Fuzzy Label Semantics for Data Mining

Zengchang Qin and Jonathan Lawry

Abstract This chapter gives a tutorial introduction on the label semantics framework for reasoning with uncertainty and several data mining models which are developed based on this framework. Modelling real world problems typically involves processing uncertainty of two distinct types. These are uncertainty arising from a lack of knowledge relating to concepts which, in the sense of classical logic, may be well defined and uncertainty due to inherent vagueness in concepts themselves. Traditionally, these two types of uncertainties are modeled in terms of probability theory and fuzzy set theory, respectively. Zadeh [31] recently argued that all the approaches for uncertainty modelling can be unified into a general theory of uncertainty (GTU). In this chapter, we will introduce an alternate approach for modelling uncertainties by using random set and fuzzy logic. This framework is referred to as label semantics where the labels could be discrete or fuzzy labels. Based on this framework, we proposed several new data mining models. These models not only give comparable accuracy to other well-known data mining models, but also high transparency by which we understand how classifications or predictions have been made instead of a black box.

1 Introduction

In some sense the world is not fuzzy. We can look out and see precisely a leaf falling from an old tree whose shadow lies on a green grassland, there are three people playing and laughing on the grassland, and not far away, there is a car parked on the side of road. Which of them are fuzzy? But this detail which we can see with our eyes is often unwanted precision when it comes to categorizing, classifying and clustering the real world into groups which we can label. We give labels to such objects as people, cars, grassland, trees and leaves, so that we can talk about these objects in terms of their common properties within their group. *Fuzzy Logic* was first proposed by Zadeh [28] as an extension of traditional binary logic. In contrast to a classical set, which has a crisp boundary, the boundary of a fuzzy set is blurred. This smooth transition is characterized by *membership functions* which give fuzzy sets flexibility in modeling linguistic expressions. In early research fuzzy logic was successfully applied in expert systems where the linguistic

M. Nikravesh et al. (eds.), *Forging the New Frontiers: Fuzzy Pioneers II.*

interpretation fuzzy sets allowed for an interface between the human user and a computer system. Because our language is fuzzy, they share the uncertainty and impreciseness: One word has many different meanings and to describe one meaning, we could use many different words. Therefore, we may use fuzzy sets to model our language. This idea provides a good way of bridging the gap between human users and computing systems, and this motivates related research into Computing with Words [29].

Almost all the labels we give to groups of objects are fuzzy. For example, friends, pretty faces, tall trees etc. An object may belong to the set of objects with a certain label, with a certain *membership value*. In traditional set theory, this membership value only has two possible values, 1 and 0, representing the case where the object belongs to or does not belong to the set, respectively. We use a fuzzy term such as 'big' to label a particular group, because they share the property of objects within this group (i.e., they are big). The objects within this group will have different membership values varying from 0 to 1 qualifying the degree to which they satisfy the concept 'big'. An object with membership of 0.8 is more likely to be described as 'big' than an object with membership of 0.4. If we consider this problem in another way. Given an object, label 'big' can be used to describe this object with some appropriateness degrees. Follow this idea, we discuss a new approach based on random set theory to interpret imprecise concepts. This framework, first proposed by Lawry [10] and is referred to as *Label Semantics*, can be regarded as an approach to Modelling with Words[1] [11].

2 Label Semantics

Vague or imprecise concepts are fundamental to natural language. Human beings are constantly using imprecise language to communicate each other. We usually say 'John is tall and strong' but not 'John is exactly 1.85 meters in height and he can lift 100kg weights'. We will focus on developing an understanding of how an intelligent agent can use vague concepts to convey information and meaning as part of a general strategy for practical reasoning and decision making. Such an agent can could be an artificial intelligence program or a human, but the implicit assumption is that their use of vague concepts is governed by some underlying internally consistent strategy or algorithm. We may notice that *labels* are used in natural language to describe what we see, hear and feel. Such labels may have different degrees of vagueness (i.e., when we say Mary is *young* and she is *female*, the label *young* is more vague than

[1] According to Zadeh [30], Modeling with Words is a new research area which emphasis "modelling" rather than "computing", however, the relation between it and Computing with Words is close is likely to become even closer. Both of the research areas are aimed at enlarging the role of natural languages in scientific theories, especially, in knowledge management, decision and control. In this chapter, the framework is mainly used for modelling and building intelligent machine learning and data mining systems. In such systems, we use words or fuzzy labels for modelling uncertainty.

the label *female* because people may have more widely different opinions on being *young* than being *female*. For a particular concept, there could be more than one label that is appropriate for describing this concept, and some labels could be more appropriate than others. Here, we will use a random set framework to interpret these facts. *Label Semantics*, proposed by Lawry [10], is a framework for modelling with linguistic expressions, or labels such as *small*, *medium* and *large*. Such labels are defined by overlapping fuzzy sets which are used to cover the universe of continuous variables. Related to fuzzy sets is the theory of possibility, which can be seen as its numerical counterpart. It is possible to build bridges between probability and fuzzy sets where the latter are viewed as possibility distributions. In particular, we shall interpret possibility measures in the framework of random sets and belief function theory and we shall consider the problem of transforming a possibility distribution into a probability distribution and vice versa.

2.1 Mass Assignment on Fuzzy Labels

The underlying question posed by label semantics is how to use linguistic expressions to label numerical values. For a variable x into a domain of discourse Ω we identify a finite set of linguistic labels $\mathcal{L} = \{L_1, \cdots, L_n\}$ 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 D_x^I to stand for the description of x given by I, as the set of labels with which it is appropriate to label x. If we allow I to vary across a population V with prior distribution P_V, then D_x^I will also vary and generate a random set denoted D_x into the power set of $\mathcal{L}$ denoted by $\mathcal{S}$. We can view the random set D_x as a description of the variable x in terms of the labels in $\mathcal{L}$. 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[2].

More formally,

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

$$\forall S \subseteq \mathcal{L}, \quad m_x(S) = P_V(\{I \in V | D_x^I = S\}) \tag{1}$$

where P_V is the prior distribution of population V. $m_x(S)$ is called the mass associated with a set of labels S and

[2] Since $\mathcal{S}$ is the power set of L, the logical representation $S \in \mathcal{S}$ can be written as $S \subseteq \mathcal{L}$. The latter representation will be used through out this chapter. For example, given $\mathcal{L} = \{L_1, L_2\}$, we can obtain $\mathcal{S} = \{\emptyset, \{L_1\}, \{L_2\}, \{L_1, L_2\}\}$. For every element in $\mathcal{S}$: $S \in \mathcal{S}$, the relation $S \subseteq \mathcal{L}$ will hold.

$$\sum_{S \subseteq \mathcal{L}} m_x(S) = 1 \tag{2}$$

Intuitively mass assignment is a distribution on appropriate label sets and $m_x(S)$ *quantifies the evidence that* S *is the set of appropriate labels for* x*. Based on the data distribution* $p(x)$*, we can calculate the prior distribution of labels by summing up the mass assignment across the database as follows:*

$$pm(S) = p(S) = \int_{\Omega} m_x(S)p(x)dx \Big/ \left(\sum_{S \subseteq \mathcal{L}} \int_{\Omega} m_x(S)p(x) \right) \tag{3}$$

However, the dominator equals to 1 *according to the definition of mass assignment and (2), so that:*

$$pm(S) = \int_{\Omega} m_x(S)p(x)dx \tag{4}$$

And in the discrete case:

$$pm(S) = \sum_{x \in \mathcal{D}} m_x(S)P(x) \tag{5}$$

For example, given a set of labels defined on the temperature outside: $\mathcal{L}_{Temp} = \{low, medium, high\}$. Suppose 3 of 10 people agree that '$medium$ is the only appropriate label for the temperature of 15° and 7 agree 'both low and $medium$ are appropriate labels'. According to def. 1,

$$m_{15}(medium) = 0.3 \; and \; m_{15}(low, medium) = 0.7$$

so that the mass assignment for 15° is $m_{15} = \{medium\} : 0.3, \{low, medium\}: 0.7$. More details about the theory of mass assignment can be found in [1].

