
ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

INTERNATIONAL JOURNAL OF APPROXIMATE REASONING

International Journal of Approximate Reasoning 32 (2003) 131–152

www.elsevier.com/locate/ijar

FS-FOIL: an inductive learning method for extracting interpretable fuzzy descriptions

Mario Drobics [a], Ulrich Bodenhofer [a,*], Erich Peter Klement [b]

[a] *Software Competence Center Hagenberg, Hauptstrasse 99, A-4232 Hagenberg, Austria*
[b] *Fuzzy Logic Laboratorium Linz-Hagenberg, Department of Algebra, Stochastics, and Knowledge-Based Math. Systems, Johannes Kepler University, A-4040 Linz, Austria*

Received 1 January 2002; accepted 1 April 2002

Abstract

This paper is concerned with FS-FOIL – an extension of Quinlan's First-Order Inductive Learning Method (FOIL). In contrast to the classical FOIL algorithm, FS-FOIL uses fuzzy predicates and, thereby, allows to deal not only with categorical variables, but also with numerical ones, without the need to draw sharp boundaries. This method is described in full detail along with discussions how it can be applied in different traditional application scenarios – classification, fuzzy modeling, and clustering. We provide examples of all three types of applications in order to illustrate the efficiency, robustness, and wide applicability of the FS-FOIL method.

Keywords: Clustering; Data mining; Fuzzy rules; Inductive learning; Interpretability; Machine learning

1. Introduction

There is no unique commonly accepted one-sentence definition of *data mining*, *machine learning*, or the more general term *information mining* that has

* Corresponding author.
E-mail address: ulrich.bodenhofer@scch.at (U. Bodenhofer).

0888-613X/02/$ - see front matter
PII: S0888-613X(02)00080-4

become fashionable in the last few years. In the authors' humble opinion, "*the non-trivial extraction of implicit, previously unknown, and potentially useful information from data*" [21] is a pretty good compromise between indispensable generality and a precise description of the matter's very core.

The fields of data mining and machine learning can roughly be divided into two large parts. On the one hand, the analysis of causal relationships between causes and effects is a major part. More specifically, one is often interested which particular input features lead to specific output values (goals). Since explicit goal information is available, such kinds of problems are often called *supervised* (a term stemming from the neural networks world). Most importantly, classification, prediction, and, less traditionally, data-driven fuzzy modeling fall into that category. On the other hand, it is often necessary to identify regions of similarity or other dependencies from unstructured data sets – without any explicit goal information. This class of data mining problems is often called *unsupervised*. Typical clustering problems like market segmentation, etc. can be assigned to this category.

Quinlan's *First-Order Inductive Learning Algorithm* (*FOIL*) [37,41] is usually assigned to supervised machine learning. FOIL tries to build up a successive coverage of those regions in the data which potentially imply a specific goal/output predicate. Since FOIL relies on binary Boolean logic, it can only process Boolean predicates. This means that non-categorical, in particular numerical, variables have to be represented by a finite set of Boolean predicates. In the case of numerical variables, this is almost always accomplished by splitting the numerical domain into a finite partition consisting of intervals. This ad-hoc granulation, however, often leads to a significant loss of information and robustness – caused by artificially sharp boundaries between the different predicates (particularly in case that the numerical data are noisy). Moreover, it is not possible to extract smooth functional relationships between numerical variables in a straightforward way. The common extension FFOIL [43] is designed for learning functional relationships in a Prolog-like fashion, however, it is also strictly based on binary logic and suffers from the same difficulties in terms of instability caused by sharp boundaries.

The given paper presents the *FS-FOIL* algorithm – a fuzzy variant of FOIL which overcomes these difficulties. By its ability to use fuzzy predicates instead of only crisp ones, FS-FOIL allows to extract linguistically expressive (interpretable) rules from both categorical and numerical data, while avoiding the problem of artificially sharp boundaries. As another highly important effect, FS-FOIL also allows to obtain smooth functional models from the rules it generates.

This paper is organized as follows. After necessary basics provided in Section 2, Sections 3 and 4 give a detailed description of FS-FOIL. Following that, we elucidate practical settings in which FS-FOIL can be employed beneficially. Section 5 is devoted to the straightforward application to classification prob-

lems. In Section 6, we demonstrate FS-FOIL's ability to model numerical functions by means of fuzzy rules. Next, we will see, however, that possible applications do not only arise in typical supervised settings, but also in unsupervised ones – we apply FS-FOIL to the problem of finding interpretable cluster descriptions in Section 7.

2. The basic setting

One of the most fundamental tasks in machine learning is the identification of input–output relationships from data samples. Assume, therefore, that we are given a set of K samples

$$\mathtt{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\}.$$

Each sample has the same $(n+m)$-dimensional structure (for all $i = 1, \ldots, K$):

$$\mathbf{x}^i = (x_1^i, \ldots, x_n^i, x_{n+1}^i, \ldots, x_{n+m}^i) \in X_1 \times \cdots \times X_n \times X_{n+1} \times \cdots \times X_{n+m}.$$

The first n dimensions/variables are the inputs; the latter m dimensions/variables are the outputs under investigation. In the following, we refer to the rth dimension $(r = 1, \ldots, n)$ as rth *input attribute*. The other m dimensions are called *goal attributes*. Ideally, the overall objective of this machine learning problem is to find a function

$$f : X_1 \times \cdots \times X_n \to X_{n+1} \times \cdots \times X_{n+m}$$

such that the inherent connection between the input attributes and the goal attribute hidden in the data set $\mathtt{X}$ is modeled as well as possible. Therefore, such machine learning problems can be regarded as some kind of data fitting.

To find such a function f, however, is not always the only objective. While statistical regression [15] or neural networks [33,46,52] allow to solve such kinds of machine learning problems, they leave the resulting function f as a *black box*, i.e. a plain function whose internals are difficult or impossible to comprehend. In many practical applications, however, qualitative insights into the structures of f are desirable. For such tasks, *rule-based systems* are most appropriate. They easily allow qualitative insight, since the function f is represented by logical rules in a close-to-natural-language manner. In the following, assume that we are not necessarily interested in the full function f, but at least in significant bits of knowledge about f and their inherent structures – *rules*.

We have not mentioned so far how input and goal attributes look like. In this paper, we would like to consider the following most important types:

Boolean categorical attributes. The domain X_i is an unstructured finite set of labels, for instance, types of car engines (gasoline, Diesel, hydrogen, electric) or classes of animals (birds, fish, mammals, etc.). The attribute values x_r^i are single elements of the label set X_i.

Fuzzy categorical attributes. There is again an unstructured finite set of labels, but with possible overlaps. Therefore, values of such kinds of variables may be fuzzy sets on this set of labels. For example, assume that we are given a finite set consisting of different grape varieties. Then blended wines (cuvees) cannot be assigned to single categories crisply.

Numerical attributes. The underlying domain X_i is the set of real numbers or a subset of these (e.g. an interval). The attribute values x_r^i are real numbers, e.g. pressures, temperatures, incomes, ratios, etc.

Note that Boolean categorical attributes are special cases of fuzzy categorical attributes, since any crisp label can be considered as a fuzzy set of labels, too.

