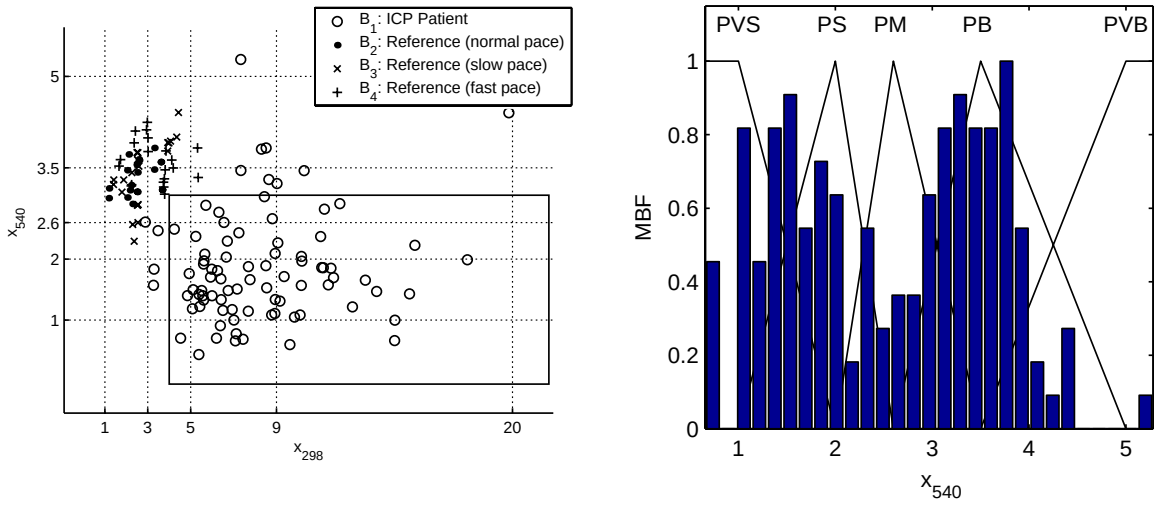

Fig. 3. Most important rule to characterize the gait pattern of ICP patients in a 2D display (left) and membership functions of feature x_{540} with histogram (right).

Such rounded parameters and automatically defined labels are a step ahead to solve the problem of intuitively understandable membership functions and terms in data-driven design. The labels always refer to relative statements for the distribution in the given training data set. A better interpretability would require a manual design—which is not feasible for real-world problems with so many features. All five terms have positive parameters which according to Section 4.1 are labeled PVS, PS, PM, PB and PVB.

Their conditional probabilities given class B_1 are

$$\hat{p}(A_{540,1\ldots5}|B_1) = (\,0.36\ \ 0.39\ \ 0.15\ \ 0.08\ \ 0.02\,). \tag{8}$$

The conditional probabilities of premise and conclusion are

$$\mathbf{R}_{B|P} = \begin{pmatrix} \hat{p}(C_1|P_1) & \hat{p}(C_1|\bar{P}_1) \\ \hat{p}(\bar{C}_1|P_1) & \hat{p}(\bar{C}_1|\bar{P}_1) \end{pmatrix} = \begin{pmatrix} 1.00 & 0.14 \\ 0.00 & 0.86 \end{pmatrix}, \tag{9}$$

$$\mathbf{R}_{P|B} = \begin{pmatrix} \hat{p}(P_1|C_1) & \hat{p}(P_1|\bar{C}_1) \\ \hat{p}(\bar{P}_1|C_1) & \hat{p}(\bar{P}_1|\bar{C}_1) \end{pmatrix} = \begin{pmatrix} 0.85 & 0.02 \\ 0.15 & 0.98 \end{pmatrix}. \tag{10}$$

These values lead to the following interpretations: There are no misclassifications $\hat{p}(C_1|P_1) = 1$. The rule covers 85% of the examples of ICP patients: $\hat{p}(P_1|C_1) = 0.85$. Rule evaluation with (4–6) and $\beta = 10$ yields a value of $Q = 0.48$ for the relevance measure ($E = 4.3$, $E_0 = 8.4$, $Q_{\mathrm{cl}} = 1.00$).

