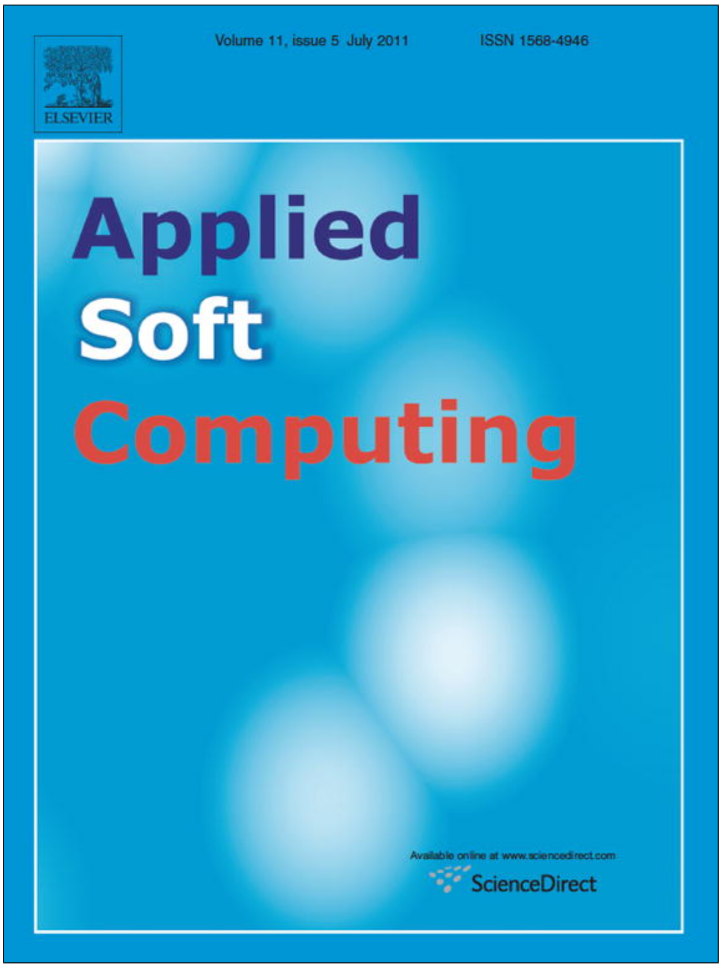
Volume 11, issue 5 July 2011
ISSN 1568-4946
ELSEVIER
Applied
Soft
Computing
Available online at www.sciencedirect.com
ScienceDirect

Applied Soft Computing 11 (2011) 3916–3928

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Prediction and query evaluation using linguistic decision trees

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

[a] Robotics Institute, Carnegie Mellon University, USA
[b] Intelligent Computing and Machine Learning Lab, School of Automation Science and Electrical Engineering, Beihang University, China
[c] Intelligent Systems Laboratory, University of Bristol, UK

ARTICLE INFO

Article history:
Received 3 June 2009
Received in revised form
15 December 2010
Accepted 13 February 2011
Available online 4 March 2011

Keywords:
Label semantics
LID3
Linguistic decision tree
Mass assignment
Random set
Linguistic query

ABSTRACT

Linguistic decision tree (LDT) is a tree-structured model based on a framework for "Modelling with Words". In previous research [15,17], an algorithm for learning LDTs was proposed and its performance on some benchmark classification problems were investigated and compared with a number of well known classifiers. In this paper, a methodology for extending LDTs to prediction problems is proposed and the performance of LDTs are compared with other state-of-art prediction algorithms such as a Support Vector Regression (SVR) system and Fuzzy Semi-Naive Bayes [13] on a variety of data sets. Finally, a method for linguistic query evaluation is discussed and supported with an example.

1. Introduction

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 and the transition is characterized by membership functions. 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. If we consider this methodology in an opposite way: given an object, fuzzy labels (sets) can be used to describe this object with some appropriateness measures. Follow this idea, we discuss a new approach based on random set theory to interpret imprecise concepts. This framework, first proposed by Lawry [13] and is referred to as *Label Semantics*, can be regarded as an approach to Modelling with Words [12].

Modeling with Words is a new research area which emphasis "modelling" rather than "computing". For example, Zadeh's theories on Perception-based Computing [30] and Precisiated Natural Language [31] are the approaches of "computing". However, the relation between it and Computing with Words [29] 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 paper, the framework we use is mainly for modelling and building intelligent machine learning and data mining systems. Therefore, the research presented here is considered as a framework for Modelling with Words.

As one of the most successful branches of Artificial Intelligence, machine learning and data mining research has developed rapidly in recent decades. However, most machine learning algorithms specialise on classification problems. For example, in the popular UCI repository [2] for machine learning and data mining research, most datasets concern classification. However, in many real-world applications, data ranging from financial analysis to weather forecasting are for prediction. A prediction model can be easily used as a classifier by setting a decision threshold. Usually, a good prediction model can be a good classifier as well. However, not all the classifier can be used for prediction. Tree induction algorithms were received a great deal of attention because of their simplicity and effectiveness. From early discrete decision trees such as ID3 [18] and C4.5 [19] to a variety types of fuzzy decision trees [8,14,23–26], most tree induction models are designed for classification but not for prediction. Although there is some research on regression trees. For example, Breiman et al.'s CART algorithm [3]. Here we present a tree induction model based on a high-level knowledge representation framework which is referred to as *Label Semantics* [10]. Label semantics is a random set semantics for mod-

* Corresponding author at: Intelligent Computing and Machine Learning Lab, School of Automation Science and Electrical Engineering, Beihang University, Xueyuan Road 37, Beijing, China
E-mail addresses: zcqin@buaa.edu.cn, zengchang.qin@gmail.com (Z. Qin), j.lawry@bris.ac.uk (J. Lawry).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.02.010

elling imprecise concepts where the degree of appropriateness of a linguistic expression as a description of a value is measured in terms of how the set of appropriate labels for that value varies across a population. Based on label semantics, *linguistic decision tree* (LDT) [15] was proposed where linguistic expressions such as *small*, *medium* and *large* are used to build a tree guided by information based heuristics. For each branch, instead of labeling it with a certain class (such as positive or negative) the probability of members of this branch belonging to a particular class is evaluated from a given training dataset. Unlabeled data is then classified by using probability estimation of classes across the whole decision tree.

Compared to other tree learning algorithms, the LDT model has following advantages: (1) the LDT model has very good transparency: A LDT can be interpreted as a set of linguistic rules based on label semantics. By applying a forward merging algorithm (see Section 3.5), we can generate much more compact trees without a significant loss of accuracy. (2) The performance of LDT model is comparable to other classifiers such as Naive Bayes and Neural Networks [17]. (3) The linguistic structure of the LDT model allows linguistic queries and information fusion (see Section 5). In this paper, the LDT classification model is extended to prediction and empirical results on several benchmark problems are presented. These problems rangers from function regression, time series prediction to real-world applications such as flood forecasting.

This paper is organized as follows: Section 2 gives a short introduction on label semantics and the corresponding methodology for analyzing data. In Section 3, the LDT model for classification is outlined and it is described how this can be extended from classification problems to prediction problems. Experimental results for the benchmark problems are given and compared with other prediction models in Section 4. In the last section the methodology for linguistic query evaluation algorithms are introduced based on a formal linguistic reasoning framework.

2. Random set semantics for Modelling with Words

Label Semantics, proposed by Lawry [10], is a random set based framework for modelling with linguistic expressions based on labels such as *small*, *large*, *short*, *tall*, *young*, *old* and so on. Such labels are defined by overlapping fuzzy sets which are used to cover the universe of continuous variables. The fundamental question posed by label semantics is how to use linguistic expressions to label numerical values. The basic idea is that when individuals make assertions, such as 'John is tall', they are essentially providing the information that the label *tall* is appropriate for describing John's height.

2.1. Label Semantics

For a variable x into a domain of discourse denoted by Ω we identify a finite set of linguistic labels $LA = \{L_1, \ldots, L_n\}$ with which to label the values of x. Then, for a specific value $x \in \Omega$, an individual I identifies a subset of LA, 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 LA. By evaluating the probability of occurrence of a particular set of labels say S, for D_x across the population then we obtain a distribution on D_x referred to as a mass assignment and denoted by m_x (see [1] for details on the Mass Assignment theory). We can view the random set D_x as a description of the variable x in terms of the labels in LA. 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 LA, denoted D_x, with associated distribution m_x, given by

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

where $m_x(S)$ is the mass associated with a set of labels S and

$$\sum_{S \subseteq LA} m_x(S) = 1$$

Intuitively $m_x(S)$ quantifies the evidence that S is the set of appropriate labels for x. For example, given a set of labels defined on a man's age $LA_{age} = \{young, middle\text{-}aged, old\}$. For a particular group of voters V and $|V| = 10$, 3 of them agree that *young* is the only suitable label for the age of 30 and 7 may agree that both *young* and *middle-aged* are suitable labels. In this case, according to Definition 1, $m_{30}(young) = 0.3$ and $m_{30}(young, middle\text{-}aged) = 0.7$ so that the mass assignment for 30 is

$$m_{30} = \{young\}: \ 0.3, \{young, middle\text{-}aged\}: \ 0.7$$

where 0.3 is the associated mass for $\{young\}$ and 0.7 is the associated mass for $\{young, middle\text{-}aged\}$.

Within this framework, *appropriateness degrees* are used to evaluate how appropriate a label is for describing a particular value of x. Given a particular value x, the appropriateness degree of L as a label for x where L is represented by fuzzy set F, is the membership value of x belonging to F. The reason we use the new term 'appropriateness degree' is partly because it more accurately reflects the underlying semantics and partly to highlight the quite distinct calculus based on this framework. It is assumed that the appropriateness of L to x, $\mu_L(x)$ is the total evidence that L is an appropriate label for x which motivates the following definition.