Beside different variants of decision trees [34,36,40,42], the *First-Order Inductive Learning Algorithm* (FOIL) and its variants [11,37,41,43,44] have emerged as standard methodologies for rule-based machine learning. These methods in their classical form, however, can only process Boolean categorical attributes. Numerical attributes can be processed in principle, but have to be converted to Boolean categorical ones by splitting the domain into a finite number of subsets (intervals most often) and assigning labels to them.

FS-FOIL is a generalization of FOIL that is capable of processing all three kinds of attributes without the need to convert any of them into Boolean categorical attributes.

3. The language of FS-FOIL

Like in the original FOIL algorithm, the language of FS-FOIL consists of first-order predicates. While FOIL works with Boolean predicates that are canonically given, since the attributes are assumed to be Boolean categorical ones, the situation in the setup of FS-FOIL is slightly more complicated. In order to deal with numerical and fuzzy categorical attributes, FS-FOIL works with fuzzy predicates instead of Boolean ones. Therefore, we have to consider how the different kinds of fuzzy predicates are defined and interpreted. A fuzzy predicate is a $X_1 \times \cdots \times X_{n+m} \to [0,1]$ mapping that maps each element $\mathbf{x} \in X_1 \times \cdots \times X_{n+m}$ to a degree of fulfillment. Since the predicates in this paper are induced by a certain kind of linguistic expressions, we will make an explicit distinction between the expressions and their corresponding semantics: similar to formal logics, we use a dummy function t to compute the actual truth value to which a sample fulfills a given predicate.

All predicates we consider in this paper are either induced by a single attribute (we call those ones *atomic predicates* in the following) or *compound predicates* that are defined as compositions of atomic predicates by means of fuzzy logical operations, such as conjunction or disjunction.

Let us consider Boolean categorical attributes first. Assume that an arbitrary, but fixed, attribute with index $r \in \{1, \ldots, n+m\}$ is belonging to that class. Its domain is given as an unstructured set of N_r labels:

$$X_r = \{L_{r,1}, \ldots, L_{r,N_r}\}.$$

Then we can define $2 \cdot N_r$ atomic predicates for attribute r, all of which are defined on the space $X_1 \times \cdots \times X_{n+m}$. Therefore, given a sample

$$\mathbf{x} \in X_1 \times \cdots \times X_n \times X_{n+1} \times \cdots \times X_{n+m},$$

the truth values to which $\mathbf{x}$ fulfills the two predicates induced by label $L_{r,j}$ ($j = 1, \ldots, N_r$) are given as

$$t(\mathbf{x} \text{ is } L_{r,j}) = \begin{cases} 1 & \text{if } x_r = L_{r,j}, \\ 0 & \text{otherwise}, \end{cases}$$

$$t(\mathbf{x} \text{ is not } L_{r,j}) = \begin{cases} 1 & \text{if } x_r \neq L_{r,j}, \\ 0 & \text{otherwise}. \end{cases}$$

Now assume that attribute r is a fuzzy categorical one. In this case, we have an unstructured set of N_r labels $\{L_{r,1}, \ldots, L_{r,N_r}\}$ again. As mentioned already, however, the data samples are fuzzy sets of labels, i.e. [1]

$$X_r = \mathbb{F}(\{L_{r,1}, \ldots, L_{r,N_r}\}) \simeq [0,1]^{N_r}.$$

A single data sample $x_r \in X_r$, therefore, can be represented by an N_r-dimensional vector of truth values from the unit interval:

$$x_r = (t_{r,1}, \ldots, t_{r,N_r}). \tag{1}$$

Note that it is often useful, albeit not necessary, to require that $\sum_{j=1}^{N_r} t_{r,j} = 1$. Hence, we can define $2 \cdot N_r$ atomic fuzzy predicates for attribute r. For a sample $\mathbf{x} \in X_1 \times \cdots \times X_{n+m}$, the truth values to which the two predicates induced by the label $L_{r,j}$ are fulfilled can be defined as follows (note that the rth component of $\mathbf{x}$ is given as in (1)):

$$t(\mathbf{x} \text{ is } L_{r,j}) = t_{r,j},$$

$$t(\mathbf{x} \text{ is not } L_{r,j}) = 1 - t_{r,j}.$$

To be able to handle numeric attributes as well, it is indispensable to define a discrete set of predicates for these kinds of attributes, too. If this quantization is done by means of partitions into crisp sets (intervals) as in traditional machine learning, small variations (e.g. noise) can cause large changes in the classification quality and instable results. This entails the demand for admitting vagueness in the assignment of samples to predicates. Fuzzy sets [49] perfectly

[1] For a given non-empty set X, $\mathbb{F}(X)$ denotes the set of fuzzy sets on X, i.e. the set of $X \to [0,1]$ mappings.

solve this problem of artificial preciseness arising from sharp interval boundaries.

Suppose that the rth attribute is numerical. This means that $X_r \subseteq \mathbb{R}$ and the values in the rth component are real numbers. We assume that, for attribute r, a family of N_r linguistic labels $M_{r,1}, \ldots, M_{r,N_r}$ is defined. Depending on the underlying context of the attribute under consideration, these labels can be natural language expressions like *very low*, *medium*, *large*. To each label $M_{r,j}$, we assign a fuzzy set with membership function $\mu_{M_{r,j}} \in \mathbb{F}(X_r)$ $(j = 1, \ldots, N_r)$. There are different ways to define the membership functions of those fuzzy sets. Firstly, it is possible to use equally sized fuzzy intervals (triangular fuzzy numbers or trapezoids). Secondly, an uneven distribution can be defined manually in case that there is specific knowledge about the particular attribute available. Thirdly, it is possible to use a clustering technique to generate unevenly distributed fuzzy sets according to the distribution of values in the data set $\mathtt{X}$. In our applications, we often use a modified c-means algorithm [30] with simple neighborhood interaction [17,18] to compute the centers of the fuzzy sets. The fuzzy sets are then arranged as trapezoids or bell-shaped functions around these centers. In any case, we strongly suggest to define families of fuzzy sets that form a partition and are in a proper order – to ensure highest interpretability of the results [9,10,12].

Given a set of linguistic labels $M_{r,1}, \ldots, M_{r,N_r}$ and their corresponding semantics modeled by fuzzy sets, we can define $4 \cdot N_r$ atomic fuzzy predicates. The degrees to which a sample $\mathbf{x} \in X_1 \times \cdots \times X_{n+m}$ fulfills these predicates can be computed as follows $(j = 1, \ldots, N_r)$:

$$
\begin{aligned}
&t(\mathbf{x} \text{ is } M_{r,j}) = \mu_{M_{r,j}}(x_r),\\
&t(\mathbf{x} \text{ is not } M_{r,j}) = 1 - \mu_{M_{r,j}}(x_r),\\
&t(\mathbf{x} \text{ is at least } M_{r,j}) = \sup\{\mu_{M_{r,j}}(u) \,|\, u \leqslant x_r\},\\
&t(\mathbf{x} \text{ is at most } M_{r,j}) = \sup\{\mu_{M_{r,j}}(u) \,|\, u \geqslant x_r\}.
\end{aligned}
$$

The two latter ordering-based predicates are not absolutely necessary, but help to improve compactness, expressiveness, and interpretability of the results [6,7,9,10,14].