2.2 Appropriateness Degrees

Consider the previous example, can we know how appropriate for a single label, 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 labeling this value with the label L, which is 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 [10]. This definition provides a relationship between mass assignments and appropriateness degrees.

Definition 2 (*Appropriateness Degrees*)

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

Consider the previous example, we then can obtain $\mu_{medium}(15) = 0.7 + 0.3 = 1$, $\mu_{low}(15) = 0.7$. Based on the underlying semantics, we can translate a set of numerical data into a set of mass assignments on appropriate labels based on the reverse of definition 2 under the following assumptions: consonance mapping, full fuzzy covering and 50% overlapping [19]. These assumptions are fully described in [19] and justified in [12]. These assumptions guarantee that there is unique mapping from appropriate degrees to mass assignments on labels.

2.3 Linguistic Translation

It is also important to note that, given definitions for the appropriateness degrees on labels, we can isolate a set of subsets of $\mathcal{L}$ with non-zero masses. These are referred to as *focal sets* and the appropriate labels with non-zero masses as *focal elements*, more formally,

Definition 3 (*Focal Set*) *The focal set of $\mathcal{L}$ is a set of focal elements defined as:*

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

Given a particular universe, we can then always find the unique and consistent translation from a given data element to a mass assignment on focal elements, specified by the function $\mu_L : L \in \mathcal{L}$. For example, Fig. 1 shows the universes of two variables x_1 and x_2 which are fully covered by 3 fuzzy sets with 50% overlap, respectively. For x_1, the following focal elements occur:

$$\mathcal{F}_1 = \{\{small_1\}, \{small_1, medium_1\}, \{medium_1\}, \{medium_1, large_1\}, \{large_1\}\}$$

Since $small_1$ and $large_1$ do not overlap, the set $\{small_1, large_1\}$ cannot occur as a focal element according to def. 3. 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* and is defined as follows:

Definition 4 (*Linguistic Translation*) *Suppose we are given a numerical data set $\mathcal{D} = \{\langle x_1(i), \ldots, x_n(i)\rangle | i = 1, \ldots, N\}$ and focal set on attribute j: $\mathcal{F}_j = \{F_j^1, \ldots, F_j^{h_j} | j = 1, \ldots, n\}$, we can obtain the following new data base by applying linguistic translation to $\mathcal{D}$:*

$$\mathcal{D} = \{A_1(i), \ldots, A_n(i) | i = 1, \ldots N\}$$

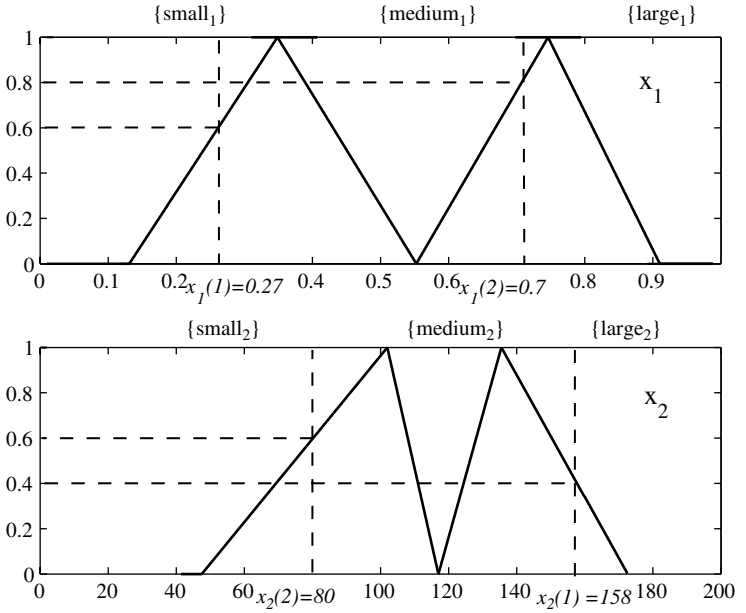

Fig. 1 A full fuzzy covering (discretization) using three fuzzy sets with 50% overlap on two attributes x_1 and x_2, respectively

$$A_j(i) = \{\langle m_{x_j(i)}(F_j^1), \ldots, m_{x_j(i)}(F_j^{h_j})\rangle\}$$

where $m_{x_j(i)}(F_j^r)$ *is the associated mass of focal element* F_j^r *as appropriate labels for data element* $x_j(i)$ *where* $r = 1, \ldots, h_j$ *and* $j = 1, \ldots, n$.

For a particular attribute with an associated focal set, linguistic translation is a process of replacing its data elements with the focal element masses of these data elements. For a variable x, it defines a unique mapping from data element $x(i)$ to a vector of associated masses $\langle m_{x(i)}(F^1), \ldots, m_{x(i)}(F^h)\rangle$.

See fig. 1. $\mu_{small_1}(x_1(1) = 0.27) = 1$, $\mu_{medium_1}(0.27) = 0.6$ and $\mu_{large_1}(0.27) = 0$. They are simply the memberships read from the fuzzy sets. We then can obtain the mass assignment of this data element according to def. 2 under the consonance assumption [19]: $m_{0.27}\ (small_1) = 0.4, m_{0.27}(small_1, medium_1) = 0.6$. Similarly, the linguistic translations for two data:

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

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

are illustrated on each attribute independently as follows:

$$\begin{bmatrix} \underline{x_1} \\ x_1(1) = 0.27 \\ x_1(2) = 0.7 \end{bmatrix} \overset{LT}{\rightarrow} \begin{bmatrix} \underline{m_x(\{s_1\}) \;\; m_x(\{s_1, m_1\}) \;\; m_x(\{m_1\}) \;\; m_x(\{m_1, l_1\}) \;\; m_x(\{l_1\})} \\ 0.4 \quad 0.6 \quad 0 \quad 0 \quad 0 \\ 0 \quad 0 \quad 0.2 \quad 0.8 \quad 0 \end{bmatrix}$$

$$\begin{bmatrix} \underline{x_2} \\ x_2(1) = 158 \\ x_2(2) = 80 \end{bmatrix} \overset{LT}{\rightarrow} \begin{bmatrix} \underline{m_x(\{s_2\}) \;\; m_x(\{s_2, m_2\}) \;\; m_x(\{m_2\}) \;\; m_x(\{m_2, l_2\}) \;\; m_x(\{l_2\})} \\ 0 \quad 0 \quad 0 \quad 0.4 \quad 0.6 \\ 0.4 \quad 0.6 \quad 0 \quad 0 \quad 0 \end{bmatrix}$$

Therefore, we can obtain:

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

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

3 Linguistic Reasoning

As a high-level knowledge representation language for modelling vague concepts, label semantics allows linguistic reasoning. This section introduces the linguistic reasoning mechanism for label semantics framework. Given a universe of discourse Ω containing a set of objects or instances to be described, it is assumed that all relevant expressions 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 5 (*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 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 6 (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) $\lambda(L_i) = \{\underline{S \subseteq \mathcal{F}} | \{L_i\} \subseteq S\}$
(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 '$\equiv_L$' for label expressions

$$\theta \equiv_L \varphi \Longleftrightarrow \lambda(\theta) = \lambda(\varphi)$$

Basically, the λ-function provides a way of transferring 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 expression θ. In this sense the imprecise linguistic restriction 'x is θ' on x corresponds to the strict constraint $D_x \in \lambda(\theta)$ on D_x. Hence, we can view label descriptions as an alternative to linguistic variables as a means of encoding linguistic constraints.

3.1 Appropriateness Measures

Based on definition 6, we can evaluate the appropriateness degree of $\theta \in LE$ is to aggregate the values of m_x across $\lambda(\theta)$. This motivates the following general definition of appropriateness measures.

Definition 7 (*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:*

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

Appropriateness degrees (def. 2) introduced at the beginning of this chapter are only a special case of the appropriateness measures where $\theta = L$ for $L \in \mathcal{L}$.

