PREDICTION TREES USING LINGUISTIC MODELING *

Zengchang Qin Jonathan Lawry
A.I. Group, Department of Engineering Mathematics, University of Bristol, Bristol BS8 1TR, U.K.
Email: {z.qin, j.lawry}@bris.ac.uk

ABSTRACT: Linguistic decision tree (LDT) is a tree-structured model based on a framework for linguistic modeling [5]. In previous research [8], 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 are compared with other state-of-art prediction algorithms such as a Support Vector Regression (SVR) system and Fuzzy Semi-Naive Bayes on two real-world applications. A forward merging algorithm for LDT prediction is also discussed for generating more compact trees.

Keyword: Label semantics, linguistic decision tree, mass assignment, forward merging.

1 INTRODUCTION

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 specialize on classification problems. But in many real-world applications, data ranging from financial analysis to weather forecasting are prediction problems. Tree induction algorithms were received a great deal of attention because of their simplicity and effectiveness. From early discrete decision trees such as ID3 and C4.5 [9] to a variety types of fuzzy decision trees [4, 6, 12], 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 [2]. Here we present a tree-structured prediction model based on a high-level knowledge representation framework which is referred to as *Label Semantics* [5].

Label semantics is a random set semantics for modeling 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. It provides us a framework for modeling uncertainty with good transparency. Based on label semantics, *Linguistic Decision Tree* (LDT) model [8] 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. So, LDT model can be regarded as a probability estimation tree model based on fuzzy labels.

In this paper, the LDT classification model is extended to prediction and empirical results on two benchmark problems are presented. The results are compared with the other three prediction models: Support vector regression system [3], Fuzzy Naive Bayes and Fuzzy Semi-Naive Bayes [10].

2 LABEL SEMANTICS

Label semantics [5] is a random set framework to capture the idea of using linguistic expressions to label imprecise concepts. The underlying question posed by label semantics is how to use linguistic expressions to label numerical values. For a variable x into a domain of discourse Ω we identify a finite set of linguistic labels $LA = \{L_1, \cdots, L_n\}$ with which to label the values of x. Then for a specific value $\alpha \in \Omega$ an individual I identifies a subset of LA, denoted D_α^I to stand for the description of α given by I, as the set of words with which it is appropriate to label α. If we allow I to vary across a population V, then D_α^I will also vary and generate a random set denoted D_α into the power set of LA. The frequency of occurrence of a particular label, say S, for D_α across the population then we obtain a distribution on D_α referred to as a *mass assignment* [1] on labels, more formally:

Definition 1 (*Mass Assignment on Labels*)

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

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

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

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

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

*This research is funded by the ORS UK and UoB research scholarship. The authors thank Nick Randon for providing part of the results for comparison studies.

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 [5]. This definition provides a relationship between mass assignments and appropriateness degrees.

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 previous example, we then can obtain

$$\mu_{medium}(15) = 0.7 + 0.3 = 1, \;\; \mu_{low}(15) = 0.7$$

Based on the underlying semantics, we can translate a set of numeric data into a set of mass assignments on appropriate labels based on the reverse of definition 2 under some assumptions: consonance mapping, full fuzzy covering and 50% overlapping [8]. These assumptions are fully described in [8] and justified in [5]. All these assumptions guarantee there is unique mapping from appropriate degrees to mass assignments on labels. Based on these assumptions, we can isolate a set of subsets of LA with non-zero mass assignments. These are referred to as *focal sets*:

Definition 3 (*Focal Set*) *The focal set of LA, $\mathscr{F}$, is a set of focal elements defined as:*

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

Figure 1 shows the universes of two variables x_1 and x_2 which are fully covered by 3 fuzzy sets, respectively. For x_1, the following focal elements occur: $\{small_1\}$, $\{small_1, medium_1\}$, $\{medium_1\}$, $\{medium_1, large_1\}$ and $\{large_1\}$. Since $small_1$ and $large_1$ do not overlap, the set $\{small_1, large_1\}$ cannot occur as a focal element according to def. 3. We can then always find the unique translation from a given data point to a mass assignment on focal elements, specified by the function μ_L; This is referred to as *linguistic translation (LT)*. For a particular attribute with an associated focal set, linguistic translation is a process of replacing data elements with masses of focal elements of these data. For example in fig. 1, $\mu_{small_1}(x_1(1) = 0.27) = 1$, $\mu_{medium_1}(0.27) = 0.6$ and $\mu_{large_1}(0.27) = 0$. They are simply the memberships read from the fuzzy sets. We then can obtain the mass assignment of this data element according to def. 2 under consonance assumption [8]:

$$m_{0.27}(small_1) = 0.4, \; m_{0.27}(small_1, medium_1) = 0.6$$

Similarly, the linguistic translation for $\mathbf{x}_1 = \langle x_1(1) = 0.27, x_2(1) = 158 \rangle$ and $\mathbf{x}_2 = \langle x_1(2) = 0.7, x_2(2) = 80 \rangle$ is illustrated on each attribute independently as follows:

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

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

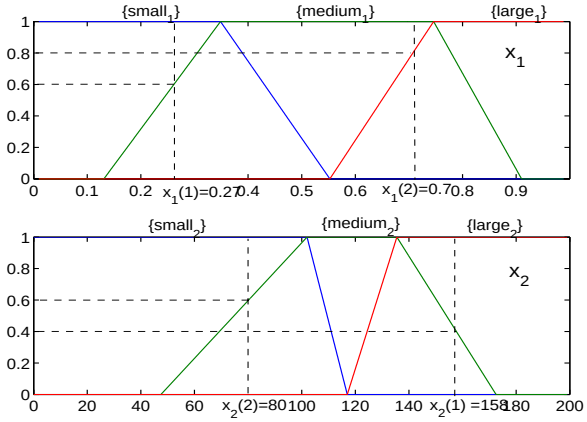

Figure 1: Full fuzzy covering (discretization) with three 50% overlapped fuzzy sets on two attributes x_1 and x_2, respectively.

3 LINGUISTIC DECISION TREES

Linguistic decision tree (LDT) [8] is a tree-structured classification model based on label semantics. The information heuristics used for building the tree are modified from ID3 in accordance with label semantics. The class probability estimation for each branch is evaluated according to the training set. Classification are made by considering the class probabilities across the whole tree. This model is fully described in [8], a concise introduction is given here for the purpose of this paper.

3.1 Linguistic decision trees for classification

Consider a database with n attributes and N instances and each instance is labeled by one of the classes: $\{C_1, \cdots, C_m\}$. A linguistic decision tree built from this database can be defined as follows:

$$LDT = \{\langle B_1, P(C_1|B_1), \cdots, P(C_m|B_1)\rangle, \cdots \\ \langle B_s, P(C_1|B_s), \cdots, P(C_m|B_s)\rangle\}$$

where $P(C_t|B)$ is the probability of class C_t given a branch B. A branch with k nodes is defined as:

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

where, $k \leq n$ and $F_j \in \mathscr{F}_j$ is one of the focal elements of attribute j. Figure 2 gives an illustration of a linguistic decision tree where each attribute is discretized by 3 fuzzy labels: *small*, *medium* and *large* with 50% overlap. For a binary classification problem, the branch

$$\langle \langle \{small_1\}, \{medium_2, large_2\} \rangle, 0.3, 0.7 \rangle$$

means the probability of class C_1 is 0.3 and C_2 is 0.7 given attribute 1 can be described as *small* and attribute 2 can only be described as *medium* and *large*.

Basically, fuzzy discretization provides an interpretation between numerical data and linguistic data based on label semantics. The effectiveness of fuzzy discretization may affect the algorithm's performance. In this paper, we will use percentile-based discretization: each attribute universe is partitioned into intervals which each contains approximately the

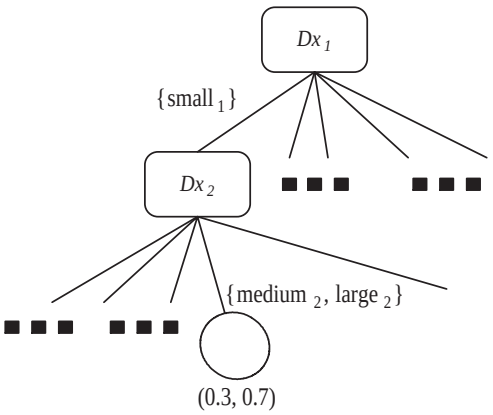

Figure 2: An illustration of a linguistic decision tree.

same number of data elements. It is a very intuitive way for generating fuzzy sets.