For convenience, from now on, we will denote all predicates with uppercase letters and use prefix notation. Assume, therefore, that we have a set of atomic predicates $\mathtt{A} = \{A_1, \ldots, A_t\}$ induced by the input attributes and a set of atomic predicates $\mathtt{C} = \{C_1, \ldots, C_s\}$ induced by the goal attributes.

It remains to clarify how compound predicates can be defined. Suppose that we are given an appropriate couple consisting of a triangular norm and its dual triangular conorm (t-norms and t-conorms are commutative, associative, and non-decreasing binary operations on the unit interval with neutral elements 1 and 0, respectively [27]; a t-conorm S is dual to a t-norm T if the equality

$1 - S(x,y) = T(1-x, 1-y)$ holds for all $x, y \in [0,1]$). Two popular possible choices are the min-max operations

$$T_{\mathbf{M}}(x,y) = \min(x,y),$$
$$S_{\mathbf{M}}(x,y) = \max(x,y)$$

or the Łukasiewicz operations:

$$T_{\mathbf{L}}(x,y) = \max(x+y-1, 0),$$
$$S_{\mathbf{L}}(x,y) = \min(x+y, 1).$$

In the following, we will restrict ourselves to the Łukasiewicz operations. Conjunctions and disjunctions of two fuzzy predicates A and B (no matter whether atomic or not), therefore, can be defined as follows:

$$t((A \wedge B)(\mathbf{x})) = T_{\mathbf{L}}(t(A(\mathbf{x})), t(B(\mathbf{x}))),$$
$$t((A \vee B)(\mathbf{x})) = S_{\mathbf{L}}(t(A(\mathbf{x})), t(B(\mathbf{x}))).$$

Note that, for obvious reasons, we strictly separate predicates belonging to input attributes (resulting from the set of atomic predicates A) and goal attributes (resulting from predicate set C).

4. The learning algorithm

The overall goal of FS-FOIL is the following: given a goal predicate $\bar{C}$ (either from C or a compound of predicates out of C), we want to find a predicate $\bar{A}$ (either from A or a compound of predicates out of A) that describes those samples in the sample set X that fulfill $\bar{C}$. FS-FOIL creates a sequential coverage of these areas by means of a set S consisting of fuzzy predicates which are conjunctions of atomic predicates (so-called *Horn clauses*). The final predicate $\bar{A}$ is then given as the disjunction of the predicates in S, i.e. FS-FOIL uses a kind of disjunctive normal form to represent the description [39]:

$$t(\bar{A}(\mathbf{x})) = \bigvee_{A \in \mathtt{S}} t(A(\mathbf{x})) = \underset{A \in \mathtt{S}}{S_{\mathbf{L}}} t(A(\mathbf{x})).$$

Before we come to the very core of FS-FOIL, let us make a few definitions. The *degree of common fulfillment* of a predicate A and the goal predicate $\bar{C}$ (for a given sample $\mathbf{x}$) is defined as $t((A \wedge \bar{C})(\mathbf{x}))$. For a given finite sample set X, the *fuzzy set of samples fulfilling a predicate A*, which we denote with $A(\mathtt{X})$, is defined as (for all $\mathbf{x} \in \mathtt{X}$)

$$\mu_{A(\mathtt{X})}(\mathbf{x}) = t(A(\mathbf{x})).$$

The *cardinality* of a fuzzy set N on an arbitrary non-empty set X with finite cardinality is defined as the sum of $\mu_N(x)$, i.e.

$$|N| = \sum_{x \in X} \mu_N(x).$$

Hence, the cardinality of samples in $\mathtt{X}$ commonly fulfilling a predicate A and the goal predicate $\bar{C}$ can be defined by

$$|(A \wedge \bar{C})(\mathtt{X})| = \sum_{\mathbf{x} \in \mathtt{X}} t((A \wedge \bar{C})(\mathbf{x})) = \sum_{\mathbf{x} \in X} T_{\mathbf{L}}(t(A(\mathbf{x})), t(\bar{C}(\mathbf{x}))).$$

The objective of FS-FOIL is to find a set of predicates $\mathtt{S}$ that fulfill two specific quality criteria – *significance* and *accuracy*. The significance of a predicate A is defined as the *common support* of a predicate A and the goal predicate $\bar{C}$, i.e. the ratio between the cardinality of samples commonly fulfilling A and $\bar{C}$ and the total number of samples:

$$\mathrm{supp}(A, \bar{C}) = \frac{|(A \wedge \bar{C})(\mathtt{X})|}{|\mathtt{X}|} = \frac{1}{K} \cdot \sum_{i=1}^{K} T_{\mathbf{L}}(t(A(\mathbf{x}^i)), t(\bar{C}(\mathbf{x}^i))).$$

The *accuracy* of a predicate A is associated with the *confidence* of predicate A with respect to $\bar{C}$, which is defined as

$$\mathrm{conf}(A, \bar{C}) = \frac{\mathrm{supp}(A, \bar{C})}{\mathrm{supp}(A)},$$

where $\mathrm{supp}(A)$ is defined as

$$\mathrm{supp}(A) = \frac{|A(\mathtt{X})|}{|\mathtt{X}|} = \frac{1}{K} \cdot \sum_{i=1}^{K} t(A(\mathbf{x}^i)).$$

Hence, the following holds:

$$\mathrm{conf}(A, \bar{C}) = \frac{|(A \wedge \bar{C})(\mathtt{X})|}{|A(\mathtt{X})|} = \frac{\sum_{i=1}^{K} t((A \wedge \bar{C})(\mathbf{x}^i))}{\sum_{i=1}^{K} t(A(\mathbf{x}^i))}.$$

In other words, the confidence of A with respect to $\bar{C}$ is the ratio between the number of samples fulfilling $\bar{C}$ that are correctly described by A (i.e. jointly fulfilling A and $\bar{C}$) and the total number of samples fulfilling A.

Outline. FS-FOIL starts with the most general predicate $\top$, the predicate that always gives a truth value of 1, and successively expands it – thereby generating more and more specific predicates – until an input predicate A is found that covers a part of the area, where the goal predicate $\bar{C}$ is fulfilled, accurately and significantly enough. This procedure is iteratively repeated as long as there are undescribed samples remaining or no accurate and significant predicates can be found anymore.

We now provide the full algorithm and discuss its internals in detail.

Algorithm 1. (FS-FOIL)

Input: goal predicate $\bar{C}$
samples $\mathtt{X} = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\}$
set of atomic input predicates $\mathtt{A}$
Output: predicate set $\mathtt{S}$
final predicate set $\mathtt{S} = \emptyset$
intermediate predicate set $\mathtt{P} = \{\top\}$
set of predicates under consideration $\mathtt{E} = \mathtt{A}$
open nodes $O = \bar{C}(\mathtt{X})$
do{
$\mathtt{P}' =$ best k predicates of $\mathtt{P}$ according to information gain measure G
$\mathtt{P} =$ expansion of all predicates in $\mathtt{P}'$ using $\mathtt{E}$
prune predicate sets $\mathtt{P}$ and $\mathtt{E}$
if a predicate $A \in \mathtt{P}$ is accurate and significant
{
add predicate A to set $\mathtt{S}$
remove nodes covered by A from the set of open nodes O
$\mathtt{P} = \{\top\}$
$\mathtt{E} = \mathtt{A}$
}
} **while** stopping condition is not fulfilled

As obvious, the final predicate set $\mathtt{S}$ is initialized with the empty set. FS-FOIL works with an intermediate set of predicates $\mathtt{P}$ which are sequentially expanded in each iteration. This set is initialized with the trivial predicate $\top$. Moreover, there is a set of predicates $\mathtt{E}$ which contains those atomic predicates that may be considered for further expansions; it is initialized with all atomic input predicates from the set $\mathtt{A}$. The fuzzy set O corresponds to the samples from $\mathtt{X}$ fulfilling $\bar{C}$ which have not yet been described by a predicate in $\mathtt{S}$. Clearly, O is initialized with all samples fulfilling $\bar{C}$, i.e. $\bar{C}(\mathtt{X})$.

