Hybrid Bayesian Estimation Trees Based on Label Semantics

Zengchang Qin and Jonathan Lawry

A.I. Group, Department of Engineering Mathematics,
University of Bristol, Bristol BS8 1TR, U.K
{z.qin, j.lawry}@bristol.ac.uk

Abstract. Linguistic decision tree (LDT) [7] is a classification model based on a random set based semantics which is referred to as label semantics [4]. Each branch of a trained LDT is associated with a probability distribution over classes. In this paper, two hybrid learning models by combining linguistic decision tree and fuzzy Naive Bayes classifier are proposed. In the first model, an unlabelled instance is classified according to the Bayesian estimation given a single LDT. In the second model, a set of disjoint LDTs are used as Bayesian estimators. Experimental studies show that the first new hybrid models has both better accuracy and transparency comparing to fuzzy Naive Bayes and LDTs at shallow tree depths. The second model has the equivalent performance to the LDT model.

1 Introduction

Tree induction algorithms have received a great deal of attention because of their simplicity and effectiveness. There are many decision tree models and tree induction algorithms that been proposed. These range from early discrete decision trees such as ID3 [9] and C4.5 [10] to a variety of fuzzy decision trees. *Linguistic decision tree* (LDT) is a tree-structured model based on a high level knowledge representation framework which is referred to as *Label semantics* [4]. Linguistic expressions (or fuzzy labels) such as *small*, *medium* and *large* are used to build a tree guided by information based heuristics. For each branch, the probability of this branch belonging to a particular class is evaluated based on the proportion of data in this class relative to all the data covered by the linguistic expressions of the branch. Therefore, LDT model can be regarded as a probability estimation tree model based on fuzzy labels. The LDT model has been shown to be an effective model for both classification and prediction. Also a LDT can be represented as a set of linguistic rules and hence provides a high level transparency. However, for complex problems, good probability estimations can only be obtained by deep LDTs, which are not good in terms of transparency. In such cases, how can we build a model which has a good probability estimation with compact LDTs (i.e. LDTs with shallow depths or with less number of branches)? This question motivates the research presented in this paper.

L. Godo (Ed.): ECSQARU 2005, LNAI 3571, pp. 896–907, 2005.

Naive Bayes is a well known and much studied algorithm in machine learning. It is a simple, effective and efficient learning method. Although Naive Bayes classification makes the unrealistic assumption that the values of the attributes of an instance are conditionally independent given the class of the instance, this model is remarkably successful in practice. In this paper, an extended version of Naive Bayes based on label semantics is introduced. The new hybrid models using Naive Bayes classification given a single LDT and a set of disjoint LDTs are proposed and tested on a number of UCI datasets [2].

2 Label Semantics

Label semantics [4] is a framework to represent the use of linguistic expressions to label a value. The underlying question posed by label semantics is how to use linguistic expressions to label numerical values. For a variable x into a domain of discourse Ω we identify a finite set of linguistic labels $\mathcal{L} = \{L_1, \cdots, L_n\}$ with which to label the values of x. Then for a specific value $\alpha \in \Omega$ an individual I identifies a subset of $\mathcal{L}$, 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 $\mathcal{L}$. The frequency of occurrence of a particular label, say S, for D_α across the population then gives a distribution on D_α referred to as a mass assignment on labels, more formally:

Definition 1 (*Mass Assignment on Labels*)

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

For example, given a set of labels defined on the temperature outside: $\mathcal{L}_{Temp} = \{low, medium, high\}$. Suppose 3 of 10 people agree that '*medium* is the only appropriate label for the temperature of 15° and 7 agree 'both *low* and *medium* are appropriate labels'. According to def. 1, $m_{15}(medium) = 0.3$ and $m_{15}(low, medium) = 0.7$ so that the mass assignment for 15° is $m_{15} = \{medium\} : 0.3$, $\{low, medium\}$: 0.7. More details about the theory of mass assignment can be found in [1].

Consider the previous example, can we know how appropriate for a single label, say *low*, to describe 15°? In this framework, *appropriateness degrees* are used to evaluate how appropriate a label is for describing a particular value of variable x. Simply, given a particular value α of variable x, the appropriateness degree for labeling this value with the label L, which is defined by fuzzy set F, is the membership value of α in F. The reason we use the new term 'appropriateness degrees' is partly because it more accurately reflects the underlying semantics and partly to highlight the quite distinct calculus based on this framework [4]. This definition provides a relationship between mass assignments and appropriateness degrees.