Definition 2 *(Appropriateness degrees).*

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

Consider the above example, the appropriate degrees for using *young* to label 30 is $\mu_{young}(30) = 0.3 + 0.7 = 1$. And similarly, $\mu_{middle\text{-}aged}(30) = 0.7$. In many real-world applications, only imprecise values can be realistically obtained due to limitations of measurement on accuracy. In the label semantics framework, values are represented by a higher level language, i.e. linguistic labels. By taking advantages of the high level representation language for its robustness and ability of coping with uncertainties, a new paradigm for data analysis and data mining is proposed.

2.2. Label Semantics for data analysis

We now make the additional assumption that value descriptions are consonant random label sets [10] which simply means that individuals in V differ regarding what labels are appropriate for a value. The consonance restriction could be justified by the idea that all individuals share a common ordering on the appropriateness of labels for a value and that the composition of D_x^I is consistent with this ordering for each I. For the purposes of data analysis, a consonance assumption is needed.

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

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

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

where $M_0 = \varnothing$ and $M_t = \{L \in LA \mid \mu_L(x) \geq \beta_t\}$ for $t = 1, 2 \ldots, k$.

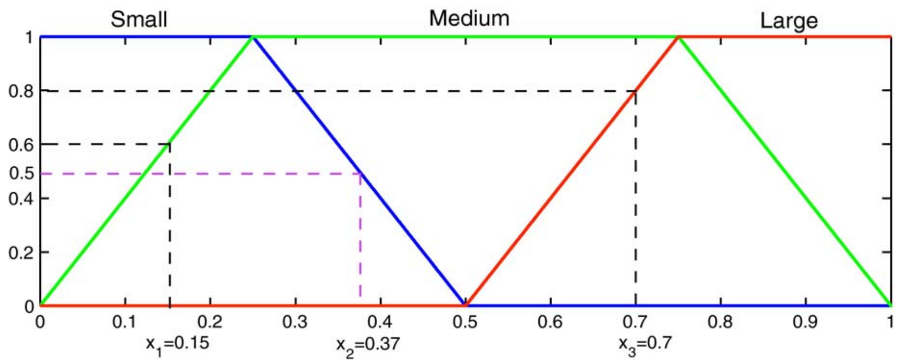

Fig. 1. An example of a full fuzzy partitioning with 3 uniformly distributed trapezoidal fuzzy sets with 50% overlap.

Definition 3 provides us with a way of calculating the mass assignment m_x from the given appropriateness degrees (see the following example). Because the appropriateness degrees are sorted under the consonance assumption the resulting mass assignments are 'nested'. Clearly, there is a unique consonant mapping to mass assignments for a given set of appropriateness degree values. We also make a full fuzzy partitioning assumption to avoid mass being allocated to the empty set. More practical to disallow this possibility as follows:

Definition 4 *(Full fuzzy partitioning).* Given a continuous discourse Ω, it is called a full fuzzy partitioning of Ω by LA if:

$$\forall x \in \Omega, \quad \exists L \in LA, \quad \mu_L(x) = 1$$

The full fuzzy partitioning assumes that, for any data element, there always exists a particular label which all the voters agree it is appropriate, though the voters may have different opinions on other labels. Fig. 1 shows a schematic illustration of a full fuzzy partitioning with 3 trapezoidal fuzzy sets. Unless otherwise stated, in this paper we will use N_F fuzzy sets with 50% overlap to cover a continuous universe. This guarantees that only two fuzzy sets overlap, so that the appropriateness degrees satisfy: $\forall x \in \Omega, \exists i \in \{1, \ldots, N_F - 1\}$ such that $\mu_{L_i}(x) = \alpha$, $\mu_{L_{i+1}}(x) = \beta$ and $\mu_{L_j}(x) = 0$ for $j < i$ or $j > i + 1$ and where $max(\alpha, \beta) = 1$. Under the full fuzzy partitioning assumption, w.l.o.g. we assume $\alpha = 1$ then m_x has the following form according to Definition 3.

$$m_x = \{L_i\} : \ 1 - \beta, \ \{L_i, L_{i+1}\} : \ \beta, \{L_j\} : \ 0 \quad \text{for } j \notin \{i, i+1\} \tag{1}$$

It is also important to note that, given definitions for the appropriateness degrees on labels, we can isolate a set of subsets of LA with non-zero masses. These are referred to as *focal sets* or *focal elements*.

Definition 5 *(Focal elements).* The set of focal elements for LA is defined by:

$$\mathcal{F} = \{S \subseteq LA | \exists x \in \Omega, m_x(S) > 0\} \tag{2}$$

For example, the focal elements generated by the fuzzy partitioning in Fig. 1 are $\mathcal{F} = \{F_1, \ldots, F_5\} = \{\{small\}, \{small, medium\}, \{medium\}, \{medium, large\}, \{large\}\}$. However, $\{small, large\}$ can not occur as a focal element since these two labels do not overlap. In other words, focal elements are the sets of labels with non-zero associated masses in describing data.

There are a few ways of fuzzy partitioning, we usually use uniform partitioning and percentile-based partitioning. Uniform partitioning splits the continuous universe into several intervals of identical length (for example, see Fig. 1). The intervals generated by percentile-based partitioning contain approximately same number of instances. Given the assumptions we have made (consonant, full fuzzy partitioning with 50% overlap) 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 $\mu_L : L \in LA$. We call this the *linguistic translation (LT).*

By applying linguistic translation, numerical values are represented by sets of appropriate labels with associated masses. For example, Fig. 1 shows a full fuzzy covering of the universe $\Omega = [0, 1]$ with three fuzzy labels: *small*, *medium* and *large* with 50% overlap where $\mathcal{F} = \{\{small\}, \{small, medium\}, \{medium\}, \{medium, large\}, \{large\}\}$. For the data element $x_1 = 0.15$, the appropriate labels are *small* and *medium*, and the appropriateness degrees (can be read from the membership values) of these labels are:

$$\mu_{small}(x_1) = 1, \qquad \mu_{medium}(x_1) = 0.6$$

The mass assignment on the appropriate labels can be calculated based on Eq. (1) to give:

$$m_{x_1} = \{small\} : \ 0.4, \{small, medium\} : \ 0.6$$

Similarly, for $x_2 = 0.37$, $x_3 = 0.7$, we obtain

$$m_{x_2} = \{small, medium\} : \ 0.5, \{medium\} : \ 0.5$$

$$m_{x_3} = \{medium, large\} : \ 0.8, \{large\} : \ 0.2$$

The linguistic translation for $\langle x_1, x_2, x_3 \rangle$ can be illustrated as follows:

$$\begin{pmatrix} x \\ \hline 0.15 \\ 0.37 \\ 0.7 \end{pmatrix} \overset{LT}{\rightarrow} \begin{pmatrix} m_x(\{s\}) & m_x(\{s,m\}) & m_x(\{m\}) & m_x(\{m,l\}) & m_x(\{l\}) \\ \hline 0.4 & 0.6 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0.2 \end{pmatrix}$$

We may notice that the linguistic translation is related to the membership functions we used. If we use different discretization techniques, we may obtain different mass assignments on labels for given numerical values. Empirical studies show that, using different discretization methods has no significant influence on the performance and stability of learning algorithms [15]. Hence, w.l.o.g, all the experiments in this paper are based on the percentile-based fuzzy partitioning method.

3. Linguistic decision trees

Linguistic decision tree, proposed by Qin and Lawry [15], is a transparent classification model based on label semantics. Consider a database with n attributes and N instances $\mathcal{D} = \{x_1(i), \ldots, x_n(i) | i = 1, \ldots, N\}$ and each instance is labeled by one of the classes: $\mathcal{C} = \{C_1, \ldots, C_{|\mathcal{C}|}\}$. A linguistic decision tree is consisted by a set of branches with associated class probabilities in the following form:

$$LDT = \{\langle B_1, P(C_1|B_1), \ldots, P(C_{|\mathcal{C}|}|B_1)\rangle, \ldots, \langle B_s, P(C_1|B_s), \ldots, P(C_{|\mathcal{C}|})|B_s)\rangle\}$$

$P(C_j|B_v)$ is the probability of belonging to class C_j when given branch B_v for $v = 1, \ldots, s$. A branch B with k nodes is defined as:

$$B = \langle F_1, \ldots, F_k\rangle$$

where $k \leq n$ and $F_j \in \mathcal{F}_j$. $\mathcal{F}_j$ is the set of focal elements for attribute j (see Definition 5).

Within a LDT, each branch has an associated probability distribution on the classes. For example the branch $\langle\langle\{large_3\}, \{small_2, large_2\}\rangle, 0.6, 0.3, 0.1\rangle$ means the probability distribution on classes C_1, C_2 and C_3 is $0.6 : 0.3 : 0.1$ when given attribute 3 that can be only described as *large* and attribute 2 can be described as *small* and *large* (attribute 1 does not appear in this branch). We need to be aware that the linguistic expressions such as *small*, *medium* or *large* for each attribute are not necessarily the same, since they are defined independently on each attribute.

