REPORT

A Fuzzy Decision Tree Algorithm Based on C4.5

Marcos E. Cintra[1], Maria C. Monard[2] and Heloisa A. Camargo[3]
[1] Exact and Natural Sciences Department – Federal Territorial University of the Semi-Arid - UFERSA – Av. Francisco Mota, 572, 59.625-900 - Mossoró - RN - Brazil
[2] Math. and Computer Science Institute - University of São Paulo - USP, Av. Trabalhador São-carlense, 400 - CEP: 13.566-590 - São Carlos - SP - Brazil
[3] Computer Science Department - Federal University of São Carlos - UFSCar,m Rod. Washington Luis, Km 235 - PO Box 676, 13565-905 - São Carlos - SP - Brazil
cintra@ufersa.edu.br, mcmonard@icmc.usp.br, heloisa@dc.ufscar.br

Decision trees have been successfully applied to many areas for tasks such as classification, regression, and feature subset selection. Decision trees are popular models in machine learning due to the fact that they produce graphical models, as well as text rules, that end users can easily understand. Moreover, their induction process is usually fast, requiring low computational resources. Fuzzy systems, on the other hand, provide mechanisms to handle imprecision and uncertainty in data, based on the fuzzy logic and fuzzy sets theory. The combination of fuzzy systems and decision trees has produced fuzzy decision tree models, which benefit from both techniques to provide simple, accurate, and highly interpretable models at low computational costs. In this paper, we expand previous experiments and present more details of the FuzzyDT algorithm, a fuzzy decision tree based on the classic C4.5 decision tree algorithm. Experiments were carried out using 16 datasets comparing FuzzyDT with C4.5. This paper also includes a comparison of some relevant issues regarding the classic and fuzzy models.

1. Introduction

Machine learning is concerned with the development of methods for the extraction of patterns from data in order to make intelligent decisions based on these patterns. A relevant concern related to machine learning methods is the issue of interpretability, which is highly desirable for end users. In this sense, Decision Trees (DT) [12] are powerful and popular models for machine learning since they are easily understandable, quite intuitive, and produce graphical models that can also be expressed as rules. The induction process of DTs is usually fast and the induced models are usually competitive with the ones generated by other interpretable machine learning methods. Also, DTs performs an embedded feature selection during their induction process.

Fuzzy systems, on the other hand, have also been successfully applied in many areas covered by machine learning. Fuzzy systems can handle uncertainty and imprecision by means of the fuzzy logic and fuzzy sets theories, producing interpretable models. A system can be considered 'fuzzy' if at least one of its attributes is defined by fuzzy sets, according to the fuzzy logic and fuzzy sets theory proposed by prof. Zadeh [9, 19]. A fuzzy system is usually composed of a Knowledge Base (KB) and an Inference Mechanism (IM). The KB contains a Fuzzy Rule Base (FRB) and a Fuzzy Data Base (FDB). The FRB has the rules that form the core of the system. These rules are constructed based on the fuzzy sets defining the attributes of the system, stored in the FDB. The FDB and FRB are used by the IM to classify new examples.

Regarding DTs, the ID3 [11], CART [1], and C4.5 [13] algorithms are among the most relevant ones. Fuzzy DTs have also been proposed in the literature [2, 7, 8, 10, 14]. Fuzzy DTs combine the powerful models of DTs with the interpretability and ability of processing uncertainty and imprecision of fuzzy systems. Moreover, fuzzy DTs inherit the desirable characteristics of DTs regarding their low computational induction cost, as well as the possibility of expressing the induced models graphically and as a set of rules.

In this work, we describe our fuzzy version of C4.5, named FuzzyDT [3], which has been applied for the induction of classifiers, presenting expanded experiments and further details on its algorithm. FuzzyDT has also been applied to a real-world problem, the prediction and control of the coffee rust disease in Brazilian crops [4]. This paper includes the experimental evaluation of FuzzyDT and C4.5 considering 16 datasets and a 10-fold cross-validation strategy.

The remainder of this paper is organized as follows. Section 2 introduces the fuzzy classification systems. Section 3 discusses DTs. The FuzzyDT algorithm is described in Section 4. Section 5 presents a comparison between classic and fuzzy DTs. Section 6 describes the experiments and comparisons, followed by the conclusions and future work in Section 7.