In contrast to the original FOIL algorithm, which performs a straightforward greedy hill climbing search, FS-FOIL employs a stepwise beam search approach to find a description (i.e. not only a single candidate, but the best k candidates are kept). This means concretely that, in the first step of the loop body, we select the best k predicates in $\mathtt{P}$ (e.g., a typical value is $k = 10$) according to the following entropy-based information gain measure G. If $\mathtt{P}$ contains less than k predicates, $\mathtt{P}'$ is set to $\mathtt{P}$ (see [43] for a detailed explanation):

$$G(A) = |(A \wedge \bar{C})(\mathtt{X})| \cdot \left(\log_2 \frac{|(A \wedge \bar{C})(\mathtt{X})|}{|A(\mathtt{X})|} - \log_2 \frac{|\bar{C}(\mathtt{X})|}{|\mathtt{X}|} \right).$$

Note that this is a slight adaptation of the original FOIL gain measure to the beam search used in FS-FOIL in order to obtain a total ordering on the set of all predicates [3].

In the next step, all predicates in $\mathtt{P}'$ are expanded by all atomic fuzzy predicates from $\mathtt{E}$. The expansion of a predicate A with an atomic fuzzy predicate B is done by means of conjunction, i.e. $A \wedge B$. In case that $\mathtt{P}$ only contains the initial trivial predicate $\top$, the expansion by an atomic predicate B is defined as $\top \wedge B = B$.

As expanding with all predicates is computationally very expensive, the third step is concerned with keeping the sets $\mathtt{P}$ and $\mathtt{E}$ as small as possible. This "pruning step" is done in the following way: all predicates $A \in \mathtt{P}$ whose support is lower than a given threshold, i.e. $\mathrm{supp}(A, \bar{C}) < \mathrm{supp}_{\min}$ are removed from $\mathtt{P}$. Moreover, all predicates that have not contributed to predicates with sufficient support upon expansion are removed from $\mathtt{E}$, i.e. we eliminate all predicates B from the set $\mathtt{E}$ for which

$$\mathrm{supp}(A \wedge B, \bar{C}) < \mathrm{supp}_{\min}$$

holds for all $A \in \mathtt{P}'$. We most often use a threshold of 1%, i.e. $\mathrm{supp}_{\min} = 0.01$. This pruning strategy does not eliminate predicates that possibly become important later, since the support of a predicate cannot increase by expansion with additional predicates. Moreover, this strategy ensures that, in each step, the set $\mathtt{P}$ only contains predicates whose supports are not smaller than $\mathrm{supp}_{\min}$.

Provided that there is a predicate $A \in \mathtt{P}$ which fulfills reasonable requirements both in terms of support and confidence, i.e.

$$\mathrm{supp}(A, \bar{C}) \geqslant \mathrm{supp}_{\min}, \tag{2}$$

$$\mathrm{conf}(A, \bar{C}) \geqslant \mathrm{conf}_{\min}, \tag{3}$$

we can add A to the final predicate set $\mathtt{S}$. In case that $\mathtt{P}$ contains more than one predicate fulfilling the above two criteria, the one with the highest information gain measure is selected. Consequently, we have to eliminate all those elements from O that have been covered by A. This is accomplished by replacing O with the intersection of the fuzzy set O and the fuzzy set of elements in $\mathtt{X}$ that have not been described by the predicate A:

$$\begin{aligned} T_{\mathbf{L}}(\mu_O(\mathbf{x}), 1 - \mu_{A(\mathtt{X})}(\mathbf{x})) &= \max(\mu_O(\mathbf{x}) + 1 - \mu_{A(\mathtt{X})}(\mathbf{x}) - 1, 0) \\ &= \max(\mu_O(\mathbf{x}) - \mu_{A(\mathtt{X})}(\mathbf{x}), 0). \end{aligned}$$

The loop terminates if either the percentage of remaining undescribed nodes $|O|/K$ falls under a certain threshold (we use a typical value of (10%) or no new significant and accurate predicates can be found by expansion anymore.

Since $\bar{A}$ only involves interpretable atomic predicates and logical operations, $\bar{A}$ can be regarded as a natural language expression which *describes* the set of those input values which also fulfill goal predicate $\bar{C}$. By applying the confi-

dence criterion (3), FS-FOIL tries to avoid that there are samples which fulfill $\bar{A}$ but do not fulfill $\bar{C}$. Therefore, the final result can be considered as a rule

$$\text{IF } \bar{A}(\mathbf{x}) \text{ THEN } \bar{C}(\mathbf{x})$$

interpreted in the sense of Mamdani [31,32], i.e. as a kind of conditional assignment instead of an implication in the strict logical sense [19,20,23,38,39].

5. The typical application: classification

The classical application of supervised machine learning is classification. Typically, the objective of classification is the following: given n input attributes (feature values) and one Boolean categorical goal attribute ($m = 1$) with N_{n+1} different labels (classes), we have to find a set of rules that is able to assign a sample to one of the N_{n+1} classes according to the feature values only. Of course, this set of rules has to be constructed such that as many samples from the sample data set $\mathtt{X}$ as possible are assigned to the correct class.

FS-FOIL can handle this task, no matter whether the goal attribute is Boolean categorical or fuzzy categorical. The typical procedure is as follows: we consider all classes independently by running FS-FOIL for N_{n+1} times with the following N_{n+1} goal predicates ($j = 1, \ldots, N_{n+1}$):

$$\bar{C}_j(\mathbf{x}) = \mathbf{x} \text{ is } L_{n+1,j}.$$

Then the result is a set of N_{n+1} predicate sets $\mathtt{S}_j$ ($j = 1, \ldots, N_{n+1}$) which can be joined into N_{n+1} compound input predicates $\bar{A}_j$ by means of disjunction. The final result is a rule base of the following form (again $j = 1, \ldots, N_{n+1}$):

$$\text{IF } \bar{A}_j(\mathbf{x}) \text{ THEN } \bar{C}_j(\mathbf{x}).$$

Example 1 (*Wine Data Set*). This data set is taken from the UCI repository [5] and contains the results of an analysis of 178 wines grown in the same region in Italy, but coming from three different vineyards. We used 80% of the data for training, i.e. $K = 141$. The analysis determined the quantities of constituents found in each of the three types of wines (attributes *Alcohol*, *Malic Acid*, *Ash*, *Alkalinity of Ash*, *Magnesium*, *Total Phenols*, *Flavonoids*, *Non-Flavonoid Phenols*, *Proanthocyanin*, *OD*280/*OD*315 *of Diluted Wines*, and *Proline*) and some optical properties (attributes *Color Intensity* and *Hue*). All together, there are $n = 13$ input attributes, all of which are numerical. The goal predicate is Boolean categorical with $N_{14} = 3$ different classes/labels corresponding to the three vineyards. Accordingly, we use fuzzy predicates induced by appropriate fuzzy sets on the domains of the numerical input attributes (see Section 3). Although the goal predicate is Boolean categorical, the use of fuzzy predicates

for the input predicates is not meaningless. The reason is that using fuzzy sets allows to model regions of overlapping goal classes easier and in a more natural way than splitting the numerical attributes into Boolean classes.