Definition 2 (*Appropriateness Degrees*)

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

Consider the previous example, we then can obtain $\mu_{medium}(15) = 0.7 + 0.3 = 1$, $\mu_{low}(15) = 0.7$. Based on the underlying semantics, we can translate a set of numerical data into a set of mass assignments on appropriate labels based on the reverse of definition 2 under the following assumptions: consonance mapping, full fuzzy covering and 50% overlapping [7]. These assumptions are fully described in [7] and justified in [4]. These assumptions guarantee that there is unique mapping from appropriate degrees to mass assignments on labels. Based on these assumptions, we can isolate a set of subsets of $\mathcal{L}$ with non-zero mass assignments. These are referred to as *focal sets*:

Definition 3 (*Focal Set*) *Given a universe Ω for variable x, the focal set of $\mathcal{L}$ is a set of focal elements defined as:*

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

The right-hand side figure of fig. 1 shows the universes of two variables x_1 and x_2 which are fully covered by 3 fuzzy sets with 50% overlap, respectively. For x_1, the following focal elements occur: $\{small_1\}$, $\{small_1, medium_1\}$, $\{medium_1\}$, $\{medium_1, large_1\}$ 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 always find a unique translation from a given data point to a mass assignment on focal elements, as specified by the function μ_L. This is referred to as *linguistic translation (LT)* and is defined as follows: *For a particular attribute with an associated focal set, 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 [7]: $m_{0.27}$ $(small_1) = 0.4$, $m_{0.27}(small_1, medium_1) = 0.6$. Similarly, the linguistic translations for $\mathbf{x}_1 = \langle x_1(1) = 0.27, x_2(1) = 158 \rangle$ and $\mathbf{x}_2 = \langle x_1(2) = 0.7, x_2(2) = 80 \rangle$ are illustrated on each attribute independently as follows:

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

$$\left[\begin{array}{c} x_2 \\ \hline x_1(2) = 158 \\ x_2(2) = 80 \end{array}\right] \overset{LT}{\rightarrow} \left[\begin{array}{ccccc} m_x(\{s_2\}) & m_x(\{s_2, m_2\}) & m_x(\{m_2\}) & m_x(\{m_2, l_2\}) & m_x(\{l_2\}) \\ \hline 0 & 0 & 0 & 0.4 & 0.6 \\ 0.4 & 0.6 & 0 & 0 & 0 \end{array}\right]$$

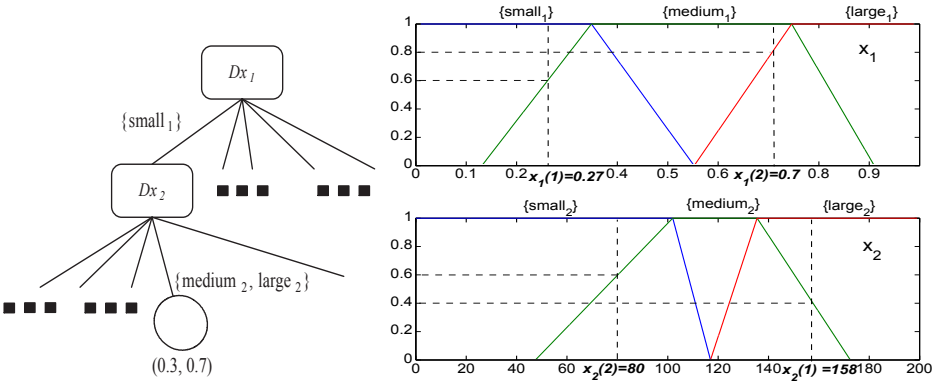

Fig. 1. Left-hand side: A schematic illustration of a linguistic decision tree. Right-hand side: A full fuzzy covering (discretization) with three fuzzy sets with 50% overlap on two attributes x_1 and x_2, respectively

3 Linguistic Decision Tree

Linguistic decision tree (LDT) [7] is a tree-structured classification model based on label semantics. The information heuristics used for building the tree are modified from Quinlan's ID3 [9] in accordance with label semantics. Given a database of which each instance is labeled by one of the classes: $\{C_1, \cdots, C_M\}$. A linguistic decision tree with S consisting branches built from this database can be defined as follows:

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

where $P(C_k|B)$ is the probability of class C_k given a branch B. A branch B with d nodes (i.e., the length of B is d) is defined as:

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

where, $d \leq n$ and $F_j \in \mathcal{F}_j$ is one of the focal elements of attribute j. The left-hand side figure of fig 1 gives an schematic illustration of a LDT for a binary classification problem. For example, consider the branch: $\langle\langle\{small_1\}, \{medium_2, large_2\}\rangle$, 0.3, 0.7$\rangle$. This means the probability of class C_1 is 0.3 and C_2 is 0.7 given attribute 1 can only be described as *small* and attribute 2 can be described as both *medium* and *large*. We may notice that different fuzzy discretization methods may result in different translations between numerical data and their linguistic models. In this paper, we will use a very intuitive method for generating fuzzy sets referred to as percentile-based (or equal-point) discretization [7, 11]. In this approach, each attribute universe is partitioned into intervals each containing approximately the same number of data elements.

Consider a training set $\mathcal{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ where each instance $\mathbf{x}$ has n attributes: $\langle x_1, \cdots, x_n\rangle$. We now describe how the relevant branch probabili-

ties for a LDT can be evaluated from a database. The probability of class C_k $(k = 1, \cdots, M)$ given B can then be evaluated as follows. First, we consider the probability of a branch B given $\mathbf{x}$:

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

where $m_{x_j}(F_j)$ for $j = 1, \cdots, d$ are mass assignments of single data element x_j. Consider the previous example, where we are given a branch $B = \langle \{small_1\}, \{medium_2, large_2\} \rangle$ in fig. 1 and data element $\mathbf{x}_1 = \langle 0.27, 158 \rangle$ (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_k given B can then be evaluated by:

$$P(C_k|B) = \frac{\sum_{i \in \mathcal{D}_k} P(B|\mathbf{x}_i)}{\sum_{i \in \mathcal{D}} P(B|\mathbf{x}_i)} \tag{2}$$

where $\mathcal{D}_k$ is the subset consisting of instances which belong to class k. In the case where the denominator is equals to 0, which may occur when the training database for the LDT is small, then there is no non-zero linguistic data covered by the branch. In this case, we obtain no information from the database so that equal probabilities are assigned to each class.

$$P(C_k|B) = \frac{1}{M} \qquad for \quad k = 1, \cdots, M \tag{3}$$

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. First we apply linguistic translation to $\mathbf{x}$ based on the fuzzy covering of the training data[1]. According to the Jeffrey's rule [3] the probabilities of class C_k given a LDT with S branches are evaluated as follows:

$$P(C_k|\mathbf{x}) = \sum_{s=1}^{S} P(C_k|B_s) P(B_s|\mathbf{x}) \tag{4}$$

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

The goal of tree-structured learning models is to generate subregions partitioned by branches that are 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

[1] In the case that a data element appears beyond the range of training data set, we then assign the appropriateness degrees of the minimum or maximum values of the universe to the data element depending on which side of the range it appears.

attribute with maximum information gain. The algorithm for developing linguistic decision trees is fully described in [7] and will not be reproduced here due to the page limitation. Similar to 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 attribute that has not appeared in this branch and that has the maximum information gain will be selected as the next node. This is will be repeated from level to level until the tree reaches the maximum specified depth or some other termination criteria are met.

4 Bayesian Estimation Trees with Fuzzy Labels

4.1 Naive Bayes Classification Based on Label Semantics

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

$$P(C_k|\mathbf{x}) = \frac{P(\mathbf{x}|C_k)P(C_k)}{P(\mathbf{x})} \tag{5}$$

Since the denominator in eq. 5 is invariant across classes, we can consider it as a normalization parameter. So, we obtain:

$$P(C_k|\mathbf{x}) \propto P(\mathbf{x}|C_k)P(C_k) \tag{6}$$

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

$$P(C_k|\mathbf{x}) \propto \prod_{j=1}^{n} P(x_j|C_k)P(C_k) \tag{7}$$

where $P(x_j|C_k)$ is often called the likelihood of the data x_j given C_k. For a qualitative attribute, it can be estimated from corresponding frequencies. For a quantitative attribute, either probability density estimation or discretization can be employed to estimate its probabilities. In label semantics framework, suppose we are given focal set $\mathcal{F}_j$ for each attribute j. Assuming that attribute x_j is numeric with universe Ω_j, then the likelihood of x_j given C_k can be represented by a density function $p(x_j|C_k)$ determine from the database $\mathcal{D}_k$ and prior density according to Jeffrey's rule [3].