The complete names of features `text` (x_1) are generated from the name of the times series and the feature type. With the building blocks expressing the probabilities in (8,9) and the respective values for

Table 4
Comparison of different classifier concepts with s_C: number of used features and their average a priori preference ($\varnothing M_{\mathrm{l,ap}}$), $r_{\max}$: number of rules, $\varnothing P_{rl}$: average number of partial premises per rule, MCE: mean classification error in [%] for training data set and for 10 × 10 crossvalidation: mean value ± standard deviation

Classifier	s_C ($\varnothing M_{\mathrm{l,ap}}$)	$r_{\max}$	$\varnothing P_{rl}$	MCE [%] Train	MCE [%] 10 × 10 CV
Complete design (CD)	8 (0.78)	6	2.0	0.7	3.9 ± 1.1
CD, MBF without (7)	7 (0.82)	7	1.9	2.7	6.2 ± 1.2
CD, only single DT with IP	5 (0.81)	5	2.2	2.7	9.2 ± 1.3
CD, no $M_{\mathrm{l,ap}}$	8 (0.58)	7	1.9	0.7	6.8 ± 2.6
CD, $\alpha = 10$ in (1)	8 (0.86)	7	1.9	2.7	6.6 ± 2.2
CD, no rule base search	13 (0.78)	20	1.9	1.4	4.0 ± 1.6
DT, IP, no $M_{\mathrm{l,ap}}$	5 (0.50)	21	2.1	1.4	8.8 ± 2.6
DT, IP, with $M_{\mathrm{l,ap}}$	7 (0.83)	33	2.7	0.7	12.7 ± 1.7
DT, no IP, no $M_{\mathrm{l,ap}}$	6 (0.38)	25	2.3	0.0	12.8 ± 3.5
DT, no IP, with $M_{\mathrm{l,ap}}$	9 (0.94)	41	2.9	0.0	9.8 ± 1.5
RPART	3 (0.73)	4	2.2	4.8	7.5 ± 4.9
CART	3 (0.66)	4	2.0	2.7	10.2 ± 9.0
ML, $s_{\mathrm{m}} = 8$, $s_{\mathrm{d}} = 2$	8 (0.54)	—	—	2.0	3.6 ± 1.5

x_{298} the following textual description of rule R_1 is produced:

> The samples for ICP patient are characterized by the range of motion of pelvis anterior–posterior tilt during stride (x_{298}). This feature is mostly bigger than otherwise (usually big (between 7 and 14.5) and sometimes medium (between 4 and 7)). In addition, these samples can be described by the maximum value of velocity of knee flexion–extension during stride (x_{540}). It is usually smaller than otherwise (sometimes very small (smaller than 1.5) and sometimes small (between 1.5 and 2.3)). From this dependencies follows a rule to describe most cases for ICP patient. If the range of motion of pelvis anterior–posterior tilt during stride (x_{298}) is medium to very big (more than 4) and the maximum value of velocity of knee flexion–extension during stride (x_{540}) is very small to medium (smaller than 3.05) follows always ICP patient.

Analogously, the remaining rules can be presented in this automatically generated textual form. By this means, the user is offered additional information, which is not comprehensible by IF–THEN rules as in Section 2.

Table 4 compares different classifiers. The proposed complete design concept (CD) with all elements to improve the interpretability is characterized by a good compromise between accuracy (MCE CV 3.9%) and interpretability. It bases on 4 decision trees for the multi-class problem as proposed in Section 4.3.

The same concept without the interpretability modification of membership functions (MBF) in (7) has a reduced accuracy (MCE CV 6.2%). This result is counter-intuitive because a better accuracy might be expected. A possible explanation is a lower stability of found MBF parameters in comparison to CD with rounded parameters during crossvalidation. In addition, its MBF parameters are more difficult to understand.

A generation of only one single decision tree (DT) separating all classes simultaneously also leads to a reduced accuracy (MCE CV 9.2%). The reason is a too small set of hypotheses for pruning and rule base search.

If a concept uses a priori preferences of features $M_{\mathrm{l,ap}}$, the mean a priori preference of the used features in the classifier is higher because the classifiers try to use similar features with higher user preferences (e.g. $\varnothing M_{\mathrm{l,ap}} = 0.58 \rightarrow 0.78$ for CD). There is no clear accuracy difference between approaches with and without $M_{\mathrm{l,ap}}$. Here, two opposite effects might interact: (1) a reduced accuracy with $M_{\mathrm{l,ap}}$ due to more interpretable, but less discriminative features, (2) an increased accuracy with $M_{\mathrm{l,ap}}$ due to exploiting user knowledge about more reliable features. The user can control the first trade-off by tuning α in (1). As an example, $\alpha = 10$ leads to a higher average a priori preference within the set of used features (e.g. $\varnothing M_{\mathrm{l,ap}} = 0.78 \rightarrow 0.86$), but to a lower accuracy.

All DT algorithms use five membership functions per feature without (7). Implicit pruning (IP) means that the development of new leaves is stopped if a statistical evaluation does not find features with significant positive information gains. Approaches without IP are overfitted with bad classification errors. All trees need many rules (21–41) to discriminate classes which is mainly caused by the five membership functions per feature.