2. Fuzzy Classification Systems

Classification is a relevant task of machine learning that can be applied to pattern recognition, decision making, and data mining, among others. The classification task can be roughly described as: given a set of objects $E = \{e_1, e_2, ..., e_n\}$, also named *examples or cases*, which are described by m features, assign a class c_i from a set of classes $C = \{C_1, C_2, ..., C_j\}$ to an object e_p, $e_p = (a_{p_1}, a_{p_2}, ..., a_{p_m})$.

Rule-based fuzzy classification systems require the granulation of the features domain by means of fuzzy sets and partitions. The linguistic variables in the antecedent part of the rules represent features, and the consequent part represents a class. A typical fuzzy classification rule can be expressed by

$$R_k : \textbf{IF } X_1 \text{ is } A_{1l_1} \textbf{ AND } ... \textbf{ AND } X_m \text{ is } A_{ml_m}$$
$$\textbf{THEN } Class = C_i$$

where R_k is the rule identifier, $X_1, ..., X_m$ are the features of the example considered in the problem (represented by linguistic variables), $A_{1l_1}, ..., A_{ml_m}$ are the linguistic values used to represent the feature values, and $C_i \in C$ is the class. The inference mechanism compares the example to each rule in the fuzzy rule base aiming at determining the class it belongs to.

The classic and general fuzzy reasoning methods [5] are widely used in the literature. Given a set of fuzzy rules (fuzzy rule base) and an input example, the classic fuzzy reasoning method classifies this input example using the class of the rule with maximum compatibility to the input example, while the general fuzzy reasoning method calculates the sum of compatibility degrees for each class and uses the class with highest sum to classify the input example.

Next section introduces the DT algorithms.

3. Decision Trees

As previously mentioned, DTs provide popular and powerful models for machine learning. Some of the relevant characteristics of DTs include the following:

- they are easily understandable and intuitive;
- the induced model can be graphically expressed, as well as a set of rules;
- they are usually competitive with more costly approaches;
- their induction process performs an embedded feature subset selection, improving the interpretability of the induced models;
- decision trees are usually robust and scalable;
- they can handle discrete and continuous data;
- DTs can be applied to datasets including a large number of examples;
- their inference and induction process require low computational cost.

C4.5 [13] is one of the most relevant and well-known DT algorithm. A fuzzy version of the classic C4.5 algorithm was proposed in [8] (in Japanese). In this work, we present our fuzzy version of C4.5, named FuzzyDT [3].

The classic C4.5 algorithm uses the information gain and entropy measures to decide on the importance of the features, which can be numerical and/or categorical. C4.5 recursively creates branches corresponding to the values of the selected features, until a class is assigned as a terminal node. Each branch of the tree can be seen as a rule, whose conditions are formed by their attributes and respective tests. In order to avoid overfitting, C4.5, as well as most DT algorithms, includes a pruning process. Specifically, C4.5 adopts a post-pruning strategy, *i.e.*, the pruning takes place after the tree is completely induced. The pruning process basically assesses the error rates of the tree and its components directly on the set of training examples [12].

To understand the process of DT pruning, assume N training examples are covered by a leaf, E of them incorrectly. This way, the error rate for this leaf is defined by E/N. Considering this set of N training cases as a sample, it is possible to estimate the probability of error over the entire population of examples covered by this leaf. This probability cannot be precisely determined. However, it has a probability distribution that is usually summarized by a pair of confidence limits. For a given confidence level CF, the upper limit of this probability can be found from the confidence limits for the binomial distribution; this upper limit is here written as $U_{CF}(E, N)$. As the upper and lower binomial distribution limits are symmetrical, the probability that the real error rate exceeds $U_{CF}(E, N)$ is $CF/2$. As pointed out by Quinlan, although one might argue that this heuristic is questionable, it frequently yields acceptable results [12].

The default confidence limits used by C4.5 is 25%. However, it is important to notice that the smaller the confidence limit, the higher the chances of pruning, while the higher the confidence limit, the smaller the chances of pruning. Thus, if the confidence limit is set to 100%, the predicted error, obtained with the examples at hand, is defined as the real error, and no pruning is performed. This idea conflicts with the natural intuition that a 25% confidence limit will produce less pruning than an 80% confidence limit, for instance. This way, one should not associate the default 25% confidence limits of C4.5 with actually pruning 25% of the generated tree. Next we detail the FuzzyDT algorithm.