Given a training set with N instances: $DB = \{\mathbf{x_1}, \cdots, \mathbf{x}_N\}$ where each instance has n attributes: $\langle x_1, \cdots, x_n \rangle$. In the following we will simply write an instance $\langle x_1 = X_1, \ldots, x_n = X_n \rangle$ as $\mathbf{x}$ to simplify notation. The probability of class C_t ($t = 1, \cdots, m$) given B can then be evaluated as follows. First, we consider the probability of a branch B given $\mathbf{x}$:

$$P(B|\mathbf{x}) = \prod_{r=1}^{k} m_{x_j}(F_j) \tag{1}$$

$m_{x_j}(F_j)$ for $j = 1, \cdots, k$ are mass assignments of single data element x_j. Consider the previous example, suppose we are given a branch $B = \langle \{small_1\}, \{medium_2, large_2\} \rangle$ in fig. 1 and data $\mathbf{x}_1 = \langle 0.27, 158 \rangle$ (the linguistic translation of $\mathbf{x}_1$ was given in last section). According to eq. 1:

$$P(B|\mathbf{x}_1) = m_{x_1}(\{small_1\}) \times m_{x_2}(\{medium_2, large_2\})$$

$$= 0.4 \times 0.4 = 0.16$$

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

$$P(C_t|B) = \frac{\sum_{i \in DB_t} P(B|\mathbf{x}_i)}{\sum_{i \in DB} P(B|\mathbf{x}_i)} \tag{2}$$

where DB_t is the subset consisting of instances which belong to class t. In the case of $\sum_{i \in DB} P(B|\mathbf{x}_i) = 0$, 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.

$$P(C_t|B) = \frac{1}{m} \qquad for \quad t = 1, \cdots, m \tag{3}$$

Now consider classifying an unlabeled instance in the form of $\mathbf{x} = \langle x_1, \cdots, x_n \rangle$ which may not be contained in the training data set DB. First we apply linguistic translation to $\mathbf{x}$ based on the fuzzy covering of the training data DB. In the case that a data element appears beyond the range of training data set $[R_{min}, R_{max}]$, 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 the probabilities of class C_t given a LDT with s branches are evaluated as follows:

$$P(C_t|\mathbf{x}) = \sum_{v=1}^{s} P(C_t|B_v) P(B_v|\mathbf{x}) \tag{4}$$

where $P(C_t|B_v)$ and $P(B_v|\mathbf{x})$ are evaluated based on equations 1 and 2 (or 3), respectively.

3.2 Linguistic decision trees for prediction

Now consider a database for prediction $DB = \{\langle x_1(i), \cdots, x_n(i), x_t(i) \rangle | i = 1, \cdots, N\}$ where $x_1, \cdots, x_n$ are potential explanatory attributes and x_t is the continuous target attribute. For the target attribute x_t: $\mathscr{F}_t = \{F_t^1, \cdots, F_t^{|\mathscr{F}_t|}\}$, we can consider each focal element of target attributes as class labels. The LDT model for prediction then have the following form:

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

The problem of considering the target focal elements as class labels is, these "classes" overlap each other so that we cannot deal them as normal discreet classes. At the stage of training, for a particular instance $\mathbf{x}_i$, ($\mathbf{x}_i \rightarrow x_t(i)$), there may be several corresponding target focal elements rather than one. The membership of $\mathbf{x}_i$ belonging to a particular target focal element F_t^u is measured by ξ as follows:

$$\xi_i^u = m_{x_t(i)}(F_t^u) \tag{5}$$

for $u = 1, \cdots, |\mathscr{F}_t|$. In another words, ξ_i^u is the associated mass of F_t^u given $x_t(i)$. The corresponding target focal elements with a membership for $\mathbf{x}_i$ are as follows: $\mathbf{x}_i \rightarrow \langle F_t^1 : \xi_i^1, \cdots, F_t^{|\mathscr{F}_t|} : \xi_i^{|\mathscr{F}_t|} \rangle$. However, since we have made an assumption of 50% overlapping on fuzzysets, so, at most two values of $\{\xi^1, \cdots, \xi^{|\mathscr{F}_t|}\}$ are non-zero. We can also consider ξ as an indicator: if $\xi_i^u \neq 0$ then F_t^u is one of the corresponding target focal elements for the data element $\mathbf{x}_i$, otherwise, F_t^u is not. Similar to equation 2, the probability of F_t^u given B is evaluated as follows:

$$P(F_t^u|B) = \frac{\sum_{i \in DB} \xi_i^u P(B|\mathbf{x}_i)}{\sum_{i \in DB} P(B|\mathbf{x}_i)} \tag{6}$$

where $F_t^u \in \mathscr{F}_t$. Equation 6 is a general version of equation 2. In classification problems, the target labels are discreet then ξ is either 0 or 1. So that $\sum_{i \in DB_u} P(B|\mathbf{x}_i) \equiv \sum_{i \in DB} \xi_i^u P(B|\mathbf{x}_i)$. Similarly, in case of $\sum_{i \in DB} P(B|\mathbf{x}_i) = 0$, we use the following equation:

$$P(F_t^u|B) = \frac{1}{|\mathscr{F}_t|} \qquad for\ u = 1, \cdots, |\mathscr{F}_t| \tag{7}$$

The probabilities of target focal elements given a data element based on a LDT with s consisting branches are evaluated by

$$P(F_t^u|\mathbf{x}) = \sum_{v=1}^{s} P(F_t^u|B_v) P(B_v|\mathbf{x}) \tag{8}$$

Thus, for a given $\mathbf{x} = \langle x_1, \cdots, x_n \rangle$ to predict its target value $\widehat{x}_t$ (i.e. $\mathbf{x}_i \rightarrow \widehat{x}_t$). We can obtain a series of probabilities on

target focal elements: $P(F_t^1|\mathbf{x}),\cdots,P(F_t^{|\mathscr{F}_t|}|\mathbf{x})$. The estimate of x_t, denoted $\widehat{x}_t$, to be the expected value:

$$\widehat{x}_t = \int_{\Omega_t} x_t \; p(x_t|\mathbf{x}) \; dx_t \tag{9}$$

where:

$$p(x_t|\mathbf{x}) = \sum_{u=1}^{|\mathscr{F}_t|} p(x_t|F_t^u) \; P(F_t^u|\mathbf{x}) \tag{10}$$

and

$$p(x_t|F_t^u) = \frac{m_{x_t}(F_t^u)}{\int_{\Omega_t} m_{x_t}(F_t^u) \; dx_t} \tag{11}$$

so that, we can obtain:

$$\widehat{x}_t = \sum_u P(F_t^u|\mathbf{x}) \; E(x_t|F_t^u) \tag{12}$$

where:

$$E(x_t|F_t^u) = \frac{\int_{\Omega_t} x_t \; m_{x_t}(F_t^u) \; dx_t}{\int_{\Omega_t} m_{x_t}(F_t^u) \; dx_t} \tag{13}$$

where the process of calculating $E(x_t|F_t^u)$ is also called defuzzification in some other literatures.

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 maximumly increased with expanding. That corresponds to selecting the attribute with maximum information gain. The algorithm for developing linguistic decision trees for prediction is same to LDTs for classification which is fully described in [8], we won't introduce it here due to the page limitation. Similar like ID3, in developing the tree, the most informative attribute will form the root of a linguistic decision tree, and the tree will expand into branches associated with all possible focal elements of this attribute. For each branch, the free attribute with maximum information gain will be the next node, from level to level, until the tree reaches the maximum specified depth or some other criteria are met.

3.3 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 [6]. 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 4 (*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*

$$T_m \geq \max_{F_t \in \mathscr{F}_t} (|Pr(F_t|B_1) - Pr(F_t|B_2)|) \tag{14}$$

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

Definition 5 (*Merged Branch*) *A merged branch MB with k nodes is defined as*

$$MB = \langle \mathscr{M}_1,\cdots,\mathscr{M}_k \rangle$$

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

$$m_x(\mathscr{M}_j) = \sum_{i=1}^{w} m_x(F_j^i) \tag{15}$$

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, {small} and {small, medium} are adjacent focal elements while {small} and {medium} are not. The probability of a merged branch given a data element $\mathbf{x} \in \Omega_1 \times,\cdots,\times\Omega_n$ can be evaluated by

$$P(MB|\mathbf{x}) = \prod_{r=1}^{k} m_{x_r}(\mathscr{M}_r) = \prod_{r=1}^{k} \left(\sum_{i=1}^{w_r} m_{x_r}(F_r^i) \right) \tag{16}$$

where k is the length of the merged branch MB and w_r for $r = 1,\cdots,k$ is the number of merged nodes of the attribute r. Based on equations 2, 3, 5, 15 and 16 we use the following equation to evaluate the probabilities on target focal elements given a merged branch.

