Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 159 (2008) 435–448
FUZZY sets and systems
www.elsevier.com/locate/fss

LFOIL: Linguistic rule induction in the label semantics framework

Zengchang Qin[a,*], Jonathan Lawry[b]

[a] *BISC, EECS Department, University of California, Berkeley, CA 94720, USA*
[b] *A.I. Group, Department of Engineering Mathematics, University of Bristol, UK*

Received 20 February 2006; received in revised form 1 October 2007; accepted 11 October 2007
Available online 26 October 2007

Abstract

Label semantics is a random set framework for modelling with words. In previous work, several machine learning algorithms based on this framework have been proposed and studied. In this paper, we introduce a new linguistic rule induction algorithm based on Quinlan's FOIL algorithm. According to this algorithm, a set of linguistic rules is generated for classification problems. The new model is empirically tested on an artificial toy problem and several benchmark problems from UCI repository. The results show that the new model can generate very compact linguistic rules while maintaining comparable accuracy to other well-known data mining algorithms.

Keywords: LFOIL; Label semantics; FOIL; Linguistic rule; Fuzzy labels

1. Introduction

The use of high-level knowledge representation in data modelling allows for enhanced transparency in the sense that the inferred models can be understood by practitioners who are not necessarily experts in the formal representation framework employed. Rule-based systems inherently tend to be more transparent than other probabilistic models such as neural networks. A set of concise and understandable rules can provide a better understanding of how the classification or prediction is made. There are two general types of algorithms for rule induction, *top-down* and *bottom-up* algorithms. Top-down approaches start from the most general rule and specialize it gradually. Bottom-up methods start from a basic fact given in training database and generalize it. In this paper we will focus on a top-down model for generating linguistic rules based on Quinlan's *first-order inductive learning* (FOIL) algorithm [10].

The FOIL algorithm is based on classical binary logic where typically attributes are assumed to be discrete. Numerical variables are usually discretized by partitioning the numerical domain into a finite number of intervals. However, because of the uncertainty involved in most real-world problems, sharp boundaries between intervals often lead to a loss of robustness and generality [4]. Fuzzy logic has been used to solve the problem of sharp transitions between two intervals. Fuzzy rule induction research has been popular in both fuzzy and machine learning communities as a means to learning robust transparent models. Many algorithms have been proposed including simple fuzzy logic rule induction [2], fuzzy

* Corresponding author. Tel.: +1 510 6434523.
E-mail addresses: zqin@cs.berkeley.edu (Z. Qin), j.lawry@bris.ac.uk (J. Lawry).

0165-0114/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2007.10.008

association rule mining [14] and first-order fuzzy rule induction based on FOIL [4,8]. In this paper, we will focus on an extension to the FOIL algorithm based on a new high-level knowledge representation framework which is referred to as *label semantics* [6]. Label semantics is a framework for linguistic reasoning based on a random set model that uses degrees of appropriateness of a label to describe a given example.

The framework used in this paper is mainly for modelling and building intelligent machine learning and data mining systems. In such systems, fuzzy labels provide a high-level mechanism of discretization and interpretation of modelling uncertainty. In label semantics, labels are assumed to be chosen from a finite predefined set of labels and the set of appropriate labels for a value is defined as a random set-valued function from a population of individuals into the set of subsets of labels [7]. In this paper, a new type of FOIL model, *linguistic FOIL* (LFOIL), is proposed for generating a set of linguistic rules based on label semantics. The experimental results on a number of real-world problems show that LFOIL has comparable accuracy to C4.5 [11] and other linguistic models which also generate compact interpretable rules.

This paper is organized as follows: Section 2 gives a general introduction to label semantics and how it can be used in data modelling. In Section 3, the formal definition of linguistic rules based on label semantics is given. The detailed linguistic rule induction algorithm with pseudo-code is also presented in this section. Section 4 gives the experimental results based on some benchmark problems and the final conclusions and discussions are in Section 5.

2. Label semantics

The fundamental notion underlying label semantics is that when individuals make assertions of the form '$\mathbf{x}$ is θ', they are essentially providing information about what labels are appropriate for the value of the underlying variable x [6]. For a variable x defined on a domain of discourse Ω we identify a finite set of linguistic labels $\mathcal{L} = \{L_1, \ldots, L_n\}$ with which to label the values of x. Then for a specific value $x \in \Omega$ an individual I identifies a subset of $\mathcal{L}$, denoted by D_x^I to stand for the description of x given by I, as the set of words (or fuzzy labels) with which it is appropriate to label x. If we allow I to vary across a population V, then D_x^I will also vary and generate a random set denoted by D_x into the power set of $\mathcal{L}$. The frequency of occurrence of a particular label, say S, for D_x across the population then gives a distribution on D_x referred to as a mass assignment on labels, or more formally:

Definition 1 (*Mass assignment on labels*). For $x \in \Omega$ the label description of x is a random set from V into the power set of $\mathcal{L}$, denoted by D_x, with associated distribution m_x, which is referred to as mass assignment:

$$\forall S \subseteq \mathcal{L}, \quad m_x(S) = P(\{I \in V : D_x^I = S\}).$$

For example, given a set of labels defined on the temperature outside: $\mathcal{L}_{\text{Temp}} = \{low, medium, high\}$. Suppose 3 of 10 people agree that '*medium* is the only appropriate label for the temperature of 15°' and 7 hold the view that 'both *low* and *medium* are appropriate labels'. According to Definition 1, the mass assignment for 15° is $m_{15}(medium) = 0.3$, and $m_{15}(low, medium) = 0.7$, or formally:

$$m_{15} = \{medium\} : 0.3,\ \{low, medium\} : 0.7.$$

More details about the theory of mass assignment can be found in [1].

Consider the previous example, can we know how appropriate for a single label is, say *low*, to describe 15°? In this framework, *appropriateness degrees* are used to evaluate how appropriate a label is for describing a particular value of variable x. Simply, given a particular value α of variable x, the appropriateness degree for describing this value with the label L, as defined by fuzzy set F, is the membership value of α in F. The reason we use the new term 'appropriateness degrees' is partly because it more accurately reflects the underlying semantics and partly to highlight the quite distinct calculus based on this framework [7]. This definition provides a relationship between mass assignments and appropriateness degrees.

Definition 2 (*Appropriateness degrees*).

$$\forall x \in \Omega,\ \forall L \in \mathcal{L}, \quad \mu_L(x) = \sum_{S \subseteq \mathcal{L}: L \in S} m_x(S).$$

In the above example, we have that the appropriateness of *medium* as a description of 15° is $\mu_{\text{medium}}(15) = 0.7 + 0.3 = 1$ and that of *low* is $\mu_{\text{low}}(15) = 0.7$.

