A Tree-structured Classification Model Based on Label Semantics

Zengchang Qin and Jonathan Lawry
Artificial Intelligence Group
Department of Engineering Mathematics
University of Bristol, BS8 1TR, UK
{z.qin, j.lawry}@bris.ac.uk

Abstract

Label Semantics is a random set based framework for Computing with Words. Imprecise concepts are modeled by the degrees of appropriateness of a linguistic expression as defined by a fuzzy set. An approach to decision tree induction based on this framework is studied and its performance when applied to real-world datasets is compared with the C4.5 and other machine learning algorithms. A method of classification under linguistic constraints was proposed and studied with experiments.

Keywords: Linguistic decision trees, Label semantics, LID3, Mass assignment, Focal sets.

1 Introduction

Traditionally machine learning and data mining research has focused on learning algorithms with high classification or prediction accuracy. However, this is not always sufficient for some application areas. We may require a model with good *transparency* from which information regarding underlying trends and relationships can be easily understood and interpret human languages rather than a black box. The research area of *Computing with Words (CW)*, proposed by Zadeh [11], provides us with a framework in which to develop such a system. Here we present, Label semantics, an alternative framework for CW which was proposed by Lawry [4]. The new framework is a random set based 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. Linguistic expressions are labels such as *small*, *medium* and *large* which are defined by fuzzy sets covering the continuous universe of the variables.

Previous research has focused on applying this framework to Bayesian learning [9]. Here we propose a new tree-structured model, *Linguistic Decision Trees (LDT)*. Like a traditional decision tree, A LDT expands from level to level guided by information content-based heuristics, until a given maximum depth is reached. For each branch, the class probabilities given this branch will be evaluated based on a linguistic training dataset, corresponding to a *linguistic translation* of the original training dataset within this framework. Unlabeled data is classified by a LDT based on Jeffrey' s rule. The case of data classification under linguistic constraints is also studied. In the last section, some experimental results are shown to support the validity of our approach.

2 Label Semantics For Data Analysis

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 (see [1] for details) on labels, more formally:

Definition 1 (*Mass Assignment*)

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

For example, given a set of labels defined on a man's age $LA_{age} = \{young(y), middle-aged(m), old(o)\}$. 3 of 10 people agree that '*young* is the only suitable label for the age of 30' and 7 agree 'both *young* and $middle-aged$ are suitable labels'. According to def. 1, $m_{30}(y) = 0.3$ and $m_{30}(y,m) = 0.7$ so that the mass assignment for 30 is

$$m_{30} = \{y\} : 0.3, \{y,m\} : 0.7$$

In this framework, *appropriateness degree* is used to evaluate how appropriate a label is for describing a particular value of variable x. This measure can be defined based on mass assignments as follows:

Definition 2 (*Appropriateness Degrees*)

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

This definition provides a relationship between mass assignments and appropriateness degrees. For example, $\mu_{young}(30) = m_{30}(y) + m_{30}(y,m) = 1$. Clearly μ_L is a function from Ω into [0,1] and therefore can technically be viewed as a fuzzy set. 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 degree' is partly because it more accurately reflects the underlying semantics and partly to highlight the quite distinct calculus based on this framework [4].

Based on the underlying semantics, we can translate a set of numeric data into a set of linguistic data, where each data value is replaced by a mass assignment label set. We need to make some assumptions for this translation. The first one is *consonance assumption*, according to which we can determine the mass assignment uniquely from the appropriateness degrees as follows. (For the justification of the consonance assumption in see [5])

Definition 3 (*Consonance Assumption*) *Let* $\{\beta_1, \beta_2, \cdots, \beta_k\} = \{\mu_L(x) | L \in LA, \mu_L(x) > 0\}$ *ordered such that* $\beta_t > \beta_{t+1}$ *for* $t = 1, 2, \cdots, k-1$ *then:*

$$m_x = M_t : \beta_t - \beta_{t-1}, t = 1, 2, \cdots, k-1,$$

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

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

Based on this assumption, there is a unique mass assignment for a given set of appropriateness degree values. For example, given $\mu_{L_1} = 0.3$ and $\mu_{L_2} = 1$, the only unique consonant mass assignment is $\{L_2\} : 0.7, \{L_1, L_2\} : 0.3$. However, it is undesirable to have mass associated with the empty set. In order to avoid this, we define a *full fuzzy covering* of the continuous universe.