Algorithm 1: Fuzzyfication of the continuous values of the training set.

```
Input: A given dataset described by m attributes and n
       examples, and the predefined fuzzy data base;
for a = 1 to m do
    for b = 1 to n do
        if Attribute Att_a is continuous then
            for x = 1 to the total number of linguistic
            values defining Att_a do
                calculate A_{Att_a x}, as the membership degree
                of the input value of attribute Att_a, example
                b, in the fuzzy set defining the x^th linguistic
                value of attribute Att_a;
            Replace the continuous value of attribute Att_a,
            example b, with the linguistic value with highest
            membership degree with it;
```

4. The FuzzyDT Algorithm

FuzzyDT[5], proposed by us in [3], uses the same measures of the classic C4.5 algorithm (entropy and information gain) to decide on the importance of the features. It also uses the same induction strategy to recursively partition the feature space creating branches until a class is assigned to each branch. However, for FuzzyDT, continuous features are defined in terms of fuzzy sets before the

[5]FuzzyDT is available at `http://dl.dropbox.com/u/16102646/FuzzyDT.zip`

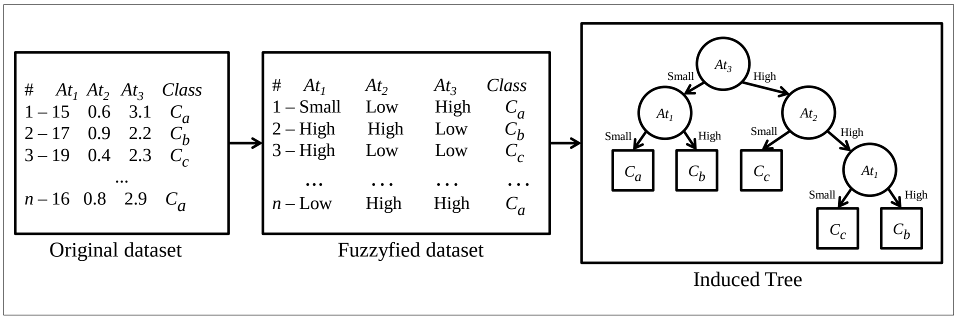

Fig 1. The FuzzyDT algorithm - a toy example

induction of the tree. This way, the process of inducing a tree using FuzzyDT takes a set of "discretized" features, *i.e.*, crispy features, since the continuous features are defined in terms of fuzzy sets and the training set is fuzzyfied before the DT induction takes place. Algorithm 1 details this fuzzyfication step.

Algorithm 2 describes FuzzyDT.

As the fuzzyfication of the training data is done before the induction of the tree, the third step of FuzzyDT corresponds to the same step of the classic DT algorithm. Figure 1 illustrates the process of data fuzzyfication and tree induction for a toy dataset with n examples, 3 attributes (At_1, At_2, and At_3), and 3 classes (C_a, C_b, and C_c).

Algorithm 2: The FuzzyDT algorithm.

1. *Define the fuzzy data base, i.e., the fuzzy granulation for the domains of the continuous features*;
2. *Replace the continuous attributes of the training set using the linguistic labels of the fuzzy sets with highest compatibility with the input values*;
3. *Calculate the entropy and information gain of each feature to split the training set and define the test nodes of the tree until all features are used or all training examples are classified*;
4. *Apply a post-pruning process, similarly to C4.5, using 25% confidence limits as default.*

The first block of Figure 1 illustrates a dataset with n examples, three attributes (At_1, At_2, and At_3) and a class attribute. The fuzzyfied version of this dataset is presented in the second block. This fuzzyfied set of examples is used to induce the final DT, illustrated in the last block of Figure 1.

It follows a detailed comparison between classic and fuzzy DTs.

5. Classic Versus Fuzzy Decision Trees

Classic and fuzzy DTs, although sharing the same basic idea of building a tree like structure by partitioning the feature spaces, also present some relevant differences. Next, we discuss some of these similarities and differences, using the classic C4.5 and FuzzyDT algorithms for the comparisons.