2.1. Fuzzy labels for data modelling

Based on the underlying semantics, we can translate a set of numerical data into a set of mass assignments on appropriate labels based on Definition 2 under the following assumptions: consonant mapping, full fuzzy covering and 50% overlapping [9].

It is certainly true that a mass assignment on D_x determines a unique appropriateness degree for μ_L for any $L \in \mathcal{L}$, but generally the converse does not hold. For example, given $\mathcal{L} = \{L_1, L_2, L_3\}$ and $\mu_{L_1} = 0.3$ and $\mu_{L_2} = 1$. We could obtain an infinite family of mass assignments:

$$\{L_1, L_2\} : \alpha,$$

$$\{L_2\} : \beta,$$

$$\{L_2, L_3\} : 0.7 - \beta,$$

$$\{L_1, L_2, L_3\} : 0.3 - \alpha,$$

for any α and β satisfying

$$0 \leqslant \alpha \leqslant 0.3, \quad 0 \leqslant \beta \leqslant 0.7.$$

Hence, the first assumption we make is that the mass assignments m_x are consonant and this allows us to determine m_x uniquely from the appropriateness degrees on labels as follows:

Definition 3 (*Consonant mass assignments on labels*). Let $\{\beta_1, \ldots, \beta_k\} = \{\mu_L(x) | L \in \mathcal{L}, \mu_L(x) > 0\}$ be ordered such that $\beta_t > \beta_{t+1}$ for $t = 1, 2, \ldots, k-1$, then

$$m_x = M_t : \beta_{t-1} - \beta_t \quad \text{for } t = 1, 2, \ldots, k-1,$$

$$M_k : \beta_k, \quad M_0 : 1 - \beta_1,$$

where $M_0 = \emptyset$ and $M_t = \{L \in \mathcal{L} | \mu_L(x) \geqslant \beta_t\}$ for $t = 1, 2 \ldots, k$.

For the previous example, given $\mu_{L_1}(x) = 0.3$ and $\mu_{L_2}(x) = 1$, we can calculate the consonant mass assignments as follows: The appropriateness degrees are ordered as $\{\beta_1, \beta_2\} = \{1, 0.3\}$ and $M_1 = \{L_2\}$, $M_2 = \{L_1, L_2\}$. We can then obtain

$$m_x = \{L_2\} : \beta_1 - \beta_2, \{L_1, L_2\} : \beta_2 = \{L_2\} : 0.7, \{L_1, L_2\} : 0.3.$$

Because the appropriateness degrees are sorted in Definition 3 the resulting mass assignments are 'nested'. Clearly then, there is a unique consonant mapping to mass assignments for a given set of appropriateness degrees. A justification of the consonance assumption can be found in [1,6].

Definition 4 (*Full fuzzy covering*). Given a continuous discourse Ω, $\mathcal{L}$ is called a full fuzzy covering of Ω if

$$\forall x \in \Omega, \ \exists L \in \mathcal{L}, \quad \mu_L(x) = 1.$$

In other words, the full fuzzy covering assumes that, for any element, there always exists a particular label which all the voters agree is appropriate to describe these data, though the voters may have different opinions on other labels. Unless otherwise stated, we will use N_F fuzzy sets with 50% overlap to cover a continuous universe (see Fig. 1), so that the appropriateness degrees satisfy: $\forall x \in \Omega$, $\exists i \in \{1, \ldots, N_F - 1\}$ such that $\mu_{L_i}(x) = \alpha$, $\mu_{L_{i+1}}(x) = \beta$ and $\mu_{L_j}(x) = 0$ for $j < i$ or $j > i + 1$ and where $\max(\alpha, \beta) = 1$. In the case that $\alpha = 1$ according to the full fuzzy covering

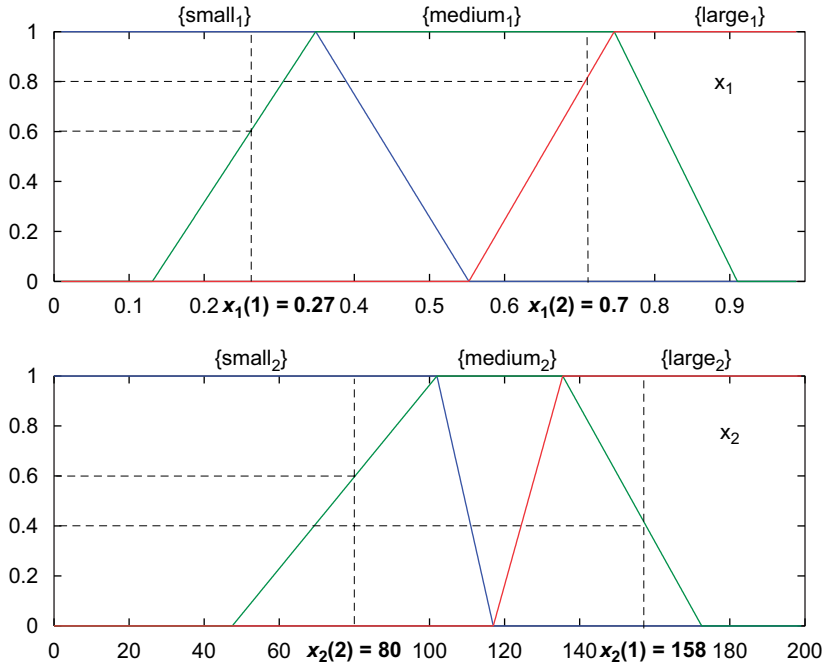

Fig. 1. A full fuzzy covering (discretization) with three fuzzy sets with 50% overlap on two attributes x_1 and x_2, respectively.

assumption, then m_x has the following form:

$$m_x = \{L_i\} : 1 - \beta, \{L_i, L_{i+1}\} : \beta. \tag{1}$$

These assumptions are fully described in [9] and justified in [6]. These assumptions guarantee that there is a unique mapping from appropriate degrees to mass assignments on labels. Based on these assumptions, we can isolate a set of subsets of $\mathcal{L}$ with non-zero mass assignments. These are referred to as *focal sets*:

Definition 5 (*Focal set*). Given a universe Ω for variable x, the focal set of $\mathcal{L}$ is a set of focal elements defined as

$$\mathcal{F} = \{S \subseteq \mathcal{L} | \exists x \in \Omega, m_x(S) > 0\}.$$

