Incremental Tuning of Fuzzy Decision Trees

Christophe Marsala
LIP6, Université Pierre et Marie Curie, Paris 6,
4 place Jussieu, 75005 Paris, France
Email: Christophe.Marsala@lip6.fr

***Abstract*—Handling stream data or temporal data is a difficult task and brings out a lot of problems to classical learning algorithms as the decision tree construction algorithms. In that context, incremental algorithms have been proposed but they often lie on the frequent reconstruction of the decision tree when this one provides a high number of misclassified examples.**

In this paper, we proposed a new algorithm to incrementally tune a fuzzy decision tree (FDT) that limit the number of reconstructions of the tree. That algorithm takes benefit of the fuzzy classification provided by a FDT to introduce a local tuning of the internal nodes of the FDT and avoid a complete reconstruction of the tree.

I. Introduction

One popular machine learning tools is the so-called fuzzy decision trees (FDT) that proposes a summarized view of a set of data. FDT have been very used in the past years as a powerful knowledge extraction tool, and nowadays it is still an active domain of researches and applications [1], [2], [3], [4], [5], [6], [7]. However, in Big Data applications, one of the main difficulties is to handle stream data or temporal data. That leads to the proposal of incremental algorithm to build (fuzzy) decision trees [5], [8]. Incremental algorithms are often based on frequent reconstructions of the tree when it provides too much misclassified examples. Every construction of a tree is very consuming in time and brought out a strong drawback for the decision trees to be used in Big Data applications.

In this paper, we proposed a new algorithm to incrementally tune a FDT. The aim of this algorithm is to reduce the number of reconstructions of the tree. That algorithm takes benefit of the fuzzy classification provided by a FDT to introduce a local tuning of the internal nodes of the FDT and avoid as much as possible a complete reconstruction of the tree.

Among existing approaches to incrementally build FDT, [9] propose an algorithm to incrementally modify a fuzzy decision tree. That approach is very interesting but lies on a set of complex operators that lead to radical changes of the FDT. Such modifications could lead to the lost of the main property (minimality) of the FDT that is ensured by a classical construction algorithm. In [10], an approach is introduced to incrementally modify a regression tree and, in particular, a gradient descent is used to optimize the test values at each internal nodes of the tree. However, this can lead to a very high complexity to tune the tree. In our approach, we avoid that complexity by a simple linear modification of one unique internal node at each time, as explained in Section III.

This paper is composed as follows. In Section II, basics on FDT are provided. In Section III, our proposed approach to incrementally tune a FDT is presented. Finally, a set of applications is proposed in Section IV and a conclusion is drawn in Section V.

II. Fuzzy Decision Trees

In this part, the algorithm to construct a FDT is recalled. An example of FDT can be seen in Fig. 1. That FDT is a binary FDT (each internal node is associated with a binary fuzzy partition automatically constructed from a set of numerical values [11] constructed from the well-known Iris database by Fischer.