Evaluation of features— For the partitioning process, both versions use the same measures, entropy and information gain, in order to select the features to be used in the test nodes of the tree;

Induction process — Both versions use the same approach: repeated subdivision of the feature space using the most informative

features until a leaf node is reached or no features or examples remain;

Handling continuous features — The classic version splits the domain into crisp intervals according to the examples at hand by minimizing entropy and maximizing information gain. This process might cause unnatural divisions that reflect on a lower interpretability of the rules. As a practical illustration, let us consider the Vehicle dataset, from UCI [6], which has *Compactness* as its first test attribute of a DT induced by C4.5. *Compactness* is a continuous attribute with real values ranging from 73 to 119. For the tree induced by C4.5, it is possible to find the following tests using *Compactness*:

1. **IF** *Compactness* is ≤ 95 **AND** ... **AND** *Compactness* is ≤ 89
2. **IF** *Compactness* is ≤ 95 **AND** ... **AND** *Compactness* is > 89
3. **IF** *Compactness* is > 95
4. **IF** *Compactness* is ≤ 102
5. **IF** *Compactness* is > 102
6. **IF** *Compactness* is ≤ 109 **AND** ... **AND** *Compactness* is ≤ 106
7. **IF** *Compactness* is ≤ 109 **AND** ... **AND** *Compactness* is > 106
8. **IF** *Compactness* is > 109
9. **IF** *Compactness* is ≤ 82 **AND** ... **AND** *Compactness* is ≤ 81
10. **IF** *Compactness* is ≤ 82 **AND** ... **AND** *Compactness* is > 81
11. **IF** *Compactness* is > 82 **AND** ... **AND** *Compactness* is ≤ 84
12. **IF** *Compactness* is > 82 **AND** ... **AND** *Compactness* is > 84

These 12 tests make it difficult to understand the rules since, for a whole understanding of the model, the user has to keep in mind the subspaces defined by each condition of the rule that uses the same attribute. A particular problem happens when the subdivisions are relatively close, strongly restraining the domain of the features (rule 10: $81 <$ *Compactness* ≤ 82).

Another issue regarding the use of continuous features by C4.5 is the number of divisions used to split continuous attributes: it cannot be predefined, even if it is previously known. In fact, for the algorithm to use a previously defined number of divisions for any attribute, such attribute needs to be discretized before the induction of the DT, since the number of divisions splitting continuous attributes is dynamically determined during the tree induction process. This way, the number determined by the DT algorithm might be different from the number of divisions used by an expert, for example. Notice that in the example provided, the DT uses 8 different splitting points for the same attribute (81, 82, 84, 89, 95, 102, 106, and 109), some of them very close to each other.

FuzzyDT, on the other hand, is able to use the partitions (in terms of fuzzy sets) defined by an expert. Furthermore, even if this information is not available, automatic methods for the generation of fuzzy partitions can be used, most of them controlling and preventing the creation of unnatural splitting points.

Reuse of features — for C4.5, the same continuous feature can be included several times in one single rule (such as feature *Compactness* in the previous example). This repetition of the same feature and subdivision of the domain degrades the interpretation of the rule. On the other hand, the induction process of FuzzyDT can be seen as inducing a DT with fuzzyfied (discretized) attributes, thus a feature is never used more than once in the same rule. This fact favours the interpretability of the generated rules.

Inference — As previously stated, a special issue regarding classic DTs is the fact that they can be seen as a set of disjunct rules in which only one rule is fired to classify a new example. For fuzzy DTs, differently from the classic model, two branches are usually fired simultaneously, each one with a degree of compatibility with an input example. This characteristic is illustrated in Figure 2, which presents the partition of attribute At_n on the left, defined by fuzzy sets S1, S2, and S3, and the membership degrees y_1 and y_2 for input x_1, as well as the fuzzy DT on the right with two triggered branches in blue, S1 and S2.

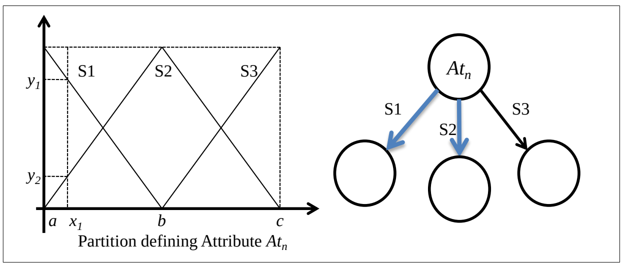

Fig 2. Partition defining attribute At_n (left) and triggered branches of a decision tree (right)

Notice that for an input value x_1, fuzzy sets S1 and S2 are intersected with membership degree values y_1 and y_2, respectively. This way, branches S1 and S2, indicated by blue (lighter) arrows, of the fuzzy DT are fired. For any input value ranging from a to b, the branches defined by fuzzy sets S1 and S2 are triggered, while for an input value ranging from b to c, the branches defined by fuzzy sets S2 and S3 are triggered.