3.1. Linguistic decision trees for classification

According to the definition of LDT, if given a branch of a LDT in the form of $B = \langle F_1, \ldots, F_k\rangle$. The probability of class C_j ($j = 1, \ldots, |\mathcal{C}|$) given B can then be evaluated from a given training set $\mathcal{D}$ as follows. First, we consider the probability of a branch B given a particular example $x \in \mathcal{D}$, where $x = \langle x_1, \ldots, x_n\rangle \in \Omega_1 \times \cdots \times \Omega_n$.

$$P(B|x) = \prod_{r=1}^{k} m_{x_r}(F_r) \tag{3}$$

$m_{x_r}(F_r)$ are the associated masses of data element x_r for $r = 1, \ldots, k$. The probability of class C_j given B can then be evaluated by:

$$P(C_j|B) = \frac{\sum_{i \in \mathcal{D}_j} P(B|x_i)}{\sum_{i \in \mathcal{D}} P(B|x_i)} \tag{4}$$

where $\mathcal{D}_j$ is the subset consisting of instances which belong to class j. In the case of $\sum_{i \in \mathcal{D}} P(B|x_i) = 0$, which can 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.

$$P(C_j|B) = \frac{1}{|\mathcal{C}|} \quad \text{for } j = 1, \ldots, |\mathcal{C}| \tag{5}$$

Now consider classifying an unlabeled instance in the form of $x = \langle x_1, \ldots, x_n\rangle$ which may not be contained in the training data set $\mathcal{D}$. First we apply linguistic translation to x based on the fuzzy covering of the training data $\mathcal{D}$. In the case that a data element appears beyond the range of training data set $[R_{min}, R_{max}]$ for a particular attribute, we assign the appropriateness degrees of R_{min} or R_{max} to the element depending on which side of the range it appears. Then, according to the Jeffrey's rule [9] the probability of class C_j given a LDT with s branches are evaluated as follows:

$$P(C_j|x) = \sum_{v=1}^{s} P(C_j|B_v)P(B_v|x) \tag{6}$$

where $P(C_j|B_v)$ and $P(B_v|x)$ are evaluated based on Eqs. (3) and (4) (or (5)), respectively. In classical decision trees, classification is made according to the class label of the branch in which the data falls. In our approach, the data for classification partially satisfies the constraints represented by a number of branches and the probability estimates across the whole decision tree are then used to obtain an overall classification. More details can be found in [15].

3.2. Linguistic decision tree for prediction

Consider a database for prediction $\mathcal{D} = \{\langle x_1(i), \ldots, x_n(i), x_t(i)\rangle \,|\, i = 1, \ldots, N\}$ where $x_1, \ldots, x_n$ 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 (x_t as well) universe and assume the set of focal elements are $\mathcal{F}_1, \ldots, \mathcal{F}_n$ and $\mathcal{F}_t$. For the target attribute x_t: $\mathcal{F}_t = \{F_t^1, \ldots, F_t^{|\mathcal{F}_t|}\}$, we can consider each focal element of target attributes as class labels. Hence, the LDT model for prediction has the following form:

$$LDT = \{\langle B_1, P(F_t^1|B_1), \ldots, P(F_t^{|\mathcal{F}_t|}|B_1)\rangle, \ldots, \langle B_s, P(F_t^1|B_s), \ldots, P(F_t^{|\mathcal{F}_t|})|B_s\rangle\}$$

Intuitively we may like to view the target focal elements as imprecise class labels. The essential difference is that, these "classes" overlap each other and this must be taken into account when evaluating branch probabilities. At the training stage, for a particular instance $x_i \in \Omega_1 \times \cdots \times \Omega_n$, where $x_i \to x_t(i)$ (i.e., $x_t(i)$ is predicted value for the instance x_i) for $i = 1, \ldots, N$, there may be several corresponding target focal elements rather than just one. The degree to which x_i belonging to a particular target focal element F_t^j is measured by ξ_i^j as follows:

$$\xi_i^j = m_{x_t(i)}(F_t^j) \tag{7}$$

where $j = 1, \ldots, |\mathcal{F}_t|$. From Eq. (7), we can see that ξ_i^j is just the associated mass of F_t^j given $x_t(i)$. Hence, we can write the corresponding target focal elements with a membership for x_i are as follows:

$$x_i \to \langle F_t^1 : \xi_i^1, \ldots, F_t^{|\mathcal{F}_t|} : \xi_i^{|\mathcal{F}_t|}\rangle \tag{8}$$

However, since we have made an assumption of 50% overlapping on fuzzy sets, so, at most two of the values $\{\xi^1, \ldots, \xi^{|\mathcal{F}_t|}\}$ are non-zero. We can also view ξ as an indicator: if and only if $\xi_i^j > 0$, F_t^j is one of the corresponding target focal elements for the data element x_i, otherwise, it is not. Based on Eq. 4, the probability of F_t^j given B is evaluated as follows:

$$P(F_t^j|B) = \frac{\sum_{i \in \mathcal{D}} \xi_i^j P(B|x_i)}{\sum_{i \in \mathcal{D}} P(B|x_i)} \tag{9}$$

where $F_t^j \in \mathcal{F}_t$. Eq. (9) is a general version of Eq. (4). In classification problems, the target labels are discreet, thus ξ is either 0 or 1. So that

$$\sum_{i \in \mathcal{D}_j} P(B|x_i) = \sum_{i \in \mathcal{D}} \xi_i^j P(B|x_i)$$

in classification problems. Example 1 shows how to calculate these probabilities. Similarly in case of $\sum_{i \in \mathcal{D}} P(B|x_i) = 0$, we use the following equation:

$$P(F_t^j|B) = \frac{1}{|\mathcal{F}_t|} \quad \text{for } j = 1, \ldots, |\mathcal{F}_t| \tag{10}$$

Based on Eq. 6, we can obtain the probabilities of target focal elements given a data element $x \in \Omega \times \cdots \times \Omega_n$ based on a LDT with s consisting branches according to the Jeffrey's rule [9]:

$$P(F_t^j|x) = \sum_{v=1}^{s} P(F_t^j|B_v)P(B_v|x) \tag{11}$$

Example 1. Consider a problem with 2 potential explanatory attributes x_1, x_2 and one target attribute x_t, where $LA_1 = \{small_1(s_1),\ large_1(l_1)\}$, $LA_2 = \{small_2(s_2),\ large_2(l_2)\}$ and $LA_t = \{small_t(s_t), large_t(l_t)\}$. We assume the focal elements defined on the attributes are $\mathcal{F}_1 = \{\{s_1\}, \{s_1, l_1\}, \{l_1\}\}$, $\mathcal{F}_2 = \{\{s_2\}, \{s_2, l_2\}, \{l_2\}\}$ and $\mathcal{F}_t = \{\{s_t\}, \{s_t, l_t\}, \{l_t\}\}$. The training database obtained by applying linguistic translation is shown in Table 1. If we are given a branch of the form:

$$B = \langle\langle\{s_1\}, \{s_2\}\rangle, P(\{s_t\}|B), P(\{s_t, l_t\}|B), P(\{l_t|B\})\rangle$$

The probabilities of target focal elements are evaluated according to Eqs. (3), (7), (9) and (10) as follows:

$$\begin{aligned}
P(\{s_t\}|B) &= \frac{\sum_{i=1}^{5} m_{x_t(i)}(\{s_t\}) \prod_{r=1,2} m_{x_r(i)}(F_r)}{\sum_{i=1}^{5} \prod_{r=1,2} m_{x_r(i)}(F_r)} \\
&= \frac{\sum_{i=1,4,5} m_{x_1(i)}(\{s_1\}) \times m_{x_2(i)}(\{s_2\}) \times m_{x_t(i)}(\{s_t\})}{\sum_{i=1}^{5} m_{x_1(i)}(\{s_1\}) \times m_{x_2(i)}(\{s_2\})} \\
&= \frac{0.4 \times 0 \times 0.9 + 0.3 \times 1 \times 0.7 + 0 \times 0.3 \times 1}{0.4 \times 0 + 0.2 \times 0.5 + 0 \times 1 + 0.3 \times 1 + 0 \times 0.3} = 0.525 \\
P(\{s_t, l_t\}, B) &= \frac{\sum_{i=1}^{5} m_{x_t}(\{s_t, l_t\}) \prod_{r=1,2} m_{x_r(i)}(F_r)}{\sum_{i=1}^{5} \prod_{r=1,2} m_{x_r(i)}(F_r)} \\
&= \frac{\sum_{i=1,2,3,4} m_{x_1(i)}(\{s_1\}) \times m_{x_2(i)}(\{s_2\}) \times m_{x_t(i)}(\{s_t, l_t\})}{\sum_{i=1}^{5} m_{x_1(i)}(\{s_1\}) \times m_{x_2(i)}(\{s_2\})} \\
&= \frac{0.4 \times 0 \times 0.1 + 0.2 \times 0.5 \times 0.8 + 0 \times 1 \times 1 + 0.3 \times 1 \times 0.3}{0.4 \times 0 + 0.2 \times 0.5 + 0 \times 1 + 0.3 \times 1 + 0 \times 0.3} = 0.425 \\
P(\{l_t\}, B) &= \frac{\sum_{i=1}^{5} m_{x_t}(\{l_t\}) \prod_{r=1,2} m_{x_r(i)}(F_r)}{\sum_{i=1}^{5} \prod_{r=1,2} m_{x_r(i)}(F_r)} \\
&= \frac{\sum_{i=2} m_{x_1(i)}(\{s_1\}) \times m_{x_2(i)}(\{s_2\}) \times m_{x_t(i)}(\{l_t\})}{\sum_{i=1}^{5} m_{x_1(i)}(\{s_1\}) \times m_{x_2(i)}(\{s_2\})} \\
&= \frac{0.2 \times 0.5 \times 0.2}{0.4 \times 0 + 0.2 \times 0.5 + 0 \times 1 + 0.3 \times 1 + 0 \times 0.3} = 0.05
\end{aligned}$$

3.3. Defuzzification

As discussed in the last section, for a given value $x = \langle x_1, \ldots, x_n\rangle$ to predict its target value $\hat{x}_t$ (i.e. $x_i \to \hat{x}_t$). We can first a series of probabilities on target focal elements: $P(F_t^1|x), \ldots, P(F_t^{|\mathcal{F}_t|}|x)$. We then take the estimate of x_t, denoted $\hat{x}_t$, to be the expected value:

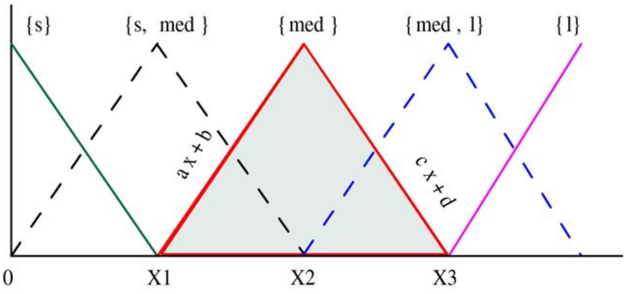

Fig. 2. Illustration of calculating the expected value for focal elements.

$$\hat{x}_t = \int_{\Omega_t} x_t\, p(x_t|x)\, dx_t \tag{12}$$

where

$$p(x_t|x) = \sum_{j=1}^{|\mathcal{F}_t|} p(x_t|F_t^j)\, P(F_t^j|x) \tag{13}$$

and

$$p(x_t|F_t^j) = \frac{m_{x_t}(F_t^j)}{\int_{\Omega_t} m_{x_t}(F_t^j)\, dx_t} \tag{14}$$

so that, we can obtain:

$$\hat{x}_t = \sum_j P(F_t^j|x)\, E(x_t|F_t^j) \tag{15}$$

where

$$E(x_t|F_t^j) = \int_{\Omega_t} x_t\, p(x_t|F_t^j)\, dx_t = \frac{\int_{\Omega_t} x_t\, m_{x_t}(F_t^j)\, dx_t}{\int_{\Omega_t} m_{x_t}(F_t^j)\, dx_t} \tag{16}$$

In practice the calculation of Eq. (16) can be illustrated by the following example.

Example 2. Suppose that the output space x_t is partitioned with a set of class labels $LA_t = \{small(s), medium(med), large(l)\}$. From this we can obtain mass assignment values across the focal sets of LA_t. For example, suppose the $m_x(\{med\})$ is defined by

$$f(x) = \begin{cases} ax + b & X_1 \le x < X_2 \\ cx + d & X_2 \le x < X_3 \end{cases} \tag{17}$$

The expected value for the focal element $\{med\}$ is evaluated as follows:

$$E(x_t|\{med\}) = \frac{f(x)}{A} \tag{18}$$

where A is the area which covered by $f(x)$ which is represented by the dark triangle. The area of the triangle can be obtained by multiplying the base and one-half the height. Here the height is 1 so that $A = (X_3 - X_1)/2$. $f(x)$ is the function of $m_x(\{med\})$ (see Fig. 2):

$$\begin{aligned}
f(x) &= \int_{X_1}^{X_2} x(ax + b) + \int_{X_2}^{X_3} x(cx + d) \\
&= \left[\frac{ax^3}{3} + \frac{bx^2}{2}\right]_{X_1}^{X_2} + \left[\frac{cx^3}{3} + \frac{dx^2}{2}\right]_{X_2}^{X_3} \\
&= X_2^3\left(\frac{a}{3} - \frac{c}{3}\right) + X_2^2\left(\frac{b}{2} - \frac{d}{2}\right) - X_1^3\frac{a}{3} - X_1^2\frac{b}{2} + X_3^3\frac{c}{3} + X_3^2\frac{d}{2}
\end{aligned}$$

Table 1
A small-scale training linguistic database for prediction, where the values are the associated masses for the corresponding focal elements on 5 given data elements.

#	Attribute 1 (x_1)			Attribute 2 (x_2)			Target attribute (x_t)		
	$\{s_1\}$	$\{s_1, l_1\}$	$\{l_1\}$	$\{s_2\}$	$\{s_2, l_2\}$	$\{l_2\}$	$\{s_t\}$	$\{s_t, l_t\}$	$\{l_t\}$
1	0.4	0.6	0	0	0.7	0.3	0.9	0.1	0
2	0.2	0.8	0	0.5	0.5	0	0	0.8	0.2
3	0	0.9	0.1	1	0	0	0	1	0
4	0.3	0.7	0	1	0	0	0.7	0.3	0
5	0	0.2	0.8	0.3	0.7	0	1	0	0

3.4. Linguistic ID3 algorithm

Linguistic ID3 (LID3) is the learning algorithm for building the linguistic decision trees from a given training database. Similar to the ID3 algorithm [18], 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 the ID3.

Definition 6 *(Branch entropy).* The entropy of branch B is given by

$$BE(B) = -\sum_{j=1}^{|\mathcal{F}_t|} P(F_t^j|B)\log_2(P(F_t^j|B)) \tag{19}$$

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 7 *(Expected entropy).*

$$EE(B, x_j) = \sum_{F_j \in \mathcal{F}_j} BE(B \cup F_j) \cdot P(F_j|B) \tag{20}$$

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:

$$P(F_j|B) = \frac{\sum_{i \in \mathcal{D}}(B \cup F_j|x_i)}{\sum_{i \in \mathcal{D}}(B|x_i)} \tag{21}$$

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

$$IG(B, x_j) = BE(B) - EE(B, x_j) \tag{22}$$

The pseudo-code is given in Fig. 3, where $\mathcal{LD}$ is the training data after linguistic translation. 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. To build a LDT, 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 a branch, the attributes which has not appeared in this branch are referred to as *free attributes*. To expand a particular branch, the free attribute with maximum information gain will be appended as the next node, from level to level, until the tree reaches the maximum specified depth or some other criteria are met.

3.5. Forward branch merging

One of the inherent disadvantages for tree induction algorithms is overfitting. There are many pruning algorithms were proposed, a good review are given in [14]. Here we present a different approach of using 'merging' instead of 'pruning' to generate compact trees. In this section, a branch merging algorithm for the LDT model is discussed. The basic idea is that, we employ breadth-first search in developing a LDT, at each depth, the adjacent branches which give similar probabilities on target focal elements are merged into one branch according to a *merging threshold*:

Definition 8 *(Merging threshold).* In a linguistic decision tree, if the maximum difference between the probabilities of target focal elements on 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

$$T_m \geq \max_{F_t \in \mathcal{F}_t}(|Pr(F_t|B_1) - Pr(F_t|B_2)|) \tag{23}$$

where $\mathcal{F}_t = \{F_t^1, \ldots, F_t^{|\mathcal{F}_t|}\}$ are focal elements for the target attribute, then B_1 and B_2 can be merged into one branch MB.

Definition 9 *(Merged branch).* A merged branch MB with k nodes is defined as

$$MB = \langle \mathcal{M}_1, \ldots, \mathcal{M}_k \rangle$$

where $\mathcal{M}_j = \{F_j^1, \ldots, F_j^w\}$ is a set of focal elements such that F_j^i is adjacent to F_j^{i+1} for $i = 1, \ldots, w-1$. The associate mass for $\mathcal{M}_j$ is given by

$$m_x(\mathcal{M}_j) = \sum_{i=1}^{w} m_x(F_j^i) \tag{24}$$

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

Where 'adjacent' means the fuzzy labels which are defined next to each other in a natural order. For the example shown in Fig. 1, {small} and {small,medium} are adjacent focal elements while

input : $\mathcal{LD}$: Linguistic dataset
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;
  $\mathcal{B}_{l+1} \leftarrow \bigcup_{r=1}^{s} \mathcal{B}'_r$.
$LDT = \mathcal{B}$

Fig. 3. Linguistic ID3 algorithm.

{small} and {medium} are not. The probability of a merged branch given a data element $x \in \Omega \times \cdots \times \Omega$ can be evaluated by

$$P(MB|x) = \prod_{r=1}^{k} m_{x_r}(\mathcal{M}_r) = \prod_{r=1}^{k} \left(\sum_{i=1}^{w_r} m_{x_r}(F_r^i) \right) \quad (25)$$

where k is the length of the merged branch MB and w_r for $r = 1, \ldots, k$ is the number of merged nodes of the attribute r for $r = 1, \ldots, s$. Based on Eqs. (4), (5), (7), (24) and (25) we use the following equation to evaluate the probabilities on target focal elements given a merged branch.

$$P(F_t^j|MB) = \frac{\sum_{i \in \mathcal{D}} \xi_i^j P(MB|x)}{\sum_{i \in \mathcal{D}} P(MB|x)} \quad (26)$$

And, the following equation is used when doing classification with a merged LDT with s branches:

$$P(F_t^j|x) = \sum_{v=1}^{s} P(F_t^j|MB_v) P(MB_v|x) \quad (27)$$

When the merging algorithm is applied in learning a linguistic decision tree, the adjacent branches meeting the merging criteria will be merged and re-evaluated according to Eq. (26). Then the adjacent branches after the first round of merging will be examined in a further round of merging, until all adjacent branches cannot be merged further. We then proceed to the next depth. The merging is applied as the tree develops from the root to the maximum depth and hence is referred to as *forward merging*.

4. Experimental studies

In this section, several benchmark prediction problems are tested with the LID3 algorithm. The prediction results obtained are compared with several the state-of-art prediction algorithms such as Support Vector Regression system (SVR), Fuzzy Naive Bayes and Fuzzy Semi-Naive Bayes (FSNB) [20]. In this paper we use ε-Support Vector Regression system (ε-SVR) with a Gaussian kernel and an ε-insensitive loss function [22]. The SVR results present here are obtained by using a Matlab toolbox for SVM implemented by Gunn [5] and the parameter settings for each problem are based on empirical research on these problems by Randon [20]. Fuzzy Naive Bayes is another linguistic model based on label semantics and Fuzzy Semi-Naive Bayes presented here is modified from Fuzzy Naive Bayes by weaken the independence assumption of Naive Bayes (more details are available in [20]). The results of Fuzzy Naive Bayes and FSNB presented in his paper is the best results so far from a set of systematic research.

