Knowledge Discovery in a Framework for Modelling with Words

Zengchang Qin[1] and and Jonathan Lawry[2]

[1] Berkeley Initiative in Soft Computing (BISC), Computer Science Division, EECS Department, University of California, Berkeley, CA 94720, US.
zqin@eecs.berkeley.edu

[2] Artificial Intelligence Group, Department of Engineering Mathematics, University of Bristol, BS8 1TR, UK.
j.lawry@bris.ac.uk

Abstract: The learning of transparent models is an important and neglected area of data mining. The data mining community has tended to focus on algorithm accuracy with little emphasis on the knowledge representation framework. However, the transparency of a model will help practitioners greatly in understanding the trends and idea hidden behind the system. In this chapter, a random set based knowledge representation framework for learning linguistic models is introduced. This framework is referred to as label semantics and a number of data mining algorithms are proposed. In this framework, a vague concept is modelled by a probability distribution over a set of appropriate fuzzy labels which is called as mass assignment. The idea of mass assignment provides a probabilistic approach for modelling uncertainty based on pre-defined fuzzy labels.

1 Introduction

Fuzzy Logic was first proposed by Zadeh [30] 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 and the transition is characterized by *membership functions*. In early research fuzzy logic was successfully applied in control systems and expert systems where the linguistic interpretation fuzzy sets allowed for an interface between the human user and a computer system. Because our language is fuzzy, the concepts represented by language is full of uncertainty and impreciseness. Therefore, fuzzy sets can be used to model language. This idea also motivates related research into Computing with Words [31] and Perception-based Reasoning [32].

Almost all the labels we give to characterize a group of objects are fuzzy. Given a fuzzy set, an object may belong to this set 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. In a fuzzy set, the membership values are continuous real values from 0 to 1. 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 [11].

Modeling with Words is a new research area which emphasis "modelling" rather than "computing". For example, Zadeh's theories on Perception-based Computing [32] and Precisiated Natural Language [34] are the approaches of "computing". However, the relation between it and Computing with Words [31] is close is likely to become even closer [33]. 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. Therefore, the research presented here is considered as a framework for modelling with words.

This chapter is organized as follows: A systematic introduction on label semantics is given in the first section. Based on the framework we introduced, we will give the details of several data mining models based on label semantics: Linguistic Decision Trees in section 3, Label semantics based Bayesian estimation in section 4, and Linguistic Rule Induction in section 5. Finally, we give the summary and discussions in the final section.

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 'Peter is tall and strong' but not 'Peter 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 al-

gorithm. 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 Peter is *young* and he is *male*, the label *young* is more vague than the label *male* because people may have more widely different opinions on being *young* than being *male*. 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.

2.1 Mass Assignment on Fuzzy Labels

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. The underlying question posed by label semantics is how to use linguistic expressions to label numerical values. 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. 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

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

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.

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,

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

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 $\mu_{medium}(15) = 0.7 + 0.3 = 1$, $\mu_{low}(15) = 0.7$. 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 $\mu_L : L \in \mathcal{L}$.

2.3 Linguistic Translation

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 [20]. Consonance assumption implies that voters are agreed with the natural order of fuzzy labels. A voter won't set 'small' and 'large' as appropriate labels without 'medium'. These assumptions are fully described in [20] and justified in [12]. These assumptions guarantee that there

is unique mapping from appropriate degrees to mass assignments on labels. For example, Figure 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. Given a particular data element, the sum of associated mass is 1. This is referred to as *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 described in Algorithm 1.

Algorithm 1: Linguistic translation

input : Given a database $\mathcal{D} = \{\langle x_1(i), \cdots, x_n(i)\rangle | i = 1, \cdots, |\mathcal{D}|\}$ with associated classes $\mathcal{C} = \{C_1, \cdots, C_{|\mathcal{C}|}\}$
output: Linguistic dataset $\mathcal{LD}$

1 **for** $j \leftarrow 1$ **to** n **do**
2 **foreach** x_j **do** : Cover the universe of x_j with N_F trapezoidal fuzzy sets with 50% overlap.
3 **for** $i \leftarrow 1$ **to** $|\mathcal{D}|$ **do**
4 **foreach** *Data element* $x_j(i)$ **do**
5 Read appropriateness degrees for $x_j(i)$ from corresponding fuzzy set.
6 Calculating corresponding mass assignments: $\mathcal{LD}_{i,j} = \langle m_{x(i)}(F_j^1), \cdots, m_{x(i)}(F_j^{h_j})\rangle$ on focal elements from appropriateness degrees.
7 Save dataset $\mathcal{LD}$ where $\mathcal{LD} = \{\mathcal{LD}_{i,j} | i = 1, \cdots, |\mathcal{D}|, j = 1, \cdots, n\}$

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. See figure 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 [20]: $m_{0.27}$ $(small_1)$ $= 0.4$, $m_{0.27}(small_1, medium_1) = 0.6$. 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:

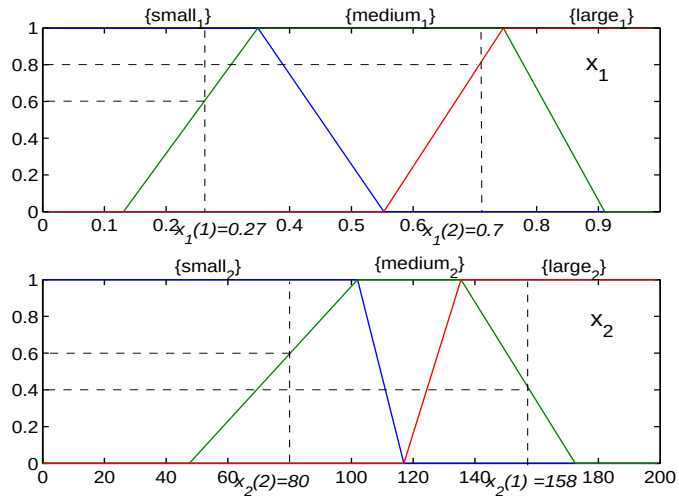

Fig. 1. A full fuzzy covering (discretization) using three fuzzy sets with 50% overlap on two attributes x_1 and x_2, respectively.

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

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

Therefore, we can obtain:

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

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

We may notice that the new mass assignment based data generated by linguistic translation is depending on the way of universe discretization. Different discretizations may result in different data. Since we will use the new data for training data mining models in the following sections. We hope our data could be as discriminate as possible. A few empirical experiments have been done in [20] and the percentile-based (or equal point) discretization is a fairly good method where each fuzzy label covers approximately the same number of data points. In this chapter, unless otherwise stated, we will use this method for discretizing the continuous universe.