Example 1. Given a continuous variable x: $\mathcal{L} = \{small, medium, large\}$, $\mathcal{F} = \{\{small\}, \{small, medium\}, \{medium\}, \{medium, large\}, \{large\}\}$. Suppose we are told that "x is **not large** but it is **small or medium**". This constraint can be interpreted as the logical expression

$$\theta = \neg large \wedge (small \vee medium)$$

According to definition 6, the possible label sets of the given logical expression θ 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 we can obtain:

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

If a prior distribution on focal elements of variable x are given as follows:

$$\{small\} : 0.1, \{small, med.\} : 0.3, \{med.\} : 0.1, \{med., large\} : 0.5, \{large\} : 0.0$$

The appropriateness measure for $\theta = \neg large \wedge (small \vee medium)$ is:

$$\begin{aligned}\mu_\theta(x) &= \sum_{S \in \lambda(\theta)} m_x(S) \\ &= m_x(\{small\}) + m_x(\{small, medium\}) + m_x(\{medium\}) \\ &= 0.1 + 0.3 + 0.1 = 0.5\end{aligned}$$

3.2 Linguistic Interpretation of the Sets of Appropriate Labels

Based on the inverse of the λ-function (def. 6), 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 sets $\{\{small\}, \{small, medium\}, \{medium\}\}$, which does not have an immediately obvious 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 8 (α-*function*)

$$\forall F \in \mathcal{F} \quad let \quad \mathcal{N}(F) = \left(\bigcup_{F' \in \mathcal{F}: F' \supseteq F} F' \right) - F \tag{6}$$

$$then \quad \alpha_F = \left(\bigwedge_{L \in F} L \right) \wedge \left(\bigwedge_{L \in \mathcal{N}(F)} \neg L \right) \tag{7}$$

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

$$\forall R \in \mathcal{F} \quad \theta_R = \bigvee_{F \in R} \alpha_F \quad where \quad \lambda(\theta_R) = R \tag{8}$$

The motivation of this mapping is as follows. Given a focal element $\{s, m\}$ (i.e. $\{small, medium\}$) 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 gives 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,

$$\theta_{\{\{m,l\},\{l\},\{l,vl\}\}} = \alpha_{\{m,l\}} \vee \alpha_{\{l\}} \vee \alpha_{\{l,vl\}} =$$

$$(m \wedge l) \vee (\neg\, m \wedge l \neg\, vl) \vee (l \wedge vl) \equiv_L large$$

where '$\equiv_L$' represents logical equivalence (see def. 6).

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

As a framework of reasoning with uncertainty, label semantics aims to model vague or imprecise concepts which can be used as a knowledge representation tool in high-level modelling tasks. We hope to develop models to be defined interms of linguistic expressions we can enhance robustness, accuracy and transparency. Transparent models should allow for a qualitative understanding of the underlying system in addition to giving quantitative predictions of behaviour. Based on label semantics, several new transparent data mining algorithms have been proposed. We found these algorithms have better transparency and comparable accuracy compared to other algorithms. These algorithms will be introduced in details in the following sections.

4 Linguistic Decision Tree

Tree induction learning models have received a great deal of attention over recent years in the fields of machine learning and data mining because of their simplicity and effectiveness. Among them, the ID3 [22] algorithm for decision trees induction has proved to be an effective and popular algorithm for building decision trees from discrete valued data sets. The C4.5 [24] algorithm was proposed as a successor to ID3 in which an entropy based approach to crisp partitioning of continuous universes was adopted. One inherent disadvantage of crisp partitioning is that it tends to make the induced decision trees sensitive to noise. This noise is not only due to the lack of precision or errors in measured features but is often present in the model itself since the available features may not be sufficient to provide a complete model of the system. For each attribute, disjoint classes are separated with clearly defined boundaries. These boundaries are 'critical' since a small change close to these points will probably cause a complete change in classification. Due to the existence of uncertainty and imprecise information in real-world problems, the class boundaries may not be defined clearly. In this case, decision trees may produce high misclassification rates in testing even if they perform well in training. To overcome this problems, many fuzzy decision tree models have been proposed [2, 9, 14, 15].

Linguistic decision tree (LDT) [19] is a tree-structured classification model based on label semantics. The information heuristics used for building the tree are modified from Quinlan's ID3 [22] in accordance with label semantics. Given a database of which each instance is labeled by one of the classes: $\{C_1, \cdots, C_M\}$. A linguistic decision tree with S consisting branches built from this database can be defined as follows:

$$T = \{\langle B_1, P(C_1|B_1), \cdots, P(C_M|B_1)\rangle, \cdots \langle B_S, P(C_1|B_S), \cdots, P(C_M|B_S)\rangle\}$$

where $P(C_k|B)$ is the probability of class C_k given a branch B. A branch B with d nodes (i.e., the length of B is d) is defined as: $B = \langle F_1, \cdots, F_d\rangle$, where $d \leq n$

and $F_j \in \mathcal{F}_j$ is one of the focal elements of attribute j. For example, consider the branch: $\langle\{\{small_1\}, \{medium_2, large_2\}\rangle$, 0.3, 0.7$\rangle$. This means the probability of class C_1 is 0.3 and C_2 is 0.7 given attribute 1 can only be described as *small* and attribute 2 can be described as both *medium* and *large*.

These class probabilities are estimated from a training set $\mathcal{D} = \{\mathrm{x}_1, \cdots, \mathrm{x}_N\}$ where each instance x has n attributes: $\langle x_1, \cdots, x_n\rangle$. We now describe how the relevant branch probabilities for a LDT can be evaluated from a database. The probability of class C_k ($k = 1, \cdots, M$) given B can then be evaluated as follows. First, we consider the probability of a branch B given x:

$$P(B|\mathrm{x}) = \prod_{j=1}^{d} m_{x_j}(F_j) \tag{9}$$

where $m_{x_j}(F_j)$ for $j = 1, \cdots, d$ are mass assignments of single data element x_j. For example, suppose we are given a branch $B = \langle\{small_1\}, \{medium_2, large_2\}\rangle$ and data $\mathrm{x} = \langle 0.27, 158\rangle$ (the linguistic translation of x_1 was given in Sect. 2.3). According to (9):

$$P(B|\mathrm{x}) = m_{x_1}(\{small_1\}) \times m_{x_2}(\{medium_2, large_2\}) = 0.4 \times 0.4 = 0.16$$

The probability of class C_k given B can then be evaluated[3] by:

$$P(C_k|B) = \frac{\sum_{i\in\mathcal{D}_k} P(B|\mathrm{x}_i)}{\sum_{i\in\mathcal{D}} P(B|\mathrm{x}_i)} \tag{10}$$

where $\mathcal{D}_k$ is the subset consisting of instances which belong to class k. According to the Jeffrey's rule [13] the probabilities of class C_k given a LDT with S branches are evaluated as follows:

$$P(C_k|\mathrm{x}) = \sum_{s=1}^{S} P(C_k|B_s)P(B_s|\mathrm{x}) \tag{11}$$

where $P(C_k|B_s)$ and $P(B_s|\mathrm{x})$ are evaluated based on (9) and (10).

[3] In the case where the denominator is equals to 0, which may occur when the training database for the LDT is small, then there is no non-zero linguistic data covered by the branch. In this case, we obtain no information from the database so that equal probabilities are assigned to each class. $P(C_k|B) = \frac{1}{M}$ for $k = 1, \cdots, M$. In the case that a data element appears beyond the range of training data set, we then assign the appropriateness degrees of the minimum or maximum values of the universe to the data element depending on which side of the range it appears.

4.1 Linguistic ID3 Algorithm

Linguistic ID3 (LID3) is the learning algorithm we propose for building the linguistic decision tree based on a given linguistic database. Similar to the ID3 algorithm [22], search is guided by an information based heuristic, but the information measurements of a LDT are modified in accordance with label semantics. The measure of information defined for a branch B and can be viewed as an extension of the entropy measure used in ID3.