Definition 4 (*Full Fuzzy Covering*) *Given a continuous discourse* Ω*,* LA *is called a full fuzzy covering of* Ω *if:*

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

Suppose we use N_F fuzzy sets with 50% overlap, so that the appropriateness degrees satisfy: $\forall x \in \Omega$, $\exists i \in \{1, \cdots, N_F - 1\}$ such that $\mu_{L_i}(x) = 1$, $\mu_{L_{i+1}} = \alpha$ and $\mu_{L_j}(x) = 0$ for $j < i$ or $j > i+1$. In this case,

$$m_x = \{L_i\} : 1 - \alpha, \{L_i, L_{i+1}\} : \alpha$$

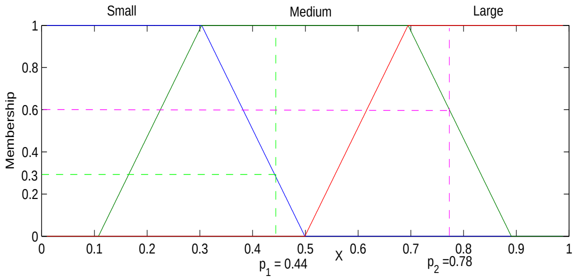

Figure 1: An example of a full fuzzy covering with 3 trapezoidal fuzzy sets with 50% overlap.

Example 1 *Figure 1 shows a full fuzzy covering of the universe with three fuzzy labels:* small*,* medium *and* large*. For the data point* $P_1 = 0.44$*, the appropriate labels are* small *and* medium (med.)*, and the appropriateness degrees of these labels are:*

$$\mu_{small}(0.44) = 0.3, \qquad \mu_{med.}(0.44) = 1$$

The mass assignment on appropriate labels is:

$$m_{0.44} = \{med.\} : 0.7, \{small, med.\} : 0.3$$

It is also interesting to note that given definitions for the appropriateness degrees on labels we can isolate a set of subsets of LA as the only values of D_x with non-zero probability. These are referred to as *focal sets*:

Definition 5 (*Focal Set*) *The focal set of LA is a set of focal elements defined as:*

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

Based on our assumption of a full fuzzy covering with 50% overlap, the following focal elements occur in example 1: {small}, {small,medium}, {medium}, {medium, large} and {large}. Since *small* and *large* do not overlap, the set {small, large} cannot occur. We can then always find the unique translation from a given data point to mass assignment on focal elements, specified by the function of μ_L; we call this the *linguistic translation (LT)*.

Definition 6 (*Linguistic Translation*) *Suppose we are given a data set* $D = \{x_1(i), \cdots, x_n(i) | i = 1, \cdots, N\}$ *with N examples and focal set on attribute j: $\mathcal{F}_j = \{F_j^1, \cdots, F_j^h | j = 1, \cdots, n\}$. (Here we assume to have same size of focal sets h for each attribute). By the linguistic translation, we then obtain linguistic data set LD defined as follow:*

$$LD = \{A_1(i), \cdots, A_n(i) | i = 1, \cdots N\}$$

$$A_j(i) = \{< m_{x_j(i)}(F_j^1), \cdots, m_{x_j(i)}(F_j^h) >\}$$

where $m_{x_j(i)}(F_j^r)$ is the associated mass of focal element F_j^r for data element $x_j(i)$ where $r = 1, \cdots, h$ and $j = 1, \cdots, n$.