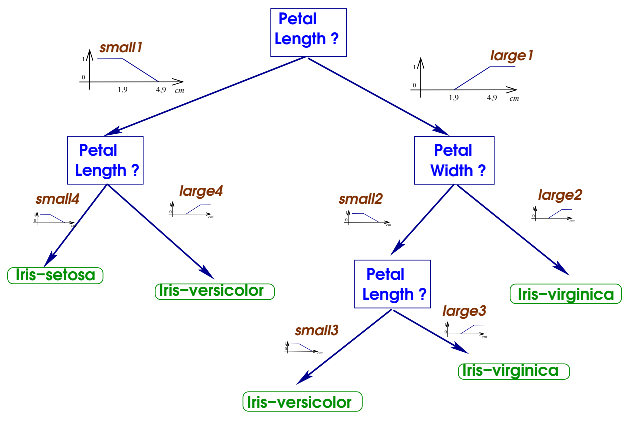

Fig. 1. Example of Fuzzy Decision Tree

A. Building Fuzzy Decision Trees

A FDT is constructed in an inductive way, as a machine learning process. A *training set* $\mathcal{E} = \{e_1, ..., e_N\}$ is given and is composed of examples. Each example is associated with a *description* which is a N_A-tuple of attribute-value pairs (A_j, v_{jl}), where each attribute A_j, from a set of attributes $\mathcal{A} = \{A_1, ..., A_{N_A}\}$, can take a (fuzzy, numerical, or symbolic) value v_{jl} in a set $\{v_{j1}, ..., v_{jm_j}\}$ of possible values. A fuzzy value v_{jl} is associated with a membership function $\mu_{v_{jl}}$. In supervised learning, a set of classes $C = \{c_1, ..., c_K\}$ is also given. Each c_k can be associated with a membership function μ_{c_k}. In the training set, the description of an example e_i is associated with a particular class c_k from C.

There exists a lot of approaches (the *Top Down Induction of Decision Tree* (TDIDT) methods) to construct them from a training data set. A tree is built from its root to its leaves, by choosing an attribute to split the training set into subsets. The splitting attribute is selected by means of a *measure* *of discrimination* H (in non-fuzzy algorithms, the Shannon entropy is used [12], [13]). As previously stated, the existing approaches to construct FDT differ mainly in their choice of H [14]. In the fuzzy settings, fuzzy measures of discrimination have been studied and a hierarchical model of validation of such a measure has been presented [14]. At each step of the construction of a FDT, the measure of discrimination H is used to value the power of discrimination of each attribute with regard to the class. Thus, it will produce a ranking of all the attributes according to their power of discrimination, and the *winner* attribute is the one ranked first.

Among measures of discrimination used to construct a FDT, the main one is the *entropy of fuzzy events* [15], [16], [17]. It corresponds to an extension of the Shannon entropy by substituting probabilities of fuzzy events [18] to classical probabilities. Other measures have been proposed [19], [20], [21], [22], [23], and an interesting one is the *measure of classification ambiguity* [19] that is defined from both a measure of fuzzy subsethood and a measure of non-specificity.

B. Construction of Fuzzy Partitions

To construct a FDT, a fuzzy partition for each numerical attribute should be provided, for instance, thanks to experts of the domain. Such a fuzzy partitions could also be automatically constructed from the training set.

For instance, in [11], an automatic algorithm [24] based on the utilization of the mathematical morphology theory is used to construct a fuzzy partition. It enables the construction of fuzzy values that are linked to the distribution of the class in relation with a given attribute. A contextual partitioning of each attribute is performed, enabling us to obtain the best partition with respect to the class.

C. Classification with Fuzzy Decision Trees

The path from the root to a leaf in a FDT is equivalent to a fuzzy decision rule. The premises of the rule are composed of tests on attributes values, and the conclusion is the value of the class that labels the leaf of the path:

$$\texttt{if } A_{l_1} = v_{l_1} \texttt{ and } \ldots \texttt{and } A_{l_p} = v_{l_p} \texttt{ then } C = c_k$$

In a FDT, a leaf can also be labelled by a set of values $\{c_1, \ldots, c_K\}$ for the class, each value c_j is associated with a weight set during the learning phase. A path is thus equivalent to the following fuzzy rule:

$\texttt{if } A_{l_1} = v_{l_1} \texttt{ and } \ldots \texttt{ and } A_{l_p} = v_{l_p} \texttt{ then}$
$C = c_1 \texttt{ with the degree } P^*(c_1|(v_{l_1}, \ldots v_{l_p})) \texttt{ and } \ldots$
$\texttt{and } C = c_K \texttt{ with the degree } P^*(c_K|(v_{l_1}, \ldots v_{l_p}))$.
where P^* is the probability measure of a fuzzy event [18].