We applied a simple clustering technique to find a priori configurations of the fuzzy sets (see Section 3 and [17,18]). For each input attribute, we computed four cluster centers and generated trapezoidal fuzzy sets around these centers. Fuzzy sets with insufficient support (i.e. covering too few samples from the data set) were omitted. As a result, six fuzzy sets were created for the *Proline* attribute and seven for the remaining twelve input attributes (see Fig. 1).

We ran FS-FOIL three times (once for each goal class) and, thereby, a compact set of five clauses was created – one or two for each goal class with thresholds of $\text{supp}_{\min} = 0.1$ and $\text{conf}_{\min} = 0.8$. Finally, three rules were obtained (see Table 1). The computations took 4 s on a Linux workstation with a 1 GHz Pentium III® processor.

To evaluate the quality of the results, we applied the rule base to an independent set of 37 samples and compared the result (each sample was crisply assigned to that class for which the highest degree of membership was obtained) with the classes to which the samples actually belong. Table 2 shows the cross validation matrix, that is, the matrix of proportions to which samples belonging to three classes (rows) were assigned to classes 1–3 by the rules we computed using FS-FOIL (columns). Out of the 37 test samples, 34 were correctly classified, two were incorrectly classified, and one was contradictorily assigned to two classes.

We compared the obtained results with those retrieved from a fuzzy variant of Quinlan's ID3 method [40], where we used the same data sets and fuzzy predicates to generate a decision tree. It showed, that the quality of the results obtained from the decision tree was slightly better (only two misclassified samples), however, the obtained rule set was much larger (five leave nodes with an average rule length of 3) compared to the results of FS-FOIL.

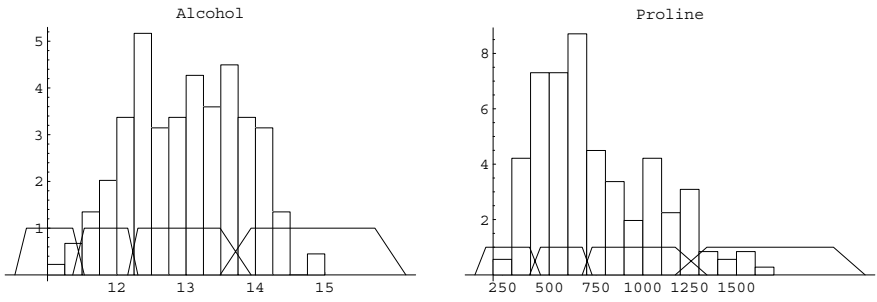

Fig. 1. Fuzzy sets for the *Alcohol* and the *Proline* attributes; the histogram bars visualize the distribution of data samples.

Table 1
Rule set computed for the Wine Data Set

	IF	THEN
Rule 1	(Flavonoids IstAtLeast High AND Proline IsAtLeast High)	Class Is 1
Rule 2	(Alcohol IsAtMost Low) OR (Flavonoids Is High AND Alcohol Is High AND Proline IsAtMost Low)	Class Is 2
Rule 3	(OD280OD315OfDilutedWines IsAtMost Low)	Class Is 3

Table 2
Cross validation matrix of the Wine Data Set with 20% test cases

	Rule 1	Rule 2	Rule 3
Class 1	1.	0.	0.
Class 2	0.071	0.727	0.119
Class 3	0.	0.061	0.859

6. Applications in fuzzy modeling

Since their inception [50,51], the broadest success of fuzzy systems has been witnessed in the control area. The practical scenario in fuzzy control is slightly different from classification, since the goal is to model a (mostly continuous) real-valued function instead of an assignment to abstract classes. The approximation/modeling of real-valued numerical functions by fuzzy systems has emerged as a discipline in its own right and is nowadays commonly called *fuzzy modeling* [2].

FS-FOIL is well-prepared for fuzzy modeling tasks. Assume that we have n numerical input attributes and one numerical output attribute ($m = 1$). If the numerical domains of all $n + 1$ attributes are covered by appropriate families of fuzzy sets (inducing corresponding fuzzy predicates), FS-FOIL can be applied as described above without any restriction: Let us assume that we have N_{n+1} linguistic labels $M_{n+1,j}$ for the goal attribute which are modeled by fuzzy sets $\mu_{M_{n+1,j}}$ ($j = 1, \ldots, N_{n+1}$). Then we can define N_{n+1} goal predicates

$$\bar{C}_i(\mathbf{x}) = \mathbf{x} \text{ is } M_{n+1,j}$$

and run FS-FOIL N_{n+1} times – once for each goal predicate. We obtain a set of N_{n+1} predicate sets S_j which can be joined into N_{n+1} compound input predicates $\bar{A}_j$ by means of disjunction ($j = 1, \ldots, N_{n+1}$). In the same way as for the classification task, the final result is a rule base:

$$\text{IF } \bar{A}_j(\mathbf{x}) \text{ THEN } \bar{C}_j(\mathbf{x}).$$

The only question that remains is how we can interpret this rule base as a real-valued function. In a most straightforward manner, we can simply employ classical *Mamdani inference* [2,16,29,31,32]; we only need to specify three components: (1) a method for "scaling" the fuzzy sets on the right-hand side with the truth values to which the antecedents of the rules are fulfilled (most often cutting or scaling); (2) a method for aggregating the fuzzy sets obtained by different firing rules (most often, maximum or sum); and (3) a defuzzification method to convert the aggregated function into one representative crisp value (e.g. center of gravity).

Due to their simplicity, easy implementation, and computational efficiency, a different type of fuzzy modeling technique has become a de-facto standard through the last few years: *Sugeno fuzzy systems* [2,48]. Such systems work with a different type of fuzzy rules with crisp functions in the consequent part. Translated to the notations of this paper, a Sugeno fuzzy system consists of a set of N_{n+1} rules of the form

$$\text{IF } \bar{A}_j(\mathbf{x}) \text{ THEN } x_{n+1} = f_j(\mathbf{x}),$$

where the functions $f_j : X_1 \times \cdots \times X_{n+1} \to X_{n+1}$ may only depend on the first n variables. Given a sample $\mathbf{x}$, the output of such a fuzzy system is computed as the following weighted sum:

$$x_{n+1} = \frac{\sum_{j=1}^{N_{n+1}} t(\bar{A}_j(\mathbf{x})) \cdot f_j(\mathbf{x})}{\sum_{j=1}^{N_{n+1}} t(\bar{A}_j(\mathbf{x}))}$$

Most often, the functions f_j are constants, i.e. $f_j(\mathbf{x}) = c_j$, or affine linear functions:

$$f_j(\mathbf{x}) = c_{0,r} + \sum_{r=1}^{n} c_{j,r} \cdot x_r.$$

For the latter case, the name *Takagi–Sugeno–Kang* (*TSK*) *fuzzy system* has become common.