For a particular attribute with an associated focal set, linguistic translation is a process of replacing data elements with masses of focal element masses these data elements. For example, consider the figure 1, the linguistic translation can be illustrated as follows.

$$\left(\begin{array}{c} Data \\ \hline 0.44 \\ 0.78 \end{array}\right) \overset{LT}{\rightarrow} \left(\begin{array}{ccccc} \{s\} & \{s,m\} & \{m\} & \{m,l\} & \{l\} \\ \hline 0 & 0.3 & 0.7 & 0 & 0 \\ 0 & 0 & 0 & 0.6 & 0.4 \end{array}\right)$$

3 Linguistic Decision Tree

The ID3 [7] algorithm for decision trees induction has proved to be an effective and popular algorithm for building decision trees from discrete valued data sets. However, it cannot cope with classification problems with continuous attribute values. The C4.5 [8] algorithm was proposed as a successor to ID3 in which an approach to crisp partitioning of continuous universe was adopted. The use of crisp partitions can be problematic since sudden and inappropriate behavior resulting from small changes to inputs will reduce the generalization capability and stability of the system. Here we propose a new decision tree induction algorithm based on label semantics that can overcome this problem. Consider a linguistic database (see Def. 6) LD with N instances. Each instance has n attributes and is labeled by one of the classes: $\{C_1, \cdots, C_m\}$. A linguistic decision tree is a decision tree where the nodes are the random sets and the branches correspond to particular focal elements. More formally:

Definition 7 (*Linguistic Decision Tree*) *A linguistic decision tree is a set of branches* *with associated class probabilities of the following form:*

$$LDT = \{< B_1, Pr(C_1|B_1), \cdots, Pr(C_m|B_1) > \\ , \cdots < B_s, Pr(C_1|B_s), \cdots, Pr(C_m)|B_s) >\}$$

and a branch B with k nodes is defined as:

$$B =< F^1_{j_1}, \cdots, F^k_{j_k} >$$

where, $k \leq n$ and $F^i_j \in \mathcal{F}_j$ for $i = 1, \cdots, k$.

A LDT is defined based on the form of the linguistic data set. Each node splits into branches according to the focal elements of this node (attribute). One attribute is not allowed to appear more than once in a branch, an attribute which is not currently part of a branch referred as a *free attribute*. The *length of a branch*, the number of consisting nodes (attributes), is less than or equal to n, the number of attributes. In a LDT, the length of the longest branch is called the *depth of the LDT*, which is also less than or equal to n.

Each branch has associated class probabilities. For example, consider the branch:

$$<< \{small_1, medium_1\}, \{big_2\} >, 0.3, 0.7 >$$

in a binary classification problem. This means the probability of class C_1 is 0.3 and C_2 is 0.7 given attribute 1 can be described as *small* & *medium* and attribute 2 can only be described as *big*.

3.1 Evaluating Class Probabilities Given a Branch

According to the definition of LDT (def. 7), if given a branch of a LDT in the form of

$$B =< F^1_{j_1}, \cdots, F^k_{j_k} >$$

The probability of Class C_t ($t = 1, \cdots, m$) given B can then be evaluated from LD.

$$Pr(C_t|B) = \frac{S(B, LD_t)}{S(B, LD)}$$

where,

$$S(B, LD_t) = \sum_{i \in LD_t} \prod_{r=1}^{k} m_{x_{j_r}(i)}(F^r_{j_r})$$

$$S(B, LD) = \sum_{i \in LD} \prod_{r=1}^{k} m_{x_{j_r}(i)}(F^r_{j_r}) \neq 0$$

LD_t is the subset consisting of instances belong to class t. In the case of $S(B, LD) = 0$, which happens when we use a small database for training a LDT, the given branch has no corresponding non-zero linguistic data. We obtain no information from the given dataset so equal probabilities are assigned to each class.

$$Pr(C_t|B) = \frac{1}{m} \qquad if: \quad S(B, LD) = 0$$

where m is the number of classes.