2.4 Linguistic Reasoning

As a high-level knowledge representation language for modelling vague concepts, label semantics allows linguistic reasoning. 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 4 (*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 5 (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) $\quad \lambda(L_i) = \{S \subseteq \mathcal{F} | \{L_i\} \subseteq S\}$
(ii) $\quad \lambda(\neg\theta) = \overline{\lambda(\theta)}$
(iii) $\quad \lambda(\theta \wedge \varphi) = \lambda(\theta) \cap \lambda(\varphi)$
(iv) $\quad \lambda(\theta \vee \varphi) = \lambda(\theta) \cup \lambda(\varphi)$
(v) $\quad \lambda(\theta \rightarrow \varphi) = \overline{\lambda(\theta)} \cup \lambda(\varphi)$

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.

2.5 High Level Label Description

In this section, we will consider how to use a high level fuzzy label to describe another fuzzy label. Here the term *high level* does not mean a hieracrhial structure. We will actually consider two set of fuzzy labels which are independently defined on the same universe. If the cardinality of a set of labels $\mathcal{L}$ is denoted by $|\mathcal{L}|$. We then can say $\mathcal{L}_1$ higher level labels of $\mathcal{L}_2$ if $\mathcal{L}_1 < \mathcal{L}_2$. We will acutally consider the methodology of using one set of fuzzy labels to represent the other set of fuzzy labels.

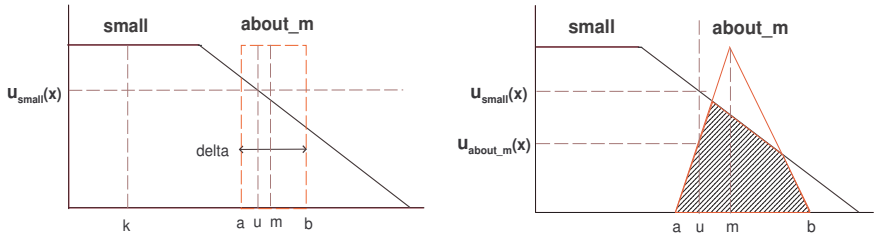

Fig. 2. The appropriateness degree of using *small* to label vague concept *about_m* is defined by the ratio of the area covered by both labels to the area covered by *about_m* only.

For example, a fuzzy concept *about_m* is defined by an interval on [a, b] (see the left-hand side figure of fig. 2), so that the appropriateness degree of using fuzzy label *small* to label *about_m* is:

$$\mu_{small}(about\_m) = \frac{1}{b-a}\int_a^b \mu_{small}(u)du \tag{3}$$

If the vagueness of the concept *about_m* depends on the interval denoted by δ where the length of the interval $|\delta| = b - a$. We then can obtain:

$$\mu_{small}(about\_m) = \frac{1}{|\delta|}\int_{u\in\delta} \mu_{small}(u)du \tag{4}$$

If *about_m* is defined by other fuzzy labels rather than an interval, for example, a triangular fuzzy set (e.g., the right-hand side figure of fig. 2). How can we define the appropriateness degrees?

We begin by considering a data element $x \in [a, b]$, the function $\mu_{about\_m}(x)$ represents the degree of x belonging to the fuzzy label F. Function $\mu_{small}(x)$ defines the appropriateness degrees of using label *small* to describe x [3]. We essentially hope to obtain the appropriateness degrees of using *small* to label *about_m*. We then consider the each elements belonging to *about_m*. If $\mu_{about\_m}(x) = 1$, which means x is absolutely belonging to *about_m*, then the appropriateness degree is just $\mu_{small}(x)$. However, if $\mu_{about\_m} < \mu_{small}(x)$, we can only say it is belonging to *about_m* in certain degrees. Logically, fuzzy operation AND is used, and in practical calculation, the $\min(\cdot)$ function is employed. The appropriateness is then defined by:

$$\mu_{small}(about\_m) = \frac{\int_{u \in \delta} \min(\mu_{small}(u), \mu_{about\_m}(u))du}{\int_{u' \in \delta} \mu_{about\_m}(u')du'} \tag{5}$$

where function $\min(x, y)$ returns the minimum value between x and y. Equation 4 is a special case of equation 5 where the following equations always hold:

$$\mu_{small}(u) = \min(\mu_{small}(u), \mu_{about\_m}(u))$$

$$|\delta| = \int_{u \in \delta} \mu_{about\_m}(u)du$$

Definition 6 *Given a vague concept (or a fuzzy label) F and a set of labels $\mathcal{L} = \{L_1, \ldots, L_m\}$ defined on a continuous universe Ω. The appropriateness degrees of using label L ($L \in \mathcal{L}$) to describe F is:*

$$\mu_L(F) = \frac{\int_{u \in \delta} \min(\mu_L(u), \mu_F(u))du}{\int_{u' \in \delta} \mu_F(u')du'} \tag{6}$$

where δ is the universe covered by fuzzy label F.

Given appropriateness degrees, the mass assignment can be obtained from the appropriateness degrees by the consonance assumption. Equation 5 is a general form for all kinds of fuzzy sets which are not limited to an interval or a triangular fuzzy sets.

3 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 [24] 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 [26] algorithm was proposed as a successor to ID3 in which an entropy based approach

[3] Here we interpret $\mu(\cdot)$ in different manners: membership function and appropriateness degrees, though they are mathematically the same.

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, 15, 16].

Linguistic decision tree (LDT) [20] is a tree-structured classification model based on label semantics. The information heuristics used for building the tree are modified from Quinlan's ID3 [24] 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: $\langle\langle\{small_1\}, \{medium_2, large_2\}\rangle, 0.3, 0.7\rangle$. 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 $\mathcal{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ where each instance $\mathbf{x}$ 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 $\mathbf{x}$:

$$P(B|\mathbf{x}) = \prod_{j=1}^{d} m_{x_j}(F_j) \tag{7}$$

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 $\mathbf{x} = \langle 0.27, 158\rangle$ (the linguistic translation of $\mathbf{x}_1$ was given in section 2.3). According to eq. 7:

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

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

$$P(C_k|B) = \frac{\sum_{i\in\mathcal{D}_k} P(B|\mathbf{x}_i)}{\sum_{i\in\mathcal{D}} P(B|\mathbf{x}_i)} \tag{8}$$

where $\mathcal{D}_k$ is the subset consisting of instances which belong to class k. 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.

According to the Jeffrey's rule [14] the probabilities of class C_k given a LDT with S branches are evaluated as follows:

$$P(C_k|\mathbf{x}) = \sum_{s=1}^{S} P(C_k|B_s)P(B_s|\mathbf{x}) \tag{9}$$

where $P(C_k|B_s)$ and $P(B_s|\mathbf{x})$ are evaluated based on equations 7 and 8.

