Enhancing the Intelligibility of Boolean Decision Trees with Concise and Reliable Probabilistic Explanations

Louenas Bounia

Heudiasyc, Université de Technologie de Compiègne (UTC), CNRS
F-60200 Compiègne, France
louenas.bounia@hds.utc.fr

Abstract. This work deals with explainable artificial intelligence (XAI), specifically focusing on improving the intelligibility of decision trees through reliable and concise probabilistic explanations. Decision trees are popular because they are considered highly interpretable. Due to cognitive limitations, abductive explanations can be too large to be interpretable by human users. When this happens, decision trees are far from being easily interpretable. In this context, our goal is to enhance the intelligibility of decision trees by using probabilistic explanations. Drawing inspiration from previous work on approximating probabilistic explanations, we propose a greedy algorithm that enables us to derive concise and reliable probabilistic explanations for decision trees. We provide a detailed description of this algorithm and compare it to the state-of-the-art SAT encoding, emphasizing the gains in intelligibility and highlighting its empirical effectiveness.

Keywords: Decision Trees · Explainable AI · Probabilistic Explanations · Greedy Algorithms

1 Introduction

Context. To explain a decision to someone is to provide the details or reasons that help that person understand why such a decision was made. When decisions are made by opaque machine learning (ML) models, such as random forests [1], boosted trees [2], Markov networks, support vector machines (SVM), and deep neural networks, generating explanations becomes a complex task. However, with the increasing number of applications relying on machine learning techniques, research in Explainable Artificial Intelligence (XAI) has become essential. It aims to develop effective methods and approaches to interpret machine learning models and explain the decisions made by these models. The XAI approaches can generally be categorized as model-agnostic methods. For example, LIME [3], SHAP [4], and Anchors [5], or formal approaches that provide abductive explanations and sufficient reasons [6]. Model-agnostic methods present a significant limitation in terms of the reliability of the explanations they generate. Indeed,

it has been reported by [7] that an explanation t can be consistent with different predicted classes, which raises concerns about their reliability.

The XAI community has studied for Boolean classifiers and the problem of generating sufficient reasons (and those of minimum-size) [17, 8]. However, the authors of [13] argued that, for practical applications, sufficient reasons may be too rigid because they are specified under worst-case conditions. In other words, t is a sufficient reason for $\boldsymbol{x}$ given a classifier $\mathcal{C}$ if each "completion" of t is classified by $\mathcal{C}$ in the same way as $\boldsymbol{x}$. Sufficient reasons in general also have an important limitation, namely, that these generated explanations can be of large size even for the restricted classifier class of decision trees. It is important to keep in mind that explanation is a social process [9, 10], where users are human beings with inherent cognitive limitations. In his seminal article [11], the psychologist **G. Miller** introduced the idea of "chunking" objects by people (i.e., grouping them into a unit) and asserted that, due to the limitations of human memory, the size of chunks is limited to 7 ± 2, and if this size exceeds 7 ± 2, then the memory does not memorize these objects well. A recent work has studied probabilistic explanations as a mechanism to reduce the size of explanations and make them more concise and suitable for explaining the real world [12], which is uncertain. The decision problem involves checking whether $\boldsymbol{x}$ admits a δ-probable reason of size k under a Boolean function f, where f is specified as a `CNF` formula, is $\mathsf{NP}^{\mathsf{PP}}$-complete [12]. This result show that the problem is hard.

Motivation. Our goal is to improve the intelligibility of decision trees using probabilistic explanations. We drew inspiration from previous work on approximating probabilistic explanations [14]. This work is inspired by the results obtained in [14]. Based on these results, we propose a greedy algorithm for generating concise and reliable probabilistic explanations for decision trees. We describe this algorithm in detail and empirically compare it to the state-of-the-art SAT encoding [13], highlighting the improvement in interpretability and emphasizing its effectiveness.

Organization of the paper. We introduce the main terms and the model used in the paper (Decision Trees), Orthogonal `DNF`, as well as sufficient reasons in Section 2, and we provide detailed definitions of δ-probable explanations 3. The greedy algorithms used will be presented in detail in Section 4, while Section 5 will be devoted to experiments. We conclude this work in Section 6.