FS-FOIL cannot be applied to the induction of Sugeno fuzzy systems in a direct manner. However, it is still possible to convert the result of FS-FOIL into such a system. Assume that we construct a rule base in the same way as described above for the Mamdani systems, i.e. such that we have N_{n+1} goal predicates $\bar{C}_j$ for the $(n+1)$st attribute and N_{n+1} compound input predicates $\bar{A}_j$. Then one possible variant to convert the FS-FOIL rule base into a Sugeno fuzzy system with constant functions f_j is to defuzzify the membership functions associated with the goal predicates, e.g., using center of gravity,

$$c_j = \frac{\int_{X_{n+1}} y \cdot \mu_{M_{n+1,j}}(y)\,\mathrm{d}y}{\int_{X_{n+1}} \mu_{M_{n+1,j}}(y)\,\mathrm{d}y}. \tag{4}$$

A second variant is to compute the values c_j from the data samples as the average of samples weighted by the degrees to which they belong to the goal predicate $\bar{C}_j$. This allows to take the distribution of samples in the individual data set into account:

$$c_j = \frac{\sum_{i=1}^{K} t(\bar{C}_j(\mathbf{x})) \cdot x_{n+1}^i}{\sum_{i=1}^{K} t(\bar{C}_j(\mathbf{x}))} = \frac{\sum_{i=1}^{K} \mu_{M_{n+1,j}}(x_{n+1}^i) \cdot x_{n+1}^i}{\sum_{i=1}^{K} \mu_{M_{n+1,j}}(x_{n+1}^i)}.$$

It is necessary to mention that FS-FOIL mainly aims at compact interpretable descriptions instead of numerical accuracy. Therefore, the methods sketched above are inferior to modern clustering-based fuzzy modeling techniques [2,13,45,47] in terms of the approximation error. However, the interpretability of the resulting rule base is much better.

Example 2 (*A two-dimensional example*). To briefly illustrate the potential of the proposed method for fuzzy modeling, we tried to reconstruct the following function from data ($n = 2$, $X_1 = X_2 = [0, 100]$, $X_3 = [-100, 100]$):

$$f(x_1, x_2) = x_2 \cdot \sin\left(\frac{2\pi x_1}{100}\right).$$

We selected $K = 500$ random samples (x_1^i, x_2^i) from the range $X_1 \times X_2 = [0, 100]^2$. The final data set was constructed as

$$\mathtt{X} = \left\{ \left(x_1^i, x_2^i, f(x_1^i, x_2^i)\right) \mid i = 1, \ldots, 400 \right\}.$$

Six fuzzy sets with bell-shaped membership functions were created for the first input attribute x_1 and two for the second input attribute x_2. The domain X_3 of the goal attribute has been covered by three fuzzy sets with trapezoidal membership functions (see Fig. 2).

Then FS-FOIL was executed to create the rule base shown in Table 3 using thresholds of $\text{supp}_{\min} = 0.01$ and $\text{conf}_{\min} = 0.6$. The computations took 6 s. In order to create the final real-valued function from the rule base, we constructed a Sugeno system by means of defuzzification of the goal fuzzy sets (employing the center of gravity formula, cf. (4)). Fig. 3 shows plots of the original function

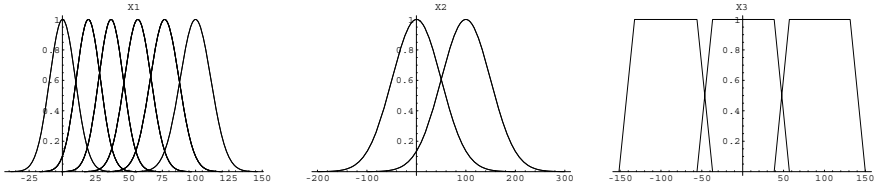

Fig. 2. Fuzzy sets for the function approximation problem.

Table 3
Rule base extracted by FS-FOIL for the function approximation problem

	IF	THEN
Rule 1	(X2 Is High AND X1 Is VeryHigh AND X1 IsAtLeast Low)	X3 Is Low
Rule 2	(X2 Is Low) OR (X1 Is High AND X1 IsAtMost Low AND X1 IsAtMost VeryHigh)	X3 Is Medium
Rule 3	(X2 Is High AND X1 Is VeryLow AND X1 IsAtMost High)	X3 Is High

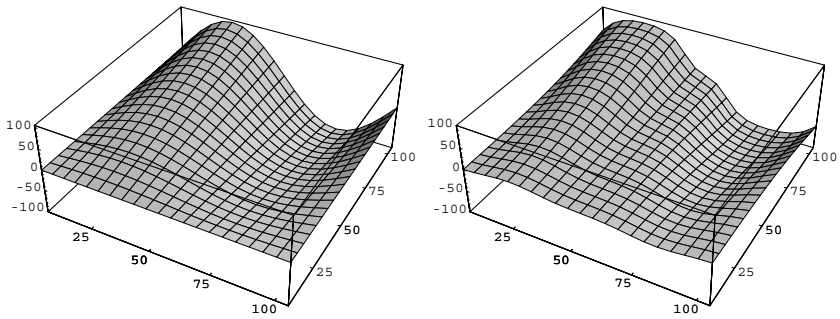

Fig. 3. Test function f (left) and the function defined by a Sugeno fuzzy system constructed by FS-FOIL (right).

f and the function defined by the resulting Sugeno system evaluated for a regular 20×20 grid.

We compared the obtained results with those retrieved from a fuzzy variant of Quinlan's ID3 method, where we used the same data sets and slightly modified fuzzy predicates to generate a decision tree. In this example, the decision tree was not able to create a suitable model, as no appropriate binary splits could be found. The overall prediction error was about 10 times larger then from the model created using the FS-FOIL method.

7. Supervised goes unsupervised: finding interpretable cluster descriptions

As already mentioned, the task of extracting, displaying, and describing previously unknown clusters of similarity in large data sets is another major issue in data mining. While there is a vast number of different clustering algorithms [1,4,25,35], interpretation of the results can be very difficult. While

domain experts may be able to interpret cluster centers and distortion matrices, the non-expert is still excluded from these insights. In order to demonstrate this substantial need, let us consider the following example: a typical application of clustering is market segmentation, i.e. the identification of significant groups of customers from data (e.g. information about customer, transactional data, order history). A salesman who is not an expert in data analysis needs to have a compact and interpretable description of the customer groups (clusters) in order to be able to take this information in his/her daily practice into account.

FS-FOIL can be used to overcome this knowledge representation bottleneck. Assume that we are given a data set consisting of K n-dimensional vectors, i.e. we have n numerical attributes. A clustering algorithm usually computes a finite set of clusters that are most often characterized by cluster centers. More advanced methods additionally use distortion matrices to handle non-spheric clusters, too [4,22,25].