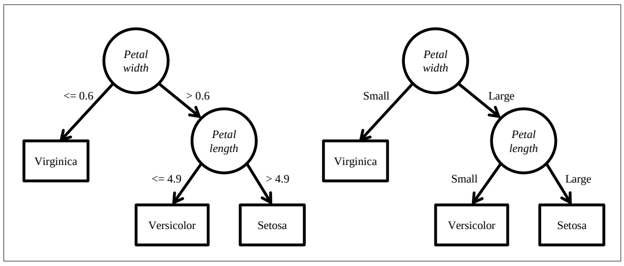

Fig 3. A classic(left) and a fuzzy (right) decision tree for the Iris dataset

Figure 4 presents the fuzzy sets (Small and Large) defining the attributes tested in the DTs of Figure 3, including the input values (*Petal Length*$= 4.9$, *Petal Width* $= 0.6$) and their corresponding membership degrees.

The inference process of the classic DT is straightforward: if *Petal Width* is ≤ 0.6, the example belongs to the *Virginica* class, otherwise, the *Petal Length* attribute is tested; if it is ≤ 4.9, the example is classified as *Versicolor*, otherwise it is classified as *Setosa*. This way, considering a new input example to be classified having *Petal Length* $= 4.9$ and *Petal Width* $= 0.6$, both values on the borderline of the crisp discretization of the classic DT, only the first rule is fired: **IF** *Petal Width* is ≤ 0.6 **THEN** *Class* is *Virginica*.

For the fuzzy DT, on the other hand, the membership degrees of the input example, shown in Figure 4 (*Petal Length Small* $= 0.66$; *Petal Length Large* $= 0.34$; *Petal Width Small* $= 0.79$; *Petal Width Large* $= 0.21$) are used to calculate the compatibility degree of the input example with each rule. For this particular example, using minimum as t-norm, the fuzzy rules and their compatibility degrees (in brackets) with the input example are:

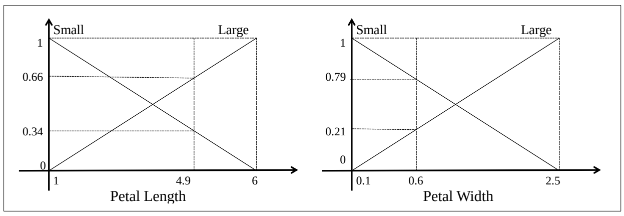

Fig 4. Fuzzy sets defining attributes *Petal Length* and *Petal Width*

1. **IF** *Petal Width* is *Small* **THEN** *Class* is *Virginica* (0.79)
2. **IF** *Petal Width* is *Large* **AND** *Petal Length* is *Small* **THEN** *Class* is *Versicolor* (0.21)
3. **IF** *Petal Width* is *Large* **AND** *Petal Length* is *Large* **THEN** *Class* is *Setosa* (0.21)

For this example, using the classic fuzzy reasoning method (best rule), the class of the first rule, which has highest compatibility degree with the input example, is used to classify the example as *Virginica*. Notice that this is the same class defined by the classic DT.

Now, let us assume that the *Petal Width* of the input example is 0.61, while the *Petal Length* remains the same. Notice that the difference in the *Petal Width* between this example and the last one is quite small (0.01). This way, we are likely to believe, intuitively, the class of such similar examples should be the same. Nevertheless, the classic

DT classifies this new example as belonging to the *Versicolor* class. The fuzzy DT, on the other hand, since it uses the compatibility degrees of the input values with the fuzzy sets defining the tree tests, still classifies this new example with the same class, *Virginica*. Notice that the same situation can occur for *Petal Length*, as well as with any continuous attribute with a crisp discretization. This robustness of fuzzy DTs is highly desirable.

Since fuzzy DTs use the compatibility degree of each rule to classify an input example, the classic and general fuzzy reasoning methods can be used in the inference process. Notice that, once multiple rules derived from the DT can be fired, an input instance can be classified with the class of the rule with highest compatibility with the input example (classic fuzzy reasoning method), or with the class with the highest combination from the set of rules with that given class (general fuzzy reasoning method).

In conclusion, the classic C4.5 algorithm and its fuzzy version, FuzzyDT, present relevant differences regarding the handling of continuous features, reuse of features, and inference procedures. C4.5 has a simpler and faster inference process, while FuzzyDT is able to avoid the reuse of features, the repeated splitting of continuous features. FuzzyDT also provides a more robust, although more costly, inference process. Next, we present the experiments and results.