The measure defined here for evaluating the prediction performance is *Average Error* (*AVE*), which scales the error according to range of output (target attribute) space is used for evaluating algorithms' performance: Given output universe defined by $\Omega_t = [a, b]$ and a training set $\mathcal{D}$, *AVE* is the average modulus error taken as a percentage of the length of the output universe, formally:

$$AVE = \frac{\sum_{i \in \mathcal{D}} |\hat{x}_t(i) - x_t(i)|}{|\mathcal{D}|(b-a)} \quad (28)$$

where $|\mathcal{D}|$ represents the number of instances in $\mathcal{D}$. The standard deviation across $\mathcal{D}$ is given by

$$\sigma_E = \sqrt{\frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} (\epsilon_i - AVE)^2} \quad (29)$$

where

$$\epsilon_i = \frac{|\hat{x}_t - x_t|}{b-a}$$

4.1. 3-D surface regression

In this example, 529 points were *uniformly* generated describing a surface defined by equation $z = sin(x \times y)$ where $x, y \in [0, 3]$ as shown on the left-hand of Fig. 4. 2209 points are sampled uniformly as the test set. The attributes are discretized uniformly by fuzzy labels, the results in the *AVE* measure with different number of fuzzy labels which are respectively defined on input and output space are listed in Table 2.

It is surprising to see that the number of fuzzy sets used for output (i.e. z) space does not cause a great difference in error. On the contrary, the number of fuzzy sets for inputs (i.e. x and y) is really matter. More fuzzy sets used for discretization, more accurate prediction surface we can obtain. Fig. 5 shows the predicted surfaces and the error surfaces, where input space are discretized with 6 fuzzy sets (left-hand column) and 7 fuzzy sets (right-hand column), respectively.

We now compare these results to those obtained from the ε-SVR with the following parameters: $\sigma = 1$, $\varepsilon = 0.05$, $C = \infty$ (the reasons for this parameter setting are in [21]). The test errors are shown in Table 3, compare to ε-SVR, LID3 is a slightly worse. As we can see from the right-hand side of Fig. 4, ε-SVR has a very good approximation to the original surface. By comparing Figs. 4 and 5, we can see that LID3 cannot accurately capture the small 'tail' on the left, while the ε-SVR can. Table 3 also shows the results of fuzzy Naive Bayes and Fuzzy Semi-Naive Bayes, among them, LID3 (7 fuzzy labels for the input and 6 labels for the output) is the second best. For such a function regression problems, higher accuracy could be obtained by increasing the number of fuzzy labels discretized for the input space. However, the computing complexity will be increased extensively with the increasing of the number of fuzzy labels. For all problems discussed in this paper, we only expected to obtain equivalent accuracy but better transparency comparing to other models.

4.2. Predicting the age of Abalone and Boston Housing problem

These two problems are taken from the UCI repository [2]. The Abalone concerns the problem of predicting the age of abalone from physical measurements. Abalones are a type of shellfish, the age of which is accurately determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope, which is a laborious and time consuming task. Boston Housing problem contains data on housing values in the suburbs of Boston, USA. The data set contains 506 instances and 13 continuous attributes (including the target attribute) and one binary attribute.

In our experiments, the instances for each data set are randomly split into two parts with approximately same number of instances, one for training and the other for test. This is referred to as 50–50 split experiments. The test errors from 10 runs of 50–50 split experiments on the two data sets are shown in Table 4 where the results obtained for the Abalone prediction test set by applying ε-SVR with a Gaussian RBF kernel with parameters: $\sigma = 1$, $\varepsilon = 0.05$ and $C = 5$. The results of LID3 are obtained from the LDTs that discretized with 3 uniformly distributed fuzzy labels at the depth 5. For Boston Housing problem the ε-SVR parameters are: $\sigma = 3$, $\varepsilon = 0.05$ and $C = 10$. The LID3 results are obtained by the LDTs with 5 uniformly distributed fuzzy labels at the depth 3. The standard deviation (Std) in Table 4 is the standard deviation of *AVE* across the experiments.

From Table 4, we can see that ε-SVR has be best performance on these two data sets. LID3 is the second best in Abalone prediction.

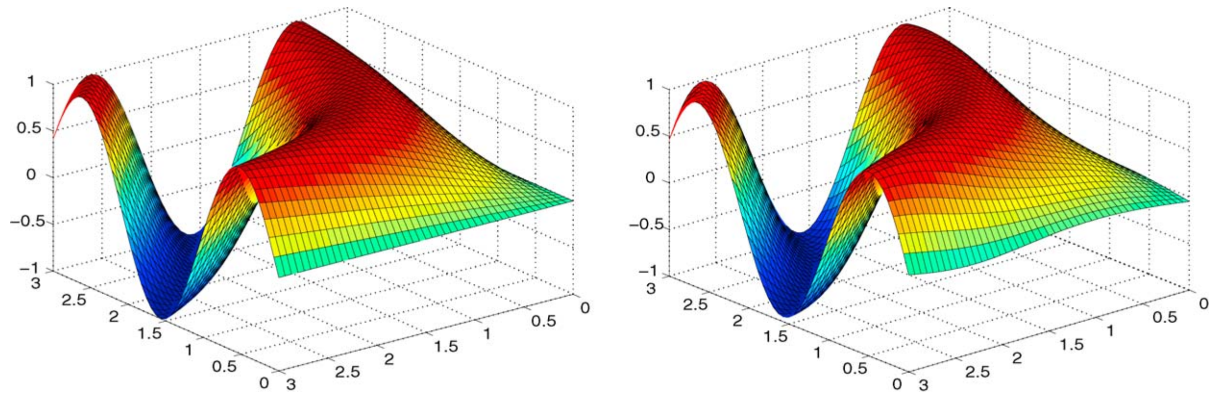

Fig. 4. Left-hand figure: the original surface of $z = sin(x \times y)$. Right-hand figure: the prediction surface by ε-SVR with a Gaussian RBF kernel.

Table 2
Average error for the $sin(x \times y)$ problem with different number of fuzzy sets (represented by N_F) for discretization on input and output space, respectively.

	The number of fuzzy sets (N_F) for output							
	Training error				Test error			
Input	4	5	6	7	4	5	6	7
N_F								
4	7.4290	7.4296	7.4254	7.4419	7.1827	7.1834	7.1785	7.1955
5	4.8314	4.8316	4.8262	4.8456	4.6772	4.6777	4.6695	4.6892
6	3.2266	3.2265	3.2160	3.2357	3.1890	3.1895	3.1776	3.1986
7	2.1734	2.1711	2.1653	2.1864	2.1560	2.1555	2.1464	2.1684

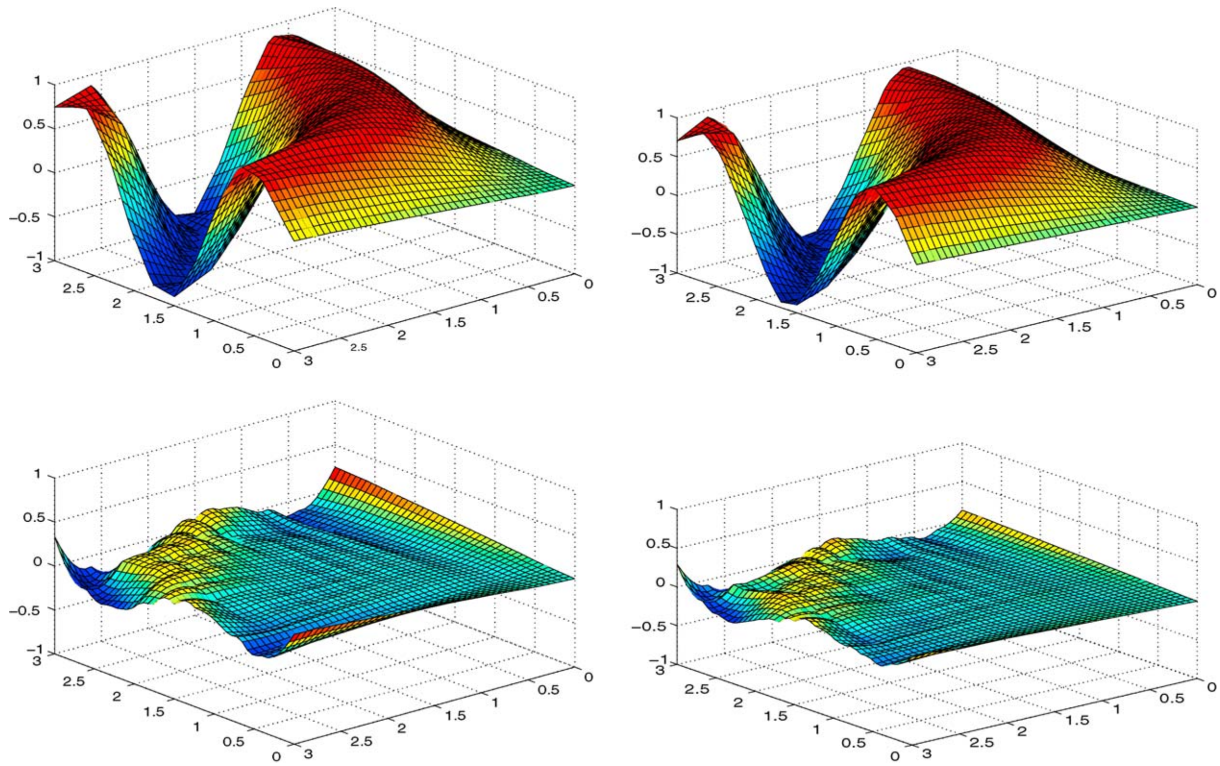

Fig. 5. Prediction surfaces (upper figures) and error surfaces (lower figures) where input spaces are discretized by 6 fuzzy sets (left-hand column) and 7 fuzzy sets (right-hand column), respectively.