$$P(F_t^j|MB) = \frac{\sum_{i\in DB} \xi_i^j P(MB|\mathbf{x})}{\sum_{i\in DB} P(MB|\mathbf{x})} \tag{17}$$

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

$$P(F_t^j|\mathbf{x}) = \sum_{v=1}^{s'} P(F_t^j|MB_v) P(MB_v|\mathbf{x}) \tag{18}$$

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 equation 17. 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

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 *DB*, *AVE* is the average modulus error taken as a percentage of the length of the output universe, formally:

$$AVE = \frac{\sum_{i\in DB} |\hat{x}_t(i) - x_t(i)|}{|DB|(b-a)} \tag{19}$$

where $|DB|$ represents the number of instances in DB. The standard deviation across DB is given by

$$\sigma_E = \sqrt{\frac{1}{|DB|}\sum_{i\in DB}(\varepsilon_i - AVE)^2} \qquad (20)$$

where:

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

In this section, the LDT results are compared with the results of ε-SVR [1] Fuzzy Naive Bayes [10] and Fuzzy Semi-Naive Bayes (FSNB) [10]. The parameter settings for other 3 models are based on the empirical research on these problems by Randon [10].

4.1 Prediction of Sunspots

This problem contains data of sunspot numbers between the years 1700-1979. For this experiment the data was organized as described in [11] 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.

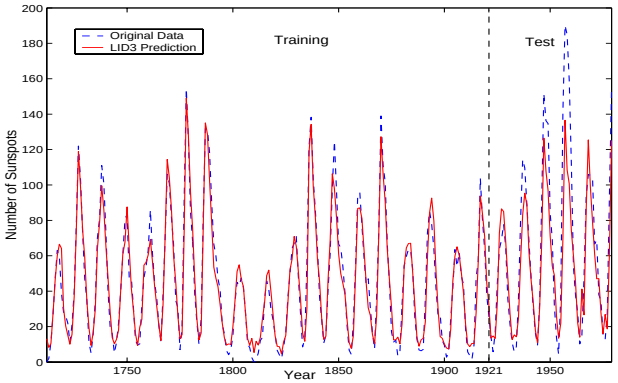

Figure 3: The prediction results obtained from LDT without merging, where the data on the left (1712-1921) are for training and the right (1921-1079) are for test.

The result comparisons in the AVE measure are shown in table 1, where the parameter setting for ε-SVR is as follows: $\sigma = 3$, $\varepsilon = 0.05$, $C = 5$ [10]. Results of LDT 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 figure 3, where the data on the left (1712-1921) are for training data and the right are (1921-1979) for test.

Table 1 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

[1] ε-Support Vector Regression system with a Gaussian kernel and an ε-insensitive loss function. The SVR results present here are obtained by using a Matlab package implemented by Gunn [3].

Table 1: Prediction results in AVE on the sunspot prediction problem.

Prediction Model	AVE %	σ_E(%)	Size
Fuzzy Naive Bayes	13.0588	13.0213	-
FSNB	10.9064	9.5208	-
ε-SVR	8.9337	9.7766	-
LDT	8.6793	8.8876	5731
LDT ($T_m = 0.05$)	8.8925	8.9437	2285
LDT ($T_m = 0.10$)	8.9649	9.1994	1493
LDT ($T_m = 0.15$)	9.8419	10.1869	757
LDT ($T_m = 0.20$)	9.8341	10.7063	204
LDT ($T_m = 0.25$)	10.5858	10.3711	81
LDT ($T_m = 0.30$)	18.9539	19.1159	5

Table 2: Average errors with standard deviations on test set of the flood forecasting problem.

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

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%.

4.2 Flood Forecasting

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 intercomparison 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 hour 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.