$$p(x_j|C_k) = \sum_{F \in \mathcal{F}_j} p(x_j|F)P(F|C_k) \tag{8}$$

From Bayes theorem:

$$p(x_j|F) = \frac{P(F|x_j)p(x_j)}{P(F)} = \frac{m_{x_j}(F)p(x_j)}{pm(F)} \tag{9}$$

where,

$$pm(F) = \int_{\Omega_j} P(F|x_j)p(x_j)dx_j = \frac{\sum_{\mathbf{x}\in\mathcal{D}} m_{x_j}(F)}{|\mathcal{D}|} \tag{10}$$

Substituting equation 9 in equation 8 and re-arranging gives

$$p(x_j|C_k) = p(x_j) \sum_{F\in\mathcal{F}_j} m_{x_j}(F)\frac{P(F|C_k)}{pm(F)} \tag{11}$$

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

$$P(F|C_k) = \frac{\sum_{\mathbf{x}\in\mathcal{D}_k} m_{x_j}(F)}{|\mathcal{D}_k|} \tag{12}$$

Here in this paper, this model is called fuzzy Naive Bayes (FNB) and more details of FNB can be found in [11].

4.2 Bayesian Estimation Given a LDT

Given a decision tree T is learnt from a training database $\mathcal{D}$. According to the Bayesian theorem: A data element $\mathbf{x} = \langle x_1, \ldots, x_n\rangle$ can be classified by:

$$P(C_k|\mathbf{x}, T) \propto P(\mathbf{x}|C_k, T)P(C_k|T) \tag{13}$$

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

$$P(\mathbf{x}|C_k, T) = P(\mathbf{x}_T|C_k, T)P(\mathbf{x}_B|C_k, T) \tag{14}$$

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

$$P(\mathbf{x}_B|C_k, T) = P(\mathbf{x}_B|C_k) = \prod_{j\in\mathbf{x}_B} P(x_j|C_k) \tag{15}$$

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

$$P(\mathbf{x}_T|C_k, T) = \frac{P(C_k|\mathbf{x}_T, T)P(\mathbf{x}_T|T)}{P(C_k|T)} \tag{16}$$

Combining equation 14, 15 and 16:

$$P(\mathbf{x}|C_k, T) = \frac{P(C_k|\mathbf{x}_T, T)P(\mathbf{x}_T|T)}{P(C_k|T)} \prod_{j\in\mathbf{x}_B} P(x_l|C_k) \tag{17}$$

Combining equation 13 and 17

$$P(C_k|\mathbf{x}, T) \propto P(C_k|\mathbf{x}_T, T)P(\mathbf{x}_T|T) \prod_{j\in\mathbf{x}_B} P(x_j|C_k) \tag{18}$$

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

$$P(C_k|\mathbf{x}, T) \propto P(C_k|\mathbf{x}_T, T) \prod_{j \in \mathbf{x}_B} P(x_j|C_k) \tag{19}$$

where $P(x_j|C_k)$ is evaluated according to eq. 11 and $P(C_k|\mathbf{x}_T, T)$ is just the class probabilities evaluated from the decision tree T according to equation 4.

The basic idea of using Bayesian estimation given a LDT is to use the LDT as one estimator and the rest of the attriubutes as other independent estimators. If we extend this idea, we use a set of small-sized LDTs as estimators, we then have the second hybird model which is described in the next section.

4.3 Bayesian Estimation from a Set of Trees

Given a training dataset, a small-sized tree (usually the depth is less than 3) can be learnt based on the method we discussed in section 3. We then learn another tree with the same size based on the rest of the attributes, i.e., the attributes which have not been used in previous trees. Successively, a set of trees can be built from training set. If we denote the trees by $\mathcal{T} = \langle T_1, \ldots, T_W \rangle$, for each tree T_w, the set of attributes $\mathbf{x}_{T_w}$ are exclusive each other for $w = 1, \ldots, W$. For a given unclassified data element $\mathbf{x}$, we can partition it into W group of disjoint set of attributes $\langle \mathbf{x}_{T_1}, \ldots, \mathbf{x}_{T_W} \rangle$. If we assume:

$$P(C_k|\mathbf{x}) = P(C_k|\mathbf{x}_{T_1}, \ldots, \mathbf{x}_{T_W}) \approx P(C_k|T_1, \ldots, T_W) \tag{20}$$