Fig. 1 shows the universes of two variables x_1 and x_2 which are fully covered by three fuzzy sets with 50% overlap, respectively. For x_1, the following focal elements occur: $\{small_1\}$, $\{small_1, medium_1\}$, $\{medium_1\}$, $\{medium_1, large_1\}$ and $\{large_1\}$. Since $small_1$ and $large_1$ do not overlap, the set $\{small_1, large_1\}$ cannot occur as a focal element according to Definition 5. We can always find a unique translation from a given data point to a mass assignment on focal elements, as specified by the function μ_L. This is referred to as *linguistic translation* (LT) and is defined as follows: For a particular attribute with an associated focal set, LT is a process of replacing data elements with masses of focal elements of these data. For example, in Fig. 1, $\mu_{\text{small}_1}(x_1(1) = 0.27) = 1$, $\mu_{\text{medium}_1}(0.27) = 0.6$ and $\mu_{\text{large}_1}(0.27) = 0$. They are simply the memberships read from the fuzzy sets. We can then obtain the mass assignment of this data element according to Definition 2 under consonance assumption [9]: $m_{0.27}(small_1) = 0.4$, $m_{0.27}(small_1, medium_1) = 0.6$. Similarly, the linguistic translations for two data:

$$\mathbf{x}_1 = \langle x_1(1) = 0.27\rangle, \langle x_2(1) = 158\rangle,$$

$$\mathbf{x}_2 = \langle x_1(2) = 0.7\rangle, \langle x_2(2) = 80\rangle$$

are illustrated on each attribute independently as follows:

$$\left[\begin{array}{c} x_1 \\ \hline x_1(1)=0.27 \\ x_1(2)=0.7 \end{array}\right] \overset{\text{LT}}{\rightarrow} \left[\begin{array}{ccccc} m_x(\{s_1\}) & m_x(\{s_1,m_1\}) & m_x(\{m_1\}) & m_x(\{m_1,l_1\}) & m_x(\{l_1\}) \\ \hline 0.4 & 0.6 & 0 & 0 & 0 \\ 0 & 0 & 0.2 & 0.8 & 0 \end{array}\right],$$

$$\left[\begin{array}{c} x_2 \\ \hline x_2(1)=158 \\ x_2(2)=80 \end{array}\right] \overset{\text{LT}}{\rightarrow} \left[\begin{array}{ccccc} m_x(\{s_2\}) & m_x(\{s_2,m_2\}) & m_x(\{m_2\}) & m_x(\{m_2,l_2\}) & m_x(\{l_2\}) \\ \hline 0 & 0 & 0 & 0.4 & 0.6 \\ 0.4 & 0.6 & 0 & 0 & 0 \end{array}\right].$$

Therefore, we can obtain the random set expressions of $\mathbf{x}_1$ and $\mathbf{x}_2$ as the following:

$$\mathbf{x}_1 \rightarrow \langle\{s_1\}:0.4, \{s_1,m_1\}:0.6\rangle, \langle\{m_2,l_2\}:0.4, \{l_2\}:0.6\rangle,$$

$$\mathbf{x}_2 \rightarrow \langle\{m_1\}:0.2, \{m_1,l_1\}:0.8\rangle, \langle\{s_2\}:0.4, \{s_2,m_2\}:0.6\rangle.$$

In such a way, all numerical data can be represented as mass assignment based on a predefined fuzzy discretization. In this paper, unless otherwise stated, we will use percentile-based (or equal points) discretization. The idea is to cover approximately the same number of data points for each fuzzy label [9]. The differences between the label semantics and other fuzzy framework are systematically introduced in [7].

2.2. Logical expressions of fuzzy labels

Given a universe of discourse Ω containing a set of objects or instances to be described, it is assumed that all relevant expression can be generated recursively from a finite set of basic labels $\mathcal{L} = \{L_1, \ldots, L_n\}$. Operators for combining expressions are restricted to the standard logical connectives of negation '$\neg$', conjunction '$\wedge$', disjunction '$\vee$' and implication '$\rightarrow$'. Hence, the set of logical expressions of labels can be formally defined as follows:

Definition 6 (*Logical expressions of labels*). The set of logical expressions, *LE*, is defined recursively as follows:

(i) $L_i \in LE$ for $i = 1, \ldots, n$.
(ii) If $\theta, \varphi \in LE$ then $\neg\theta, \theta \wedge \varphi, \theta \vee \varphi, \theta \rightarrow \varphi \in LE$.

Basically, we interpret the main logical connectives as follows: $\neg L$ means that L is not an appropriate label, $L_1 \wedge L_2$ means that both L_1 and L_2 are appropriate labels, $L_1 \vee L_2$ means that either L_1 or L_2 are appropriate labels and $L_1 \rightarrow L_2$ means that L_2 is an appropriate label whenever L_1 is. As well as labels for a single variable, we may want to evaluate the appropriateness degrees of a complex logical expression $\theta \in LE$. Consider the set of logical expressions *LE* obtained by recursive application of the standard logical connectives in $\mathcal{L}$. In order to evaluate the appropriateness degrees of such expressions we must identify what information they provide regarding the appropriateness of labels. In general, for any label expression θ we should be able to identify a maximal set of label sets, $\lambda(\theta)$, that are consistent with θ so that the meaning of θ can be interpreted as the constraint $D_x \in \lambda(\theta)$.

Definition 7 (λ-*Function*). Let θ and φ be expressions generated by recursive application of the connectives $\neg$, $\vee$, $\wedge$ and $\rightarrow$ to the elements of $\mathcal{L}$ (i.e. $\theta, \varphi \in LE$). Then the set of possible label sets defined by a linguistic expression can be determined recursively as follows:

(i) $\lambda(L_i(x)) = \{S \subseteq \mathcal{F} | \{L_i\} \subseteq S\}$.
(ii) $\lambda(\neg\theta) = \overline{\lambda(\theta)}$.
(iii) $\lambda(\theta \wedge \varphi) = \lambda(\theta) \cap \lambda(\varphi)$.
(iv) $\lambda(\theta \vee \varphi) = \lambda(\theta) \cup \lambda(\varphi)$.
(v) $\lambda(\theta \rightarrow \varphi) = \overline{\lambda(\theta)} \cup \lambda(\varphi)$.

It should also be noted that the λ-function provides us with notion of logical equivalence for label expressions:

$$\theta \equiv_{\text{L}} \varphi \iff \lambda(\theta) = \lambda(\varphi).$$

Basically, the λ-function provides a way of mapping from logical expressions of labels (or linguistic rules) to random set descriptions of labels (i.e. focal elements). $\lambda(\theta)$ corresponds to those subsets of $\mathcal{F}$ identified as being possible values of D_x by the expression θ.