A CD stopped after pruning without a search for a rule base with cooperating rules has an acceptable accuracy but a lower interpretability due to much more features and rules in comparison to the CD.

Other decision tree algorithms as RPART and CART [7] tend to oversimplified solutions resulting in higher classification errors.

The maximum-likelihood classifier (ML; feature selection of $s_{\mathrm{m}} = 8$ features by a multivariate analysis of variances—MANOVA, dimension reduction to $s_{\mathrm{d}} = 2$ aggregated features by a discriminant analysis, maximum likelihood classification with estimated parameters of Gaussian distributions for each class) has the best accuracy due to a small number of unknown parameters and only limited deviations from a Gaussian distribution in this problem. Nevertheless, it is hardly interpretable due to using probability distributions in linear-transformed features spaces. In addition, it also prefers features with low-user preferences.

Altogether, the interpretability modifications lead in the discussed problem to small rule bases with only few rules, a small feature set with high a priori preferences and with understandable membership functions—without any significant loss of classifier accuracy. As a consequence, they contribute to an improved interpretability according to all aspects discussed in Section 1.

6. Conclusion

Interpretability is the main advantage of fuzzy systems in applications like knowledge extraction from data and decision support. A data-based design of fuzzy systems has to include measures to enhance interpretability according to the initially stated criteria. The paper presents a method for an automatic and complete design which makes little assumptions on a priori information besides the learning data set. However, additional information, e.g. on the preference of features or their membership functions, can be included in the design. Interpretability of the generated fuzzy system is obtained by structural choices regarding the type of membership functions, rules and inference mechanism on the one hand, and including interpretability criteria in the rule/rule base evaluation, on the other hand. Moreover, the developed algorithm enables the user to control the trade-off between accuracy and interpretability. As

a result, the learning algorithm produces relevant individual rules from which a rather small subset of cooperating rules is selected for the rule base. To improve the acceptance of the method itself and the results, rules and additional information are presented in a textual form. Experience gained in the field of Instrumented Gait Analysis, from which the presented example is taken, is very promising.

Acknowledgements

We would like to thank Tobias Loose, Sebastian Beck, Markus Reischl, Georg Bretthauer from Forschungszentrum Karlsruhe, and our project partners from the University of Heidelberg (Department of Orthopaedic Surgery), especially Rüdiger Rupp, Leonhard Döderlein, and Hans Jürgen Gerner, for inspiring discussions.

References

[1] K.-P. Adlassnig, A fuzzy logical model of computer-assisted medical diagnosis, Methods Inform. Med. 19 (1980) 141–148.
[2] R. Babuška, Fuzzy Modeling for Control, Kluwer Academic Press, Boston, 1998.
[3] U. Bodenhofer, P. Bauer, A formal model of interpretability of linguistic variables, in: J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling, Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 2002.
[4] A. Bonarini, Evolutionary learning of general fuzzy rules with biased evaluation functions: competition and cooperation, in: Proc. 1st IEEE Conf. on Evolutionary Computation, vol. 1, IEEE Press, Piscataway, NJ, 1994, pp. 51–56.
[5] P.P. Bonissone, K.S. Decker, Selecting uncertainty calculi and granularity: an experiment in trading-off precision and complexity, in: L.N. Kanal, J.F. Lemmer (Eds.), Uncertainty in Artificial Intelligence, North-Holland, Amsterdam, 1986, pp. 217–247.
[6] X. Boyen, L. Wehenkel, Automatic induction of fuzzy decision trees and its application to power system security assessment, Fuzzy Sets and Systems 102 (1) (1999) 3–19.
[7] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth, Belmont, CA, 1984.
[8] J. Buckley, Y. Hayashi, Fuzzy neural networks: a survey, Fuzzy Sets and Systems 66 (1994) 1–13.
[9] J. Casillas, O. Cordón, F. Herrera, L. Magdalena, Finding a balance between interpretability and accuracy in fuzzy rule-based modeling: an overview, in: J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling, Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 2002.
[10] J. Casillas, O. Cordón, F. Herrera, L. Magdalena (Eds.), Trade-off between Accuracy and Interpretability in Fuzzy Rule-Based Modelling, Studies in Fuzziness and Soft Computing, Physica, Heidelberg, 2002.
[11] M.-Y. Chen, D.A. Linkens, Rule-base self-generation and simplification for data-driven fuzzy models, Fuzzy Sets and Systems 142 (2) (2004) 243–265.
[12] K.J. Cios, W. Pedrycz, R.W. Swiniarski, Data Mining Methods for Knowledge Discovery, Kluwer Academic Press, Boston, 1998.
[13] O. Cordón, F. Herrera, Identification of linguistic fuzzy models by means of genetic algorithms, in: H.H.D. Driankov (Ed.), Fuzzy Model Identification, Selected Approaches, Springer, Berlin, 1997, pp. 215–250.
[14] O. Cordón, F. Herrera, A proposal for improving the accuracy of linguistic modeling, IEEE Trans. Fuzzy Systems 8 (3) (2000) 335–344.
[15] O. Cordón, F. Herrera, I. Zwir, A hierarchical knowledge-based environment for linguistic modeling: models and iterative methodology, Fuzzy Sets and Systems 138 (2) (2003) 307–341.
[16] M. Delgado, A.F. Gómez-Skarmeta, F. Martín, A methodology to model fuzzy systems using fuzzy clustering in a rapid-prototyping approach, Fuzzy Sets and Systems 97 (1998) 287–301.
[17] M. Drobics, U. Bodenhofer, E.P. Klement, FS-FOIL: an inductive learning method for extracting interpretable fuzzy descriptions, Internat. J. Approx. Reasoning 32 (2–3) (2003) 131–152.