3.1 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 [24], 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 7 (*Branch Entropy*) *The entropy of branch B given a set of classes $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$ is*

$$E(B) = -\sum_{t=1}^{|\mathcal{C}|} P(C_t|B)\log_2 P(C_t|B) \tag{10}$$

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:

$$EE(B, x_j) = \sum_{F_j\in\mathcal{F}_j} E(B \cup F_j)\cdot P(F_j|B) \tag{11}$$

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:

$$P(F_j|B) = \frac{\sum_{i\in\mathcal{D}} P(B \cup F_j|\mathbf{x}_i)}{\sum_{i\in\mathcal{D}} P(B|\mathbf{x}_i)} \tag{12}$$

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

$$IG(B, x_j) = E(B) - EE(B, x_j) \tag{13}$$

Algorithm 2: Decision Tree Learning

input : $\mathcal{LD}$: Linguistic dataset obtained from Algorithm 1.
output: LDT: Linguistic Decision Tree

Set a maximum depth M_{dep} and a threshold probability T.
for $l \leftarrow 0$ **to** M_{dep} **do**
  $\mathcal{B} \leftarrow \emptyset$ when $l = 0$
  The set of branches of LDT at depth l is $\mathcal{B}_l = \{B_1, \cdots, B_{|\mathcal{B}_l|}\}$
  for $v \leftarrow 1$ **to** $|\mathcal{B}|$ **do**
    foreach B_v **do** :
    for $t \leftarrow 1$ **to** $|\mathcal{C}|$ **do**
      foreach t **do** Calculating conditional probabilities: $P(C_t|B_v) = \sum_{i\in\mathcal{D}_t} P(B_v|\mathbf{x}_i)/\sum_{i\in\mathcal{D}} P(B_v|\mathbf{x}_i)$
      if $P(C_t|B_v) \geq T$ **then**
        break (step out the loop)
    if $\exists$ x_j*: x_j is free attribute* **then**
      foreach x_j **do** : Calculate: $IG(B_v, x_j) = E(B_v) - EE(B_v, x_j)$
      $IG_{max}(B_v) = \max_{x_j}[IG(B_v, x_j)]$
      Expanding B_v with x_{max} where x_{max} is the free attribute we can obtain the maximum IG value IG_{max}.
      $\mathcal{B}'_v \leftarrow \bigcup_{F_j\in\mathcal{F}_j}\{B_v \cup F_j\}$.
    else
      exit;
  $\mathcal{B}_{l+1} \leftarrow \bigcup_{r=1}^{s} \mathcal{B}'_r$.
$LDT = \mathcal{B}$

The pseudo-code is listed in Algorithm 2. 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 has been reached.

3.2 Degrees of Fuzziness

Through linguistic translation, all numerical data can be represented as mass assignments based on a predefined fuzzy discretization method. In this section, unless otherwise stated, we will use a percentile-based (or equal points) discretization. The idea is to cover approximately the same number of data points for each fuzzy label. The justification for using this discretization method is given in [20].

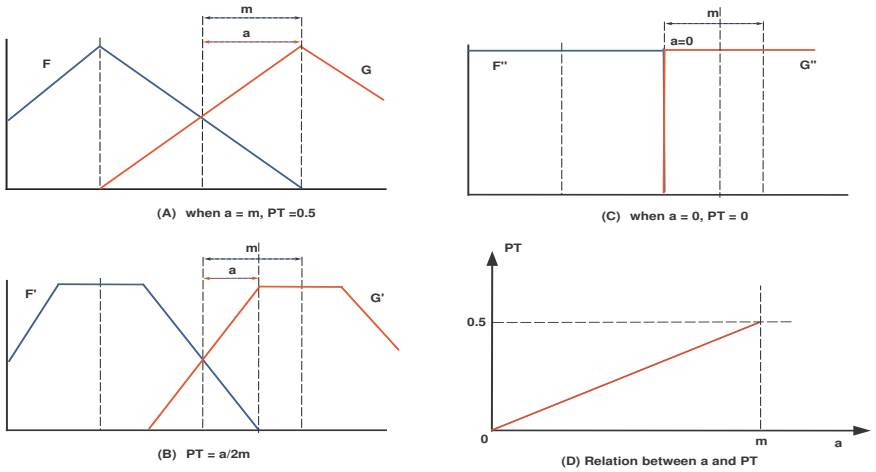

Fig. 3. A schematic illustration of calculating the overlap parameter PT given different degrees of overlaps.

Basically, fuzzy discretization provides an interpretation between numerical data and their corresponding linguistic data based on label semantics. We may notice that different fuzzy discretization may result in different linguistic data. We introduce a new parameter PT by which to measure the degrees of overlapping between fuzzy labels. As we can see from figure 3, given two fuzzy labels F and G, m is the distance between the weighting centers of a fuzzy labels to the meeting point of their membership functions. a is actually the length of the overlapping area. PT is calculated as follows:

$$PT = a/2m \tag{14}$$

$PT = 0.5$ represents 50% of overlapping between each two neighboring fuzzy labels (e.g., figure 3-A). $PT = 0$ represents no overlapping at all (figure 3-C),

i.e., the labels are discrete but not fuzzy. Figure 3-B shows a situation that the degree of overlapping is between 0 and 0.5. Figure 3-D also shows the linear relation of parameter a and PT.

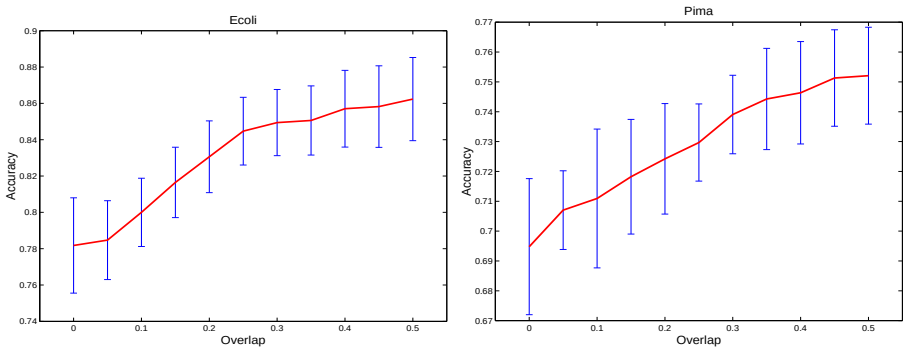

Fig. 4. Monotonically increased performance for linguistic decision trees with increasing degrees of fuzziness.

As we can see from these two figures, the performance these two datesets are roughly monotonic increased with the increase of PT. It implies that more fuzziness tends to increase the robustness of the LDT model and get better performance. From all the results, we can see that LDTs with fuzzy labels generally outperform the ones with discrete labels (where $PT = 0$). Due to the page limit, we cannot put all the results but they are available in [23]. Therefore, in summary, for the case of LDT model, we can say that fuzziness will bring greater performance. The increase is almost monotonically. But the optimal overlapping degrees are depends on the dataset you tested.