Example 1. Given a continuous variable x shown in Fig. 1 and $\mathcal{L}_x = \{small, medium, large\}$ and $\mathcal{F} = \{\{small\}, \{small, medium\}, \{medium\}, \{medium, large\}, \{large\}\}$ suppose we are told that 'x is *not large* but it is *small or medium*'. This constraint can be interpreted as the logical expression

$$\theta_x = \neg large \wedge (small \vee medium).$$

According to Definition 7, the possible label sets of the given logical expression θ_x are calculated as follows:

$$\lambda(\neg large) = \{\{small\}, \{small, medium\}, \{medium\}\},$$

$$\lambda(small) = \{\{small\}, \{small, medium\}\},$$

$$\lambda(medium) = \{\{small, medium\}, \{medium\}, \{medium, large\}\},$$

so that

$$\lambda(\theta_x) = \lambda(\neg large \wedge (small \vee medium)) = \{\{small\}, \{small, medium\}, \{medium\}\} \cap (\{\{small\}, \{small, medium\}\} \cup \{\{small, medium\}, \{medium\}, \{medium, large\}\}) = \{\{small\}, \{small, medium\}, \{medium\}\}.$$

Based on Definition 7, we can evaluate the appropriateness degree of $\theta \in LE$ by aggregating the values of m_x across $\lambda(\theta)$ so that, for example,

$$\mu_{\neg \text{large} \wedge (\text{small} \vee \text{medium})}(x) = m_x(\{small\}) + m_x(\{small, medium\}) + m_x(\{medium\}).$$

This motivates the following general definition of appropriateness measures.

Definition 8 (*Appropriateness measures*). $\forall \theta \in LE, \forall x \in \Omega$ the measure of appropriateness degrees of θ as a description of x is given by

$$\mu_\theta(x) = \sum_{S \in \lambda(\theta)} m_x(S).$$

2.3. Linguistic interpretation of appropriate labels

Based on the inverse of the λ-function (Definition 7), a set of linguistic rules (or logical label expressions) can be obtained from a given set of possible label sets. For example, suppose we are given the possible label set $\{\{small\}, \{small, medium\}, \{medium\}\}$, which does not have an immediately obvious logical interpretation. However, by using the α-function, we can convert this set into a corresponding linguistic expression $(small \vee medium) \wedge \neg large$ or its logical equivalence.

Definition 9 (α-*Function*).

$$\forall F \in \mathcal{F} \quad \text{let } \mathcal{N}(F) = \left(\bigcup_{F' \in \mathcal{F} : F' \supseteq F} F' \right) - F, \tag{2}$$

then

$$\alpha_F = \left(\bigwedge_{L \in F} L \right) \wedge \left(\bigwedge_{L \in \mathcal{N}(F)} \neg L \right). \tag{3}$$

We can then map a set of focal sets to label expressions based on the α-function as follows:

$$\forall R \in \mathcal{F}, \quad \theta_R = \bigvee_{F \in R} \alpha_F \quad \text{where } \lambda(\theta_R) = R. \tag{4}$$

The motivation of this mapping is a follows. Given a focal set $\{s, m\}$ this states that the labels appropriate to describe the attribute are exactly *small* and *medium*. Hence, they include s and m and exclude all other labels that occur in focal sets that are supersets of $\{s, m\}$. Given a set of focal sets $\{\{s, m\}, \{m\}\}$ this provides the information that the set of labels is either $\{s, m\}$ or $\{m\}$ and hence the sentence providing the same information should be the disjunction of the α sentences for both focal sets. The following example illustrates the calculation of the α-function.

Example 2. Let $\mathcal{L} = \{very\ small\ (vs),\ small\ (s),\ medium\ (m),\ large\ (l),\ very\ large\ (vl)\}$ and $\mathcal{F} = \{\{vs, s\}, \{s\}, \{s, m\}, \{m\}, \{m, l\}, \{l\}, \{l, vl\}\}$. For calculating $\alpha_{\{l\}}$, we obtain

$$F' \in \mathcal{F} : F' \supseteq \{l\} = \{\{m, l\}, \{l\}, \{l, vl\}\} = \{m, l, vl\},$$

$$\mathcal{N}(\{l\}) = \left(\bigcup_{F' \in \mathcal{F} : F' \supseteq \{l\}} F' \right) - \{l\} = \{l, vl, m\} - \{l\} = \{vl, m\},$$

$$\alpha_{\{l\}} = \left(\bigwedge_{L \in F} L \right) \wedge \left(\bigwedge_{L \in \mathcal{N}(F)} \neg L \right) = (l) \wedge (\neg m \wedge \neg vl) = \neg m \wedge l \wedge \neg vl.$$

Also we can also obtain

$$\alpha_{\{m,l\}} = m \wedge l, \quad \alpha_{\{l,vl\}} = l \wedge vl.$$

Hence, a set of label sets $\{\{m, l\}, \{l\}, \{l, vl\}\}$ can be represented by a linguistic expression as follows:

$$\begin{aligned} \theta_{\{\{m,l\},\{l\},\{l,vl\}\}} &= \alpha_{\{m,l\}} \vee \alpha_{\{l\}} \vee \alpha_{\{l,vl\}} \\ &= (m \wedge l) \vee (\neg m \wedge l \neg vl) \vee (l \wedge vl) \equiv_{\mathrm{L}} large, \end{aligned}$$

where '$\equiv_{\mathrm{L}}$' represents logical equivalence (see Definition 7).

Basically, α-function provides a way of obtaining logical expressions from a random set description of labels. It is an inverse process of λ-function.

3. Linguistic rule induction

In the previous section, a basic introduction of label semantics is given and how it can be used for data modelling is discussed. In this section, we will describe a linguistic rule induction model based on label semantics. We begin by clarifying the definition of a *linguistic rule*. Based on Definition 6, a linguistic rule is a rule that can be represented as a multi-dimensional logical expressions of fuzzy labels.

Definition 10 (*Multi-dimensional logical expressions of labels*). $MLE^{(n)}$ is the set of all multi-dimensional label expressions that can be generated from the logical label expression $LE_j : j = 1, \ldots, n$ and is defined recursively by

(i) If $\theta \in LE_j$ for $j = 1, \ldots, n$ then $\theta \in MLE^{(n)}$.
(ii) If $\theta, \varphi \in MLE^{(n)}$ then $\neg\theta, \theta \wedge \varphi, \theta \vee \varphi, \theta \rightarrow \varphi \in MLE^{(n)}$.