[18] F. Eshragh, E.H. Mamdani, A general approach to linguistic approximation, Internat. J. Man-Mach. Stud. 11 (1979) 501–519.
[19] J. Espinosa, J. Vandervalle, Constructing fuzzy models with linguistic integrity from numerical data—AFREILI algorithm, IEEE Trans. Fuzzy Systems 8 (5) (2000) 591–600.
[20] T.P. Hong, S.S. Tseng, A generalised version space learning algorithm for noisy and uncertain data, IEEE Trans. Knowledge Data Eng. 9 (1997) 336–340.
[21] F. Höppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis, Wiley, Chichester, 1999.
[22] H. Ichibuchi, K. Nozaki, H. Tanaka, Efficient fuzzy partition of pattern space for classification problems, Fuzzy Sets and Systems 59 (1993) 295–304.
[23] J. Jäkel, L. Gröll, R. Mikut, Tree-oriented hypothesis generation for interpretable fuzzy rules, in: Proc. 7th Europ. Congr. on Intelligent Techniques and Soft Computing EUFIT'99, September 13–16, 1999, Aachen,1999, pp. 279–280 CD-ROM.
[24] J. Jäkel, L. Gröll, R. Mikut, Automatic generation and evaluation of interpretable fuzzy rules, in: M. Mohammadian (Ed.), New Frontiers in Computational Intelligence and its Applications, IOS Press, Amsterdam, 2000, pp. 1–10.
[25] J. Jäkel, L. Gröll, R. Mikut, On fuzzy connectives in the context of automatic rule generation, in: Proc 9th Zittau Fuzzy Colloquium, 17.09.2001–19.09.2001, 2001, pp. 230–237.
[26] J.-S.R. Jang, C.-T. Sun, Neuro-fuzzy modeling and control, Proc. IEEE 83 (3) (1995) 378–406.
[27] Y. Jin, Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement, IEEE Trans. Fuzzy Systems 8 (2) (2000) 212–221.
[28] Y. Jin, W. von Seelen, B. Sendhoff, An approach to rule-based knowledge extraction, in: Proc. IEEE Conf. on Fuzzy Systems, Anchorage, Alaska, 1998, pp. 1188–1193.
[29] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.
[30] F. Klawonn, D. Nauck, R. Kruse, Generating rules from data by fuzzy and neuro-fuzzy methods, in: Proc. Fuzzy-Neuro-Systeme '95, Darmstadt 1995, pp. 223–230.
[31] A. Krone, H. Kiendl, An evolutionary concept for generating relevant fuzzy rules from data, Internat. J. Knowledge-based Intell. Eng. Systems 1 (4) (1997) 207–213.
[32] A. Krone, H. Taeger, Data-based fuzzy rule test for fuzzy modelling, Fuzzy Sets and Systems 123 (3) (2001) 343–358.
[33] T. Loose, J. Jäkel, R. Mikut, Datenbasierte Generierung natürlichsprachlicher Erklärungstexte am Beispiel der Instrumentellen Ganganalyse, in: Proc. 12th Workshop Fuzzy on Systems, Dortmund, Forschungszentrum Karlsruhe, FZKA 6767, 2002, pp. 43–57.
[34] T. Loose, H. Malberg, R. Mikut, R. Rupp, J. Simon, S. Wolf, L. Döderlein, Quantitative evaluation of CP gait after botulinum toxin therapy on the basis of a normalcy measure, Gait Posture 16 (Suppl. 1) (2002) 176–177.
[35] T. Loose, R. Mikut, H. Malberg, J. Simon, M. Schablowski, R. Rupp, L. Döderlein, A computer based method to assess gait data, in: IFMBE Proc. 2nd European Medical and Biological Engineering Conference, EMBEC, 2002, pp. 798–799.
[36] C. Marsala, B. Bouchon-Meunier, Choice of a method for the construction of fuzzy decision trees, in: Proc. IEEE Internat. Conf. on Fuzzy Systems, FUZZ-IEEE'03, St Louis (USA), May 2003, pp. 584–589.
[37] R. Mikut, J. Jäkel, L. Gröll, Automatic design of interpretable membership functions, in: Proc. 8th Zittau Fuzzy Colloquium, September 6–8, 2000, Hochschule Zittau/Görlitz, 2000, pp. 103–111.
[38] R. Mikut, J. Jäkel, L. Gröll, Inference methods for partially redundant rule bases, in: R. Hampel, M. Wagenknecht, N. Chaker (Eds.), Fuzzy Control: Theory and Practice, Advances in Soft Computing, Physica, Heidelberg, 2000, pp. 177–185.
[39] R. Mikut, T. Loose, J. Jäkel, Rule-oriented information acquisition from biological time series in clinical decision making, in: Proc. 10th Zittau Fuzzy Colloquium, Hochschule Zittau/Görlitz, 2002, pp. 300–307.
[40] G.A. Miller, Note on the bias of information estimates, in: H. Quastler (Ed.), Information Theory in Psychology, Free Press, Glencoe, IL, 1955, pp. 95–100.
[41] G.A. Miller, The magical number seven plus or minus two: some limits on our capacity to process information, Psychol. Rev. 63 (1956) 81–97.
[42] J. Valente de Oliveira, Semantic constraints for membership function optimization, IEEE Trans. Systems Man Cybernetics—Part A: Systems and Humans 29 (1) (1999) 128–138.
[43] J. Perry, Gait Analysis, Normal and Pathological Function, Slack Inc., Thorofare, 1992.
[44] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[45] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.