Then, according to the Bayesian theorem:

$$P(C_k|\mathcal{T}) = P(C_k|T_1, \ldots, T_W) = \frac{P(T_1, \ldots, T_W|C_k)P(C_k)}{P(T_1, \ldots, T_W)} \tag{21}$$

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

$$P(T_1, \ldots, T_W|C_k) = \prod_{w=1}^{W} P(T_w|C_k) \tag{22}$$

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

$$P(T_w|C_k) = \frac{P(C_k|T_w)P(T_w)}{P(C_k)} \tag{23}$$

So that,

$$\prod_{w=1}^{W} P(T_w|C_k) = \frac{\prod_{w=1}^{W} P(C_k|T_w) \prod_{i=1}^{W} P(T_w)}{P(C_k)^W} \tag{24}$$

Combine eq. 21, 22 and 24, we obtain

$$P(C_k|\mathcal{T}) \propto \frac{\prod_{w=1}^{W} P(C_k|T_w) \prod_{w=1}^{W} P(T_w)}{P(C_k)^{W-1}} \tag{25}$$

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

$$P(C_k|\mathcal{T}) \propto \frac{\prod_{w=1}^{W} P(C_k|T_w)}{P(C_k)^{W-1}} \tag{26}$$

where $P(C_k|T_w)$ is evaluated according to eq. 4.

5 Experimental Studies

We evaluated the LDT model, single LDT with Bayesian estimation (denoted by BLDT) and Bayesian estimation with a set of trees (denoted by FLDT - a forest of LDTs) on 10 datasets taken from the UCI Machine Learning repository [2]. The descriptions are shown in table 1. Unless otherwise stated, attributes are discretized by 2 trapezoidal fuzzy sets with 50% overlap based on equal-point discretization (see section 3), and classes are evenly split into two sub-datasets randomly, one half for training and the other half for testing, this is referred to as a 50-50 split experiment. For each dataset, we ran 50-50 experiment with random split for 10 times and the average test accuracies with standard deviations are shown against depths of the trees are shown in figures 2. The results of C4.5 [2] Fuzzy Naive Bayes (FNB), FLDT and the best resutls of LDT and BLDT are shown in table 2, where d for LDT and BLDT represents the depth at which the best results are obtained.

From all the figures, we can see that the BLDT model generally performs better at shallow depths than LDT model. However, with the increasing of the tree depth, the performance of the BLDT model remains constant or decreases, while the accuracy curves for LDT increase. For datasets Balance, Ecoli, Wisconsin-Cancer (Wcancer) and Wine, BLDT model performs better at most of depths. For Iris and Heptitis, the differences are insignificant at all depths. For Pima, LDT model performs better than BLDT model in most the depths and the differences are significant. For the rest of the datasets, the accuracy curves cross somewhere in the middle and the differences are not significant.

Table 1. Descriptions of the datasets for experiments selected from the UCI machine learning repository [2]

Dataset	Classes	Size	Attributes	Dataset	Classes	Size	Attributes
Balance	3	625	4	**Ecoli**	8	336	8
Glass	6	214	9	**Heptitis**	2	155	19
Iris	3	150	4	**Liver**	2	345	6
Pima	2	768	8	**Sonar**	2	208	60
Wine	3	178	14	**Wcancer**	2	699	9

[2] The results are obtained by WEKA [12] machine toolkit with default settings.