Any n-dimensional logical expression θ identifies a subset of $2^{\mathcal{L}_1} \times \cdots \times 2^{\mathcal{L}_n}$, denoted by $\lambda^{(n)}(\theta)$ (see Example 3), constraining the cross-product of logical descriptions on each variable: $D_{x_1} \times \cdots \times D_{x_1}$. In such a way the imprecise constraint θ on n variables can be interpreted as the precise constraint $D_{x_1} \times \cdots \times D_{x_1} \in \lambda^{(n)}(\theta)$.

Given a particular data element, how can we evaluate if a linguistic rule is appropriate for describing it? Based on the one-dimensional case, we now extend the concepts of appropriateness degrees to the multi-dimensional case as follows:

Definition 11 (*Multi-dimensional appropriateness degrees*). Given a set of n-dimensional label expressions $MLE^{(n)}$:

$$\forall \theta \in MLE^{(n)}, \quad \forall x_j \in \Omega_j : j = 1, \ldots, n,$$

$$\mu_\theta^n(\mathbf{x}) = \mu_\theta^n(x_1, \ldots, x_n) = \sum_{\langle F_1, \ldots, F_n \rangle \in \lambda^{(n)}(\theta)} (F_1, \ldots, F_n) = \sum_{\langle F_1, \ldots, F_n \rangle \in \lambda^{(n)}(\theta)} \prod_{j=1}^{n} m_{x_j}(F_j).$$

The appropriateness degrees in one dimension are for evaluating a single label for describing a single data element, while in multi-dimensional cases they are for evaluating a linguistic rule for describing a data vector.

Example 3. Consider a modelling problem with two variables x_1 and x_2 for which $\mathcal{L}_1 = \{small(s), medium(med), large\ (lg)\}$ and $\mathcal{L}_2 = \{low\ (lo), moderate\ (mod),\ high\ (h)\}$. Also suppose the focal elements for $\mathcal{L}_1$ and $\mathcal{L}_2$ are

$$\mathcal{F}_1 = \{\{s\}, \{s, med\}, \{med\}, \{med, lg\}, \{lg\}\},$$

$$\mathcal{F}_2 = \{\{lo\}, \{lo, mod\}, \{mod\}, \{mod, h\}, \{h\}\}.$$

According to the multi-dimensional generalization of Definition 7 we have that

$$\lambda^{(2)}((med \wedge \neg s) \wedge \neg lo) = \lambda^{(2)}(med \wedge \neg s) \cap \lambda^{(2)}(\neg lo) = \lambda(med \wedge \neg s) \times \lambda(\neg lo).$$

Now, the set of possible label sets is obtained according to the λ-function:

$$\lambda(med \wedge \neg s) = \{\{med\}, \{med, lg\}\},$$

$$\lambda(\neg lo) = \{\{mod\}, \{mod, h\}, \{h\}\}.$$

Hence, based on Definition 7 we can obtain

$$\lambda^{(2)}((med \wedge \neg s) \wedge \neg lo) = \{\langle\{med\}, \{mod\}\rangle, \langle\{med\}, \{mod, h\}\rangle, \langle\{med\}, \{h\}\rangle, \langle\{med, lg\}, \{mod\}\rangle, \langle\{med, lg\}, \{mod, h\}\rangle, \langle\{med, lg\}, \{h\}\rangle\}.$$

The above calculation on random set interpretation of the given rule based on λ-function is illustrated in Fig. 2: given focal sets $\mathcal{F}_1$ and $\mathcal{F}_2$, we can construct a two-dimensional space where the focal elements have corresponding focal cells. Representation of the multi-dimensional λ-function of the logical expression of the given rule is represented by grey cells.

Given $\mathbf{x} = \langle x_1, x_2 \rangle = \langle x_1 = \{med\} : 0.6, \{med, lg\} : 0.4\rangle, \langle x_2 = \{lo, mod\} : 0.8, \{mod\} : 0.2\rangle$, we obtain

$$\mu_\theta(\mathbf{x}) = (m(\{med\}) + m(\{med, lg\})) \times (m(\{mod\}) + m(\{mod, h\}) + m(\{h\})) = (0.6 + 0.4) \times (0.2 + 0 + 0) = 0.2.$$

And according to Definition 7

$$\mu_{\neg\theta}^n(\mathbf{x}) = 1 - \mu_\theta(\mathbf{x}) = 0.8.$$

In other words, we can say that the linguistic expression θ covers the data $\mathbf{x}$ to degree 0.2 and θ can be considered as a linguistic rule. This interpretation of appropriateness will be highlighted in the next section on rule induction.

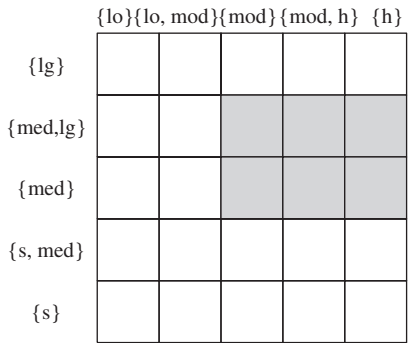

Fig. 2. Representation of the multi-dimensional λ-function (grey cells) of the logical expression $(med \wedge \neg s) \wedge \neg lo$ by showing the focal cells $\mathcal{F}_1 \times \mathcal{F}_2$.

3.1. Information heuristics for LFOIL

In the last section, we have shown how to evaluate the appropriateness of using a linguistic rule to describe a data vector. In this section, a new algorithm for learning a set of linguistic rules is proposed based on the FOIL algorithm [10]. This new algorithm is referred to as LFOIL. Generally, the heuristics for a rule learning model are for assessing the usefulness of a literal as the next component of the rule. The heuristics used for LFOIL are similar but modified from the FOIL algorithm [10] so as to incorporate linguistic expressions based on label semantics. Consider a classification rule of the form

$$R_i = \theta \rightarrow C_k \quad \text{where } \theta \in MLE^{(n)}.$$

Given a data set $\mathcal{D}$ and a particular class C_k, the data belonging to class C_k are referred to as *positive examples* and the rest of them are *negative examples*. For the given rule R_i, the coverage of positive data is evaluated by

$$T_i^+ = \sum_{l \in \mathcal{D}_k} \mu_\theta(\mathbf{x}_l) \tag{5}$$

and the coverage of negative examples is given by

$$T_i^- = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_\theta(\mathbf{x}_l), \tag{6}$$

where $\mathcal{D}_k$ is the subset of the database which consists of the data belonging to class C_k. The information for the original rule R_i can be evaluated by

$$I(R_i) = -\log_2 \left(\frac{T_i^+}{T_i^+ + T_i^-} \right). \tag{7}$$

Suppose we then propose to add another label expression φ to the body of R_i to generate a new rule

$$R_{i+1} = \varphi \wedge \theta \rightarrow C_k,$$

where $\varphi, \theta \in MLE^{(n)}$. By adding the new literal φ, the positive and negative coverage becomes

$$T_{i+1}^+ = \sum_{l \in \mathcal{D}_k} \mu_{\theta \wedge \varphi}(\mathbf{x}_l), \tag{8}$$

$$T_{i+1}^- = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_{\theta \wedge \varphi}(\mathbf{x}_l). \tag{9}$$