3.3 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 $\boldsymbol{\theta} = \langle \theta_1, \cdots, \theta_n \rangle$, 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

$$\forall F_j \in \mathcal{F}_j \quad m_{\theta_j}(F_j) = \begin{cases} \frac{pm(F_j)}{\sum_{F_j \in \lambda(\theta_j)} pm(F_j)} & if : F_j \in \lambda(\theta_j) \\ 0 & otherwise \end{cases} \tag{15}$$

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:

$$pm(F_j) = \int_{\Omega_j} m_x(F_j) p(x_j) dx_j \tag{16}$$

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

$$pm(F_j) \propto \int_{\Omega_j} m_x(F_j) dx_j \tag{17}$$

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

$$P(B|\boldsymbol{\theta}) = \prod_{r=1}^{|B|} m_{\theta_{j_r}}(F_{j_r}) \tag{18}$$

and therefore, by Jeffrey's rule [14]

$$P(C_t|\boldsymbol{\theta}) = \sum_{v=1}^{|LDT|} P(C_t|B_v) P(B_v|\boldsymbol{\theta}) \tag{19}$$

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.

3.4 Classification given fuzzy data

In previous sections LDTs have only been used to classify crisp data where objects are described in terms of precise attribute values. However, in many real-world applications limitations of measurement accuracy means that only imprecise values can be realistically obtained. In this section we introduce the idea of fuzzy data and show how LDTs can be used for classification in this context. Formally, a fuzzy database is defined to be a set of elements or objects each described by linguistic expressions rather than crisp values. In other words

$$\mathcal{FD} = \{\langle \theta_1(i), \ldots, \theta_n(i) \rangle : i = 1, \ldots, N\}$$

Currently there are very few benchmark problems of this kind with fuzzy attribute values. This is because, traditionally only crisp data values are recorded even in cases where this is inappropriate. Hence, we have generated a fuzzy database from a toy problem where the aim is to identify the interior of a figure of eight shape. Specifically, 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 figure 5). 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).

To form the fuzzy database we first generated a crisp database by uniformly sampling 961 points across $[-1.6, 1.6]^2$. Then each data vector $\langle x_1, x_2 \rangle$

was converted to a vector of linguistic expressions $\langle \theta_1, \theta_2 \rangle$ as follows: $\theta_j = \theta_{R_j}$ where $R_j = \{F \in \mathcal{F}_j : m_{x_j}(F) > 0\}$ A LDT was then learnt by applying the LID3 algorithm to the crisp database. This tree was then used to classify both the crisp and fuzzy data.

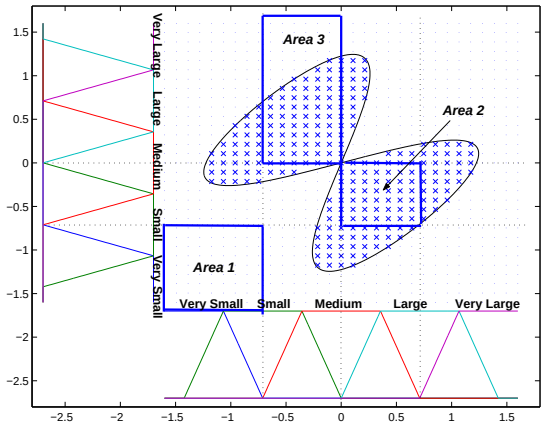

Fig. 5. Testing on the 'eight' problem with linguistic constraints $\boldsymbol{\theta}$, where each attribute is discretized by 5 trapezoidal fuzzy sets: *very small, small, medium, large and very large.*

Suppose a LDT is trained on the 'eight' database where each attribute is discretized by five fuzzy sets uniformly: *verysmall* (vs), *small* (s),*medium* (m), *large* (l) and *verylarge* (vl). Further, suppose we are given the following description of data points:

$\boldsymbol{\theta}_1 = \langle x \text{ is } vs \vee s \wedge \neg m, y \text{ is } vs \vee s \wedge \neg m \rangle$
$\boldsymbol{\theta}_2 = \langle x \text{ is } m \wedge l, y \text{ is } s \wedge m \rangle$
$\boldsymbol{\theta}_3 = \langle x \text{ is } s \wedge m, y \text{ is } l \vee vl \rangle$

Experimental results obtained based on the approach introduced in 3.3 are as follows:

$Pr(C_1|\boldsymbol{\theta}_1) = 1.000 \quad Pr(C_2|\boldsymbol{\theta}_1) = 0.000$
$Pr(C_1|\boldsymbol{\theta}_2) = 0.000 \quad Pr(C_2|\boldsymbol{\theta}_2) = 1.000$
$Pr(C_1|\boldsymbol{\theta}_3) = 0.428 \quad Pr(C_2|\boldsymbol{\theta}_3) = 0.572$

As we can see from figure 5, the above 3 linguistic constraints roughly correspond to the area 1, 2 and 3, respectively. By considering the occurrence of legal and illegal examples within these areas, we can verify the correctness of our approach.

3.5 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).

$$P(F_t^j|\mathbf{x}) = \sum_{v=1}^{|LDT|} P(F_t^j|B_v)P(B_v|\mathbf{x}) \tag{20}$$

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

$$\widehat{x}_t = \int_{\Omega_t} x_t \; p(x_t|\mathbf{x}) \; dx_t \tag{21}$$

where:

$$p(x_t|\mathbf{x}) = \sum_{j=1}^{|\mathcal{F}_t|} p(x_t|F_t^j) \; P(F_t^j|\mathbf{x}) \tag{22}$$

and

$$p(x_t|F_t^j) = \frac{m_{x_t}(F_t^j)}{\int_{\Omega_t} m_{x_t}(F_t^j) \; dx_t} \tag{23}$$

so that, we can obtain:

$$\widehat{x}_t = \sum_j \; P(F_t^j|\mathbf{x}) \; E(x_t|F_t^j) \tag{24}$$

where:

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

$$E(x_t|F_t^j) = \int_{\Omega_t} x_t \; p(x_t|F_t^j) \; dx_t = \frac{\int_{\Omega_t} x_t \; m_{x_t}(F_t^j) \; dx_t}{\int_{\Omega_t} m_{x_t}(F_t^j) \; dx_t} \tag{25}$$

We test our model on a toy problem of surface regression: 529 points were *uniformly* generated describing a surface defined by equation $z = sin(x \times y)$ where $x, y \in [0, 3]$. 2209 points are sampled uniformly as the test set. The attributes are discretized uniformly by fuzzy labels, the detailed results with different number of fuzzy labels are available in [21]. We compared the prediction surface by the LDT model and the original surface in figure in 6. As we can see from the figures that these results are quite comparable though LDT didn't capture the small change at the tail. In this experiment, we use 7 fuzzy labels for discretization. If we use more labels, we can get the results as good as as we want, but it just needs more computational time.