Definition 9 (*Branch Entropy*) *The entropy of branch B given a set of classes $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$ is*

$$E(B) = -\sum_{t=1}^{|\mathcal{C}|} P(C_t|B) \log_2 P(C_t|B) \tag{12}$$

Now, given a particular branch B suppose we want to expand it with the attribute x_j. The evaluation of this attribute will be given based on the *Expected Entropy* defined as follows:

Definition 10 (*Expected Entropy*)

$$EE(B, x_j) = \sum_{F_j \in \mathcal{F}_j} E(B \cup F_j) \cdot P(F_j|B) \tag{13}$$

where $B \cup F_j$ represents the new branch obtained by appending the focal element F_j to the end of branch B. The probability of F_j given B can be calculated as follows:

$$P(F_j|B) = \frac{\sum_{i \in \mathcal{D}} P(B \cup F_j|\mathrm{x}_i)}{\sum_{i \in \mathcal{D}} P(B|\mathrm{x}_i)} \tag{14}$$

We can now define the Information Gain (IG) *obtained by expanding branch B with attribute x_j as:*

$$IG(B, x_j) = E(B) - EE(B, x_j) \tag{15}$$

The goal of tree-structured learning models is to make subregions partitioned by branches be less "impure", in terms of the mixture of class labels, than the unpartitioned dataset. For a particular branch, the most suitable free attribute for further expanding (or partitioning), is the one by which the "pureness" is maximally increased with expanding. That corresponds to selecting the attribute with maximum information gain. As with ID3 learning, the most informative attribute will form the root of a linguistic decision tree, and the tree will expand into branches associated with all possible focal elements of this attribute. For each branch, the free attribute with maximum information gain will be the next node, from level to level, until the

tree reaches the maximum specified depth or the maximum class probability reaches the given threshold probability.

4.2 Forward Merging Algorithm

From the empirical studies on UCI data mining repository [4], we showed that LID3 performs at least as well as and often better than three well-known classification algorithms across a range of datasets (see [19]). However, even with only 2 fuzzy sets for discretization, the number of branches increases exponentially with the depth of the tree. Unfortunately, the transparency of the LDT decreases with the increasing number of branches. To help to maintain transparency by generating more compact trees, a forward merging algorithm based on the LDT model is proposed in this section and experimental results are given to support the validity of our approach.

In a full developed linguistic decision tree, if any of two adjacent branches have sufficiently similar class probabilities according to some criteria, so that these two branches may give similar classification results and therefore can then be merged into one branch in order to obtain a more compact tree. We employ a *merging threshold* to determine whether or not two adjacent branches can be merged.

Definition 11 (*Merging Threshold*) *In a linguistic decision tree, if the maximum difference between class probabilities of two adjacent branches B_1 and B_2 is less than or equal to a given merging threshold T_m, then the two branches can be merged into one branch. Formally, if*

$$T_m \geq \max_{c \in \mathcal{C}} |P(c|B_1) - P(c|B_2)| \tag{16}$$

where $\mathcal{C} = \{C_1, \cdots, C_{|\mathcal{C}|}\}$ is the set of classes, then B_1 and B_2 can be merged into one branch MB.

Definition 12 (*Merged Branch*) *A merged branch MB nodes is defined as*

$$MB = \langle \mathcal{M}_{j_1}, \cdots, \mathcal{M}_{j_{|MB|}} \rangle$$

$|MB|$ are the number of nodes for the branch MB where the node is defined as:

$$\mathcal{M}_j = \{F_j^1, \cdots, F_j^{|\mathcal{M}_j|}\}$$

Each node is a set of focal elements such that F_j^i is adjacent to F_j^{i+1} for $i = 1, \cdots, |\mathcal{M}_j| - 1$. If $|\mathcal{M}| > 1$, it is called compound node, which means it is a compound of more than one focal elements because of merging. The associate mass for $\mathcal{M}_j$ is given by

$$m_x(\mathcal{M}_j) = \sum_{i=1}^{|\mathcal{M}_j|} m_x(F_j^i) \tag{17}$$

where w is the number of merged adjacent focal elements for attribute j.

Based on (9), we can obtain:

$$P(C_t|\mathrm{x}) = \prod_{r=1}^{|MB|} m_{x_r}(\mathcal{M}_r) \tag{18}$$

Therefore, based on (10) and (17) we use the following formula to calculate the class probabilities given a merged branch.

$$P(C_t|MB) = \frac{\sum_{i\in\mathcal{D}_t} P(C_t|\mathrm{x}_i)}{\sum_{i\in\mathcal{D}} P(C_t|\mathrm{x}_i)} \tag{19}$$

For example, Fig. 2 shows the change in test accuracy and the number of leaves (or the number of rules interpreted from a LDT) for different T_m on the wis-cancer

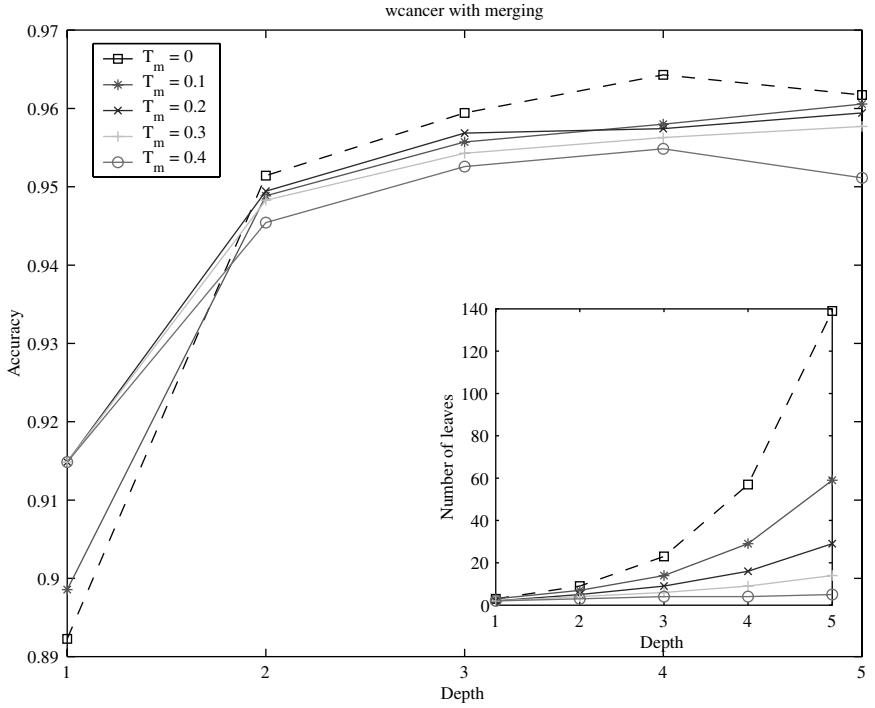

Fig. 2 The change in accuracy and number of leaves as T_m varies on the wis-cancer dataset with $N_F = 2$. While the dot trial $T_m = 0$ is with $N_F = 2$

dataset. It shows that the accuracy is not greatly influenced by merging, but the number of branches is greatly reduced. This is especially true for the curve marked by '+' corresponding to $T_m = 0.3$ where applying forward merging, the best accuracy (at the depth 4) is only reduced by approximately 1%, whereas, the number of branches is reduced by roughly 84%.