2 Decision Tree, Orthogonal `DNF`, Abductive Explanations

2.1 Formal preliminaries

For an integer n, let $[n]$ be the set $\{1, \ldots, n\}$. By $\mathcal{F}_n$ we denote the class of all Boolean functions from $\{0,1\}^n$ to $\{0,1\}$, and we use $X_n = \{x_1, \ldots, x_n\}$ to denote the set of input Boolean variables, corresponding to the features under consideration. Any assignment $\boldsymbol{x} \in \{0,1\}^n$ is called an *instance*. If $f(\boldsymbol{x}) = 1$ for some $f \in \mathcal{F}_n$, then $\boldsymbol{x}$ is called a *model* of f.

We refer to f as a *propositional formula* when it is described using the Boolean connectives $\wedge$ (conjunction), $\vee$ (disjunction), and $\neg$ (negation), together with the Boolean constants 1 (true) and 0 (false). Other connectives, like material implication $\rightarrow$ can also be considered. As usual, a *literal* ℓ is a variable x_i (a positive literal) or its negation $\neg x_i$, also denoted $\overline{x}_i$ (a negative literal). x_i and $\overline{x}_i$ are complementary literals. A positive literal x_i is associated with a positive feature (i.e., x_i is set to 1), while a negative literal $\overline{x}_i$ is associated with a negative feature (i.e., x_i is set to 0). A *term* (or *monomial*) t is a conjunction of literals, and a *clause* c is a disjunction of literals. A term is usually viewed as a (conjunctively-interpreted) set of literals, while a clause is viewed as a (disjunctively-interpreted) set of literals. A `DNF` *formula* is a disjunction of terms and a `CNF` *formula* is a conjunction of clauses. Often, a `DNF` formula is viewed as a (disjunctively-interpreted) set of terms, while a `CNF` formula is viewed as a (conjunctively-interpreted) set of clauses. The set of variables occurring in a formula f is denoted $Var(f)$ ($Lit(f)$ is the set of literals of f). A formula f is *consistent* if and only if it has a model. A `CNF` formula is *monotone* whenever every literal over a given variable in the formula has the same polarity (i.e., whenever a literal occurs in the formula, the complementary literal has no occurrence in the formula). A formula f_1 *implies* a formula f_2, noted $f_1 \models f_2$, if and only if every model of f_1 is a model of f_2. Two formulae f_1 and f_2 are *equivalent*, noted $f_1 \equiv f_2$ whenever they have the same models.

Given an instance $\boldsymbol{z} \in \{0,1\}^n$, the corresponding term is defined as

$$t_{\boldsymbol{z}} = \bigwedge_{i=1}^{n} x_i^{z_i} \text{ where } x_i^0 = \overline{x}_i \text{ and } x_i^1 = x_i$$

A term t *covers* an instance $\boldsymbol{z}$ if $t \subseteq t_{\boldsymbol{z}}$. An *implicant* of a Boolean function f is a term that implies f. A *prime implicant* of f is an implicant t of f such that no proper subset of t is an implicant of f. Dually, an *implicate* of a Boolean function f is a clause that is implied by f, and a *prime implicate* of f is an implicate c of f such that no proper subset of c is an implicate of f.

A partial instance is a tuple $\boldsymbol{y} \in \{0,1,\bot\}^n$. Intuitively, if $y[i] = \bot$, then the value of the i-th feature is undefined. $\mathbf{Comp}(\mathbf{y})$ denotes the set of completions of $\boldsymbol{y}$. We say that $\boldsymbol{y}$ is subsumed by $\boldsymbol{x}$ if it is possible to obtain $\boldsymbol{y}$ from $\boldsymbol{x}$ by exchanging some undefined values with values from $\boldsymbol{x}$, denoted $\boldsymbol{x} \subseteq \boldsymbol{y}$. We define $|\boldsymbol{y}|_\bot = |\{i \in \{1,\ldots,n\} : \boldsymbol{y}[i] = \bot\}|$.

2.2 Decision Tree

Binary Decision Tree. A (Binary) decision tree over X_n is a binary tree T, each of whose internal nodes is labeled with one of n input Boolean variables from X_n, and whose leaves are labeled 0 or 1. Every variable is assumed to appear at most once on any root-to-leaf path (read-once property). The value $T(\boldsymbol{x}) \in \{0,1\}$ of T on an input instance $\boldsymbol{x}$ is given by the label of the leaf reached from the root as follows: at each node, go to the left or right child depending

on whether the input value of the corresponding variable is 0 or 1, respectively. The size of T, denoted $|T|$, is given by the number of its nodes.