Table 3
Comparisons of prediction models in average error on the $sin(x \times y)$ problem.

	Fuzzy Naive Bayes	FSNB	ε-SVR	LID3
AVE $\pm\, \sigma_E$	16.042 ± 12.817	2.815 ± 2.268	1.452 ± 0.746	2.146 ± 1.795

Table 4
Prediction results in *AVE* from 10 runs 50–50 split experiments on the Abalone prediction and the Boston Housing prediction problem, respectively.

Prediction model	Abalone			Boston Housing		
	AVE %	σ_E (%)	Std	AVE %	σ_E (%)	Std
Fuzzy Naive Bayes	7.9660	7.2010	0.6638	8.2437	9.0864	0.5034
FSNB	7.0141	6.9277	0.5225	7.7059	8.9876	0.5766
ε-SVR	5.6921	6.0034	0.0894	5.4508	6.7989	0.3874
LID3	6.4327	6.3247	0.3145	8.2022	8.1502	0.4579

But, it does not perform very well in Boston Housing problem where LID3 gives the equivalent average errors to Fuzzy Naive Bayes.

4.3. Prediction of sunspots

This problem is taken from the Time Series Data Library [7] and contains data of sunspot numbers between the years 1700 and 1979. For this experiment the data was organized as described in [27] using a sliding window and the validation set of 35 examples (1921–1955) was merged into the test set of 24 examples (1956–1979). This is because a validation set is not required in this framework. Hence, a training set of 209 examples (1712–1920) and a test set of 59 examples (1921–1979) are used in this paper. 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, ε-SVR, Fuzzy Naive Bayes [21] and Fuzzy Semi-Naive Bayes in the *AVE* measure are shown in Table 5, where the parameter setting for ε-SVR is as follows: $\sigma=3$, $\varepsilon=0.05$, $C=5$ and the results for FSNB are the best results from a range of FSNB parameter settings [21]. Results of LID3 present here are obtained from LDTs discretized by 4 fuzzy labels by percentile-based method (both on input and output spaces) and at the depth of 5. The comparison between the prediction data and the original data are shown in Fig. 6, where the data on the left (1712–1921) are for training data and the right are (1921–1979) for test.

Table 5 also shows the results of LID3 by applying forward branch merging where the merging threshold varies from 0.05 to 0.30. From the table, we can see that ε-SVR gives the best results and the LID3 gives the second best. If we increase the merging threshold T_m, the size of LDT (i.e. the number of branches) is reduced greatly while the error rate only changes slightly. For example, compare $T_m=0$ (no merging) and $T_m=0.25$, the tree reduced about 98.6% in size and the error rate only increases 1.91%. Fig. 7 shows the scatter plot of the actual sunspot number against the predicted number on 59 test data by using Fuzzy Naive Bayes, ε-SVR, non-merged LDT and merged LDT with $T_m=0.25$. 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.

4.4. Flood forecasting

In this section, a flood forecasting problem is investigated. We attempt to model the stream flow characteristics of a river. The database we shall investigate here describes the Bird Creek river basin in Oklahoma, USA. The data was collected to form part of a real-time hydrological model inter-comparison exercise conducted in Vancouver, Canada in 1987 and reported by World Meteorological Organization (WMO) in 1992. The database describing the Bird Creek catchment area gives information on two attributes: the average rainfall (U) given in mm, derived from 12 rainfall gauges situated in or near the catchment area and the river's stream flow (Y) given in m^3/s, measured using a continuous stage recorder. Both values are recorded in the database at 6-h intervals. In this paper only a subset of the original flood data is used. This is comprised of 2090 training examples and 1030 examples for test.

Flood forecasting is a typical problem of prediction and several models had been developed based on the Bird-Creek data. By using windowing techniques, Clukie and Han [4] extensively developed the Weather Radar Information Processor System (WRIP) [6]. A Fuzzy Semi-Naive Bayes model is also used to study this problem

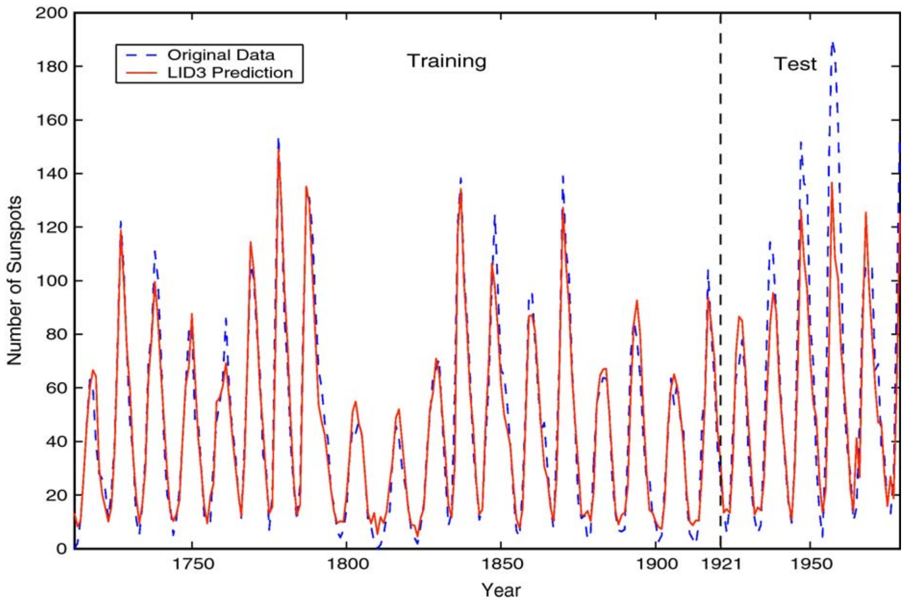

Fig. 6. The prediction results obtained from LID3 without merging, where the data on the left (1712–1921) are for training and the right (1921–1079) are for test.

Table 5
Prediction results in *AVE* on the sunspot prediction problem.

Prediction model	AVE %		σ_E (%)		Tree size
	Training	Test	Training	Test	LDT only
Fuzzy Naive Bayes	9.5514	13.0588	10.7682	13.0213	–
FSNB	5.1301	10.9064	5.4943	9.5208	–
ε-SVR	5.6988	8.9337	5.8328	9.7766	–
LID3	3.7557	8.6793	3.1859	8.8876	5731
LID3 (T_m = 0.05)	3.9146	8.8925	3.3100	8.9437	2285
LID3 (T_m = 0.10)	4.1259	8.9649	3.5013	9.1994	1493
LID3 (T_m = 0.15)	4.9315	9.8419	4.3850	10.1869	757
LID3 (T_m = 0.20)	5.9327	9.8341	5.1525	10.7063	204
LID3 (T_m = 0.25)	7.2166	10.5858	5.9409	10.3711	81
LID3 (T_m = 0.30)	14.0175	18.9539	12.4700	19.1159	5

by Randon [21] with and without windowing techniques. In order to make direct comparisons with other river flow modelling techniques we shall initially use the same training and test data as in previous studies. In this paper, windowing technique is not used. The rainfall values, $\langle U_{T-2}, U_{T-2}, U_T \rangle$ and stream flow value $\langle Y_{T-2}, Y_{T-1}, Y_T \rangle$ are used to produce six steps ahead prediction on stream flow value $\hat{Y}_{T+6}$. The results obtained from LID3 are compared with the results of Fuzzy Semi-Naive Bayes and ε-SVR. The results in terms of average errors are shown in Table 6, where the results of ε-SVR are based on parameters: $\sigma = 3$, $\varepsilon = 0.05$ and $C = 5$. The LID3 results are obtained based on the linguistic translation by which each attribute is discretized uniformly by 3 fuzzy labels and the LDT extends with the maximum depth 6.

As we can see from Table 6, LID3 outperforms the other models on this problem. However, the size of the LDT is still be very large (2133 branches without merging). By applying forward merging, the errors increase only slightly while the number of branches are significantly reduced. With $T_m = 0.30$, the LID3 still gives better

