Fuzziness and Performance: An Empirical Study with Linguistic Decision Trees

Zengchang Qin[1] and Jonathan Lawry[2]

[1] Berkeley Initiative in Soft Computing
Computer Science Division, EECS Department
University of California, Berkeley CA 94720, USA
zqin@cs.berkeley.edu
[2] Artificial Intelligence Group
Engineering Mathematics Department
University of Bristol, BS8 1TR, UK
j.lawry@bris.ac.uk

Abstract. Generally, there are two main streams of theories for studying uncertainties. One is probability theory and the other is fuzzy set theory. One of the basic ideas of fuzzy set theory is how to define and interpret membership functions. In this paper, we will study tree-structured data mining model based on a new interpretation of fuzzy theory. In this new theory, fuzzy labels will be used for modelling. The membership function is interpreted as appropriateness degrees for using labels to describe a fuzzy concept. Each fuzzy concept is modelled by a distribution on the appropriate fuzzy label sets. Previous work has shown that the new model outperforms some well-known data mining models such as Naive Bayes and Decision trees. However, the fuzzy labels used in previous works were predefined. We are interested in study the influences on the performance by using fuzzy labels with different degrees of overlapping. We test a series of UCI datasets and the results show that the performance of the model increased almost monotonically with the increase of the overlapping between fuzzy labels. For this empirical study with the LDT model, we can conclude that more fuzziness implies better performance.

1 Introduction

Uncertainty is a nature of our world. Generally, there are two main streams for modelling uncertainties. One is probability theory and the other is fuzzy set theory. Since the first paper published by Zadeh in 1965 [9], fuzzy logic has become an important branch in artificial intelligence as well as some engineering areas such as intelligent control. One of the basic ideas of fuzzy set theory is how to define and interpret membership functions. There are a few different interpretation of fuzziness [8]. In this paper, we will study tree-structured data mining model based on a new interpretation of fuzzy theory. In this new theory, which is referred to as Label Semantics [2], fuzzy labels will be used for modelling.

P. Melin et al. (Eds.): IFSA 2007, LNAI 4529, pp. 407–416, 2007.

One inherent disadvantage of classical decision trees is that the model is sensitive to noise. As pointed out by Quinlan [6]: "the results of (traditional) decision trees are categorical and so do not convey potential uncertainties in classification. Small changes in the attribute values of a case being classified may result in sudden and inappropriate changes to the assigned class. Missing or imprecise information may apparently prevent a case being classified at all". This noise is not only due to the lack of precision or errors in measured features but is often present in the model itself since the available features may not be sufficient to provide a complete model of the system. To overcome this problem, some probabilistic or soft decision trees were proposed. The first fuzzy decision tree reference can be back to in 1977. Since then, There are more than forty references on either on fuzzy tree learning or fuzzy rule learning. All these algorithms highlight advantage of using fuzzy rules for classification applications is to maintain transparency as well as a high accuracy rate. According to Olaru and Wehenkel [3]: these fuzzy decision tree algorithms can be roughly divided into two categories:

1. Enable the use of decision trees to manage fuzzy information in the forms of fuzzy inputs, fuzzy classes or fuzzy rules.
2. Using fuzzy logic to improve their predictive accuracy.

Previous work by Lawry and Qin [4] has shown that the LDT model outperforms some well-known data mining models such as Naive Bayes and classical decision trees such as C4.5 [7]. It also can handle fuzzy information and has better transparency comparing to other models. However, the fuzzy labels used in previous works were predefined under some assumptions. We are interested in study the influences of different degrees of overlapping between neighboring fuzzy labels.

2 Linguistic Decision Trees

Linguistic decision tree (LDT) [4] is a tree-structured classification model based on label semantics. The information heuristics used for building the tree are modified from Quinlan's ID3 [5] in accordance with label semantics. The nodes of a LDT are linguistic descriptions of variables and leaves are sets of appropriate labels. In such decision trees, the probability estimates for branches across the whole tree is used for classification, instead of the majority class of the single branch into which the examples fall. Linguistic expressions such as *small*, *medium* and *large* are used to learn from data and build a linguistic decision tree guided by information based heuristics. For each branch, instead of labeling it with a certain class (such as positive or negative in binary classification) 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.