The class of decision trees over X_n is denoted $\texttt{DT}_n$. It is well-known that any decision tree $T \in \texttt{DT}_n$ can be transformed in linear time into an equivalent disjunction of terms, denoted $\texttt{DNF}(T)$ (This DNF is **an orthogonal** DNF.), where each term corresponds to a path from the root to a leaf labeled with 1. Dually, T can be transformed in linear time into a conjunction of clauses, denoted $\texttt{CNF}(T)$, where each clause is the negation of the term describing a path from the root to a leaf labeled with 0 (see [15, 16]).

2.3 Orthogonal DNF

A classical problem in Boolean theory is to derive an orthogonal disjunctive normal form of an arbitrary Boolean function. Let's consider the DNF:

$$\phi = \bigvee_{k=1}^{m} \left(\bigwedge_{i \in A_k} x_i \bigwedge_{j \in B_k} \overline{x_j} \right) \tag{1}$$

Where $A_k \cap B_k = \emptyset$ for all $k = 1, 2, \ldots, m$. $C_k = \left(\bigwedge_{i \in A_k} x_i \bigwedge_{j \in B_k} \overline{x_j} \right)$ is the k-th term of the DNF.

Definition 1 (Orthogonal DNF). *A DNF of the form 1 is said to be orthogonal if $(A_k \cap B_l) \cup (A_l \cap B_k) \neq \emptyset$ for all $k, l \in \{1, 2, \ldots, m\}$ and $k \neq l$.*

Proposition 1. *Let ϕ be a orthogonal DNF of the form 1, then the number of its models is equal to:*

$$w(\phi) = \sum_{k=1}^{m} 2^{n-|A_k|-|B_k|} = \sum_{k=1}^{m} \alpha_k$$

where $\alpha_k = 2^{n-|A_k|-|B_k|}$ for each term C_k of the DNF formula ϕ.

Remark 1. For an instance $\boldsymbol{x}$ and a decision tree $T \in \texttt{DT}_n$ such that $T(\boldsymbol{x}) = 1$, let t be a subterm of $t_{\boldsymbol{x}}$. The conditioning of $\texttt{DNF}(T)$ by a term t, denoted $\texttt{DNF}(T) \wedge t$, remains a DNF orthogonal formula [20].

Example 1. The decision tree of the figure 1 assigning bank loans using the following features: x_1: "has a permanent contract", x_2: "less than 40 years old", x_3: "annual income greater than $35K$", and x_4: "repaid a previous loan".

A DNF representation of T is given by :

$$\texttt{DNF}(T) = \{x_1 \wedge x_2 \wedge x_3,\ x_1 \wedge x_2 \wedge \overline{x_3} \wedge x_4,\ x_1 \wedge \overline{x_2} \wedge x_3 \wedge x_4,\ x_1 \wedge \overline{x_2} \wedge \overline{x_3} \wedge x_4,\ \overline{x_1} \wedge x_2 \wedge \overline{x_3}\}$$

Dually, a CNF representation of T is given by :

$$\texttt{CNF}(T) = \{\overline{x_1} \vee \overline{x_2} \vee x_3 \vee x_4,\ \overline{x_1} \vee x_2 \vee \overline{x_3} \vee x_4,\ \overline{x_1} \vee x_2 \vee x_3 \vee x_4,\ x_1 \vee x_2,\ x_1 \vee \overline{x_2} \vee \overline{x_3}\}$$

The $\texttt{DNF}(T)$ is an orthogonal DNF because it satisfies definition 1, and the number of models of $\texttt{DNF}(T)$ (denoted ϕ) is : $w(\phi) = 2^1 + 2^1 + 2^0 + 2^0 + 2^0 = 4 + 3 = 7$.

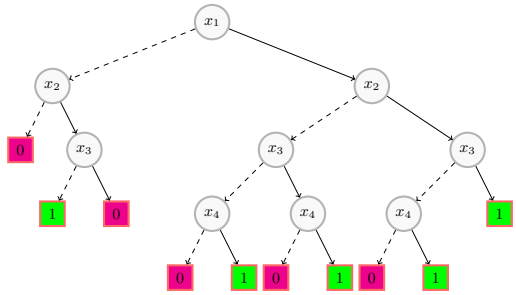

Fig. 1: A decision tree T for assigning bank loans over $\{x_1, x_2, x_3, x_4\}$.