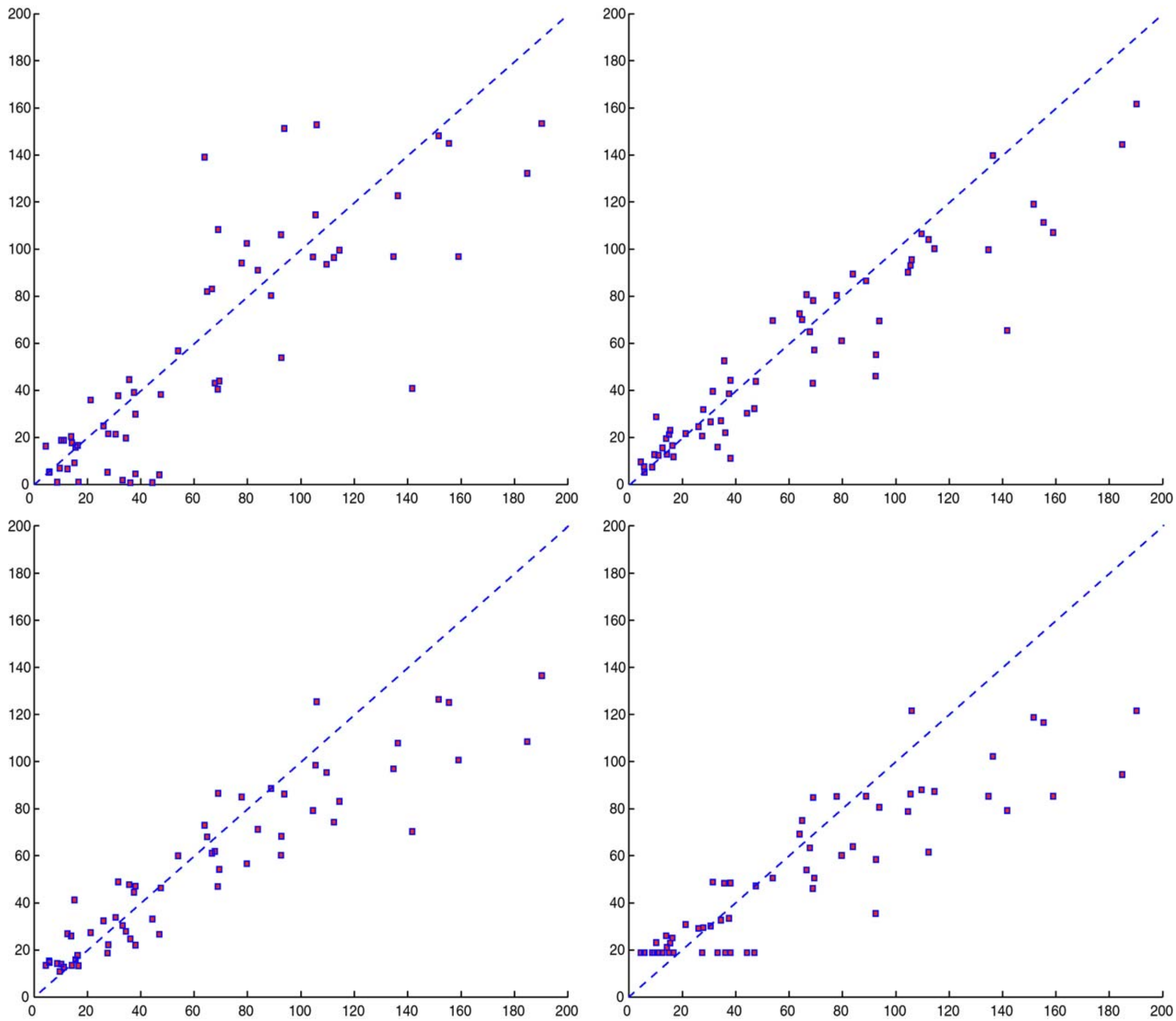

Fig. 7. Scatter plot showing original data verses prediction data on sunspot prediction problems. Upper left: Fuzzy Naive Bayes; upper right: SVR; lower left: non-merged LDT; lower right: merged LDT with $T_m = 0.25$.

Table 6
Average errors with standard deviations on test set of the flood forecasting problem.

Prediction model	AVE %	σ_E (%)	Tree size
Fuzzy Naive Bayes	2.9922	7.3017	–
FSNB	2.9219	7.1798	–
ε-SVR	3.3555	7.6602	–
LID3	2.5625	6.9160	2133
LID3 (T_m = 0.05)	2.5596	6.8865	815
LID3 (T_m = 0.10)	2.5576	6.1244	652
LID3 (T_m = 0.15)	2.6523	6.9574	389
LID3 (T_m = 0.20)	2.7932	6.9225	225
LID3 (T_m = 0.25)	2.7935	6.9258	203
LID3 (T_m = 0.30)	2.8227	7.0835	118
LID3 (T_m = 0.35)	2.9368	7.5019	79
LID3 (T_m = 0.40)	2.9769	7.7628	37

accuracy to Fuzzy Semi-Naive Bayes. However, the tree has only 108 branches and comparing to LID3 without merging, the tree size has been reduced nearly 94%. The performance on the test set can be seen from Fig. 8. Although LID3 over-estimates at some peaks, it still captures the original data well.

5. Linguistic query evaluation

For many practical applications it is not sufficient that a data model only provides information regarding classifications or predictions. Often we are interested in using our model to infer relationships and test hypothesis. Here in this section, a methodology for evaluating linguistic queries using linguistic decision trees within the label semantics framework is proposed. The linguistic decision trees can be represented in label expressions in the form of a vector $\vec{\theta} = \langle \theta_1, \ldots, \theta_n \rangle$. θ is linguistic expression of labels which are joining by logical connectives, for example, $\theta = (small_1 \vee medium_2) \wedge \neg large_3$.

Definition 10 *(Label expressions).* The set of label expressions of *LA*, LE, is defined recursively as follows:

(i) $\forall i\ L_i \in LE$

(ii) If $\theta, \varphi \in LE$ then $\neg\varphi, \theta\wedge\varphi, \theta\vee\varphi, \theta\rightarrow\varphi \in LE$

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. If we consider label expressions formed from *LA* by recursive application of the connectives then an expression θ identifies a set of *possible label sets* according to the λ-function.

Definition 11 *(λ-function).* Let θ and φ be expressions generated by recursive application of the connectives ¬, ∨, ∧ and → to the elements of *LA*. Then the set of possible label sets defined by a linguistic expression can be determined recursively as follows:

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

Intuitively, λ(θ) corresponds to those subsets of $\mathcal{F}$ identified as being possible values of D_x by expression θ. In this sense the

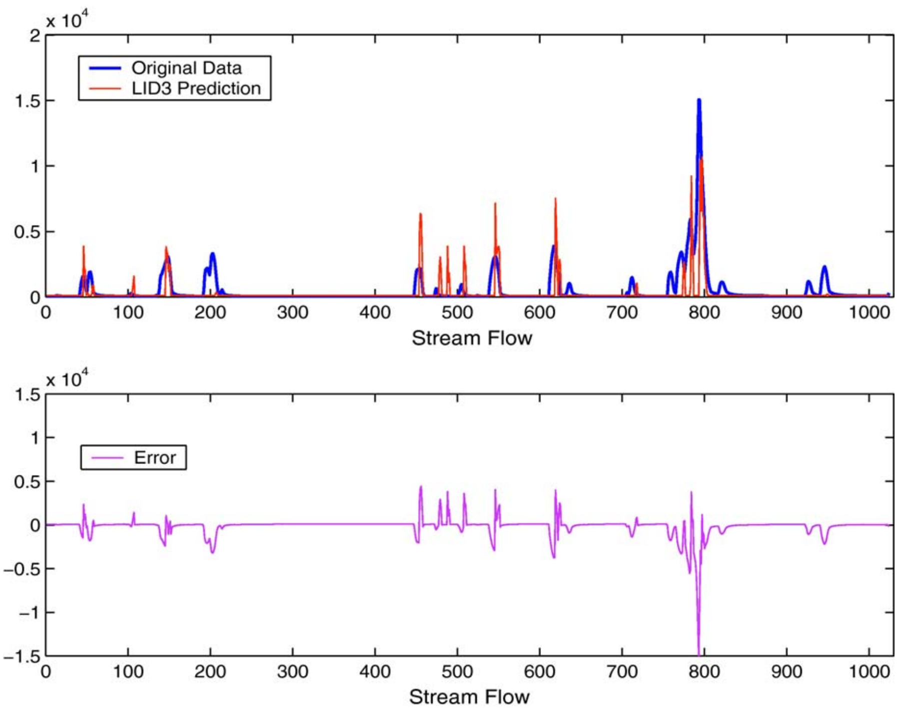

Fig. 8. The stream flow prediction with a merged LDT with T_m = 0.3.

imprecise linguistic restriction 'x is θ' on x corresponds to the strict constraint $D_x \in \lambda(\theta)$ on D_x [11].

Example 3. Given a variable h representing John's height and LA_h = {*veryshort*, *short*, *medium*, *tall*, *verytall*}, suppose we are told that "John is **not very tall** but it is **medium to tall**". This constraint can be interpreted as the logical expression

$$\theta_h = \neg very\ tall \wedge (medium \vee tall)$$

According to Definition 11, the possible label sets of the given linguistic constraint θ_h are

$$\lambda(\theta_h) = \lambda(\neg very\ tall \wedge (medium \vee tall))$$
$$= \{\{medium\}, \{medium, tall\}, \{tall\}\}$$

Two kinds of queries are discussed in this paper: single queries and compound queries and the evaluation methods are given as follows.

Single queries $F_t : \langle \theta_1, \ldots, \theta_n \rangle$

This represents the question: *Do elements satisfying* $\vec{\theta}$ *have a value of* x_t *with description* F_t? Consider the vector of linguistic expression $\vec{\theta} = \langle \theta_1, \ldots, \theta_n \rangle$, where θ_j is the linguistic expression on attribute j. The probability value for F_t conditional on this information using a given a linguistic decision tree can be evaluated through the following steps:

$$m_{\theta_j}(F_j) = \begin{cases} \dfrac{pm(F_j)}{\sum_{F_j \in \lambda(\theta_j)} pm(F_j)} & \text{if } F_j \in \lambda(\theta_j) \\ 0 & \text{otherwise} \end{cases} \tag{30}$$

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:

$$pm(F_j) = \int_{\Omega_j} m_x(F_j) p(x_j)\, dx_j \tag{31}$$

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

$$pm(F_j) \propto \int_{\Omega_j} m_x(F_j)\, dx_j \tag{32}$$

For example, given LA_x={small, large} and x *is small* (i.e. θ = *small*). By applying the λ function (Definition 11), we can generate the possible label sets for x, so that:

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

Suppose the prior mass assignments are

$$pm = \{small\} : 0.3, \{small, large\} : 0.2, \{large\} : 0.5$$

According to Eq. (30) we then obtain,

$$m_\theta = \{small\} : \frac{0.3}{(0.3+0.2)}, \{small, large\} : \frac{0.2}{(0.2+0.3)}$$
$$= \{small\} : 0.6, \{small, large\} : 0.4$$