6. Experiments

FuzzyDT was compared to C4.5 using 16 datasets from the UCI - Machine Learning Repository [6] and a 10-fold cross-validation strategy. C4.5 was selected for the comparisons since FuzzyDT presents many similarities with C4.5, while adding the advantages of fuzzy systems, regarding the processing of uncertainty and imprecision, as well as interpretability, to the induced models.

Table 1 summarizes the characteristics of the datasets, presenting the number of examples (E), features, including the number of continuous (c) and discrete (d) features in brackets, number of classes (C), as well as the majority error, *i.e.*, the error of the algorithm that always predicts the majority class.

The experiments were carried out using the classic fuzzy reasoning method [5] and post-pruning with a default confidence level of 25%.

Dataset	Examples	Features	(c	d)	C	ME	Dataset	E	Features	(c	d)	C	ME
Breast	682	9	(9	0)	2	65.10	Iono	351	34	(34	0)	2	35.90
Credit	653	15	(6	9)	2	45.33	Iris	150	4	(4	0)	3	66.67
Cylinder	277	32	(19	13)	2	35.74	Liver	345	7	(7	0)	2	57.97
Diabetes	769	8	(8	0)	2	34.90	Segment	210	19	(19	0)	7	85.71
Gamma	19020	10	(10	0)	2	64.84	Spam	4601	57	(57	0)	2	60.60
Glass	220	9	(9	0)	7	65.46	Steel	1941	29	(29	0)	7	32.82
Haberman	306	3	(3	0)	2	73.53	Vehicle	846	18	(18	0)	4	74.23
Heart	270	13	(13	0)	2	44.44	Wine	178	13	(13	0)	3	59.74

Table 1. Characteristics of the datasets

Table 2 presents the error rates of the experiments, including the average error rates. The error standard deviations are presented in brackets.

Table 3 presents the average number of rules (column Rules) and the average of the total number of conditions in each model (column Cond.). The standard deviations for the number of rules, as well as number of conditions, are presented in brackets (columns SD). The best (smallest) rates, are dark-gray shaded.

As can be observed, FuzzyDT obtained better results than C4.5, *i.e.* smaller error rates, for 10 of the 16 datasets. All the models obtained better error rates than the majority error of the datasets included in the experiments. In order to check for statistically significant differences, we executed the Mann-Whitney test [15], which, with a 95% confidence, found no statistically significant differences between FuzzyDT and C4.5.

Dataset	FuzzyDT		C4.5		Dataset	FuzzyDT		C4.5	
Breast	1.49	(0.00)	5.13	(3.03)	Ionosphere	3.99	(7.35)	11.40	(3.83)
Credit	7.81	(0.00)	13.00	(2.98)	Iris	8.00	(2.67)	5.33	(5.81)
Cylinder	6.16	(4.51)	30.65	(5.91)	Liver	36.76	(4.65)	32.74	(6.57)
Diabetes	21.84	(1.27)	25.52	(2.63)	Segmentation	12.38	(2.33)	2.86	(1.12)
Gamma	21.13	(0.70)	15.02	(0.58)	Spam	28.98	(0.00)	7.87	(1.35)
Glass	39.13	(0.00)	30.37	(6.87)	Steel	20.78	(0.93)	23.13	(2.48)
Haberman	26.67	(0.00)	29.09	(6.10)	Vehicle	25.37	(1.80)	27.07	(4.10)
Heart	14.44	(5.60)	23.11	(5.94)	Wine	5.00	(6.43)	7.25	(6.59)

Table 2. Error rates

For the evaluation of the interpretability of the models, we compared the average number of rules and the average of the total number of conditions in the models induced by FuzzyDT and C4.5. In this work, we adopt the average number of conditions in the induced models as the measure of the syntactic complexity of the models.

Regarding the number of rules of the induced models, C4.5 and FuzzyDT tie, both presenting better results (smallest number of rules) for 8 datasets. It is interesting to notice that for the *Gamma* dataset, although the error rate obtained by the model induced by C4.5 is smaller than the one of FuzzyDT, the number of rules for the C4.5 model is 8 times larger than for FuzzyDT: 43.00 rules for FuzzyDT against 328.70 for C4.5. No other discrepancies are present in the results regarding the average number of rules of the induced models.