4.3 Linguistic Constraints

Here we assume that the linguistic constraints take the form of $\theta = \langle x_1 \text{ is } \theta_1, \ldots, x_n \text{ is } \theta_n \rangle$, where θ_j represents a label expression based on $\mathcal{L}_j : j = 1, \ldots, n$. Consider the vector of linguistic constraint $\vec{\theta} = \langle \theta_1, \cdots, \theta_n \rangle$, where θ_j is the linguistic constraints on attribute j. We can evaluate a probability value for class C_t conditional on this information using a given linguistic decision tree as follows. The mass assignment given a linguistic constraint θ is evaluated by

$$\forall F_j \in \mathcal{F}_j \quad m_{\theta_j}(F_j) = \begin{cases} \frac{pm(F_j)}{\sum_{F_j \in \lambda(\theta_j)} pm(F_j)} & if : F_j \in \lambda(\theta_j) \\ 0 & otherwise \end{cases} \tag{20}$$

where $pm(F_j)$ is the prior mass for focal elements $F_j \in \mathcal{F}_j$ derived from the prior distribution $p(x_j)$ on Ω_j as follows:

$$pm(F_j) = \int_{\Omega_j} m_x(F_j) p(x_j) dx_j \tag{21}$$

Usually, we assume that $p(x_j)$ is the uniform distribution over Ω_j so that

$$pm(F_j) \propto \int_{\Omega_j} m_x(F_j) dx_j \tag{22}$$

For branch B with s nodes, the probability of B given $\vec{\theta}$ is evaluated by

$$P(B|\vec{\theta}) = \prod_{r=1}^{|B|} m_{\theta_{j_r}}(F_{j_r}) \tag{23}$$

and therefore, by Jeffrey's rule [13]

$$P(C_t|\vec{\theta}) = \sum_{v=1}^{|LDT|} P(C_t|B_v) P(B_v|\vec{\theta}) \tag{24}$$

The methodology for classification under linguistic constraints allows us to fuse the background knowledge in linguistic form into classification. This is one of the

advantages of using high-level knowledge representation language models such as label semantics.

4.4 Linguistic Decision Trees for Predictions

Consider a database for prediction $\mathcal{D} = \{\langle x_1(i), \cdots, x_n(i), x_t(i)\rangle \,|i = 1, \cdots, |\mathcal{D}|\}$ where $x_1, \cdots, x_n$ are potential explanatory attributes and x_t is the continuous target attribute. Unless otherwise stated, we use trapezoidal fuzzy sets with 50% overlap to discretized each continuous attribute individually (x_t) universe and assume the focal sets are $\mathcal{F}_1, \cdots, \mathcal{F}_n$ and $\mathcal{F}_t$. For the target attribute x_t: $\mathcal{F}_t = \{F_t^1, \cdots, F_t^{|\mathcal{F}_t|}\}$. For other attributes: x_j: $\mathcal{F}_j = \{F_j^1, \ldots, F_j^{|\mathcal{F}_j|}\}$. The inventive step is, to regard the focal elements for the target attribute as class labels. Hence, the LDT[4] model for prediction has the following form: A linguistic decision tree for prediction is a set of branches with associated probability distribution on the target focal elements of the following form:

$$\begin{aligned} LDT = \{&\langle B_1, P(F_t^1|B_1), \cdots, P(F_t^{|\mathcal{F}_t|}|B_1)\rangle, \cdots, \\ &\langle B_{|LDT|}, P(F_t^1|B_{|LDT|}), \cdots, P(F_t^{|\mathcal{F}_t|})|B_{|LDT|})\rangle\} \end{aligned}$$

where $F_t^1, \cdots, F_t^{|\mathcal{F}_t|}$ are the target focal elements (i.e. the focal elements for the target attribute or the output attribute).

$$P(F_t^j|\mathrm{x}) = \sum_{v=1}^{|LDT|} P(F_t^j|B_v)P(B_v|\mathrm{x}) \tag{25}$$

Given value $\mathrm{x} = \langle x_1, \cdots, x_n\rangle$ we need to estimate the target value $\widehat{x_t}$ (i.e. $\mathrm{x}_i \rightarrow \widehat{x_t}$). This is achieved by initially evaluating the probabilities on target focal elements: $P(F_t^1|\mathrm{x}), \cdots, P(F_t^{|\mathcal{F}_t|}|\mathrm{x})$ as described above. We then take the estimate of x_t, denoted $\widehat{x_t}$, to be the expected value:

$$\widehat{x_t} = \int_{\Omega_t} x_t \; p(x_t|\mathrm{x})\, dx_t \tag{26}$$

where:

$$p(x_t|\mathrm{x}) = \sum_{j=1}^{|\mathcal{F}_t|} p(x_t|F_t^j)\; P(F_t^j|\mathrm{x}) \tag{27}$$

[4] We will use the same name 'LDT' for representing both linguistic decision trees (for classification) and linguistic prediction trees.

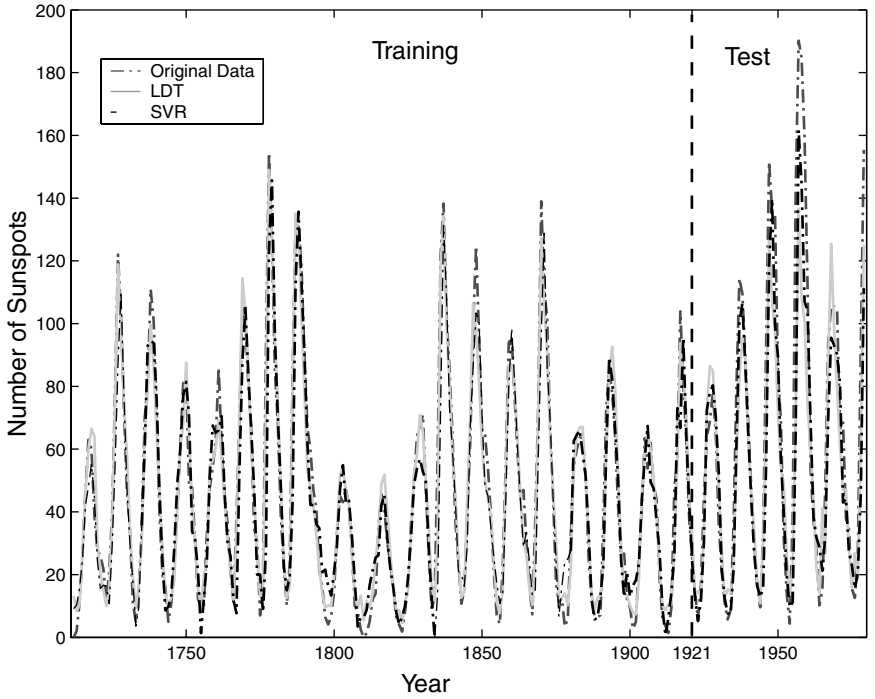

Fig. 3 The prediction results obtained from SVR and LID3 without merging, where the data on the left (1712-1921) are for training and the right (1921-1079) are for test

and

$$p(x_t|F_t^j) = \frac{m_{x_t}(F_t^j)}{\int_{\Omega_t} m_{x_t}(F_t^j)\, dx_t} \tag{28}$$

so that, we can obtain:

$$\widehat{x_t} = \sum_j P(F_t^j|\mathrm{x})\; E(x_t|F_t^j) \tag{29}$$

where:

$$E(x_t|F_t^j) = \int_{\Omega_t} x_t\; p(x_t|F_t^j)\, dx_t = \frac{\int_{\Omega_t} x_t\, m_{x_t}(F_t^j)\, dx_t}{\int_{\Omega_t} m_{x_t}(F_t^j)\, dx_t} \tag{30}$$

We tested our model with a real-world problem taken from the Time Series Data Library [8] and contains data of sunspot numbers between the years 1700-1979. The input attributes are x_{T-12} to x_{T-1} (the data for previous 12 years) and the output (target) attribute is x_T, i.e. one-year-ahead. The experimental results for LID3 and ε-SVR [6] are compared in Fig. 3. We can see the results are quite comparable. More details are available in [20].

5 Bayesian Estimation Tree Based on Label Semantics

Bayesian reasoning provides a probabilistic approach to inference based on the Bayesian theorem. Given a test instance, the learner is asked to predict its class according to the evidence provided by the training data. The classification of unknown example x by Bayesian estimation is on the basis of the following probability,

$$P(C_k|\mathrm{x}) = \frac{P(\mathrm{x}|C_k)P(C_k)}{P(\mathrm{x})} \tag{31}$$

Since the denominator in (31) is invariant across classes, we can consider it as a normalization parameter. So, we obtain:

$$P(C_k|\mathrm{x}) \propto P(\mathrm{x}|C_k)P(C_k) \tag{32}$$

Now suppose we assume for each variable x_j that its outcome is independent of the outcome of all other variables given class C_k. In this case we can obtain the so-called naive Bayes classifier as follows:

$$P(C_k|\mathrm{x}) \propto \prod_{j=1}^{n} P(x_j|C_k)P(C_k) \tag{33}$$

where $P(x_j|C_k)$ is often called the likelihood of the data x_j given C_k. For a qualitative attribute, it can be estimated from corresponding frequencies. For a quantitative attribute, either probability density estimation or discretization can be employed to estimate its probabilities.