Each value v_i can be either precise or fuzzy, and is described by means of a membership function μ_{v_i}. When an example described by means of a set of values $\{A_1 = w_1; ...; A_n = w_n\}$, has to be classified, its description is compared to premises of rule r. For each premise, a similarity degree $\texttt{Deg}(w_{l_i}, v_{l_i})$ between w_{l_i} and the edge value v_{l_i} is valued. At the end, according to rule r, the example is classified and associated with the class c_j with a *final degree* $\texttt{Fdeg}_r(c_j)$. This degree is obtained by means of an aggregation of the degrees $\texttt{Deg}(w_{l_i}, v_{l_i})$ thanks to the *minimum*[1]:

$$\texttt{Fdeg}_r(c_j) = \min_{i=1\ldots p} \texttt{Deg}(w_{l_i}, v_{l_i}).P^*(c_j|(v_{l_1}, \ldots v_{l_p})) \quad (1)$$

To classify an example, each c_j is associated with the membership degree $\texttt{Fdeg}(c_j)$, from $[0, 1]$, obtained from the whole set of rules. If n_ρ is the number of rules from the FDT:

$$\texttt{Fdeg}(c_j) = \max_{r=1\ldots n_\rho} \texttt{Fdeg}_r(c_j) \quad (2)$$

At the end, the example is classified in the class c_k that has the highest degree $\texttt{Fdeg}(c_k)$.

III. Tuning Fuzzy Decision Trees

Tuning FDT occurs when the tree is not sufficiently accurate to class forthcoming examples. First of all, when a new example e occurs, the local counter n_e of seen examples is incremented. When classifying e, two cases could arise. If e is correctly classified, the tree $\mathcal{T}$ is accurate and there is no need to change it. On the contrary, if e is misclassified, $\mathcal{T}$ should be tuned in order to take into account future cases like e. With a FDT, there is two kinds of misclassification:

- e is misclassified as c_k with a degree $\texttt{Fdeg}(c_k) = 1$;
- e is misclassified as c_k with a degree $\texttt{Fdeg}(c_k) < 1$.

In the first case, the fuzzy labels encountered on paths that leads to the decision are not incriminated in the wrong decision. The problem of $\mathcal{T}$ lies in its global form that necessitates a global tuning of the tree. To perform that tuning, e is added with a time stamp to the windowed training memory of examples. The counter n_w of misclassified examples is incremented by 1 in order to take it into account.

In the second case, the fuzzy labels encountered on paths used to classify e are highly involved in the wrong decision. Thus, they should be tuned in order to take into account e and enables a change in the decision. Here, a local tuning of the fuzzy partitions should be done.

A. Global Tuning

The global tuning of the tree involved a new construction of the FDT. This global tuning is done when the rate of wrongly classified examples $\frac{n_w}{n_e}$ goes over a fixed threshold (for instance, a 10% rate). When it occurs, it is time to reshape the fuzzy decision tree.

Let $\mathcal{T}$ be the old FDT and let $\mathcal{T}'$ be the new FDT. $\mathcal{T}'$ is built by means of the examples stored in the windowed training memory during the use of $\mathcal{T}$. That set of examples is added to the training set of examples that enables the construction of $\mathcal{T}$. Time stamps associated to examples is used to only keep in the training set the newer examples and to forget the older ones (for instance, in order to have a constant number of examples used to build the FDT).

The construction of $\mathcal{T}'$ is done thanks to the algorithm described in Section II-A. The training set used is stored for the next global tuning that will occur in the future. The windowed training memory is deleted and the counters n_e and n_w are put to 0.

[1]Any operators can also be used, we focus here on the Zadeh's operators.

B. Local Tuning of Fuzzy Partitions

The local tuning of the fuzzy partitions of a FDT $\mathcal{T}$ is done when an example has been misclassified by $\mathcal{T}$ but with a degree $\texttt{Fdeg}(c_k) < 1$. In our approach, we propose to tune only a single node after such a misclassification. First of all, the rule with the highest $\texttt{Fdeg}(c_k)$ (see eq. (2)) is selected. In that rule, the node that set the lowest degree $\texttt{Fdeg}_r(c_k)$ (see eq. (1)) is selected. The fuzzy partition of that node is thus tuned to change the membership degrees for the example.