	FuzzyDT		C4.5		FuzzyDT		C4.5	
Dataset	Rules	SD	Rules	SD	Cond.	SD	Cond.	SD
Breast	15.00	(0.00)	12.30	(3.32(	50.00	(0.00)	52.70	(19.83)
Credit	7.80	(1.83)	19.30	(6.48)	21.40	(5.50)	90.90	(33.76)
Cylinder	45.80	(4.87)	42.80	(9.45)	198.50	(26.16)	248.50	(102.69)
Diabetes	13.40	(5.50)	23.60	(7.55)	42.60	(22.00)	150.20	(64.15)
Gamma	43.00	(0.00)	328.70	(31.36)	228.00	(0.00)	3,634.80	(435.18)
Glass	24.00	(0.00)	24.10	(2.17)	99.00	(0.00)	137.80	(20.35)
Haber	4.60	(1.20)	3.10	(1.64)	8.30	(2.10)	6.90	(4.18)
Heart	22.40	(2.84)	23.60	(3.67)	78.90	(13.87)	95.70	(23.08)
Iono	21.00	(0.00)	13.90	(1.45)	89.00	(0.00)	72.40	(13.15)
Iris	5.00	(0.00)	4.60	(0.66)	9.00	(0.00)	12.10	(3.08)
Liver	2.40	(1.28)	24.50	(4.92)	3.70	(2.00)	139.90	(34.60)
Segment	28.80	(1.89)	41.80	(2.96)	127.30	(13.52)	314.70	(35.73)
Spam	55.00	(0.00)	100.40	(10.43)	660.00	(0.00)	1,045.50	(142.89)
Steel	225.60	(2.85)	159.90	(6.74)	1,761.70	(24.48)	1,909.00	(125.86)
Vehicle	82.00	(6.47)	66.30	(7.20)	530.50	(62.02)	503.00	(79.81)
Wine	15.40	(0.80)	5.10	(0.30)	48.20	(4.40)	12.50	(1.20)

Table 3. Number of rules and conditions

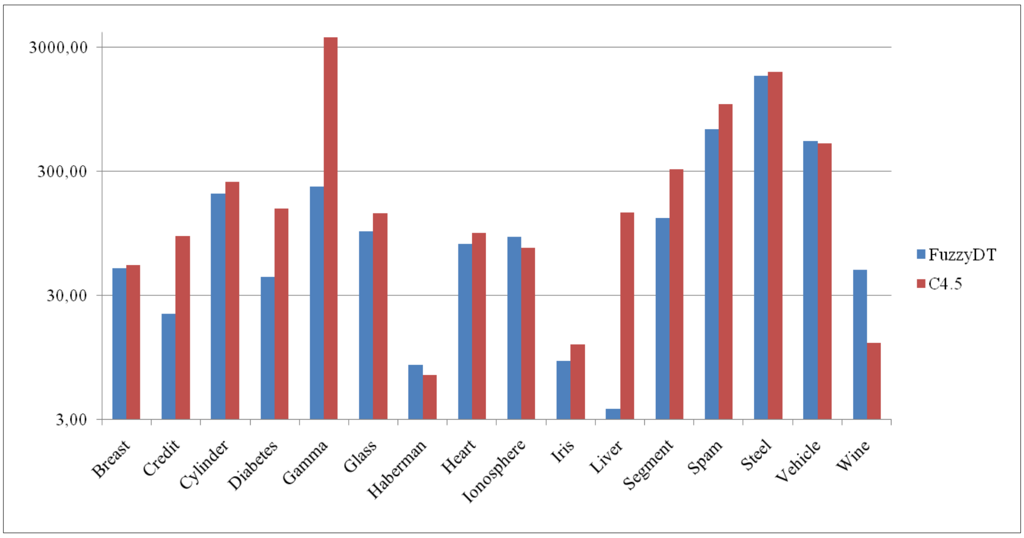

Fig 5. Average of the total number of conditions

Figure 5 present the average number of conditions in a graphical manner. Notice that a 10-base logarithm scale is used for the number of conditions. Thus, the lower portion of the graph varies from 3 to 30 conditions, while the upper area varies from 300 to 3,000 conditions.

Regarding the average number of conditions in the models, the ones induced by FuzzyDT were smaller than those of C4.5 for 12 of the 16 datasets. Moreover, notice in Figure 5 that for the *Credit*, *Gamma*, and *Liver* datasets the number of conditions of the C4.5 models is larger than for FuzzyDT. In fact, the number of conditions of the C4.5 models is 3 times larger than FuzzyDT for *Credit*, 16 times larger for *Gamma*, and 38 times larger for the *Liver* dataset.