3.2 Evaluating Class Probabilities Given a Data Element

Consider classifying a given data element in the form of $\vec{y} =< y_1, y_2, \cdots, y_n >$ which may not be contained in the training data set D. Firstly, we need to translate $\vec{y}$ to linguistic data based on the fuzzy covering of the training data. 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 depends on which side of the range it appears. Then, by Jeffrey's rule we can evaluate the probabilities of class C_t given a LDT with s consisting branches as follows,

$$Pr(C_t|\vec{y}) = \sum_{v=1}^{s} Pr(B_v|\vec{y}) Pr(C_t|B_v)$$

where

$$Pr(B|\vec{y}) = \prod_{r=1}^{k} m_y(F^r_{j_r})$$

3.3 Classification Under Linguistic Constraints

The linguistic model has the advantage of allowing for data to be classified when some background knowledge about attributes are available in the form of *linguistic constraints*. Linguistic constraints are represented by compound label expressions. We interpret the main logical connectives in the following manners: $\neg L$ 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. The linguistic constraints take the form of $\theta =< x_1 = LS_1, \cdots, x_n = LS_n >$, where LS represents a set of expressions generated by application of the connectives to the labels. More generally, 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 $\lambda(\theta)$.

Definition 8 (*Possible Label Sets*) *Let θ and ψ be expressions generated by recursive application of the connectives $\neg, \vee, \wedge$ and $\rightarrow$ to the elements of LA. Then the set of possible label sets defined by a linguistic expression can be determined recursively as follows:*

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

Intuitively, $\lambda(\theta)$ corresponds to those subsets of LA identified as being possible values of D_x by expression θ. In this sense the imprecise linguistic restriction 'x is θ' on x corresponds to the strict constraint $D_x \in \lambda(\theta)$ on D_x.

Example 2 *Given a continuous variable x and $LA = \{small, medium, large\}$, and we are told '$x$ is **not large** but it is **between small and medium**.' It can be interpreted into a logical expression*

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

According to Definition 8, possible label sets of the given linguistic constraint θ_x is

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

Consider the vector of linguistic constraints $\vec{\theta} =< \theta_1, \cdots, \theta_n >$, where θ_j is the linguistic constraints on attribute j. We can evaluate a probability value for C_t conditional on this information using a given linguistic decision tree as follows:

$$\forall F_j \in \mathcal{F}_j \quad m_{\theta_j} = \frac{pm(F_j)}{\sum_{F_j \in \lambda(\theta_j)} pm(F_j)}$$
$$= 0 \quad otherwise$$

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:

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

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

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

Then for branch B

$$Pr(B|\vec{\theta}) = \prod_{j=1}^{k} m_{\theta_j}(F_j)$$

and therefore , by Jeffrey's rule

$$Pr(C_t|\vec{\theta}) = \sum_{v=1}^{s} Pr(C_t|B_v) Pr(B_v|\vec{\theta})$$

Consider the Example 2, if the prior mass assignment is

$$\{small\} : 0.2, \{small, medium\} : 0.1, \{medium\} : 0.2, \{medium, large\} : 0.15, \{large\} : 0.35.$$

With the previous given linguistic constraints θ_x, we then obtain:

$$m_{\theta_x} = \{small\} : 0.2/(0.2 + 0.2 + 0.1) = 0.4, \{small, medium\} : 0.2/(0.2 + 0.2 + 0.1) = 0.4, \{medium\} : 0.1/(0.2 + 0.2 + 0.1) = 0.2, \{medium, large\} : 0, \{large\} : 0$$

Based on this method, we can classify *fuzzy data* with a LDT. Compared to linguistic data, fuzzy data is represented only by a set of appropriate labels, but without associated masses on labels. E.g., in the Example 1, P_1 can be interpreted as a fuzzy data $\{small\ \&\ medium\}$. This is equivalent to given a linguistic constraint $\theta_{P_1} = (small \wedge medium)$, according to which we can classify it using the method described above.