5.1 Fuzzy Naive Bayes

In label semantics framework, suppose we are given focal set $\mathcal{F}_j$ for each attribute j. Assuming that attribute x_j is numeric with universe Ω_j, then the likelihood of x_j given C_k can be represented by a density function $p(x_j|C_k)$ determine from the database $\mathcal{D}_k$ and prior density according to Jeffrey's rule [13].

$$p(x_j|C_k) = \sum_{F \in \mathcal{F}_j} p(x_j|F)P(F|C_k) \tag{34}$$

From Bayes theorem, we can obtain:

$$p(x_j|F) = \frac{P(F|x_j)p(x_j)}{P(F)} = \frac{m_{x_j}(F)p(x_j)}{pm(F)} \tag{35}$$

where,

$$pm(F) = \int_{\Omega_j} P(F|x_j)p(x_j)dx_j = \frac{\sum_{\mathrm{x}\in\mathcal{D}} m_{x_j}(F)}{|D|} \tag{36}$$

Substituting 35 in 34 and re-arranging gives

$$p(x_j|C_k) = p(x_j) \sum_{F\in\mathcal{F}_j} m_{x_j}(F)\frac{P(F|C_k)}{pm(F)} \tag{37}$$

where $P(F|C_k)$ can be derived from $\mathcal{D}_k$ according to

$$P(F|C_k) = \frac{\sum_{\mathrm{x}\in\mathcal{D}_k} m_{x_j}(F)}{|\mathcal{D}_k|} \tag{38}$$

This model is called fuzzy Naive Bayes (FNB). If we weaken the independence assumption, we can obtain a fuzzy semi-Naive Bayes (FSNB). More details of FNB and FSNB can be found in [25].

5.2 Hybrid Bayesian Estimation Tree

Based on previous two linguistic models, a hybrid model was proposed in [18]. Given a decision tree T is learnt from a training database $\mathcal{D}$. According to the Bayesian theorem: A data element $\mathrm{x} = \langle x_1, \ldots, x_n\rangle$ can be classified by:

$$P(C_k|\mathrm{x}, T) \propto P(\mathrm{x}|C_k, T)P(C_k|T) \tag{39}$$

We can then divide the attributes into 2 disjoint groups denoted by $\mathrm{x}_T = \{x_1, \cdots, x_m\}$ and $\mathrm{x}_B = \{x_{m+1}, \cdots, x_n\}$, respectively. x_T is the vector of the variables that are contained in the given tree T and the remaining variables are contained in x_B. Assuming conditional independence between x_T and x_B we obtain:

$$P(\mathrm{x}|C_k, T) = P(\mathrm{x}_T|C_k, T)P(\mathrm{x}_B|C_k, T) \tag{40}$$

Because x_B is independent of the given decision tree T and if we assume the variables in x_B are independent of each other given a particular class, we can obtain:

$$P(\mathrm{x}_B|C_k, T) = P(\mathrm{x}_B|C_k) = \prod_{j\in\mathrm{x}_B} P(x_j|C_k) \tag{41}$$

Now consider x_T. According to Bayes theorem,

$$P(\mathrm{x}_T|C_k, T) = \frac{P(C_k|\mathrm{x}_T, T)P(\mathrm{x}_T|T)}{P(C_k|T)} \tag{42}$$

Combining (40), (41) and (42):

$$P(\mathrm{x}|C_k, T) = \frac{P(C_k|\mathrm{x}_T, T)P(\mathrm{x}_T|T)}{P(C_k|T)} \prod_{j \in \mathrm{x}_B} P(x_l|C_k) \tag{43}$$

Combining (39) and (43)

$$P(C_k|\mathrm{x}, T) \propto P(C_k|\mathrm{x}_T, T)P(\mathrm{x}_T|T) \prod_{j \in \mathrm{x}_B} P(x_j|C_k) \tag{44}$$

Further, since $P(\mathrm{x}_T|T)$ is independent from C_k, we have that:

$$P(C_k|\mathrm{x}, T) \propto P(C_k|\mathrm{x}_T, T) \prod_{j \in \mathrm{x}_B} P(x_j|C_k) \tag{45}$$

where $P(x_j|C_k)$ is evaluated according to 37 and $P(C_k|\mathrm{x}_T, T)$ is just the class probabilities evaluated from the decision tree T according to 11.

We tested this new model with a set of UCI [4] data sets. Figure 4 is a simple result. More results are available in [18]. From Figs. 4, we can see that the BLDT model generally performs better at shallow depths than LDT model. However, with the increasing of the tree depth, the performance of the BLDT model remains constant or decreases, while the accuracy curves for LDT increase. The basic idea of using Bayesian estimation given a LDT is to use the LDT as one estimator and the rest of the attributes as other independent estimators. Consider the two extreme cases for 45. If all the attributes are used in building the tree (i.e. $\mathrm{x}_T = \mathrm{x}$), the probability estimations are from the tree only, that is:

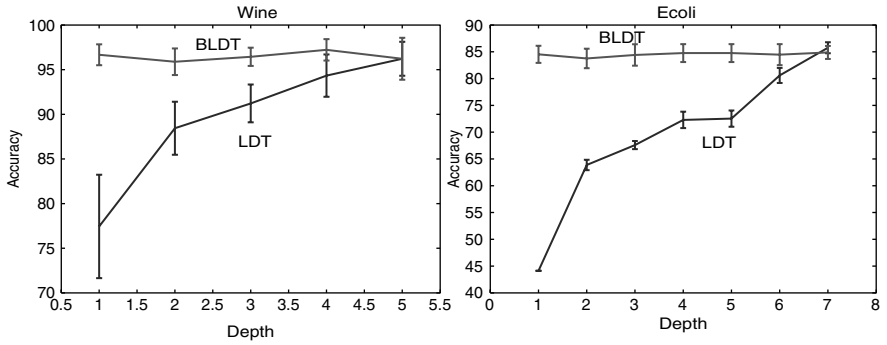

Fig. 4 Results for single LDT with Bayesian estimation: average accuracy with standard deviation on each dataset against the depth of the tree

$$P(C_k|\mathrm{x}, T) \propto P(C_k|\mathrm{x}_T, T)$$

If none of the attributes are used in developing the tree (i.e. $\mathrm{x} = \mathrm{x}_B$), the probability estimation will become:

$$P(C_k|\mathrm{x}, T) \propto \prod_{j \in \mathrm{x}_B} P(x_j|C_k)$$

which is simply a Naive Bayes classifier.

6 Linguistic Rule Induction

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 practioners who are not necessarily experts in the formal representation framework employed. Rule based systems inherently tend to be more transparent than other models such as neural networks. A set of concise understandable rules can provide a better understanding of how the classification or prediction is made. Generally, 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 star 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 [23].

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. 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 [3], fuzzy association rule mining [27] and first-order fuzzy rule induction based on FOIL [5, 16]. In this paper, we will focus on an extension to the FOIL algorithm based on label semantics.

6.1 Linguistic Rules in Label Semantics

In Sect. 2 and 3, 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. Now, we begin by clarifying the definition of a linguistic rule. Based on def. 5, a linguistic rule is a rule can be represented as a multi-dimensional logical expressions of fuzzy labels.