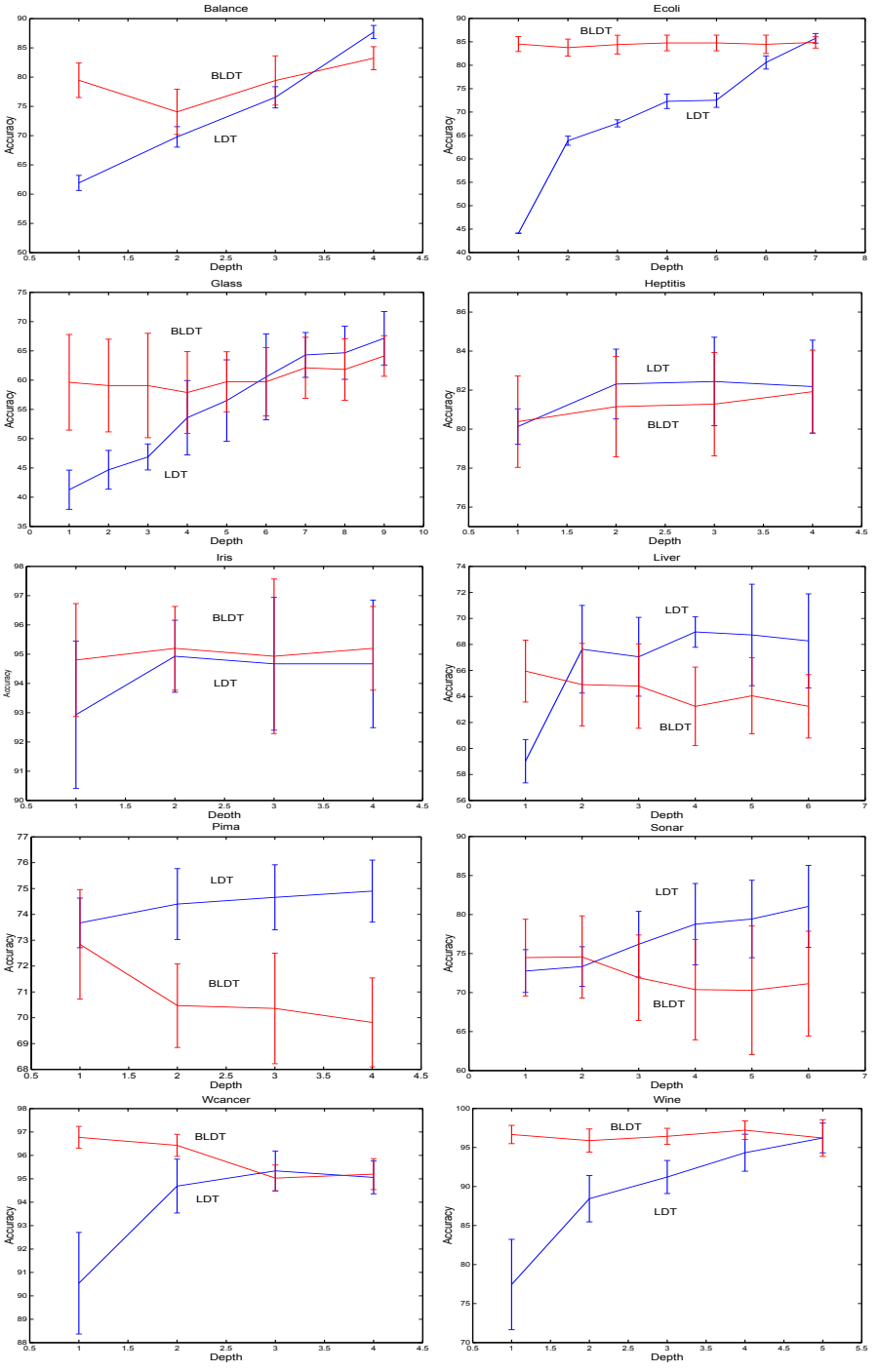

Fig. 2. Results for single LDT with Bayesian estimation: average accuracy with standard deviation on each dataset against the depth of the tree

Table 2. Experimental results on 10 UCI datasets: average accuracy with standard deviation from 10 runs of random 50-50 split experiments

	C4.5	**FNB**	**LDT**		**BLDT**		**FLDT**	
Database	Acc	Acc	Acc	d	Acc	d	Acc(d=1)	Acc(d=2)
Balance	79.20±1.53	73.77±2.43	87.70±1.13	4	83.23±1.97	4	66.26±2.81	79.42±1.99
Ecoli	78.99±2.23	76.53±4.19	85.76±1.03	7	84.53±1.60	1	80.18±3.45	78.76±1.60
Glass	64.77±5.10	48.35±6.80	59.17±3.70	9	64.13±3.47	9	52.94±8.74	58.53±5.28
Heptitis	76.75±4.68	80.13±2.28	82.44±2.27	3	81.92±2.13	4	80.26±3.15	79.26±0.41
Iris	93.47±3.23	93.73±2.60	94.93±1.23	2	95.20±1.43	2	93.73±1.89	92.00±3.38
Liver	65.23±3.86	63.35±2.38	68.96±3.18	4	65.95±2.38	1	62.43±4.62	59.65±2.09
Pima	72.16±2.80	72.29±2.25	74.90±1.20	4	72.84±2.12	1	72.40±1.48	66.07±1.04
Sonar	70.38±5.23	74.76±4.96	81.05±5.24	6	74.57±5.26	2	76.48±4.82	75.62±2.21
Wcancer	94.38±1.42	96.74±0.54	95.34±0.85	3	96.77±0.47	1	97.17±0.93	98.77±0.85
Wine	88.09±4.14	96.22±1.67	96.22±1.90	5	97.22±1.20	4	96.11±0.79	98.56±1.66