4 LID3 Algorithm

Linguistic ID3 (LID3) is the learning algorithm for building a linguistic decision tree. As with ID3, search is guided by an information based heuristics, but the information measurements of LDT are modified in accordance with label semantics. The underlying search heuristic is based on the measure of information defined for a branch B and can be viewed as an extension of entropy equation of the ID3 algorithm:

Definition 9 (*Branch Entropy*) *The entropy of branch B is given by*

$$E(B) = -\sum_{t=1}^{m} Pr(C_t|B)\log_2(Pr(C_t|B))$$

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

Definition 10 (*Expected Entropy*)

$$EE(B, x_j) = \sum_{F_j^r \in \mathcal{F}_j} E(B \cup F_j^r) \cdot Pr(F_j^r|B)$$

where $B \cup F_j^r$ represents adding the node F_j^r to branch B. The probability of F_j^r given B can be calculated as follows:

$$Pr(F_j^r|B) = \frac{S(B \cup F_j^r|LD)}{S(B, LD)}$$

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

$$IG(B, x_j) = E(B) - EE(B, x_j)$$

As with ID3 learning, the most informative attribute will form the root of a linguistic decision tree, and the tree will expand into branches associated with all possible focal elements of this attribute. For each branch, the free attribute with maximum information gain will be the next node, from level to level, until the tree reaches the maximum depth or other termination conditions are satisfied.

Table 1: Description of datasets.

Dataset	Cases	Classes	Features
Breast-w	699	2	9
Diabetes	768	2	8
Glass	214	6	9
Iris	150	3	4
Ionosphere	351	2	34
Sonar	208	2	60
Wine	178	3	14

5 Experimental Studies

In this section, we present a number of examples showing how the LDT model performs on real world problems. Table 1 gives the descriptions of the datasets we used for the experiments. These datasets are from UCI machine learning repository [2].

5.1 Accuracy Comparisons

Here in this paper, attributes are discretized uniformly by 2 trapezoidal fuzzy sets with 50% overlap, and sub-classes are evenly splited into two sub-datasets, one half for training and the other half for testing (50-50 split). We ran LID3, C4.5, Naive Bayes Learning and Neural Network[1] with 10 runs of 50-50 split experiments on each dataset and the average accuracies with standard deviation are shown in the table 5. We then do the paired t-tests [6] with 95% confidence to compare LID3 with other 3 models. If LID3 wins, we mark the corresponding data set with $\surd$. For data set without marking, it means that LID3 has equivelent (not significant better or worse) accuracy according to the t-test.

Comparing with C4.5, LID3 obtained significant better (confidence level is greater than 95%) results on 5 datasets of 7, and LID3 also performs better though not statistically significant on other two datasets. Comparing with Naive Bayes learning, LID3 is significant better on 4 datasets. Comparing with

[1] WEKA[10] is used to generate the results of J48 (C4.5 in WEKA version) unpruned tree, Navie Bayes Learning and Nerual Network with default parameter settings.

Table 2: Accuracy comparisons between LID3 and three other learning algorithms. Where each attributes are uniformly discretized by 2 fuzzysets.

	Results from 10 runs of 50%-50% split experiments (%)				Whether LID3 wins		
Data	**C4.5**	**N.B.**	**N.N.**	**LID3**	**C4.5**	**N.B.**	**N.N.**
Breast-w	94.38 ± 1.42	96.28 ± 0.73	94.95 ± 0.80	96.00 ± 0.65	√		√
Diabetes	72.16 ± 2.80	75.05 ± 2.37	74.64 ± 1.41	76.22 ± 1.81	√		
Glass	64.77 ± 5.10	45.98 ± 7.00	64.30 ± 3.38	66.06 ± 3.89		√	
Ionosphere	89.13 ± 2.13	82.97 ± 2.50	87.78 ± 2.88	88.98 ± 2.23		√	
Iris	93.46 ± 3.23	94.53 ± 2.62	95.87 ± 2.70	96.53 ± 1.29	√	√	
Sonar*	70.48 ± 0.00	70.19 ± 0.00	81.05 ± 0.00	86.54 ± 0.00	√	√	√
Wine	88.09 ± 4.14	96.29 ± 2.12	96.85 ± 1.57	95.33 ± 1.80	√		

* A particular single split of the original dataset is used, so, the standard deviation is 0.