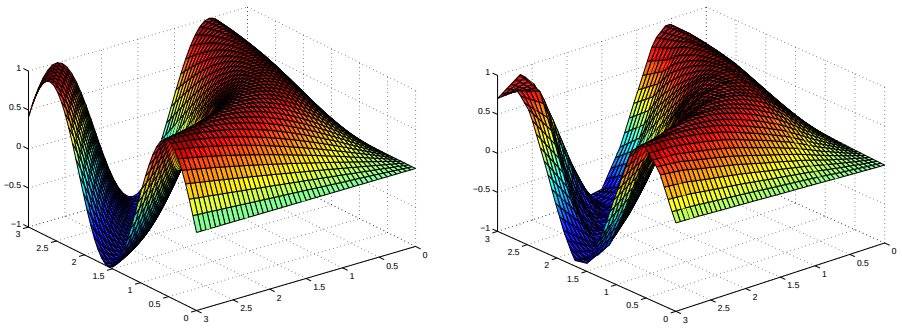

Fig. 6. Left-hand: the surface of $z = sin(x \times y)$. Right-hand: the prediction surface by linguistic decision trees.

4 Bayesian Estimation 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 $\mathbf{x}$ by Bayesian estimation is on the basis of the following probability,

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}|C_k)P(C_k)}{P(\mathbf{x})} \tag{26}$$

Since the denominator in eq. 26 is invariant across classes, we can consider it as a normalization parameter. So, we obtain:

$$P(C_k|\mathbf{x}) \propto P(\mathbf{x}|C_k)P(C_k) \tag{27}$$

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:

$$P(C_k|\mathbf{x}) \propto \prod_{j=1}^{n} P(x_j|C_k)P(C_k) \tag{28}$$

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.

4.1 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 [14].

$$p(x_j|C_k) = \sum_{F \in \mathcal{F}_j} p(x_j|F)P(F|C_k) \tag{29}$$

From Bayes theorem, we can obtain:

$$p(x_j|F) = \frac{P(F|x_j)p(x_j)}{P(F)} = \frac{m_{x_j}(F)p(x_j)}{pm(F)} \tag{30}$$

where,

$$pm(F) = \int_{\Omega_j} P(F|x_j)p(x_j)dx_j = \frac{\sum_{\mathbf{x} \in \mathcal{D}} m_{x_j}(F)}{|\mathcal{D}|} \tag{31}$$

Substituting equation 30 in equation 29 and re-arranging gives

$$p(x_j|C_k) = p(x_j) \sum_{F \in \mathcal{F}_j} m_{x_j}(F) \frac{P(F|C_k)}{pm(F)} \tag{32}$$

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

$$P(F|C_k) = \frac{\sum_{\mathbf{x} \in \mathcal{D}_k} m_{x_j}(F)}{|\mathcal{D}_k|} \tag{33}$$

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 [27].

4.2 Fuzzy Semi-Naive Bayes

The main advantage of using Semi-Naive Bayes over Naive Bayes is that it allows us to solve non-decomposable problems such as XOR by weakening the independence assumption of Naive Bayes. However, in order to utilize Semi-Naive Bayes it is necessary to find effective groupings of attributes within which dependencies must be taken into account. In this chapter, we present and evaluate a number of heuristic search algorithms for finding such groups of attributes.

Given a set of attributes: $x_1, x_2, \cdots, x_n$, they are partitioned into subsets $S_1, \cdots, S_w$ where $w \geq n$ and for each S_i a joint mass assignment $m_{i,j}$ is determined as follows: suppose, w.l.o.g $S_i = \{x_1, \cdots, x_v\}$ then the join mass assignment is

$$\forall T_1 \times \cdots \times T_v \in 2^{LA_1} \times \cdots \times 2^{LA_v} \tag{34}$$

$$m_{i,j}(T_1, \cdots, T_v) = \frac{1}{|DB_j|} \sum_{k \in \mathcal{D}} \prod_{r=1}^{w} m_{r,j}(T_i : x_i \in S_r) \tag{35}$$

Hence the prototype describing C_j is defined as $\langle m_{i,j}, \cdots, m_{w,j} \rangle$. A prototype of this form naturally defines a joint mass assignment m_j on the whole cross product space $2^{LA_1} \times \cdots \times 2^{LA_n}$ conditional on C_j as follows:

$$\forall T_1 \times \cdots \times T_n \in 2^{LA_1} \times \cdots \times 2^{LA_n} m_j(T_1, \cdots, T_n) = \prod_{r=1}^{w} m_{r,j}(T_i : x_i \in S_r) \tag{36}$$

In this formulation we are encoding variable dependence within the variable groupings $S_i : i = 1, \cdots w$ and assuming independence between the groups.

In order to estimate classification probabilities given input vectors of rea attribute values we need a mechanism for mapping from mass assignments on label space onto density functions on attribute space.

Definition 8 (Conditional Density Given a Mass Assignment) *Let x be a variable into Ω with prior distribution $p(x)$, LA be a set of labels for x and m be a posterior mass assignment for the set of appropriate labels of x inferred from some database $\mathcal{D}$. Then the posterior distribution of x conditional on m is given by*

$$\forall x \in \Omega,\ p(x|m) = p(x) \sum_{S \subseteq LA} \frac{m(S)}{pm(S)} m_x(S) \tag{37}$$

where $pm(S)$ is the prior mass assignment generated by the prior distribution $p(x)$ according to

$$pm(S) = \int_{\Omega} m_x(S) p(x) dx \tag{38}$$

This definition is motivated by the following argument based on the theorem of total probability which for a mass assignment, describing variables x on Ω.