Table 3. Result comparisons (with LDT, BLDT and FLDT are at depth 2) based on t-test with 90% confidence, where '$\surd$' represents significant better, '$-$' represents equivalence and '$\times$' represents significant worse

Database	BLDT vs C4.5	BLDT vs FNB	BLDT vs LDT	FLDT vs C4.5	FLDT vs FNB	FLDT vs LDT
Balance	$\surd$	$\surd$	$\surd$	$-$	$\surd$	$\surd$
Ecoli	$\surd$	$\surd$	$\surd$	$-$	$-$	$\surd$
Glass	$-$	$\surd$	$\surd$	$-$	$\surd$	$\surd$
Heptitis	$\surd$	$-$	$-$	$-$	$-$	$\times$
Iris	$-$	$-$	$-$	$-$	$-$	$-$
Liver	$-$	$-$	$-$	$-$	$-$	$\times$
Pima	$-$	$-$	$\times$	$-$	$-$	$\times$
Sonar	$-$	$-$	$-$	$-$	$-$	$-$
Wcancer	$\surd$	$-$	$\surd$	$\surd$	$\surd$	$\surd$
Wine	$\surd$	$-$	$\surd$	$\surd$	$-$	$\surd$

We performed t-tests with a confidence level of 90% [3] to compare the models at depth 2 (except for C4.5 and FNB) and the results are shown in table 3. We can see that BLDT and FLDT models are better than Fuzzy Naive Bayes and C4.5. However, if we compare BLDT and FLDT with LDT, we can find that the BLDT model outperforms LDT at shallow depths and FLDT model has the equivalent performance. From fig. 2, we found that most best results for BLDT are obtained at shallow depths, but for LDTs the best results are always obtained with deep depths. So, we can conclude that BLDT model is

[3] We generally believe that the confidence level of 90 % is enough to be significant for comparisons among different learning models given these relatively simple data sets.

more efficient than LDT. Compare to BLDT, the FLDT model performs relative worse and less efficient, the reasons are probably because that small-trees are not good estimators. But this still needs more further investigation.

6 Conclusions

In this paper, we propsed two hybrid models by combining Naive Bayes classifier and linguistic decision trees based on label semantics. Through experimental studies, we found that the BLDT (the Bayesian estimation model given a LDT) model outperforms fuzzy naive Bayes, C4.5 and the linguistic decision tree model at shallow tree depths. However, the FLDT (using a set of small-size LDTs as Bayesian estimators) model outperforms fuzzy Naive Bayes classifier and C4.5 but has equivalent accuracy to LDTs. Further research focus on investigating the reasons that FLDTs are not good Bayesian estimators and testing on more datasets.

References

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. C. Blake and C.J. Merz. UCI machine learning repository. *http://www.ics.uci.edu/ mlearn/MLRepository.html*
3. R.C.Jeffrey. *The Logic of Decision*, Gordon & Breach Inc., New York, 1965.
4. J. Lawry. A framework for linguistic modelling, *Artificial Intelligence*, 155: pp. 1-39, 2004.
5. C. X. Ling. Decision tree with better ranking. *Proceedings of International Conference on Machine Learning (ICML2003)*. Washington DC, 2003.
6. F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*. 52, pp.199-215, 2003.
7. Z. Qin and J. Lawry. Decision Tree Learning with Fuzzy Labels. To appear in *Information Sciences*, 2005.
8. Z. Qin and J. Lawry. ROC analysis of a linguistic decision tree merging algorithm. *The Pro. of UK Workshop on Computational Intelligence*, Loughborough, UK, 2004.
9. J.R. Quinlan. Induction of decision trees. *Machine Learning* 1: 81-106. 1986
10. J.R. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann, 1993.
11. N.J. Randon and J. Lawry. Classification and query evaluation using modelling with words. *Information Sciences, Special Issue - Computing with Words: Models and Applications*, To appear.
12. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999. http://www.cs.waikato.ac.nz/∼ml/weka/