Assume that we have obtained a certain number of clusters from the n-dimensional data set. Let us denote this number with N_{n+1}. Moreover, we can assign a label $L_{n+1,j}$ to each cluster. No matter which clustering method we have employed, it is in any case possible to determine the degree $t_{n+1,j}(\mathbf{x})$ to which a given sample $\mathbf{x}$ belongs to the jth cluster. If the clustering method is crisp, we can consider these functions as Boolean predicates over the set of cluster labels $\{L_{n+1,1},\ldots,L_{n+1,N_{n+1}}\}$. Therefore, we can construct an $n+1$-dimensional data set by adding the cluster memberships as $(n+1)$st attribute, i.e.

$$\mathtt{X} = \left\{(x_1^i,\ldots,x_n^i,x_{n+1}^i)\,|\,i=1,\ldots,K\right\}, \tag{5}$$

where x_{n+1}^i is the label of the cluster to which the ith sample $\mathbf{x}^i$ belongs. If we employ a fuzzy clustering method, we can add an $(n+1)$st fuzzy categorical attribute instead of a Boolean one. More specifically, this means that the construction (5) still applies, but each feature x_{n+1}^i is a fuzzy set on the set of cluster labels $\{L_{n+1,1},\ldots,L_{n+1,N_{n+1}}\}$ that is defined as follows:

$$x_{n+1}^i = \left(t_{n+1,1}(\mathbf{x}^i),\ldots,t_{n+1,N_{n+1}}(\mathbf{x}^i)\right).$$

In order to summarize, this means that we have added the cluster membership as a goal attribute. In case that the clustering method is crisp, this attribute is Boolean categorical. If we use a fuzzy clustering method, this $(n+1)$st attribute is fuzzy categorical. Then FS-FOIL can be employed without any restriction. Applying it to all N_{n+1} goal predicates results in N_{n+1} compound input predicates $\bar{A}_1,\ldots,\bar{A}_{N_{n+1}}$ that describe the regions in the data that belong to the different clusters. Since FS-FOIL employs atomic predicates and fuzzy logical operations to build up the predicates $\bar{A}_j$, these can be understood as close-to-natural-language descriptions of the clusters.

In [17,18], an approach to descriptive data analysis is presented which performs exactly this trick to create descriptions of the clusters. It is worth to mention, however, that the clustering is not performed on the data set as is.

Instead, a *self-organizing map* (*SOM*) [28] is computed first to reduce the amount of data and to eliminate noise, missing values, and outliers. The node values of the self-organizing map are then used as input data X. For more details and examples, we refer to [17,18].

Example 3 (*Image segmentation*). A typical application of clustering in computer vision is image segmentation. Therefore, it seemed interesting to apply the three-stage approach to this problem, too. Moreover, the possibility to describe segments with natural language expressions gives rise to completely new opportunities in image understanding and content-based image retrieval.

The example in Fig. 4 shows a noisy RGB color image with $170 \times 256 = 43520$ pixels. As input attributes, the coordinates (attributes X and Y), the RGB values (*Red*, *Green*, and *Blue*), and HSL features (attributes *Hue*, *Saturation* and *Lightness*) were used. First, the data were mapped onto a SOM with $10 \times 10 = 100$ nodes. Hence, the data set consisted of $K = 100$ samples with $n = 8$ attributes. By applying a modified fuzzy c-means method [17,18], a set of four clusters was generated. After adding the cluster membership as the fuzzy categorical goal attribute as described above, FS-FOIL was executed to compute descriptions of the four clusters. Table 4 shows these descriptions. They can be interpreted as follows: the first cluster corresponds to the blue sky. The second cluster mainly contains the black pants of the two skiers. The snow is contained in the third cluster. Finally, the jackets and faces are contained in

Fig. 4. Original image (left) and its segmentation (right).

Table 4
Cluster descriptions for the image segmentation problem

	Description
Cluster 1	(Blue Is High) OR (Red IsAtMost Low AND Blue IsAtLeast VeryHigh)
Cluster 2	Lightness IsAtMost Dark
Cluster 3	Lightness IsAtLeast Light
Cluster 4	(Hue Is Orange) OR (Hue Is Red) OR (Hue Is Yellow) OR (Hue Is Green AND Lightness Is Normal)

Table 5
Evaluation matrix for the image segmentation problem

	Desc. 1	Desc. 2	Desc. 3	Desc. 4
Cluster 1	0.97	0.	0.	0.
Cluster 2	0.	0.98	0.	0.
Cluster 3	0.07	0.	0.98	0.
Cluster 4	0.02	0.	0.	0.99

the fourth cluster. As easy to see, the descriptions in Table 4 perfectly describe these four areas by means of their dominant colors.

The four segments can also be visualized using a segmentation image: we assign four different gray values to the four clusters and mark each pixel with the gray value that corresponds to the cluster to which the pixel belongs to the highest degree. For the given example, the segmentation image is shown in Fig. 4 on the right-hand side.

Analogously to Example 1, Table 5 displays the evaluation matrix of the cluster descriptions. These values show that the descriptions are accurate and significant.

Finally, let us mention that the computation of the self-organizing map took approximately 6 s, clustering 1 s, and the computation of the descriptions by FS-FOIL took approximately 2 s.

8. Concluding remarks

This paper has presented FS-FOIL, an inductive learning method that is able to construct interpretable fuzzy rules from data. In contrast to other inductive learning methods based on FOIL, FS-FOIL is able to deal with numerical and

fuzzy categorical attributes as well. Three different application scenarios – classification, fuzzy modeling, and descriptive clustering – have demonstrated the wide application potential of the this method.

Future extensions of FS-FOIL will aim in two directions. Firstly, since FS-FOIL can handle virtually any kind of input or goal predicate, it may be beneficial to work with relational predicates that involve two or more attributes as well. That would open completely new opportunities in terms of compact and interpretable descriptions. More concretely, one may think of similarity predicates like x_i *is similar to* x_j that may be modeled by fuzzy equivalence relations [26,29] or ordering-based predicates like x_i *is at least as large as* x_j that may be modeled by fuzzy orderings [7,8]. Secondly, an appropriate combination of FS-FOIL with optimization techniques for finding optimal configurations of fuzzy sets (e.g. RENO [24]) might lead to significant improvements of approximation accuracy in fuzzy modeling applications, while maintaining the superior properties of FS-FOIL in terms of interpretability.

Acknowledgements

This work has been done in the framework of the K*plus* Competence Center Program which is funded by the Austrian Government, the Province of Upper Austria, and the Chamber of Commerce of Upper Austria.

References

[1] M.R. Anderberg, Cluster Analysis for Applications, Academic Press, New York, 1973.
[2] R. Babuška, Fuzzy Modeling for Control, Kluwer Academic Publishers, Boston, 1998.
[3] R.J. Bayardo Jr., R. Agrawal, Mining the most interesting rules, in: S. Chaudhuri, D. Madigan (Eds.), Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 1999, pp. 145–154.
[4] J.C. Bezdek, J. Keller, R. Krishnapuram, N.K. Pal, in: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing, The Handbooks of Fuzzy Sets, vol. 4, Kluwer Academic Publishers, Boston, 1999.
[5] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, University of California, Irvine, Department of Information and Computer Sciences, 1998, http://www.ics.uci.edu/~mlearn/MLRepository.html.
[6] U. Bodenhofer, The construction of ordering-based modifiers, in: G. Brewka, R. Der, S. Gottwald, A. Schierwagen (Eds.), Fuzzy-Neuro Systems '99, Leipziger Universitätsverlag, 1999, pp. 55–62.
[7] U. Bodenhofer, in: A Similarity-Based Generalization of Fuzzy Orderings, Schriftenreihe der Johannes-Kepler-Universität Linz, vol. C 26, Universitätsverlag Rudolf Trauner, 1999.
[8] U. Bodenhofer, A similarity-based generalization of fuzzy orderings preserving the classical axioms, Int. J. Uncertain. Fuzziness Knowledge-Based Systems 8 (5) (2000) 593–610.