2.1 Introduction to Label Semantics

Label semantics is a methodology of using linguistic expressions or fuzzy labels to describe numerical values. For a variable x into a domain of discourse Ω we identify a finite set of fuzzy labels $\mathcal{L} = \{L_1, \cdots, L_n\}$ with which to label the values of x. Then for a specific value $x \in \Omega$ an individual I identifies a subset of $\mathcal{L}$, denoted D_x^I to stand for the description of x given by I, as the set of labels with which it is appropriate to label x. If we allow I to vary across a population V with prior distribution P_V, then D_x^I will also vary and generate a random set denoted D_x into the power set of $\mathcal{L}$ denoted by $\mathcal{S}$. We can view the random set D_x as a description of the variable x in terms of the labels in $\mathcal{L}$. The frequency of occurrence of a particular label, say S, for D_x across the population then gives a distribution on D_x referred to as a mass assignment on labels. More formally,

Definition 1 (*Label Description*). *For $x \in \Omega$ the label description of x is a random set from V into the power set of $\mathcal{L}$, denoted D_x, with associated distribution m_x, which is referred to as mass assignment:*

$$\forall S \subseteq \mathcal{L}, \quad m_x(S) = P_V(\{I \in V | D_x^I = S\}) \tag{1}$$

where $m_x(S)$ is called associated mass of S and $\sum_{S \subseteq \mathcal{L}} m_x(S) = 1$. Intuitively mass assignment is a distribution on appropriate label sets and $m_x(S)$ quantifies the evidence that S is the set of appropriate labels for x.

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

Algorithm 1. Linguistic translation

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

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

quite distinct calculus based on this framework [2]. This definition provides a relationship between mass assignments and appropriateness degrees.

Definition 2 (*Appropriateness Degrees*)

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

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. These assumptions are fully described in [4] and justified in [2]. These assumptions guarantee that there is unique mapping from appropriate degrees to mass assignments on labels. For example, given $\mu_{middleAged}(30) = 0.3$ and $\mu_{young}(30) = 1$ which are the memberships of being *middleAged* and *young* given a value of 30 (a person's age). The corresponding mass assignment is: $m_{30} = \{young, middleAged\} : 0.3, \{young\} : 0.7$ (More details of mass assignment calculations are available in [2] and [4]). Given a database, we can translate each data element into its mass assignment representation. This process is called *linguistic translation*. The pseudo-code is given in algorithm 1.

2.2 Degrees of Overlapping

Through linguistic translation, all numerical data can be represented as mass assignments based on a predefined fuzzy discretization method. In this paper, unless otherwise stated, we will use a percentile-based (or equal points) discretization. The idea is to cover approximately the same number of data points for each fuzzy label. The justification for using this discretization method is given in [4].

Basically, fuzzy discretization provides an interpretation between numerical data and their corresponding linguistic data based on label semantics. We may notice that different fuzzy discretization (fuzzification of a continuous universe) may result in different linguistic data. We introduce a new parameter PT by which to measure the degrees of overlapping between fuzzy labels. As we can see from figure 1, given two fuzzy labels F and G, m is the distance between the weighting centers of a fuzzy labels to the meeting point of their membership functions. a is actually the length of the overlapping area. PT is calculated as follows:

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

$PT = 0.5$ represents 50% of overlapping between each two neighboring fuzzy labels (e.g., figure 1-A). $PT = 0$ represents no overlapping at all (figure 1-C), i.e., the labels are discrete but not fuzzy. Figure 1-B shows a situation that the degree of overlapping is between 0 and 0.5. Figure 1-D also shows the linear relation of parameter a and PT.

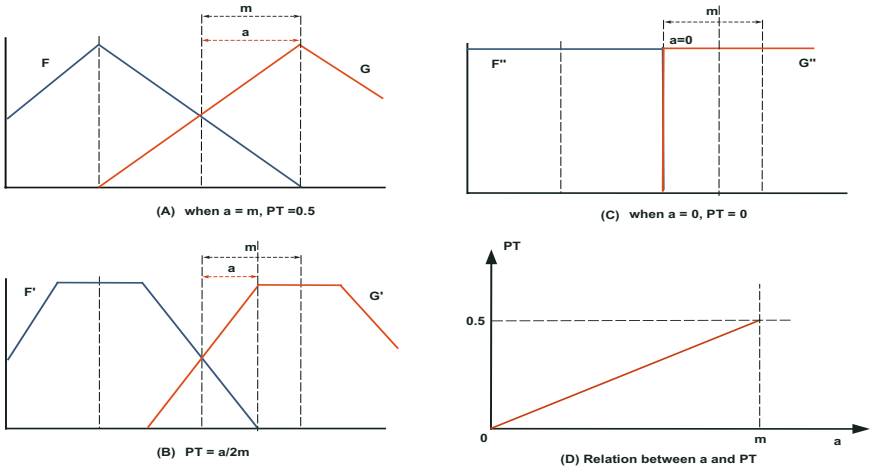

Fig. 1. A schematic illustration of calculating the overlap parameter PT given different degrees of overlaps

2.3 Classification

Given a database of which each instance is labeled by one of the classes: $\mathcal{C} = \{C_1, \cdots, C_{|\mathcal{C}|}\}$. A linguistic decision tree with S branches built from this database can be defined as follows:

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

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 are *focal elements* of attribute j. Focal elements are the appropriate label sets with non-zero masses [2]. 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*.