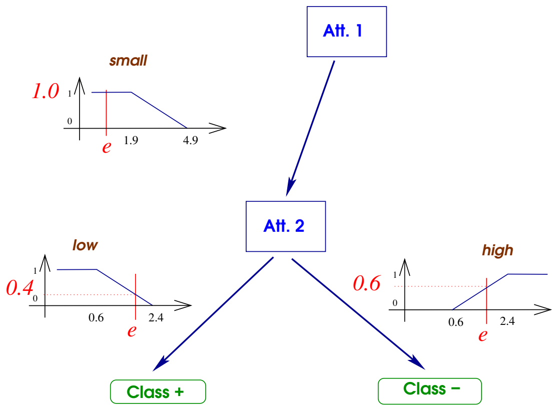

Fig. 2. Classification paths

For instance in Fig. 2, the example e has been associated with a membership degree of 1 to the first vertex of the path (associated to attribute $Att.1$), and it has been associated to a membership degree of 0.4 to the left vertex that goes from the node labeled by $Att.2$ and a corresponding membership degree of 0.6 to the right vertex from that node. For the 2 pathes from the root to the leaves that are represented in Fig. 2, we have:

- for the left path: $\texttt{Fdeg}_l(c_+) = \min(1, 0.4) = 0.4$
- for the right path: $\texttt{Fdeg}_h(c_-) = \min(1, 0.6) = 0.6$

The final class associated to e is c_- with the degree $\max(0.4, 0.6) = 0.6$ as seen in Section II-C. If e is considered as misclassified, the fuzzy partition associated with the node labeled by $Att.2$ will change because it is this fuzzy partition that leads to the final degree of 0.6.

To illustrate that change, for the sake of simplicity, we focus on a binary fuzzy partition (see Fig. 3) but the approach can be generalized to other kinds of partition. The left fuzzy set of the partition (labeled "low") is defined by means of the highest value of its kernel x_l and the highest value of its support x_h. The right fuzzy set of the partition (labeled "high") is defined by means of the lowest value of its kernel x_h and the lowest value of its support x_l.

Let e be the value of example e for the involved attribute. We denote $\mu_l(e)$ the membership degree of e to the fuzzy set "low", and $\mu_h(e)$ the membership degree of e to the fuzzy set "high". In Fig. 3, we consider that $\mu_l(e) < \mu_h(e)$.

As e is misclassified, we aim to change the fuzzy partition in order to have $\mu_l(e) > \mu_h(e)$ that could lead to an accurate classification for e. To enable that, the membership functions are translated thanks to a coefficient δ that is valued as follows.

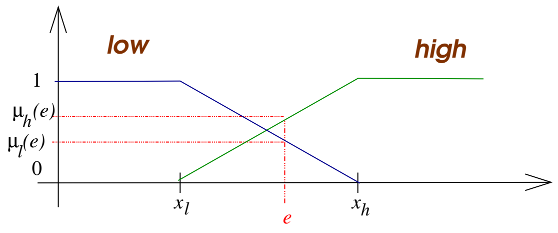

Fig. 3. Original fuzzy partition

Let $d = sgn(\mu_h(e) - \mu_l(e))$ be the direction of the translation. When $d = 1$ then the fuzzy sets of the fuzzy partition are translated to the right, otherwise ($d \leq 0$) it is translated to the left. It can be easily seen that $d > 0$ implies $\mu_h(e) > 0.5$ (as we have a fuzzy partition, $\mu_l(e) = 1 - \mu_h(e)$) and thus, if e has been misclassified because $\mu_l(e) < \mu_h(e)$, a translation to the right should be performed in order to have $\mu_l(e) > \mu_h(e)$. The amplitude of the translation is valued as:

$$\delta = \frac{\max(e - x_l, x_h - e)}{1 + f} \tag{3}$$

with $f \geq 0$ that values the amplitude of the translation: the lower f the stronger the translation. δ enables to increase of the membership degree of e to the fuzzy set that has the lowest degree before the tuning and to decrease the highest degree (see Fig. 4, after the tuning we have $\mu_l(e) > \mu_h(e)$).

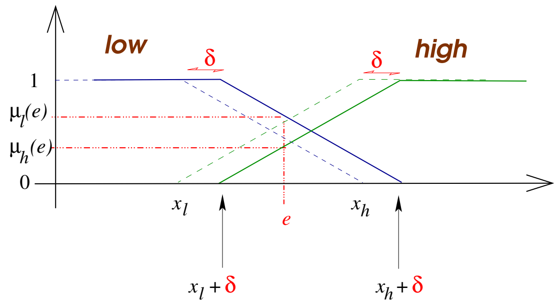

Fig. 4. Tuned fuzzy partition with a translation to the right

When $f = 0$, the amplitude of translation is optimum. It enables the example e to be added completely in the fuzzy set. When $f > 0$, the translation is more smooth and moves the boundary of the fuzzy set close to e to increase its membership.