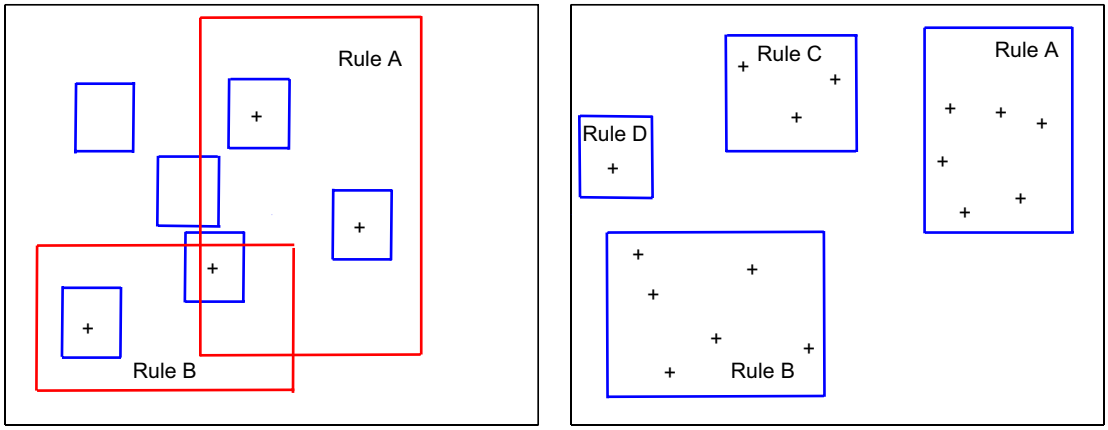

Fig. 3. Illustrations of LFOIL algorithms. Left-hand side figure: each data set has associated masses and it may not be fully covered. Notice that the positive data lie in the overlapping area of Rules A and B, while there is still a small square that cannot be covered due to the limitation of rule base. Right-hand figure: in order to avoid generating too specified rules, we set a threshold to determine the purity in terms of a mixture of positives and negatives covered for a rule.

Therefore, the information becomes

$$I(R_{i+1}) = -\log_2\left(\frac{T_{i+1}^+}{T_{i+1}^+ + T_{i+1}^-}\right). \tag{10}$$

Then we can evaluate the information gain from adding expression φ by

$$G(\varphi) = T_{i+1}^+(I(R_i) - I(R_{i+1})). \tag{11}$$

We can see that the measure of information gain consists of two components. T_{i+1}^+ is the coverage of positive data by the new rule R_{i+1} and $(I(R_i) - I(R_{i+1}))$ is the increase of information. The probability of C_k given a linguistic rule R_i is evaluated by

$$P(C_k|R_i) = \frac{\sum_{l\in\mathcal{D}_k}\mu_\theta(\mathbf{x}_l)}{\sum_{l\in\mathcal{D}}\mu_\theta(\mathbf{x}_l)} = \frac{T_i^+}{T_i^+ + T_i^-}. \tag{12}$$

When $P(C_k|R_{i+1}) > P(C_k|R_i)$ (i.e. by appending a new literal, more positive examples are covered), we can obtain that $(I(R_i) - I(R_{i+1})) > 0$. By choosing a literal φ with maximum G value, we can form the new rule which covers more positive examples and thus increasing the accuracy of the rule.

3.2. Linguistic FOIL

We define a prior knowledge base $KB \subseteq MLE^{(n)}$ and a probability threshold $PT \in [0, 1]$. KB consists of fuzzy label expressions based on labels defined on each attribute. For example, given fuzzy labels $\{small_1\ large_1\}$ to describe Attribute 1 and $\{small_2\ large_2\}$ to describe Attribute 2. A possible knowledge base for the given two variables is $KB = \{small_1, \neg small_1, large_1, \neg large_1, small_2, \neg small_2, large_2, \neg large_2\}$.

The idea for FOIL is as follows: From a general rule, we specify it by adding new literals in order to increase the relative coverage of positive to negatives so that the information gain is reduced. After developing one rule, the positive examples covered by this rule are deleted from the original database. We then need to find a new rule based on this reduced database until all positive examples are covered. In this paper, because of the fuzzy linguistic nature of the expressions employed, typically data will be only partially covered by a given rule. For this reason we need a probability threshold PT as part of the decision process concerning rule coverage.

Fig. 3 qualitatively illustrates the LFOIL algorithm. Each small box represents a data element in the left-hand figure, which for most cases cannot be fully covered. For example, see the left-hand figure of Fig. 3, the data lying in the overlapping area of Rules A and B cannot be fully covered due to the limitations of the rule base. We consequently

need to find the rules with better generalization as well as good accuracy. In the right-hand side figure of Fig. 3: for example, Rule D is a rule that only covers positive examples and here may not be good compared with Rules C and A. Although Rule A covers a negative example, it also covers a large area of positive examples fairly well. Such a rule has a better generalization than Rule D. If we set PT too high, it may possibly result in too many rules similar to Rule D, which may have high training accuracy but low test accuracy because of the lack of generalization. A pseudo-code of LFOIL consists of two parts which are described as follows:

Generating a rule:

- Let rule $R_i = \theta_1 \wedge \cdots \wedge \theta_d \to C_k$ be the rule at step i, we then find the next literal $\theta_{d+1} \in KB - \{\theta_1, \ldots, \theta_d\}$ for which $G(\theta_{d+1})$ is maximal.
- Replace rule R_i with $R_{i+1} = \theta_1 \wedge \cdots \wedge \theta_d \wedge \theta_{d+1} \to C_k$.
- If $P(C_k|\theta_1 \wedge \cdots \wedge \theta_{i+1}) \geqslant PT$ then terminate else repeat.