Neural Network, LID3 performs significantly better only on 2 datasets and equivelent on 5 datasets. So far by our experiments, we can say that the LDT model has comparable accuracy to Neural Network and performs better than C4.5 and Naive Bayes learning.

5.2 Linguistic Constraints Testing on the 'Eight' Problem

The 'eight' problem is a toy problem defined as follows: A figure of eight shape was generated according to the equation $x = 2^{(-0.5)}(sin(2t) - sin(t))$ and $y = 2^{(-0.5)}(sin(2t)+sin(t))$ where $t \in [0, 2\pi]$. (See figure 2). Points in $[-1.6, 1.6]^2$ are classified as legal if they lie within the 'eight' shape (marked with ×) and illegal if they lie outside (marked with points). The database consisted of 961 examples generated from a regular grid on $[-1.6, 1.6]^2$ for training, and 961 unseen examples from the same distribution as the test dataset.

Example 3 *Suppose a LDT is trained on the 'Eight' database where each attribute is discretized by five fuzzy sets uniformly:* very small (vs)*,* small (s)*,* medium (m)*,* large (l) *and* very large (vl)*. Further, suppose we are given the following description of data points:*
$\theta_1 =< x = vs \vee s \wedge \neg m, y = vs \vee s \wedge \neg m >$
$\theta_2 =< x = m \wedge l, y = s \wedge m >$
$\theta_3 =< x = s \wedge m, y = l \vee vl >$

Probabilities of illegal ($\cdot$) and legal ($\times$) given the corresponding linguistic constraints are:

$$\begin{array}{ll} Pr(\cdot|\theta_1) = 1.000 & Pr(\times|\theta_1) = 0.000 \\ Pr(\cdot|\theta_2) = 0.000 & Pr(\times|\theta_2) = 1.000 \\ Pr(\cdot|\theta_3) = 0.428 & Pr(\times|\theta_3) = 0.572 \end{array}$$

As we can see from figure 2, the above 3 linguistic constraints are roughly correspond to the area 1, area 2 and area 3, respectively. By examining the occurrence of legal and illegal examples, we verified the correctness of our results.

As we have discussed in section 3.3, we test our model based on fuzzy data (FD), on the training set of the 'eight' problem and obtain results that are shown in table 3, together with the results based on linguistic data (LD). Even without associated masses, our model still gives a reasonable approximating of the legal data area, though it is not as accurate as testing on linguistic data, for example, see the figure 3 and 4. The accuracy increases with N_F the number of fuzzy sets used for discretization, and obviously, the model works uniformly better on linguistic data than on fuzzy data. This example shows that LDT model still can perform well in dealing with real fuzzy and ambiguous data.

Table 3: Classification accuracy comparisons based on linguistic data and fuzzy data without masses on the 'eight' problem.

N_F	3	4	5	6	7
LD	87.72%	94.17%	95.94%	97.29%	98.54%
FD	79.29%	85.85%	89.39%	94.17%	95.01%

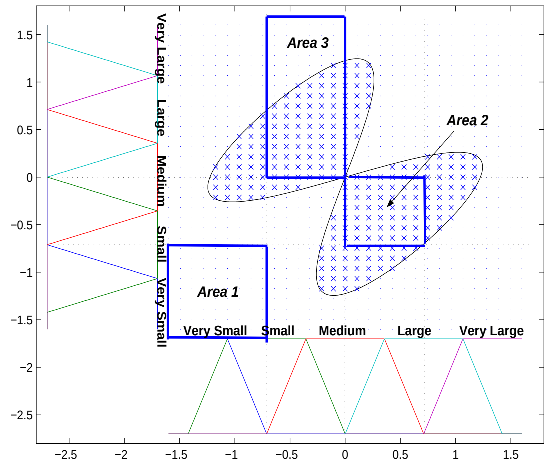

Figure 2: Testing on the 'eight' problem with linguistic constraints, where each attribute is discretized by 5 fuzzy sets.