In order to smooth the tuning of the fuzzy partition, it should be interesting to value f by means of the number of training examples that have provided the original fuzzy partition. That means that, for each fuzzy partition, the number of training examples t_l (resp. t_h) that belongs to the lowest (resp. highest) fuzzy set of the partition is stored with the FDT. Thus, a good way to set f is to take into account the difference of sizes (in number of training examples) between the two fuzzy sets. For instance: $f = \frac{|t_l - t_h|}{t_l + t_h}$ that leads to have a low value for f (and thus a high amplitude for the translation) when the subsets are balanced. f increases when the difference of number of examples between the sets increases. This aims to reproduce a kind of "gravitation strength" of a large set with regards to a smaller set and enables small translations for sets constructed by means of a high number of training examples.

C. Incremental Tuning FDT Algorithm

The algorithm is given in Algorithm 1.

Algorithm 1 Incremental Tuning FDT Algorithm

```
Require:
    $\mathcal{T}$: a fuzzy decision tree,
    $\mathcal{X}$: training set used to build $\mathcal{T}$,
    $\mathcal{W}$: windowed training memory,
    $n_e$ : number of examples classified by $\mathcal{T}$ ,
    $n_w$ : number of examples misclassified by $\mathcal{T}$ ,
 1: while $\exists$ example $e$ to classify do
 2:   Classification of $e$ with $\mathcal{T}$
 3:   $n_e \longleftarrow n_e + 1$
 4:   if $e$ is misclassified into class $c_k$ then
 5:     if $\texttt{Fdeg}(c_k) = 1$ then
 6:       $n_w \longleftarrow n_w + 1$
 7:       Add $e$ to $\mathcal{W}$
 8:       if $n_w$ is greater than a fixed threshold then
 9:         $\mathcal{X} \longleftarrow \mathcal{X} \cup \mathcal{W}$ (with time stamp selection)
10:         Build the FDT $\mathcal{T}'$ with $\mathcal{X}$
11:         $\mathcal{T} \longleftarrow \mathcal{T}$', $n_e \longleftarrow 0$, $n_w \longleftarrow 0$, and $\mathcal{W} \longleftarrow \emptyset$
12:       end if
13:     else
14:       Local tuning of $\mathcal{T}$ (see Section III-B)
15:     end if
16:   end if
17: end while
```

IV. Applications

The proposed Algorithm 1 can be applied in domain when there is a flow of examples to handle. For instance, mining streams as in [5], or temporal data sets.

Such data set could be obtained for instance by dayly querying RSS feeds from news websites and associated to each news a class (a label like sports, politics,...). More detailed on the application of our algorithm to such kind of data will be published later.

V. Conclusion and Future Work

In this paper, a new algorithm to incrementally tune a fuzzy decision tree is proposed. That algorithm is based on two different kinds of tuning.

A global tuning of the FDT is done when a rate of completely misclassified examples has been reached. That global tuning is based on a complete construction of a new FDT using all misclassified examples seen previously.

A local tuning of the fuzzy partitions involved in internal nodes of the FDT is performed when an example is not completely misclassified, taking benefit here of the membership degree of classification a FDT is able to provide.

This algorithm will be tested on several application where the data are provided by means of a stream (RSS feeds) or as a temporal data set. Such kind of applications will enable us to value the number of global tuning and local tuning involved in such kind of problem.

References

[1] X. Wang and C. Borgelt, "Information measures in fuzzy decision trees," in *Proc. of the IEEE International Conference on Fuzzy Systems*, vol. 1, July 2004, pp. 85–90.

[2] W. Pedrycz and Z. Sosnowski, "Genetically optimized fuzzy decision trees," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 3, pp. 633–641, June 2005.

[3] X. Liu and W. Pedrycz, "The development of fuzzy decision trees in the framework of axiomatic fuzzy set logic," *Applied Soft Computing*, vol. 7, pp. 325–342, 2007.