Generating a rule base: Let $\Delta_i = \{\varphi_1 \to C_k, \ldots, \varphi_t \to C_k\}$ be the rule base at step i where $\varphi \in MLE$. We evaluate the coverage of Δ_i as follows:

$$CV(\Delta_i) = \frac{\sum_{l \in \mathcal{D}_k} \mu_{\varphi_1 \vee \cdots \vee \varphi_t}(\mathbf{x}_l)}{|\mathcal{D}_k|}. \tag{13}$$

We define a coverage function $\delta : \Omega_1 \times \cdots \times \Omega_n \to [0, 1]$ according to

$$\begin{aligned}\delta(\mathbf{x}|\Delta_i) &= \mu_{\neg\Delta_i}(\mathbf{x}) = \mu_{\neg(\varphi_1 \vee \cdots \vee \varphi_t)}(\mathbf{x}) \\ &= 1 - \mu_{(\varphi_1 \vee \cdots \vee \varphi_t)}(\mathbf{x}) = 1 - \sum_{w=1}^{t} \mu_{R_w}(\mathbf{x}),\end{aligned} \tag{14}$$

where $\delta(\mathbf{x}|\Delta_i)$ represents the degree to which $\mathbf{x}$ is *not* covered by a given rule base Δ_i. If CV is less than a predefined coverage threshold $CT \in [0, 1]$:

$$CV(\Delta_i) < CT,$$

then we generate a new rule for class C_k according to the above rule generation algorithm to form a new rule base Δ_{i+1} but where the entropy calculations are amended such that, for a rule $R = \theta \to C_k$,

$$T^+ = \sum_{l \in \mathcal{D}_k} \mu_\theta(\mathbf{x}_l) \times \delta(\mathbf{x}_l|\Delta_i), \tag{15}$$

$$T^- = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_\theta(\mathbf{x}_l). \tag{16}$$

The algorithm terminates when $CV(RB_{i+1}) \geqslant CT$ or $CV(RB_{i+1}) - CV(RB_i) < \varepsilon$ where $\varepsilon \in [0, 1]$ is a very small value, i.e. if there are no improvements in covering positive examples, we will stop the algorithm to avoid an infinite-loop calculation.

3.3. Class probabilities given a rule base

Given a rule base $\Delta_i = \{\varphi_1 \to C_k, \ldots, \varphi_t \to C_k\}$ and unclassified data $\mathbf{x}$, we can estimate the probability of C_k, $P(C_k|\mathbf{x})$, as follows: Firstly, we determine the rule $R_{\max} = \varphi_j \to C_k$ for which $\mu_{\varphi_j}(\mathbf{x})$ is maximal:

$$\varphi_j = \arg \max_{k \in \Delta_i} \mu_{\varphi_k}. \tag{17}$$

Therefore, given the unclassified data $\mathbf{x}$, rule $R_{\max}$ is the most appropriate rule from the rule base we learned. For the rule $R_{\max} \to C_k$ we evaluate two probabilities $p_{\max}$ and $q_{\max}$ where

$$p_{\max} = P(C_k|\varphi_j), \tag{18}$$

$$q_{\max} = P(C_k|\neg\varphi_j). \tag{19}$$

We then use Jeffrey's rule [5] to evaluate the class probability by

$$P(C_k|\mathbf{x}) = p_{\max} \times \mu_{\varphi_j}(\mathbf{x}) + q_{\max} \times (1 - \mu_{\varphi_j}(\mathbf{x})). \tag{20}$$

4. Experimental studies

In this section we first test the LFOIL algorithm with a toy problem described as follows: A figure of eight shape was generated according to the equation

$$x = 2^{(-0.5)}(\sin(2t) - \sin(t))$$

and

$$y = 2^{(-0.5)}(\sin(2t) + \sin(t)),$$

where $t \in [0, 2\pi]$ (see Fig. 4). Points in $[-1.6, 1.6]^2$ are classified as legal if they lie within the 'eight' shape (marked with $\times$) and illegal if they lie outside (marked with points). The database consists of 961 examples generated from a regular grid on $[-1.6, 1.6]^2$ for training, and 961 unseen examples from the same distribution as the test data set.

The following rules are generated by LFOIL algorithm with $PT = 0.7$, $CV = 0.9$ and $\varepsilon = 0.005$:

R_1: x is $\neg$ very small $\wedge$ small $\wedge$ medium $\wedge \neg$ large and y is $\neg$ small $\wedge$ medium $\rightarrow$ legal.
R_2: x is $\neg$ small $\wedge$ medium and y is $\neg$ very small $\wedge$ small $\wedge$ medium $\wedge \neg$ large $\rightarrow$ legal.
R_3: x is medium $\wedge \neg$ large and y is large $\wedge$ very large $\rightarrow$ legal.
R_4: x is large $\wedge$ very large and y is medium $\wedge \neg$ large $\rightarrow$ legal.
R_5: x is very small $\wedge$ small $\wedge \neg$ medium and y is medium $\wedge \neg$ large $\rightarrow$ legal.
R_6: x medium $\wedge \neg$ large and y is very small $\wedge$ small $\wedge \neg$ medium $\rightarrow$ legal.

These rules are symmetric and, as we can see from Fig. 4, the rules capture the legal area very well. For example, the area covered by R_1 is marked by a box shown in Fig. 4.

We also test LFOIL on some benchmark problems taken from UCI machine learning repository [3]. The data descriptions are listed in the left-hand side of Table 1. For each variable, we used 3 fuzzy labels with percentile-based discretization (see Section 2.1). For each data set, we randomly split it into two equal parts with one-half for training and the other half for test. This is referred to as 50–50 split experiments [9]. We ran 10 50–50 experiments on each data set and the average accuracy is listed in Table 1 together with the number of linguistic rules on the right. The results of LFOIL are compared with other two linguistic models: LID3 (linguistic decision tree learning algorithm) [9] and

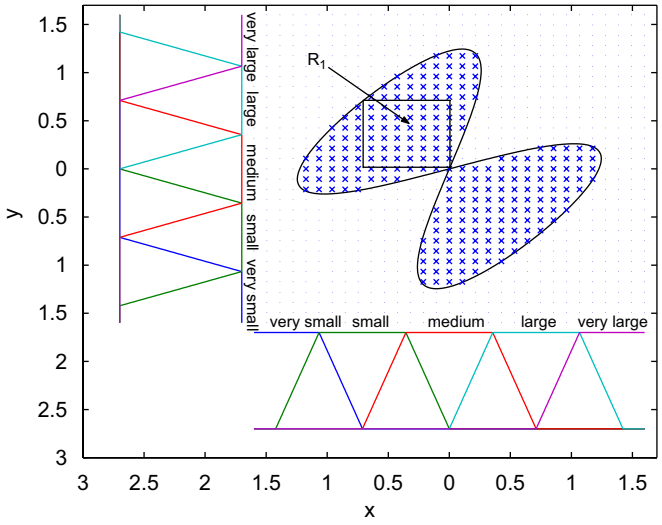

Fig. 4. An illustration of the 'eight' problem, where each attribute is discretized by 5 fuzzy labels: *very small*, *small*, *medium*, *large* and *very large*, respectively.