Hence, $m_\theta(\{small\}) = 0.6$ and $m_\theta(\{small, large\}) = 0.4$ according to the given the linguistic constraint θ = *small*. For branch B with k nodes, the probability of B given $\vec{\theta}$ is evaluated by

$$P(B|\vec{\theta}) = \prod_{r=1}^{k} m_{\theta_r}(F_r) \tag{33}$$

and therefore, by the Jeffrey's rule [9]

$$P(F_t|\vec{\theta}) = \sum_{v=1}^{s} P(F_t|B_v) P(B_v|\vec{\theta}) \tag{34}$$

Compound queries $\theta_t : \langle \theta_1, \ldots, \theta_n \rangle$

This represents the question: *Do elements satisfying* $\vec{\theta}$ *have a value of* x_t *satisfies the linguistic expression* θ_t? Given a linguistic expression $\vec{\theta} = \langle \theta_1, \ldots, \theta_n \rangle$, where θ_j for $j = 1, \ldots, n$ is the linguistic expression on attribute j, and θ_t (the linguistic expression on the target attribute). The evaluation method for compound queries is based on the single queries.

$$P(\theta_t|\vec{\theta}) = \sum_{F_t \in \lambda(\theta_t)} P(F_t|\vec{\theta}) \tag{35}$$

Example 4. Consider the $y = sin(x \times y)$ problem, 7 fuzzy labels are defined on input attributes (i.e., x and y) and target attribute z, respectively. $LA_x = LA_y = LA_z$ = {*extremely small* (*es*), *very small* (*vs*), *small* (*s*), *medium* (*m*), *large* (*l*), *very large* (*vl*), *extremely large* (*el*)}. From this we obtain the focal elements describing each attribute: $\mathcal{F}_x = \mathcal{F}_y = \mathcal{F}_z = \{\{es, vs\}, \{vs\}, \{vs, s\}, \{s\}, \{s, m\}, \{s, m\}, \{m\}, \{m, l\}, \{m, l\}, \{l, vl\}, \{vl\}, \{vl, el\}\}$.

Suppose we are given:

$$\theta_x = \neg\ very\ small\ \wedge\ small\ \wedge\ \neg\ medium$$
$$\theta_y = \neg\ large\ \wedge\ (very\ large\ \vee\ extremely\ large)$$

Given the query for evaluation $F_z^i : \langle \theta_x, \theta_y \rangle$ for $i = 1 : |\mathcal{F}_z|$. According to the above Eqs. (30), (33) and (34), we obtain:

$$P(\{es, vs\}|\theta) = P(\{vs\}|\theta) = P(\{s\}|\theta) = P(\{s, m\}|\theta) = 0$$
$$P(\{m\}|\theta) = 0.0003, \quad P(\{m, l\}|\theta) = 0.0006, \quad F = P(\{l\}|\theta) = 0.0152,$$
$$P(\{l, vl\}|\theta) = 0.1646, \quad P(\{vl\}|\theta) = 0.2125, \quad P(\{vl, el\}|\theta) = 0.2338$$

Suppose the query for evaluation is a compound query

$$\theta_z = \neg large \wedge very\ large$$

According to the λ-function, we obtain:

$$\lambda(\theta_z) = \{\{very\ large\}, \{very\ large, extremely\ large\}\}$$

Then, according to Eq. (35) we obtain:

$$P(\theta_z|\langle \theta_x, \theta_y \rangle) = P(\{vl\}|\langle \theta_x, \theta_y \rangle) + P(\{vl, el\}|\langle \theta_x, \theta_y \rangle)$$
$$= 0.2125 + 0.2338 = 0.4463$$

The above example is also used in Section 4.1 for studying the performance of the LDT model. In this section, we emphasize its ability of supporting linguistic queries. Combining these two experiments, we can see its superiority in both accuracy and transparency. That is also the significance of using label semantics for designing data mining models. Some recent research in linguistic rule induction using label semantics also yields similar results [16].

6. Conclusions and discussions

In this paper, a tree-structured prediction model based on a framework for Modelling with Words has been described. Linguistic decision tree was proposed as a classification model for its advantages of handling uncertainties and being transparent. In this paper, the methodology of using LDT to do prediction was proposed and tested on several benchmark problems such as function regression, time series prediction and real-world forecasting applications. By empirical studies, we show that LDT model has equivalent prediction ability comparing to several state-of-art prediction model such as ε-SVR and Fuzzy Semi-Naive Bayes. A forward merging has been described to increase transparency without a great sacrifices on accuracy. Finally, we discuss the method to evaluate linguistic queries by LDT and tested on a toy problem.

We are not arguing that the LDT model is a best algorithm in terms of accuracy. Although we cannot say LDT model outperform others, we may say that LDT model has equivalent prediction performance comparing to other prediction algorithms mentioned in this paper. On the other hand, LDT model has better transparency in the following two aspects: (1) unlike other black-box prediction models, a LDT can be interpreted as a set of linguistic rules, which may provides the information how the predictions are made. (2) The high-level knowledge representation structure of the LDT model allows us to evaluate linguistic queries based on label semantics framework.

Acknowledgments

Most of the work has been done when the first author was with the University of Bristol. The first author is also funded by the Fundamental Research Funds for the Central Universities, the NCET and the China Scholar Council.

References

[1] J.F. Baldwin, T.P. Martin, B.W. Pilsworth, Fril-Fuzzy and Evidential Reasoning in Artificial Intelligence, John Wiley & Sons Inc., 1995.
[2] C. Blake, C.J. Merz, UCI Machine Learning Repository, 2000, http://www.ics.uci.edu/~mlearn/MLRepository.html.
[3] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, Wadsworth Inc., 1984.
[4] I.D. Cluckie, D. Han, Dendritic river modelling system in WRIP, in: Fifth International Symposium on Hydraulically Application of Weather Radar, Heian-Kaikan, Keyoto, Japan, November 19–22, 2001.
[5] S.R. Gunn, Support Vector Machines for Classification and Regression, Technical Report of Dept. of Electronics and Computer Science, University of Southampton, May 1998, http://www.isis.ecs.soton.ac.uk/resources/svminfo.
[6] D. Han, Weather radar information processing and real-time flood forecasting, PhD Thesis, University of Salford, 1991.
[7] R. Hyndman, M. Akram, Time Series Data Library, Monash University, http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/.
[8] C.Z. Janikow, Fuzzy decision trees: issues and methods., IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 28 (1) (1998) 1–14.
[9] R.C. Jeffrey, The Logic of Decision, Gordon & Breach Inc., New York, 1965.
[10] J. Lawry, A framework for linguistic modelling, Artificial Intelligence 155 (2004) 1–39.
[11] J. Lawry, J.W. Hall, R. Bovey, Fusion of expert and learnt knowledge in a framework of fuzzy labels, Journal of Approximate Reasoning 36 (2004) 151–198.
[12] J. Lawry, J. Shanahan, A. Ralescu, Modelling with Words: Learning, Fusion, and Reasoning Within a Formal Linguistic Representation Framework, LNAI 2873, Springer-Verlag, 2003.
[13] J. Lawry, Modelling and Reasoning with Vague Concepts, Springer, 2006.
[14] C. Olaru, L. Wehenkel, A complete fuzzy decision tree technique, Fuzzy Sets and Systems 138 (2003) 221–254.
[15] Z. Qin, J. Lawry, Z. Qin, J. Lawry, Decision tree learning with fuzzy labels, Information Sciences 172 (1–2) (2005) 91–129.
[16] Z. Qin, J. Lawry, LFOIL: linguistic rule induction in the label semantics framework, Fuzzy Sets and Systems. 159 (2008) 435–448.
[17] Z. Qin, J. Lawry, Fuzziness and performance: an empirical study with linguistic decision trees, in: IFSA World Congress, LNAI 4529, 2007, pp. 407–416.
[18] J.R. Quinlan, Induction of decision trees, Machine Learning 1 (1986) 81–106.
[19] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, 1993.
[20] N.J. Randon, J. Lawry, Linguistic Modelling Using a Semi-Naive Bayes Framework, IPMU-2002, Annecy, France, 2002.
[21] N.J. Randon, Fuzzy and Random Set Based Induction Algorithms, PhD Thesis, Dept. of Engineering Mathematics, University of Bristol, 2004.
[22] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[23] Y. Yuan, M.J. Shaw, Induction of fuzzy decision trees, Fuzzy Sets and Systems 69 (1995) 125–139.
[24] X.-Z. Wang, C.R. Dong, Improving generalization of fuzzy if-then rules by maximizing fuzzy entropy, IEEE Transactions on Fuzzy Systems 17 (3) (2009) 556–567.
[25] X.-Z. Wang, J.-H. Zhai, S.-X. Lu, Induction of multiple fuzzy decision trees based on rough set technique, Information Sciences 178 (2008) 3188–3202.
[26] X.-Z. Wang, Y. DS, E.C.C. Tsang, A comparative study on heuristic algorithms for generating fuzzy decision trees., IEEE Transactions on Systems, Man, and Cybernetics, Part B 31 (2) (2001) 215–226.
[27] A.A. Weigend, B.A. Huberman, D.E. Rumelhart, Predicting sunspots and exchange rates with connectionist networks, in: M. Casdagli, S. Eubank (Eds.), Non-linear Modelling and Forecasting, SFI Studies in the Science of Complexity, Proceedings, vol. XII, Addison-Wesley, 1992, pp. 395–432.
[28] L.A. Zadeh, Fuzzy sets, Information and Control 8 (1965) 338–353.
[29] L.A. Zadeh, Fuzzy logic = computing with words, IEEE Transaction on Fuzzy Systems 4 (2) (1996) 103–111.
[30] L.A. Zadeh, Toward a perception-based theory of probabilistic reasoning with imprecise probabilities, Journal of Statistical Planning and Inference 105 (2002) 23–64.
[31] L.A. Zadeh, Toward a generalized theory of uncertainty (GTU) - an outline, Information Sciences 172 (1–2) (2005) 1–40.