[4] K. Crockett, Z. Bandar, and D. Mclean, "On the optimization of t-norm parameters within fuzzy decision trees," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, July 2007, pp. 1–6.

[5] T. Wang, Z. Li, Y. Yan, and H. Chen, "An incremental fuzzy decision tree classification method for mining data streams," in *Proceeding of MLDM'07, LNAI 4571*, Granada, Spain, July 2007, pp. 91–103.

[6] L.-C. Dong, D. Wang, and X.-Z. Wang, "An improved sample selection algorithm in fuzzy decision tree induction," in *Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernectics*, San Antonio, TX (USA), October 2009, pp. 629–634.

[7] N. Abu-Halaweh and R. Harrison, "Practical fuzzy decision trees," in *Proc. of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, Nashville, (USA), March 2009, pp. 211 – 216.

[8] P. E. Utgoff, "Incremental induction of decision trees," *Machine Learning*, vol. 4, pp. 161–185, 1989.

[9] M. Guetova, S. Hölldobler, and H.-P. Störr, "Incremental fuzzy decision trees," in *Proceedings of the 25th German Conference on Artificial Intelligence (KI2002*, 2002.

[10] A. Lemos, W. Caminhas, and F. Gomide, "Fuzzy evolving linear regression trees," *Evolving Systems*, vol. 2, pp. 1–14, 2011.

[11] C. Marsala and B. Bouchon-Meunier, "An adaptable system to construct fuzzy decision trees," in *Proc. of the NAFIPS'99 (North American Fuzzy Information Processing Society)*, New York, USA, 1999, pp. 223–227.

[12] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification And Regression Trees*. New York: Chapman and Hall, 1984.

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

[14] C. Marsala and B. Bouchon-Meunier, "Quality of measures for attribute selection in fuzzy decision trees," in *Proc. of the International Conf. on Fuzzy Systems (WCCI-2010)*, Barcelona, Spain, July 2010.

[15] C. Z. Janikow, "Fuzzy decision trees: Issues and methods," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 28, no. 1, pp. 1–14, 1998.

[16] M. Ramdani, "Une approche floue pour traiter les valeurs numériques en apprentissage," in *Journées Francophones d'apprentissage et d'explication des connaissances*, 1992.

[17] R. Weber, "Fuzzy-ID3: A class of methods for automatic knowledge acquisition," in *IIZUKA'92 Proceedings of the 2nd International Conference on Fuzzy Logic*, 1992, pp. 265–268.

[18] L. Zadeh, "Probability measures of fuzzy events," *Journal Math. Anal. Applic.*, vol. 23, 1968, reprinted in "*Fuzzy Sets and Applications: selected papers by L. A. Zadeh*", R. R. Yager, S. Ovchinnikov, R. M. Tong and H. T. Nguyen eds, pp. 45–51.

[19] Y. Yuan and M. Shaw, "Induction of fuzzy decision trees," *Fuzzy Sets and systems*, vol. 69, pp. 125–139, 1995.

[20] C. Olaru and L. Wehenkel, "A complete fuzzy decision tree technique," *Fuzzy Sets and Systems*, vol. 138, no. 2, pp. 221–254, September 2003.

[21] X. Boyen and L. Wehenkel, "Automatic induction of fuzzy decision tree and its application to power system security assessmen," *Fuzzy Sets and Systems*, vol. 102, no. 1, pp. 3–19, 1999.

[22] K. Cios and L. Sztandera, "Continuous ID3 algorithm with fuzzy entropy measures," in *Proceedings of the first International IEEE Conference on Fuzzy Systems*, San Diego, 1992.

[23] X. Wang, B. Chen, G. Qian, and F. Ye, "On the optimization of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 112, no. 1, pp. 117–125, May 2000.

[24] C. Marsala and B. Bouchon-Meunier, "Fuzzy partioning using mathematical morphology in a learning scheme," in *Proceedings of the 5th IEEE Int. Conf. on Fuzzy Systems*, vol. 2, New Orleans, USA, September 1996, pp. 1512–1517.