Given a training set $\mathcal{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ where each instance $\mathbf{x}$ has n attributes: $\langle x_1, \cdots, x_n\rangle$. The class probability of C_k given a particular branch B is calculated by the proportion of data covered by this branch and belonging to C_k to all the data covered by this branch:

$$P(C_k|B) = \frac{\sum_{i\in\mathcal{D}_k} P(B|\mathbf{x}_i)}{\sum_{i\in\mathcal{D}} P(B|\mathbf{x}_i)} \tag{3}$$

where $\mathcal{D}_k = \sum_{i:\mathbf{x}_i \to C_k} \mathbf{x}_i$ is the subset consisting of instances which belong to class k. The probability of a branch B given $\mathbf{x}$ can be regarded as the proportion of the data $\mathbf{x}$ covered by branch B and it is evaluated by:

$$P(B|\mathbf{x}) = \prod_{j=1}^{d} m_{x_j}(F_j) \tag{4}$$

where $m_{x_j}(F_j)$ for $j = 1, \cdots, d$ are mass assignments of the single data element x_j. Now consider classifying an unlabelled instance in the form of $\mathbf{y} = \langle y_1, \cdots, y_n \rangle$ from the test set. First we apply linguistic translation to $\mathbf{y}$ based on the fuzzy covering of the training data $\mathcal{D}$. According to the Jeffrey's rule the probabilities of class C_k given a LDT with S branches are evaluated as follows:

$$P(C_k|\mathbf{y}) = \sum_{s=1}^{S} P(C_k|B_s)P(B_s|\mathbf{y}) \tag{5}$$

where $P(C_k|B_s)$ and $P(B_s|\mathbf{x})$ are evaluated based on equations 3 and 4.

Algorithm 2. Decision Tree Learning

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

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

2.4 LID3 Algorithm

Linguistic ID3 (LID3) is the learning algorithm proposed for building the linguistic decision tree. Similar to the ID3 algorithm [5], search is guided by an

information based heuristic, but the information measurements of a LDT are modified in accordance with label semantics. The measure of information defined for a branch B and can be viewed as an extension of the entropy measure used in ID3. The branch entropy of a branch B is given by

$$E(B) = -\sum_{k=1}^{|\mathcal{C}|} P(C_k|B) \log_2(P(C_k|B)) \tag{6}$$

where $|\mathcal{C}|$ is the number of classes. Now, given a particular branch B suppose we want to expand it with the attribute x_j. The evaluation of this attribute will be given based on the expected entropy defined as follows:

$$EE(B, x_j) = E[E(x_j|B)] = \sum_{F_j \in \mathcal{F}_j} E(B \cup F_j) P(F_j|B) \tag{7}$$

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:

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

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

$$IG(B, x_j) = E(B) - EE(B, x_j) \tag{9}$$

The pseudo-code of the LID3 algorithm are shown in Algorithm 2.

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

Dataset	Classes	Size	Attributes	Dataset	Classes	Size	Attributes
Balance	3	625	4	**Breast-cancer**	2	286	9
Ecoli	8	336	8	**Glass**	6	214	9
Heart-C	2	303	13	**Heart-S**	2	270	13
Heptitis	2	155	19	**Iris**	3	150	4
Liver	2	345	6	**Pima**	2	768	8
Wcancer	2	699	9	**Wine**	3	178	14

3 Experiments

In this section, we investigate the influences of overlapping degrees on the accuracy by some empirical studies. First of all, we need to specify the parameter settings for the LDT model. In the following experiments, we use 3 trapezoidal fuzzy sets for discretization (i.e., Alg. 1 line 2: $N_F = 3$). Probability threshold $T = 1$ (Alg. 2 line 1) and we set $M_{dep} = n$ in order to develop a complete LDT