[46] S. Renooij, C.L.M. Witteman, Talking probabilities: communicating probabilistic information with words and numbers, Internat. J. Approx. Reasoning 22 (1999) 169–194.
[47] J. Rives, FID3: Fuzzy induction decision tree, in: Proc 1st Internat. Symp. Uncertainty, Modelling and Analysis, IEEE Computer Society Press, Los Alamitos, CA, 1990, pp. 457–462.
[48] M. Setnes, R. Babuška, U. Kaymak, H.R.V. Nauta Lemke, Similarity measures in fuzzy rule base simplification, IEEE Trans. Systems Man Cybernetics—Part B: Cybernetics 28 (3) (1998) 376–386.
[49] M. Setnes, J. Roubos, GA-fuzzy modeling and classification: complexity and performance, IEEE Trans. Fuzzy Systems 8 (5) (2000) 509–522.
[50] M. Sugeno, T. Yasukawa, A fuzzy-logic-based approach to qualitative modeling, IEEE Trans. Fuzzy Systems 1 (1) (1993) 7–31.
[51] C.-H. Wang, J.-F. Liu, T.-P. Hong, S.-S. Tseng, A fuzzy inductive learning strategy for modular rules, Fuzzy Sets and Systems 103 (1999) 91–105.
[52] R.R. Yager, On the construction of hierarchical fuzzy systems models, IEEE Trans. Systems Man Cybernet.—Part C 28 (1) (1998) 55–66.
[53] J. Yen, L. Wang, Simplifying fuzzy rule-based models using orthogonal transformation methods, IEEE Trans. Sytems Man Cybernet.—Part B 29 (1) (1999) 13–24.
[54] Y. Yoshinari, W. Pedrycz, K. Hiroto, Construction of fuzzy models through clustering techniques, Fuzzy Sets and Systems 54 (3) (1993) 157–165.
[55] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (2) (1995) 125–139.
[56] L.A. Zadeh, Probability measures of fuzzy events, J. Math. Anal. Appl. 10 (1968) 421–427.
[57] L.A. Zadeh, From computing with numbers to computing with words—from manipulation of measurements to manipulation of perceptions, IEEE Trans. on Circuits and Systems, I: Fundamental Theory and Applications 45 (1999) 105–119.
[58] L. Zadeh, J. Kacprzyk (Eds.), Computing with Words in Information/Intelligent Systems, vol. 1: Foundations, vol. 2: Applications, Physica-Verlag, Heidelberg, 2000.