In summary, FuzzyDT presented smaller error rates than C4.5 for most of the datasets, as well as smaller syntactic complexity for 12 datasets. The models induced by C4.5 for three datasets were considerably more complex than those induced by FuzzyDT. Next, we present the conclusions and future work.

7. Conclusions

DTs have been successfully applied to many areas for tasks such as classification, regression, and feature subset selection, among others. DTs are popular in machine learning due to the fact that they produce graphical models, as well as textual rules, that end users can easily understand. The induction process of DTs is usually fast, requiring low computational resources.

Fuzzy systems, on the other, provide mechanisms to handle imprecision and uncertainty in data based on the fuzzy logic and fuzzy sets theory. The combination of fuzzy systems and DTs has produced fuzzy DT models, which benefit from both techniques to provide simple, accurate, and highly interpretable models at low computational costs.

In this paper, we detailed the FuzzyDT algorithm, a fuzzy DT based on the classic C4.5 DT algorithm and expanded previous experiments. We also provided a thorough comparison of some relevant issues regarding the classic and the fuzzy models, and discussed the use of FuzzyDT for feature subset selection. FuzzyDT was experimentally evaluated and compared to C4.5 using 16 datasets and a 10-fold cross-validation strategy. FuzzyDT obtained smaller error for 10 datasets and was also able to induce models with less rules and less conditions in the rules when compared to C4.5.

As future work we intend to compare FuzzyDT with other fuzzy DTs, as well as with other classic DTs. We also intend to further evaluate FuzzyDT for the task of feature subset selection.

Acknowledgments We thank FAPESP, Sao Paulo State Research Foundation, for the financial support

References

[1] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth & Books, Pacific Grove, CA, 1984.

[2] R. L. P. Chang and T. Pavlidis. Fuzzy decision tree algorithms. IEEE Transactions on Systems Man and Cybernetics, 7:28–35, 1977.

[3] M. E. Cintra and H. A. Camargo. Feature subset selection for fuzzy classification methods. Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems - IPMU, 80:318–327, 2010. Communications in Computer and Information Science - Springer.

[4] M. E. Cintra, C. A. A. Meira, M. C. Monard, H. A. Camargo, and L. H. Rodrigues. The use of fuzzy decision trees for coffee rust warning in Brazilian crop. In Int. Conf. Int. Sys. Design & Applications, volume 1, pages 1347–1352, 2011.

[5] O. Cordon, M. J. del Jesus, and F. Herrera. A proposal on reasoning methods in fuzzy rule-based classification systems. International Journal of Approximate Reasoning, 20(1):21–45, 1999.

[6] A. Frank and A. Asuncion. UCI Machine Learning Repository, 2010.

[7] C. Z. Janikow. Fid 4.1: an overview. In Proc. of the North American Fuzzy Information Processing Society - NAFIPS 2004, pages 877–881, 2004.

[8] Hori Kazunor, Umano motohide, Sato Hiroshi, and Uno Yuushi. Fuzzy C4.5 for generating fuzzy decision trees and its improvement. Faji Shisutemu Shinpojiumu Koen Ronbunshu, 15:515–518, 1999.

[9] G. J. Klir and Bo Yuan. Fuzzy Sets and Fuzzy Logic - Theory and Applications. Prentice-Hall, 1995.

[10] Cristina Olaru and Louis Wehenkel. A complete fuzzy decision tree technique. Fuzzy Sets and Systems, 138(2):221–254, 2003.

[11] J. R. Quinlan. Induction of Decision Trees. Machine Learning, 1:81–106, 1986. Reprinted in Shavlik and Dieterich (eds.) Readings in Machine Learning.

[12] J. Ross Quinlan. C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning). Morgan Kaufmann, 1 edition, January 1993.

[13] J. Ross Quinlan. Bagging, Boosting and C4.5. In Proc. of the 13th Conf. on Artificial Intelligence, pages 725–730, 1996.

[14] M. Tokumaru and N. Muranaka. Kansei impression analysis using fuzzy c4.5 decision tree. In Int. con. on Kansei engineering and emotion research, 2010.

[15] Frank Wilcoxon. Individual comparisons by ranking methods. Biometrics Bulletin, 1:80–83, 1945.

[16] L.A. Zadeh. Fuzzy sets. Information and Control, 8:338–353, 1965.