Definition 13 (*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 $2^{\mathcal{L}_1} \times \ldots \times 2^{\mathcal{L}_n}$, denoted $\lambda^{(n)}(\theta)$ (see example 3), constraining the cross product of logical descriptions on each variable: $D_{x_1} \times \ldots \times D_{x_1}$. In such a way the imprecise constraint θ on n variables can be interpret as the precise constraint $D_{x_1} \times \ldots \times D_{x_1} \in \lambda^{(n)}(\theta)$

Given a particular data, how can we evaluated 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 14 (*Multi-dimensional Appropriateness Degrees*) *Given a set of n-dimensional label expressions $MLE^{(n)}$:*

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

$$\mu^n_\theta(\mathrm{x}) = \mu^n_\theta(x_1, \cdots, x_n) = \sum_{\langle F_1, \cdots, F_n \rangle \in \lambda^{(n)}(\theta)} (F_1, \cdots, F_n)$$

$$= \sum_{\langle F_1, \cdots, F_n \rangle \in \lambda^{(n)}(\theta)} \prod_{j=1}^{n} m_{x_j}(F_j)$$

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 6 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 def. 6 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. 5: given focal set $\mathcal{F}_1$ and $\mathcal{F}_2$, we can construct a 2-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 are represented by grey cells.

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

$$\mu_\theta(\mathrm{x}) = (m(\{med\}) + m(\{med, lg\})) \times (m(\{mod\}) + m(\{mod, h\}) + m(\{h\}))$$

$$= (0.6 + 0.4) \times (0.2 + 0 + 0) = 0.2$$

And according to def. 6:

$$\mu^n_{\neg\theta}(\mathrm{x}) = 1 - \mu_\theta(\mathrm{x}) = 0.8$$

In another words, we can say that the linguistic expression θ covers the data x to degree 0.2 and θ can be considered as a linguistic rule. This interpretation of appropriateness is highlighted in next section on rule induction.

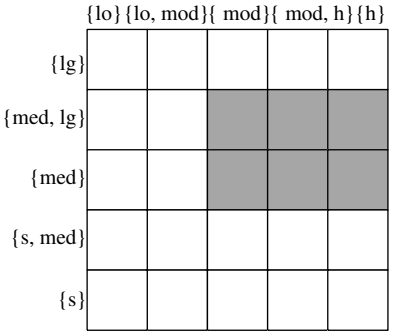

Fig. 5 Representation of the multi-dimensional λ-function of the logical expression $\theta = (med \wedge \neg s) \wedge \neg lo$ showing the focal cells $\mathcal{F}_1 \times \mathcal{F}_2$

6.2 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 [23], it is referred to as *Linguistic FOIL* (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 [23] so as to incorporate linguistic expressions based on labels semantics. Consider a classification rule of the form:

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

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

$$T_i^+ = \sum_{l \in \mathcal{D}_k} \mu_\theta(\mathbf{x}_l) \tag{46}$$

and the coverage of negative examples is given by

$$T_i^- = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_\theta(\mathbf{x}_l) \tag{47}$$

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

$$I(R_i) = -\log_2 \left(\frac{T_i^+}{T_i^+ + T_i^-} \right) \tag{48}$$

Suppose we then propose to 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:

$$T_{i+1}^+ = \sum_{l \in \mathcal{D}_k} \mu_{\theta \wedge \varphi}(\mathbf{x}_l) \tag{49}$$

$$T_{i+1}^- = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_{\theta \wedge \varphi}(\mathbf{x}_l) \tag{50}$$

Therefore, the information becomes,

$$I(R_{i+1}) = -\log_2\left(\frac{T_{i+1}^+}{T_{i+1}^+ + T_{i+1}^-}\right) \tag{51}$$

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

$$G(\varphi) = T_{i+1}^+(I(R_i) - I(R_{i+1})) \tag{52}$$

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:

$$P(C_k|R_i) = \frac{\sum_{l\in\mathcal{D}_k}\mu_\theta(\mathrm{x}_l)}{\sum_{l\in\mathcal{D}}\mu_\theta(\mathrm{x}_l)} = \frac{T_i^+}{T_i^+ + T_i^-} \tag{53}$$

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.

6.3 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 cover more positive and less negative examples according to the heuristics introduced in last section. 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.

A pseudo-code of LFOIL are consists of two parts which are described follows:

Generating a Rule

- Let rule $R_i = \theta_1 \wedge \cdots \wedge \theta_d \rightarrow C_k$ be the rule at step i, we then find the next literal $\theta_{d+1} \in KB - \{\theta_1, \cdots, \theta_d\}$ for which $G(\theta_{d+1})$ is maximal.

- Replace rule R_i with $R_{i+1} = \theta_1 \wedge \cdots \wedge \theta_d \wedge \theta_{d+1} \rightarrow C_k$
- If $P(C_k|\theta_1 \wedge \cdots \wedge \theta_{i+1}) \geq PT$ then terminate else repeat.

Generating a Rule Base

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

$$CV(\Delta_i) = \frac{\sum_{l \in \mathcal{D}_k} \mu_{\varphi_1 \vee \cdots \vee \varphi_t}(\mathrm{x}_l)}{|\mathcal{D}_k|} \tag{54}$$

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

$$\delta(\mathrm{x}|\Delta_i) = \mu_{\neg\Delta_i}(\mathrm{x}) = \mu_{\neg(\varphi_1 \vee \cdots \vee \varphi_t)}(\mathrm{x}) \tag{55}$$

$$= 1 - \mu_{(\varphi_1 \vee \cdots \vee \varphi_t)}(\mathrm{x}) = 1 - \sum_{w=1}^{t} \mu_{R_w}(\mathrm{x})$$

where $\delta(\mathrm{x}|\Delta_i)$ represents the degree to which 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 $R = \theta \rightarrow C_k$,

$$T^+ = \sum_{l \in \mathcal{D}_k} \mu_\theta(\mathrm{x}_l) \times \delta(\mathrm{x}_l|\Delta_i) \tag{56}$$

$$T^- = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_\theta(\mathrm{x}_l) \tag{57}$$

The algorithm terminates when $CV(RB_{i+1}) \geq CT$ or $CV(RB_{i+1}) - CV(RB_i) < \epsilon$ where $\epsilon \in [0, 1]$ 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.

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

$$\varphi_j = \arg\max_{k \in \Delta_i} \mu_{\varphi_k} \tag{58}$$

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

$$p_{max} = P(C_k|\varphi_j) \tag{59}$$

$$q_{max} = P(C_k|\neg\varphi_j) \tag{60}$$

We then use Jeffrey's rule [13] to evaluate the class probability by:

$$P(C_k|\mathrm{x}) = p_{max} \times \mu_{\varphi_j}(\mathrm{x}) + q_{max} \times (1 - \mu_{\varphi_j}(\mathrm{x})) \tag{61}$$

We tested this rule learning algorithms with some toy problems and some real-world problems. Although it does not give us very good accuracy but we obtained some comparable performance to decision tree but with much better transparency. More details are available in [21].

7 Conclusions and Discussions

In this chapter, label semantics, a higher level knowledge representation language, was used for modeling imprecise concepts and building intelligent data mining systems. In particular, several linguistic models have been proposed including: Linguistic Decision Trees (LDT) (for both classification and prediction), Bayesian estimation trees and Linguistic FOIL (LFOIL).

Through previous empirical studies, we have shown that in terms of accuracy the linguistic decision tree model tends to perform significantly better than both C4.5 and Naive Bayes and has equivalent performance to that of the Back-Propagation neural networks. However, it is also the case that this model has much better transparency than other algorithms. Linguistic decision trees are suitable for both classification and prediction. Some benchmark prediction problems have been tested with the LDT model and we found that it has comparable performance to a number of state-of-art prediction algorithms such as support vector regression systems. Furthermore, a methodology for classification with linguistic constraints has been proposed within the label semantics framework.