Table 1
Experimental results on 7 numerical data sets from UCI repository [3]

Data set	Test accuracy (%)				Num. of rules	
	C4.5	FNB	LID3	LFOIL	LID3	LFOIL
BreastCancer	69.16	68.22	73.02	64.10	20	17
Breast-W	94.38	96.74	96.20	95.63	59	8
Heart-c	75.50	76.85	76.87	74.55	48	22
Heart-stalog	75.78	78.34	76.52	71.89	42	17
Hepatitis	76.75	80.13	82.94	75.64	27	8
Liver	65.23	63.35	56.86	54.98	85	3
Pima	72.16	72.29	76.54	71.67	31	4

Table 2
t-Test on experimental results with 90% confidence

Data set	LFOIL vs. C4.5	LFOIL vs. FNB	LFOIL vs. LID3
BreastCancer	–	–	×
Breast-W	–	–	–
Heart-c	–	–	–
Heart-stalog	–	×	×
Hepatitis	–	×	×
Liver	×	×	–
Pima	–	–	×

'–' represents equivalence and '×' represents worse.

fuzzy naive Bayes (FNB) [12]. The parameter settings for LFOIL are $PT = 0.7$, $CV = 0.9$ and $\varepsilon = 0.005$. We also compare LFOIL with C4.5 [1] [11], the results of the t-test with 90% confidence are shown in Table 2.

As we can see from Tables 1 and 2, though the accuracy of LFOIL tends to be marginally worse than LID3 and FNB, it is fairly comparable to C4.5. The most important advantage of LFOIL is that a small number of rules for classification from a large database are extracted. For the LID3 algorithm, given a linguistic decision tree, each path from root to leaves can be interpreted as a linguistic rule. Such a rule can be represented in logical expression by using logical connectives [9]. Therefore, the logical expressions of LID3 rules are as comprehensive as the ones generated by LFOIL. So, it is fair to compare the number of rules generated by these two algorithms.

For example, consider the Pima Indian problem. The database contains the details of 768 females from the population of Pima Indians living near Phoenix, AZ, USA. The diagnostic binary-valued variable investigated is whether the patient shows sign of diabetes according to the World Health Organization criteria. We use 3 fuzzy labels: *low*, *medium* and *high* for each of the eight attributes. We can obtain the following rules that decide whether a patient has the signs of diabetes:

R_1: *Plasma concentration* (Attribute 2) is low $\wedge$ medium and *the number of times pregnant* (Attribute 1) is medium $\wedge$ $\neg$ high.
R_2: *Plasma concentration* is medium and *age* (Attribute 8) is $\neg$ low.
R_3: *Plasma concentration* is low $\wedge$ medium and *the number of times pregnant* is high.
R_4: *Plasma concentration* is $\neg$ medium $\wedge$ high and *diabetes pedigree function* (Attribute 7) is medium.

Although the accuracy for the Pima Indian problem is statistically worse than LID3, the transparency is greatly improved by using only four rules (while the linguistic decision tree has 31 branches) that give a much better understanding of this problem. For some real-world situations, these rules could be much more useful than a black-box model for an expert practitioner.

[1] The results for C4.5 are obtained from a free machine learning toolkit Weka [13] with default settings. The parameter settings for LID3 are according to [9].

5. Conclusions and discussions

In this paper, we introduce a random set-based framework for data mining. In particular, a new algorithm is proposed based on FOIL algorithm and tested on a toy problem and some benchmark problems from UCI repository. The results show that very compact linguistic rules can be learned that reflect the essence of the problem. Although the new algorithm does not necessarily achieve best accuracy among the other models in this framework, it has much better transparency and comparable accuracy to C4.5.

The main contribution of this paper is to describe a method of evaluating linguistic rules through label semantics and to propose a new FOIL-based algorithm for linguistic rule induction. In this algorithm, we use an information-based heuristics to guide the rule construction. This is not the only way of constructing good rules. Another approach is to search exhaustively through the knowledge base *KB*. Assuming that we do not use too many fuzzy labels for discretization, this approach may also be computationally tractable. The rules which cover less positive examples will be discarded according to a predefined threshold. Baldwin and Xie [2] report similar idea for generating simple fuzzy logic (IF–THEN) rules. The complexity of this algorithm is proportional to the number of fuzzy labels we used for discretization and the thresholds. This approach could also be used in linguistic rule learning. In this paper, the parameter setting of LFOIL is from trial-and-error experiments; some further study is necessary to study the influence of parameter settings.

References

[1] J.F. Baldwin, T.P. Martin, B.W. Pilsworth, Fril-fuzzy and Evidential Reasoning in Artificial Intelligence, Wiley, New York, 1995.
[2] J.F. Baldwin, D. Xie, Simple fuzzy logic rules based on fuzzy decision tree for classification and prediction problem, in: Z. Shi, Q. He (Eds.), Intelligent Information Processing II, Springer, Berlin, 2004.
[3] C. Blake, C.J. Merz, UCI machine learning repository, ⟨http://www.ics.uci.edu/∼mlearn/MLRepository.html⟩.
[4] M. Drobics, U. Bodenhofer, E.P. Klement, FS-FOIL: an inductive learning method for extracting interpretable fuzzy descriptions, Internat. J. Approx. Reason. 32 (2003) 131–152.
[5] R.C. Jeffrey, The Logic of Decision, Gordon & Breach, New York, 1965.
[6] J. Lawry, A framework for linguistic modelling, Artificial Intelligence 155 (2004) 1–39.
[7] J. Lawry, Modelling and Reasoning with Vague Concepts, Springer, Berlin, 2006.
[8] H. Prade, G. Richard, M. Serrurier, Enriching relational learning with fuzzy predicates, in: Proceedings of PKDD 2003, Lecture Notes in Artificial Intelligence, vol. 2838, 2003, pp. 399–410.
[9] Z. Qin, J. Lawry, Decision tree learning with fuzzy labels, Inform. Sci. 172 (1–2) (2005) 91–129.
[10] J.R. Quinlan, Learning logical definitions from relations, Mach. Learning 5 (1990) 239–266.
[11] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, 1993.
[12] N.J. Randon, J. Lawry, Classification and query evaluation using modelling with words, Inform. Sci. Special Issue—Comput. with Words: Models and Appl. 176 (2006) 438–464.
[13] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, Los Altos, CA, 1999, ⟨http://www.cs.waikato.ac.nz/∼ml/weka/⟩.
[14] D. Xie, Fuzzy associated rules discovered on effective reduced database algorithm, in: Proceedings of IEEE-FUZZ, Reno, USA, 2005, pp. 779–784.