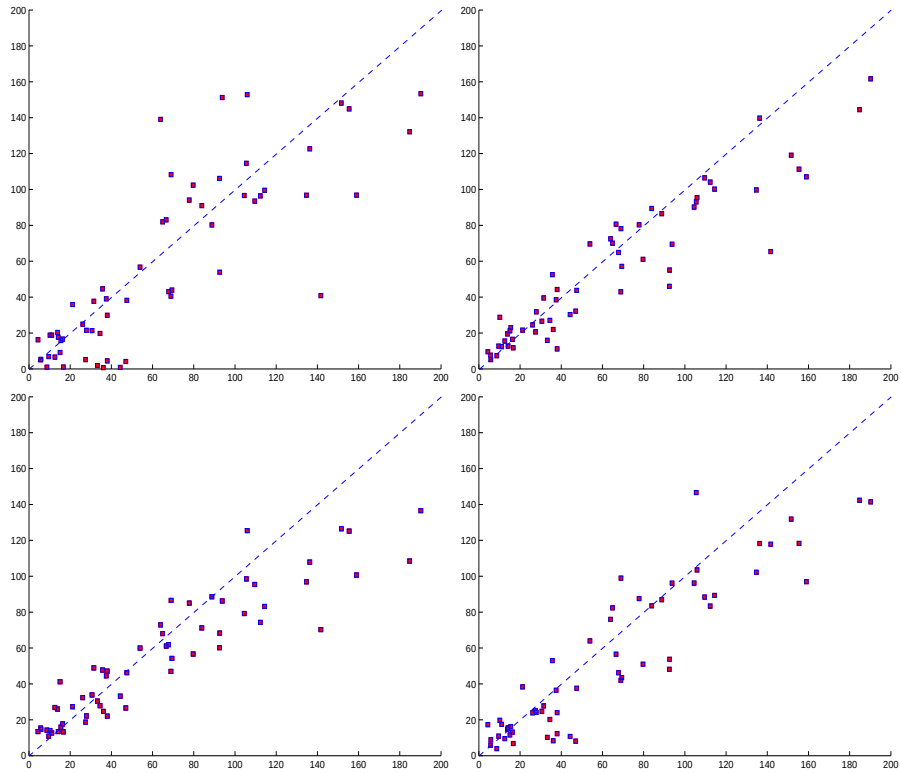

Fig. 7. Scatter plot showing original data verses prediction data on sunspot prediction problems. Upper left: Fuzzy Naive Bayes; upper right: Support Vector Regression; lower left: non-merged LDT with 5 fuzzy labels; lower right: Semi-naive Bayes.

We now consider methods for finding attribute groupings that increase discrimination in the model. Two measures has been proposed in [27]:

Definition 9 (Importance Measure) *Let the joint mass assignment for S_i given C_j be denoted $m_{i,j}$. For any input vector S_i the probability of cloass C_j can be estimated using Bayes theorem where*

$$P(C_j|S_i) = \frac{p(S_i|m_{i,j})|C_j|}{p(S_i|m_{i,j})|C_j| + p(S_i|m_{i,\neg j})|C_{\neg j}|} \tag{39}$$

where $m_{i,\neg j}$ denotes the mass assignments for S_j given $\neg C_j$. The importance measured of group S_i for class C_j is then defined by

$$IM_j(S_i) = \frac{\sum_{k\in\mathcal{D}_j} P(C_j|S_i(k))}{\sum_{k\in\mathcal{D}} P(C_j|S_i(k))} \tag{40}$$

Effectively, $IM_j(S_i)$ is a measure of the importance of the set of variables S_i as discriminators of C_j from the other classes.

Definition 10 (Correlation Measure) *Let $\mathcal{F}_1$ be the focal sets for S_1 and $\mathcal{F}_2$ the focal sets for S_2. Now let $m_{1,2,j}$ be the joint mass of $S_1 \cup S_2$ given C_j*

$$C(S_1, S_2) = \sqrt{\frac{1}{|\mathcal{F}_1||\mathcal{F}_1|} \sum_{R\subseteq\mathcal{F}_1} \sum_{T\subseteq\mathcal{F}_2} (m_{1,2,j}(R,T) - m_{1,j}(R)m_{2,j}(T))^2} \tag{41}$$

Here a threshold must be used to determine whether attributes should be grouped. The nearer the correlation measure gets to 1 the higher the correlation between attribute groups.

We tested our models 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, Fuzzy Naive Bayes, Semi-Naive Bayes and ε-SVR [6] are compared in figure 7. We can see the results are quite comparable. In these graphs, for an error free prediction all points will fall on the line defined by $y = x$. Roughly, from the illustration, we can see that SVR and non-merged LDT have better performance, because predicted values distributed closer to $y = x$ than other two models.

4.3 Hybrid Bayesian Estimation Tree

Based on previous two linguistic models, a hybrid model was proposed in [19]. Given a decision tree T is learnt from a training database $\mathcal{D}$. According to the Bayesian theorem: A data element $\mathbf{x} = \langle x_1, \ldots, x_n\rangle$ can be classified by:

$$P(C_k|\mathbf{x}, T) \propto P(\mathbf{x}|C_k, T)P(C_k|T) \tag{42}$$

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

$$P(\mathbf{x}|C_k, T) = P(\mathbf{x}_T|C_k, T)P(\mathbf{x}_B|C_k, T) \tag{43}$$

Because $\mathbf{x}_B$ is independent of the given decision tree T and if we assume the variables in $\mathbf{x}_B$ are independent of each other given a particular class, we can obtain:

$$P(\mathbf{x}_B|C_k,T) = P(\mathbf{x}_B|C_k) = \prod_{j\in\mathbf{x}_B} P(x_j|C_k) \tag{44}$$

Now consider $\mathbf{x}_T$. According to Bayes theorem,

$$P(\mathbf{x}_T|C_k,T) = \frac{P(C_k|\mathbf{x}_T,T)P(\mathbf{x}_T|T)}{P(C_k|T)} \tag{45}$$

Combining equation 43, 44 and 45:

$$P(\mathbf{x}|C_k,T) = \frac{P(C_k|\mathbf{x}_T,T)P(\mathbf{x}_T|T)}{P(C_k|T)} \prod_{j\in\mathbf{x}_B} P(x_l|C_k) \tag{46}$$

Combining equation 42 and 46

$$P(C_k|\mathbf{x},T) \propto P(C_k|\mathbf{x}_T,T)P(\mathbf{x}_T|T) \prod_{j\in\mathbf{x}_B} P(x_j|C_k) \tag{47}$$

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

$$P(C_k|\mathbf{x},T) \propto P(C_k|\mathbf{x}_T,T) \prod_{j\in\mathbf{x}_B} P(x_j|C_k) \tag{48}$$

where $P(x_j|C_k)$ is evaluated according to eq. 32 and $P(C_k|\mathbf{x}_T,T)$ is just the class probabilities evaluated from the decision tree T according to equation 9.

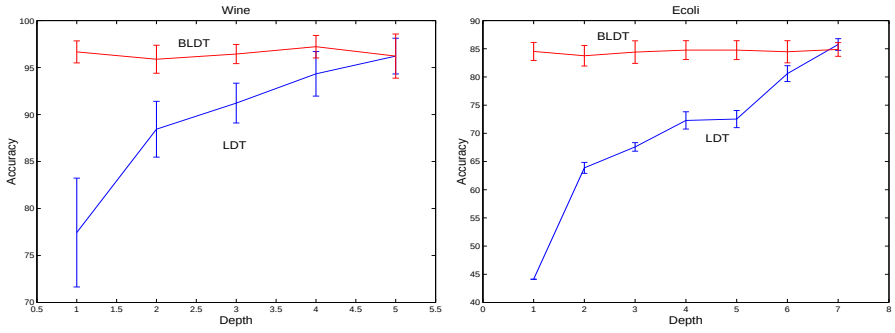

Fig. 8. Results for single LDT with Bayesian estimation: average accuracy with standard deviation on each dataset against the depth of the tree.