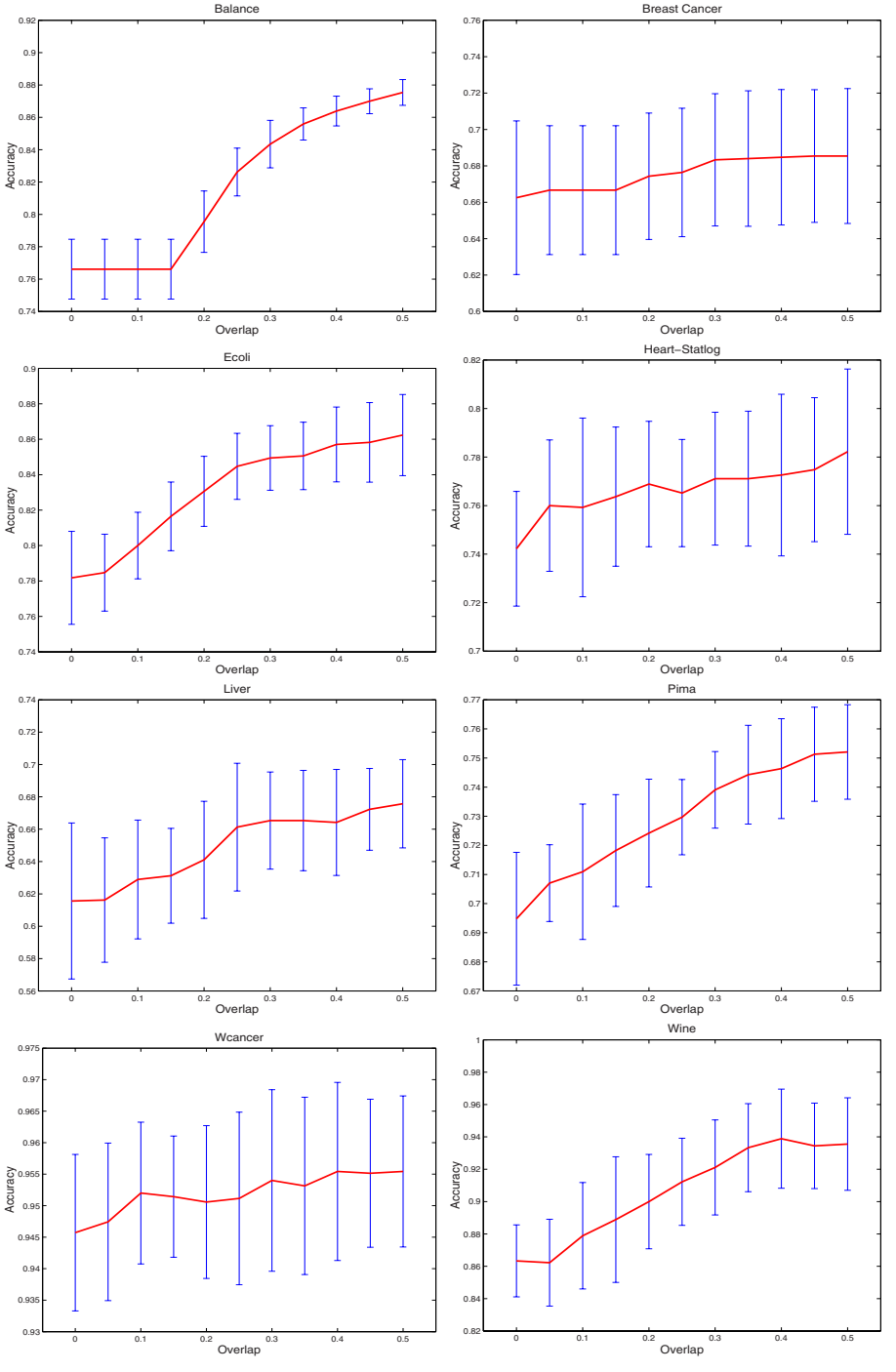

Fig. 2. Monotonically increased performance

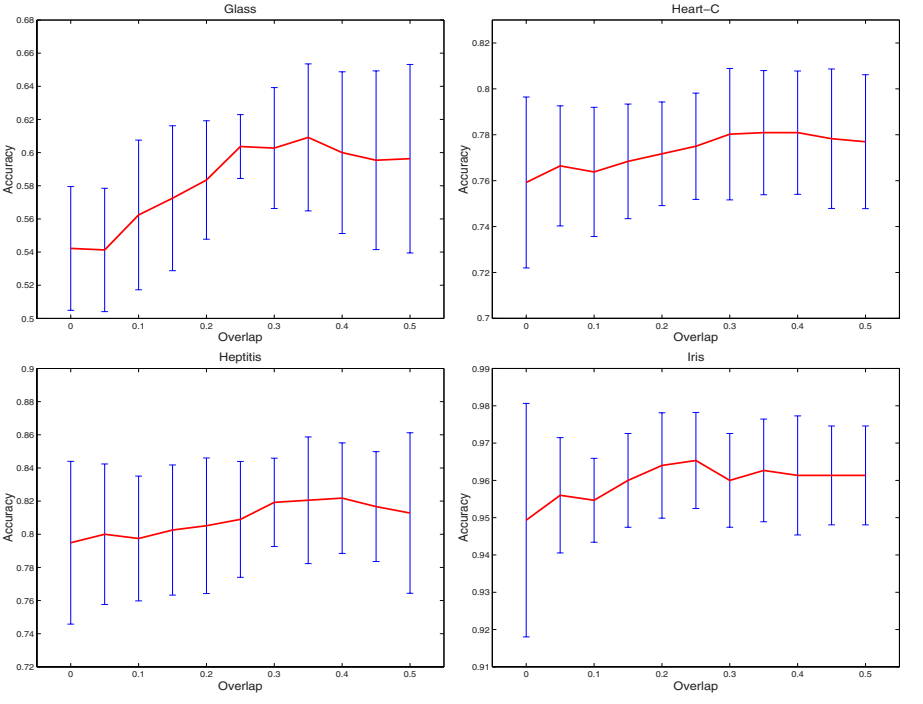

Fig. 3. More overlapping does not guarantee the better performance for these datasets

(the growth of LDT will be stopped if all attributes have been used, see Alg 2 line 11). These settings are justified in [4]. We tested 12 datasets taken from UCI [1] machine learning repository. For each experiment, the dataset is partitioned into two parts that the data belonging to the same class are evenly split. One part of the data is for training and the other for test. We will randomly do the split for 10 times and the average results with standard deviation will be calculated. This is referred to as 50-50 split experiments [4]. The experimental results on the given data sets are shown in figures 2 and 3, respectively.

As we can see from these figures, the performance of 8 of the 12 datesets are roughly monotonic increased with the increase of PT. It implies that more fuzziness tends to increase the robustness of the LDT model and get better performance. However, from the results in figure 3, we can tell that more overlapping does not guarantee the better performance. For some datasets, 30% of overlapping maybe is enough. More overlapping would not be necessary and it may give worse results sometime. From all the results, we can see that LDTs with fuzzy labels generally outperform the ones with discrete labels (where $PT = 0$). Therefore, in summary, for the case of LDT model, we can say that fuzziness will bring greater performance. The increase is almost monotonically. But the optimal overlapping degrees are depends on the dataset you tested.