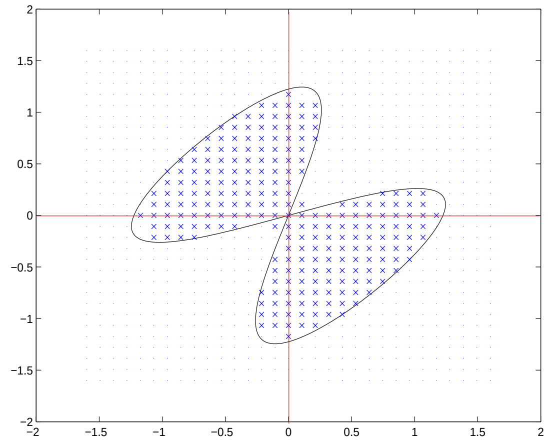

Figure 3: Classification on linguistic dataset, where each attribute is discretized uniformly by 7 fuzzy sets.

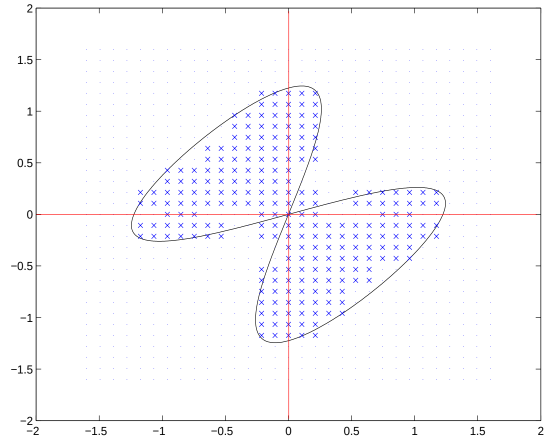

Figure 4: Classification on fuzzy data without masses, where each attribute is discretized uniformly by 7 fuzzy sets.

6 Conclusion

In this paper, we proposed a new decision tree learning algorithm based on label semantics and its performance on some UCI datasets was studied. The model has comparable classification accuracy to the Neural Network and better than C4.5 and Naive Bayes learning with statistical significance on datasets listed in table 1. The test of classifying under linguistic constraints on a toy problem shows validity of our approach and it is feasible for building a transparent machine learning system with this approach.

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] J. Lawry. Query Evaluation from Linguistic Prototypes, *Proceedings of 10'th IEEE International Conference on Fuzzy Systems*, 2001.

[4] J. Lawry. Label Semantics: A formal framework for modeling with words. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Lecture Notes in Artificial Intelligence: 374-384*, Springer-Verlag, 2001.

[5] J. Lawry. A framework for linguistic modelling, to appeared in *Artificial Intelligence*.

[6] T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[7] J.R. Quinlan. Induction of decision trees. *Machine Learning* 1: 81-106. 1986

[8] J.R. Quinlan. *C4.5: Programs for machine learning*. San Mateo: Morgan Kaufmann, 1993.

[9] N.J. Randon and J. Lawry. Linguistic modeling using a semi-Naive Bayes Framework. *IPMU-2002*, Annecy, France, 2002.

[10] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999. http://www.cs.waikato .ac.nz/∼ ml/weka/

[11] L.A.Zadeh. Fuzzy logic = computing with words. *IEEE Transaction on Fuzzy Systems*. Vol. 4, No. 2: 103-111.