We tested this new model with a set of UCI [4] data sets. Figure 8 is a simple result. More results are available in [19]. From figures 8, 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 eq. 48. If all the attributes are used in building the tree (i.e. $\mathbf{x}_T = \mathbf{x}$), the probability estimations are from the tree only, that is:

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

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

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

which is simply a Naive Bayes classifier.

4.4 Bayesian Estimation From a Set of Trees

Given a training dataset, a small-sized tree (usually the depth is less than 3) can be learnt based on the method we discussed in section 3. We then learn another tree with the same size based on the remaining attributes, i.e., the attributes which have not been used in previous trees. In this manner, a set of trees can successively be built from training set. We denote this set of trees by $\mathcal{T} = \langle T_1, \ldots, T_W \rangle$ and where the set of attributes $\mathbf{x}_{T_w}$ for $w = 1, \ldots, W$ for a partition of $\{x_1, \ldots, x_n\}$ (see fig. 9 for a schematic illustration). For a given unclassified data element $\mathbf{x}$, we can partition it into W groups of disjoint set of attributes $\langle \mathbf{x}_{T_1}, \ldots, \mathbf{x}_{T_W} \rangle$. If we assume:

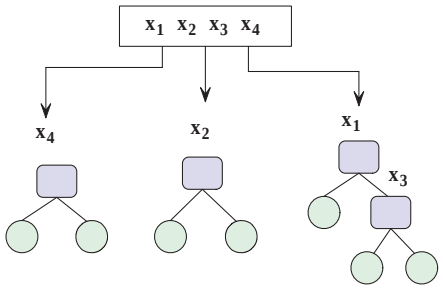

Fig. 9. An schematic illustration of Bayesian estimation from a set of linguistic decision trees.

$$P(C_t|\mathbf{x}) = P(C_t|\mathbf{x}_{T_1}, \ldots, \mathbf{x}_{T_W}) \approx P(C_t|T_1, \ldots, T_W) \tag{49}$$

Then, according to the Bayes theorem:

$$P(C_t|\mathcal{T}) = P(C_t|T_1,\ldots,T_W) = \frac{P(T_1,\ldots,T_W|C_t)P(C_t)}{P(T_1,\ldots,T_W)} \tag{50}$$

Assuming that the trees are generated independently then it is reasonable to assume that the groups of attributes are conditionally independent of each other. Hence,

$$P(T_1,\ldots,T_W|C_t) = \prod_{w=1}^{W} P(T_w|C_t) \tag{51}$$

For a particular tree T_w for $w = 1,\ldots,W$, we have

$$P(T_w|C_t) = \frac{P(C_t|T_w)P(T_w)}{P(C_t)} \tag{52}$$

So that,

$$\prod_{w=1}^{W} P(T_w|C_t) = \frac{\prod_{w=1}^{W} P(C_t|T_w) \prod_{i=1}^{W} P(T_w)}{P(C_t)^W} \tag{53}$$

Combining eq. 50, 51 and 53, we obtain

$$P(C_t|\mathcal{T}) \propto \frac{\prod_{w=1}^{W} P(C_t|T_w) \prod_{w=1}^{W} P(T_w)}{P(C_t)^{W-1}} \tag{54}$$

Since $\prod_{w=1}^{W} P(T_w)$ is independent from C_t, we finally obtain:

$$P(C_t|\mathcal{T}) \propto \frac{\prod_{w=1}^{W} P(C_t|T_w)}{P(C_t)^{W-1}} \tag{55}$$

where $P(C_t|T_w)$ is evaluated according to eq. 9.

5 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 [25].

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 [29] and first-order fuzzy rule induction based on FOIL [5, 17]. In this paper, we will focus on an extension to the FOIL algorithm based on label semantics.

5.1 Generalized Appropriateness Measures

Based on definition 5, 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 11 (*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 $L \in \mathcal{L}$.

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 5, 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:

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

5.2 Linguistic Rules in Label Semantics

In sections 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. 4, a linguistic rule is a rule can be represented as a multi-dimensional logical expressions of fuzzy labels.

Definition 12 (*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 $\lambda^{(n)}(\theta)$, 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 13 (*Multi-dimensional Appropriateness Degrees*) *Given a set of n-dimensional label expressions $MLE^{(n)}$:*

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

$$\begin{aligned}\mu^n_\theta(\mathbf{x}) \ = \mu^n_\theta(x_1, \cdots, x_n) &= \sum_{\langle F_1, \cdots, F_n \rangle \in \lambda^{(n)}(\theta)} (F_1, \cdots, F_n) \\ &= \sum_{\langle F_1, \cdots, F_n \rangle \in \lambda^{(n)}(\theta)} \prod_{j=1}^{n} m_{x_j}(F_j)\end{aligned}$$

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.

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 5 we have that

$$\begin{aligned}\lambda^{(2)}((med \wedge \neg s) \wedge \neg lo) &= \lambda^{(2)}(med \wedge \neg s) \cap \lambda^{(2)}(\neg lo) \\ &= \lambda(med \wedge \neg s) \times \lambda(\neg lo)\end{aligned}$$

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. 5 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. 10: 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.

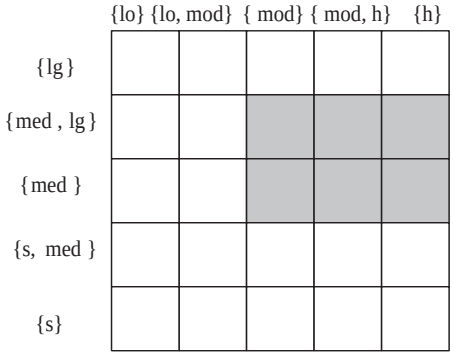

Fig. 10. Representation of the multi-dimensional λ-function of the logical expression $\theta = (med \wedge \neg\ s) \wedge \neg lo$ showing the focal cells $\mathcal{F}_1 \times \mathcal{F}_2$.

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 def. 5:

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

In another 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 is highlighted in next section on rule induction.