[9] U. Bodenhofer, P. Bauer, Towards an axiomatic treatment of "interpretability", in: Proceedings of the 6th International Conference on Soft Computing (IIZUKA2000), Iizuka, October 2000, pp. 334–339.
[10] 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-Verlag, Heidelberg, 2002 (to appear).
[11] I. Bratko, S. Muggleton, Applications of inductive logic programming, Commun. ACM 38 (11) (1995) 65–70.
[12] 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-Verlag, Heidelberg, 2002 (to appear).
[13] S.L. Chiu, Fuzzy model identification based on cluster estimation, J. Intelligent and Fuzzy Systems 2 (3) (1994).
[14] M. De Cock, U. Bodenhofer, E.E. Kerre, Modelling linguistic expressions using fuzzy relations, in: Proceedings of the 6th International Conference on Soft Computing (IIZUKA2000), Iizuka, October 2000, pp. 353–360.
[15] N.R. Draper, H. Smith, Applied Regression Analysis, Wiley, New York, 1981.
[16] D. Driankov, H. Hellendoorn, M. Reinfrank, An Introduction to Fuzzy Control, Springer, Heidelberg, 1993.
[17] M. Drobics, U. Bodenhofer, W. Winiwarter, Mining clusters and corresponding interpretable descriptions – a three-stage approach, Expert Systems 19 (3) (2002) (to appear).
[18] M. Drobics, U. Bodenhofer, W. Winiwarter, E.P. Klement, Data mining using synergies between self-organizing maps and inductive learning of fuzzy rules, in: Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, July 2001, pp. 1780–1785.
[19] D. Dubois, H. Prade, What are fuzzy rules and how to use them, Fuzzy Sets and Systems 84 (1996) 169–185.
[20] D. Dubois, H. Prade, L. Ughetto, Fuzzy logic, control engineering and artificial intelligence, in: H.B. Verbruggen, H.-J. Zimmermann, R. Babuška (Eds.), Fuzzy Algorithms for Control, International Series in Intelligent Technologies, Kluwer Academic Publishers, Boston, 1999, pp. 17–57.
[21] W. Frawley, G. Piatetsky-Shapiro, C. Matheus, Knowledge discovery in databases – an overview, AI Magazine 13 (1992) 57–70.
[22] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: Proceedings of the IEEE International Conference on Decision and Control, San Diego, CA, 1979, pp. 761–766.
[23] P. Hájek, in: Metamathematics of Fuzzy Logic, Trends in Logic, vol. 4, Kluwer Academic Publishers, Dordrecht, 1998.
[24] J. Haslinger, U. Bodenhofer, M. Burger, Data-driven construction of Sugeno controllers: analytical aspects and new numerical methods, in: Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, Vancouver, July 2001, pp. 239–244.
[25] F. Hoeppner, F. Klawonn, R. Kruse, T.A. Runkler, Fuzzy Cluster Analysis – Methods for Image Recognition, Classification, and Data Analysis, Wiley, Chichester, 1999.
[26] F. Klawonn, Fuzzy sets and vague environments, Fuzzy Sets and Systems 66 (1994) 207–221.
[27] E.P. Klement, R. Mesiar, E. Pap, in: Triangular Norms, Trends in Logic, vol. 8, Kluwer Academic Publishers, Dordrecht, 2000.
[28] T. Kohonen, Self-Organizing Maps, second ed., Springer, Berlin, 1997.
[29] R. Kruse, J. Gebhardt, F. Klawonn, Foundations of Fuzzy Systems, Wiley, New York, 1994.
[30] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Comm. 28 (1) (1980) 84–95.

[31] E.H. Mamdani, Application of fuzzy logic to approximate reasoning using linguistic synthesis, IEEE Trans. Comput. 26 (1977) 1182–1191.
[32] E.H. Mamdani, S. Assilian, An experiment in linguistic synthesis with a fuzzy logic controller, Int. J. Man-Mach. Stud. 7 (1975) 1–13.
[33] J.L. McClelland, D.E. Rumelhart (Eds.), Parallel Distributed Processing – Exploration in the Microstructures of Cognition, Volume II: Psychological and Biological Models, MIT Press, Cambridge, MA, 1986.
[34] R.S. Michalski, I. Bratko, M. Kubat, Machine Learning and Data Mining, Wiley, Chichester, 1998.
[35] R.S. Michalski, R.E. Stepp, Clustering, in: S.C. Shapiro (Ed.), Encyclopedia of Artificial Intelligence, Wiley, Chichester, 1992, pp. 168–176.
[36] T.M. Mitchell (Ed.), Machine Learning, McGraw Hill, Dordrecht, 1997.
[37] S. Muggleton, L. De Raedt, Inductive logic programming: Theory and methods, J. Logic Program. 19&20 (1994) 629–680.
[38] V. Novák, I. Perfilieva, J. Močkoř, Mathematical Principles of Fuzzy Logic, Kluwer Academic Publishers, Boston, 1999.
[39] I. Perfilieva, Normal forms for fuzzy logic functions and their approximation ability, Fuzzy Sets and Systems 124 (2001) 371–384.
[40] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.
[41] J.R. Quinlan, Learning logical definitions from relations, Machine Learning 5 (3) (1990) 239–266.
[42] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
[43] J.R. Quinlan, Learning first-order definitions of functions, J. Artificial Intelligence Res. 5 (1996) 139–161.
[44] J.R. Quinlan, R.M. Cameron-Jones, Induction of logic programs: FOIL and related systems, New Generation Computing 13 (1995) 287–312.
[45] J.-S. Roger Jang, ANFIS: Adaptive-network-based fuzzy inference systems, IEEE Trans. Systems Man Cybernet. 23 (3) (1993) 665–685.
[46] D.E. Rumelhart, J.L. McClelland, Parallel Distributed Processing – Exploration in the Microstructures of Cognition, Volume I: Foundations C4.5:, MIT Press, Cambridge, MA, 1986.
[47] M. Setnes, R. Babuška, Fuzzy relational classifier trained by fuzzy clustering, IEEE Trans. Systems Man Cybernet., Part B: Cybernetics 29 (5) (1999) 619–625.
[48] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Trans. Systems Man Cybernet. 15 (1) (1985) 116–132.
[49] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[50] L.A. Zadeh, Toward a theory of fuzzy systems, in: R.E. Kalman, N. De Claris (Eds.), Aspects of Network and System Theory, Holt, Rinehart and Winston, New York, 1970.
[51] L.A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Systems Man Cybernet. 3 (1) (1973) 28–44.
[52] J.M. Zurada, Introduction to Artificial Neural Networks, West Publishing, St. Paul, 1992.