A Fuzzy Semi-Naive Bayes model is also used to study this problem by Randon [10] with and without windowing techniques. In order to make direct comparisons with other river flow modelling techniques we shall initially use the same

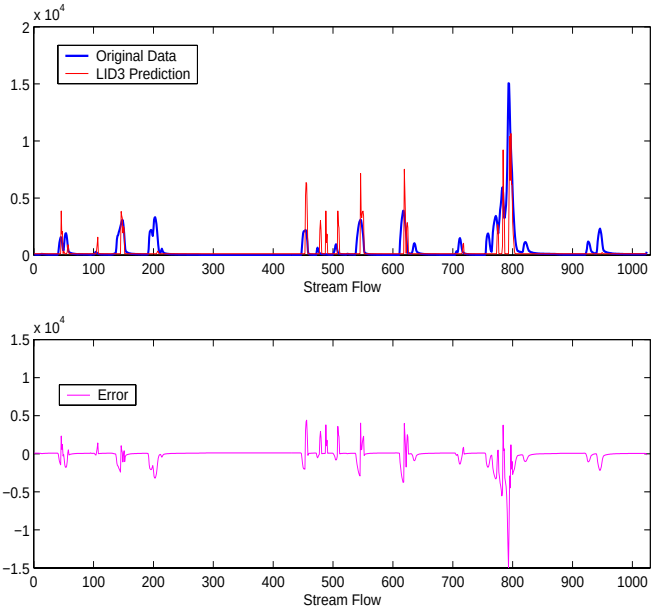

Figure 4: The stream flow prediction (upper figure) with a merged LDT with $T_m = 0.3$ and the error are shown in the below figure.

training and test data as in previous studies. 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 2, where the results of ε-SVR are based on parameters: $\sigma = 3$, $\varepsilon = 0.05$ and $C = 5$. The LDT 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 2, LDT 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 accuracy to Fuzzy Semi-Naive Bayes. However, the tree has only 108 branches and comparing to LDT without merging, the tree size has been reduced nearly 94%. The performance on the test set can be seen from figure 4. Although LID3 over-estimates at some peaks, it still captures the original data well.

5 CONCLUSIONS

Linguistic decision tree is a classification model for its advantages of handling uncertainties and being transparent. In this paper, a methodology of extending linguistic decision tree from classification to prediction is proposed. We tested on two benchmark problems: sunspot prediction and real-world flood forecasting. 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. More compact trees can also be obtained by applying forward merging while the performances of the algorithm are not significantly influenced with small merging thresholding.

However, 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 unlike other black-box prediction models, a LDT can be interpreted as a set of linguistic rules, which may provides the information about how the predictions are made.

REFERENCE

[1] J.F. Baldwin, T.P. Martin and B.W. Pilsworth. Fril-Fuzzy and Evidential Reasoning in Artificial Intelligence. John Wiley & Sons Inc, 1995.

[2] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth Inc., 1984.

[3] S. R. Gunn. “Support vector machines for classification and regression”. Technical Report of Dept. of Electronics and Computer Science, University of Southampton, May 1998.

[4] C.Z. Janikow. “Fuzzy decision trees: issues and methods”. IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 28, No. 1, Feb. 1998.

[5] J. Lawry. “A framework for linguistic modelling”. Artificial Intelligence, 155: pp. 1-39, 2004.

[6] C. Olaru and L. Wehenkel. “A complete fuzzy decision tree technique”. Fuzzy Sets and Systems. 138: 221-254, 2003.

[7] F. Provost and P. Domingos. “Tree induction for probability-based ranking”. Machine Learning. 52, 199-215, 2003.

[8] Z. Qin and J. Lawry. “A tree-structured model classification model based on label semantics”. The Proceedings of IPMU-04, Perugia, Italy 2004.

[9] J.R. Quinlan. C4.5: Programs for Machine Learning. San Mateo: Morgan Kaufmann, 1993.

[10] N.J. Randon. Fuzzy and Random Set Based Induction Algorithms. PhD Thesis, Dept. of Engineering Mathematics, University of Bristol, 2004.

[11] A. A. Weigend, B.A. Huberman, and D.E. Rumelhart. “Predicting sunspots and exchange rates with connectionist networks”. Non-linear Modelling and Forecasting, SFI Studies in the Science of Complexity,Vol. XII, pp395-432, Addison-Wesley, 1992.

[12] Y. Yuan and M. J. Shaw. “Induction of fuzzy decision trees”. Fuzzy Sets and Systems, 69: pp. 125-139, 1995.