4 Conclusions

In this paper, we extended the previous work on linguistic decision trees to study the influences on performance by using fuzzy labels with different degrees of overlapping. We tested the LDT model on a series of UCI datasets and the results show that the performance increased almost monotonically with the increase of the overlapping between fuzzy labels. For this empirical study with the LDT model, we can conclude that more fuzziness does imply better performance. However, the optimal overlapping degrees are depends on datasets.

Acknowledgements

Qin is British Telecommunications (BT) Research Fellow in BISC Group. This research was partly funded by BT/BISC Fellowship.

References

1. C. Blake and C.J. Merz. UCI machine learning repository, *http://www.ics.uci.edu/∼mlearn/ MLRepository.html*.
2. J. Lawry, *Modelling and Reasoning with Vague Concepts*, Springer, 2006.
3. C. Olaru and L. Wehenkel, A complete fuzzy decision tree technique, *Fuzzy Sets and Systems*, Vol. 138: pp.221-254, 2003.
4. Z. Qin and J. Lawry. Decision tree learning with fuzzy labels, *Information Sciences*, Vol. 172/1-2, pp. 91-129, 2005.
5. J. R. Quinlan. Induction of decision trees, *Machine Learning* 1: 81-106. 1986
6. J. R. Quinlan, Decision trees at probabilistic classifiers, *Proceeding of 4th International Workshop on Machine Learning*, pp. 31-37, Morgan Kauffman, 1987.
7. J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo: Morgan Kaufmann, 1993.
8. P. Wang, Interpretations on fuzziness, *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol 26(2): 321-326, 1996.
9. L. A. Zadeh, Fuzzy sets, *Information and Control*, Vol 8: pp. 338-353, 1965.