5.3 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 [25], 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 [25] so as to incorporate linguistic expressions based on labels semantics. Consider a classification rule of the form:

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

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

$$T_i^+ = \sum_{l \in \mathcal{D}_k} \mu_\theta(\mathbf{x}_l) \tag{56}$$

and the coverage of negative examples is given by

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

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

$$I(R_i) = -\log_2 \left(\frac{T_i^+}{T_i^+ + T_i^-} \right) \tag{58}$$

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:

$$T_{i+1}^{+} = \sum_{l \in \mathcal{D}_k} \mu_{\theta \wedge \varphi}(\mathbf{x}_l) \tag{59}$$

$$T_{i+1}^{-} = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_{\theta \wedge \varphi}(\mathbf{x}_l) \tag{60}$$

Therefore, the information becomes,

$$I(R_{i+1}) = -\log_2 \left(\frac{T_{i+1}^{+}}{T_{i+1}^{+} + T_{i+1}^{-}} \right) \tag{61}$$

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

$$G(\varphi) = T_{i+1}^{+}(I(R_i) - I(R_{i+1})) \tag{62}$$

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:

$$P(C_k|R_i) = \frac{\sum_{l \in \mathcal{D}_k} \mu_\theta(\mathbf{x}_l)}{\sum_{l \in \mathcal{D}} \mu_\theta(\mathbf{x}_l)} = \frac{T_i^{+}}{T_i^{+} + T_i^{-}} \tag{63}$$

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.

5.4 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:

$$CV(\Delta_i) = \frac{\sum_{l \in \mathcal{D}_k} \mu_{\varphi_1 \vee \cdots \vee \varphi_t}(\mathbf{x}_l)}{|\mathcal{D}_k|} \tag{64}$$

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

$$\begin{aligned} \delta(\mathbf{x}|\Delta_i) = \mu_{\neg \Delta_i}(\mathbf{x}) = \mu_{\neg(\varphi_1 \vee \cdots \vee \varphi_t)}(\mathbf{x}) \\ = 1 - \mu_{(\varphi_1 \vee \cdots \vee \varphi_t)}(\mathbf{x}) = 1 - \sum_{w=1}^{t} \mu_{R_w}(\mathbf{x}) \end{aligned} \tag{65}$$

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 $R = \theta \rightarrow C_k$,

$$T^+ = \sum_{l \in \mathcal{D}_k} \mu_\theta(\mathbf{x}_l) \times \delta(\mathbf{x}_l|\Delta_i) \tag{66}$$

$$T^- = \sum_{l \in (\mathcal{D} - \mathcal{D}_k)} \mu_\theta(\mathbf{x}_l) \tag{67}$$

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 $\mathbf{x}$, we can estimate the probability of C_k, $P(C_k|\mathbf{x})$, as follows: Firstly, we determine the rule $R_{max} = \varphi_j \rightarrow C_k$ for which $\mu_{\varphi_j}(\mathbf{x})$ is maximal:

$$\varphi_j = \arg\max_{k \in \Delta_i} \mu_{\varphi_k} \tag{68}$$

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

$$p_{max} = P(C_k|\varphi_j) \tag{69}$$

$$q_{max} = P(C_k|\neg\varphi_j) \tag{70}$$

We then use Jeffrey's rule [14] to evaluate the class probability by:

$$P(C_k|\mathbf{x}) = p_{max} \times \mu_{\varphi_j}(\mathbf{x}) + q_{max} \times (1 - \mu_{\varphi_j}(\mathbf{x})) \tag{71}$$

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 [22].

6 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, a number of linguistic data mining models have been proposed including: Linguistic Decision Trees (LDT) (for both classification and prediction), Bayesian estimation models (Fuzzy Naive Bayes, Semi-Naive Bayes, Bayesian Estimation Trees) and Linguistic Rule Induction (Linguistic FOIL).

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 [20]. 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 re-constructed 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.

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 constraints and classification for fuzzy data. Other information fusion methods are discussed in [12]. How to efficiently use background knowledge is an important challenge in machine learning. For example, Wang [28] 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.

Acknowledgements

The authors thank Prof Lotfi Zadeh for some insightful comments on this research. The first author also thanks Masoud Nikravesh, Marcus Thint, Ben Azvine and Trevor Martin for their interests in this research and support. The writing of this chapter is partly funded BT/BISC fellowship.

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. J Lawry (2006), *Modelling and Reasoning with Vague Concepts*, Springer.
14. R. C. Jeffrey (1965), *The Logic of Decision*, Gordon & Breach Inc., New York.
15. C. Olaru and L. Wehenkel (2003), A complete fuzzy decision tree technique. *Fuzzy Sets and Systems*. 138: pp.221-254.
16. Y. Peng, P. A. Flach (2001), Soft discretization to enhance the continuous decision trees. *ECML/PKDD Workshop: IDDM*.
17. H. Prade, G. Richard, and M. Serrurier (2003), Enriching relational learning with fuzzy predicates, *Proceedings of PKDD*, LNAI 2838, pp. 399-410.
18. 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.
19. 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.
20. Z. Qin and J. Lawry (2005), Decision tree learning with fuzzy labels, *Information Sciences*, Vol. 172/1-2: pp. 91-129.
21. Z. Qin and J. Lawry (2005), Prediction trees using linguistic modelling, *the Proceedings of International Fuzzy Association World Congress-05*, September 2005, Beijing, China.

22. 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.
23. Z. Qin and J. Lawry (2007), Fuzziness and performance: an empirical study with linguistic decision trees. To appear in IFSA-2007, Cuncun, Mexico.
24. J. R. Quinlan (1986), Induction of decision trees, *Machine Learning*, Vol 1: pp. 81-106.
25. J. R. Quinlan (1990), Learning logical definitions from relations, *Machine Learning*, 5: 239-266.
26. J. R. Quinlan (1993), *C4.5: Programs for Machine Learning*, San Mateo: Morgan Kaufmann.
27. 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*, Vol. 176: pp 438-464.
28. Pei Wang (2004), The limitation of Bayesianism, *Artificial Intelligence* 158(1): pp. 97-106.
29. D. Xie (2005), Fuzzy associated rules discovered on effective reduced database algorithm, *Proceedings of IEEE-FUZZ*, pp. 779-784, Reno, USA.
30. L. A. Zadeh (1965), Fuzzy sets, *Information and Control*, Vol 8: pp. 338-353.
31. L. A. Zadeh (1996), Fuzzy logic = computing with words, *IEEE Transaction on Fuzzy Systems*. Vol. 4, No. 2: pp. 103-111.
32. L. A. Zadeh (2002), Toward a perception-based theory of probabilistic reasoning with imprecise probabilities, *Journal of Statistical Planning and Inference*, Vol. 105: pp. 233264.
33. L.A. Zadeh (2003), Foreword for modelling with words, *Modelling with Words*, LNAI 2873, Ed., J. Lawry, J. Shanahan, and A.Ralescu, Springer.
34. L.A. Zadeh (2005), Toward a generalized theory of uncertainty (GTU) an outline, *Information Sciences*, Vol. 172/1-2, pp. 1-40.