In order to reduce complexity and enhance transparency, a forward merging algorithm has been proposed to merge the branches which give sufficiently similar probability estimations. With merging, the partitioning of the data space is reconstructed and more appropriate granules can be obtained. Experimental studies show that merging reduces the tree size significantly without a significant loss of accuracy. In order to obtain a better estimation, a new hybrid model combining the LDT model and Fuzzy Naive Bayes has been investigated. The experimental studies show that this hybrid model has comparable performance to LID3 but with

much smaller trees. Finally, a FOIL based rule learning system has been introduced within label semantics framework. In this approach, the appropriateness of using a rule to describe a data element is represented by multi-dimensional appropriateness measures. Based on the FOIL algorithm, we proposed a new linguistic rule induction algorithm according to which we can obtain concise linguistic rules reflecting the underlying nature of the system.

It is widely recognized that most natural concepts have non-sharp boundaries. These concepts are vague or fuzzy, and one will usually only be willing to agree to a certain degree that an object belongs to a concept. Likewise, in machine learning and data mining, the patterns we are interested in are often vague and imprecise. To model this, in this chapter, we have discretized numerical attributes with fuzzy labels by which we can describe real values. Hence, we can use linguistic models to study the underlying relationships hidden in the data. The linguistic models proposed in this chapter have advantages in the following respects[5]:

Interpretability

A primary motivation for the development of linguistic modeling is to provide an interface between numerical scales and a symbolic scale which is usually composed of linguistic terms. Transparency for a model is hard to define. In this chapter, we employ an intuitive way of judging the transparency for decision trees - the number of branches. By forward merging, the number of branches or the size of the tree is reduced, so that we may conclude the transparency of trees is enhanced. We also provide a methodology by which random sets of labels can be interpreted as logical expressions and vice versa.

Robustness

It is often claimed that fuzzy or other 'soft' approaches are more robust than discrete or 'crisp' approaches. In machine learning and data mining problems, the robustness can be considered as the insensitivity of predictive performance of models to small variations in the training data. In decision tree learning, soft boundaries are less sensitive to small changes than sharp boundaries. Hence, the performance of the linguistic models tends to be better than the corresponding discrete models because of the inherent robustness of these soft boundaries.

Information Fusion

One of the distinctive advantages of linguistic models is that they allow for information fusion. In this chapter, we discussed methods for classification with linguistic

[5] H*ü*llermeier [7] argues that these aspects are the potential contributions of fuzzy set theory to machine learning and data mining research.

constraints and linguistic queries based on linguistic decision trees. Other information fusion methods are discussed in [12]. How to efficiently use background knowledge is an important challenge in machine learning. For example, Wang [26] argues that Bayesian learning has limitations in combining the prior knowledge and new evidence. We also need to consider the inconsistency between the background knowledge and new evidence. We believe that it will become a popular research topic in approximate reasoning.

Acknowledgments This research was conduced when the first author was with the AI Group, Department of Engineering Mathematics of Bristol University, UK. The first author thanks Prof Lotfi Zadeh for some insightful comments on this research. He also thanks Drs Masoud Nikravesh, Marcus Thint and Ben Azvine for their interests in this research and support. At last, we thank BT/BISC fellowship for funding the publication of this chapter.

References

1. J.F. Baldwin, T.P. Martin and B.W. Pilsworth (1995) *Fril-Fuzzy and Evidential Reasoning in Artificial Intelligence*. John Wiley & Sons Inc.
2. J. F. Baldwin, J. Lawry and T.P. Martin (1997) Mass assignment fuzzy ID3 with applications. *Proceedings of the Unicom Workshop on Fuzzy Logic: Applications and Future Directions*, London pp. 278–294.
3. J. F. Baldwin and D. Xie (2004), Simple fuzzy logic rules based on fuzzy decision tree for classification and prediction problem, *Intelligent Information Processing II*, Z. Shi and Q. He (Ed.), Springer.
4. C. Blake and C.J. Merz. UCI machine learning repository. *http://www.ics.uci.edu/∼mlearn/MLRepository.html*
5. M. Drobics, U. Bodenhofer and E. P. Klement (2003), FS-FOIL: an inductive learning method for extracting interpretable fuzzy descriptions, *International Journal of Approximate Reasoning*, 32: pp. 131–152.
6. S. R. Gunn (1998), Support vector machines for classification and regression. Technical Report of Dept. of Electronics and Computer Science, University of Southampton. *http://www.isis.ecs.soton.ac.uk/resources/svminfo*
7. E. Hullermeier (2005), Fuzzy methods in machine learning and data mining: status and prospects, to appear in *Fuzzy Sets and Systems*.
8. R. Hyndman and M Akram. Time series Data Library. Monash University. *http://www-personal.buseco.monash.edu.au/∼hyndman/TSDL/index.htm*
9. C. Z. Janikow (1998), Fuzzy decision trees: issues and methods. *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol. 28, No. 1.
10. J. Lawry (2001), Label semantics: A formal framework for modelling with words. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, LNAI 2143: pp. 374–384, Springer-Verlag.
11. J. Lawry, J. Shanahan, and A. Ralescu (2003), *Modelling with Words: Learning, fusion, and reasoning within a formal linguistic representation framework*. LNAI 2873, Springer-Verlag.
12. J. Lawry (2004), A framework for linguistic modelling, *Artificial Intelligence*, 155: pp. 1–39.
13. R. C. Jeffrey (1965), *The Logic of Decision*, Gordon & Breach Inc., New York.
14. C. Olaru and L. Wehenkel (2003), A complete fuzzy decision tree technique. *Fuzzy Sets and Systems*. 138: pp.221–254.
15. Y. Peng, P. A. Flach (2001), Soft discretization to enhance the continuous decision trees. *ECML/PKDD Workshop: IDDM*.

16. H. Prade, G. Richard, and M. Serrurier (2003), Enriching relational learning with fuzzy predicates, *Proceedings of PKDD*, LNAI 2838, pp. 399–410.
17. Z. Qin and J. Lawry (2004), A tree-structured model classification model based on label semantics, *Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU-04)*, pp. 261–268, Perugia, Italy.
18. Z. Qin and J. Lawry (2005), Hybrid Bayesian estimation trees based on label semantics, L. Godo (Ed.), *Proceedings of Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Lecture Notes in Artificial Intelligence 3571, pp. 896–907, Springer.
19. Z. Qin and J. Lawry (2005), Decision tree learning with fuzzy labels, *Information Sciences*, Vol. 172/1–2: pp. 91–129.
20. Z. Qin and J. Lawry (2005), Prediction trees using linguistic modelling, *the Proceedings of International Fuzzy Association World Congress-05*, September 2005, Beijing, China.
21. Z. Qin and J. Lawry (2005), Linguistic rule induction based on a random set semantics, *the Proceedings of International Fuzzy Association World Congress-05*, September 2005, Beijing, China.
22. J. R. Quinlan (1986), Induction of decision trees, *Machine Learning*, Vol 1: pp. 81–106.
23. J. R. Quinlan (1990), Learning logical definitions from relations, *Machine Learning*, 5: 239–266.
24. J. R. Quinlan (1993), *C4.5: Programs for Machine Learning*, San Mateo: Morgan Kaufmann.
25. N. J. Randon and J. Lawry (2006), Classification and query evaluation using modelling with words, *Information Sciences, Special Issue - Computing with Words: Models and Applications*.
26. Pei Wang (2004), The limitation of Bayesianism, *Artificial Intelligence* 158(1): pp. 97–106.
27. D. Xie (2005), Fuzzy associated rules discovered on effective reduced database algorithm, To appear in the *Proceedings of IEEE-FUZZ*, Reno, USA.
28. L. A. Zadeh (1965), Fuzzy sets, *Information and Control*, Vol 8: pp. 338–353.
29. L. A. Zadeh (1996), Fuzzy logic = computing with words, *IEEE Transaction on Fuzzy Systems*. Vol. 4, No. 2: pp. 103–111.
30. L.A. Zadeh (2003), Foreword for modelling with words, *Modelling with Words*, LNAI 2873, Ed., J. Lawry, J. Shanahan, and A.Ralescu, Springer.
31. L.A. Zadeh (2005), Toward a generalized theory of uncertainty (GTU)– an outline, *Information Sciences*, Vol. 172/1–2, pp. 1–40.