2.4 Abductive explanations

As a salient characteristic, decision trees provide a single explicit abductive explanation for classifying any input instance $\boldsymbol{x}$: its direct reason (denoted P_x^T) [15]. P_x^T corresponding to the unique root-to-leaf path of T that is compatible with $\boldsymbol{x}$, i.e., the path-restricted explanation for $\boldsymbol{x}$ given T. In general, this reason differs from the instance itself (but may nevertheless coincide with it).

Another important notion of abductive explanations corresponds to the following concept of sufficient reason [17].

Definition 2 (Sufficient reason). *Let $T \in \mathtt{DT}_n$ and $\boldsymbol{x} \in \{0,1\}^n$ such that $T(\boldsymbol{x}) = 1$ (resp. $T(\boldsymbol{x}) = 0$). A sufficient reason for $\boldsymbol{x}$ given T is a prime implicant t of T (resp. $\neg T$) that covers T. A minimum-size sufficient reason for $\boldsymbol{x}$ given T is a sufficient reason for $\boldsymbol{x}$ given T that contains a minimal number of literals.*

Unlike sufficient reasons that are subset-minimal abductive explanations, direct reasons may contain arbitrarily many redundant features.

Example 2. Given the tree T in figure 1, and for the instance $\boldsymbol{x} = (1, 1, 1, 1)$, we observe that $T(\boldsymbol{x}) = 1$. The direct reason for $\boldsymbol{x}$ given T is $p_{\boldsymbol{x}}^T = x_1 \wedge x_2 \wedge x_3$, $x_1 \wedge x_4$ and $x_1 \wedge x_2 \wedge x_3$ are two sufficient reasons for $\boldsymbol{x}$ given T. $x_1 \wedge x_4$ is the unique minimum-size sufficient reason for $\boldsymbol{x}$ given T.

Due to cognitive limitations, abductive explanations are often too large to be interpretable, even for decision trees (see [16, 15, 18]). In these cases, we need to reduce the size of abductive explanations while still determining the predicted label with high probability. In the remainder of this article, we shed light on probabilistic explanations.

3 Probabilistic Explanations

The concept of sufficient reason is often considered a natural concept for explaining the result of a classifier, but it imposes a strict restriction by requiring that all completions of a partial instance be classified in the same way, and these reasons can also be of large size. To relax this limitation, a probabilistic generalization of explanations has been proposed by [12].

Definition 3 (δ-probable reason). *Let $\delta \in (0,1]$, a δ-probable reason for $\boldsymbol{x}$ given a Boolean function $f \in F_n$ such that $f(\boldsymbol{x}) = 1$ is a partial term t_y such that $t_y \subseteq t_x$ and:*

$$\mathbb{P}_z[f(z) \mid t \subseteq t_z] = \frac{|\{z \in \mathbf{Comp(y)} \mid f(z) = f(\boldsymbol{x})\}|}{2^{|y|_\perp}} = \frac{h(\boldsymbol{y})}{2^{|y|_\perp}} \geq \delta \tag{2}$$

In the case where f is represented by a decision tree T, equation 2 can be rewritten as:

$$\frac{w(\phi \wedge t_y)}{2^{|y|_\perp}} \geq \delta \tag{3}$$

$\phi = \texttt{DNF}(T)$ and $t_y = \bigwedge_{i=1}^{n} x_i^{y_i}$, $|y|_\perp = n - |y|$, and $w(\phi \wedge t_y)$ is the number of models of $\phi \wedge t_y$.

Example 3. Let T be the tree in figure 1, and let $\boldsymbol{x} = (1,1,1,1)$ ($T(\boldsymbol{x}) = 1$). We observe that $t_{\{x_1,x_2\}} = x_1 \wedge x_2$ and $t_{\{x_1,x_3\}} = x_1 \wedge x_3$ are $\frac{3}{4}$**-probable reasons** for $\boldsymbol{x}$ given T, and $t_{\{x_1,x_4\}} = x_1 \wedge x_4$ is a 1-probable reason (sufficient reason).

We delve deeper into the computational challenges of probabilistic explanations after an in-depth exploration of their fundamental concepts. To address these challenges, we introduce a tailor-made greedy algorithm, offering a promising route to efficiently derive these explanations.

4 Greedy algorithms

The decision problem of determining whether $\boldsymbol{x}$ admits a δ-probable reason of a certain size k for a Boolean function f, where f is represented by a CNF formula, is $\mathsf{NP}^{\mathsf{PP}}$-complete [19], and NP-hard when f is a decision tree [13]. These results demonstrate the difficulty of deriving a probabilistic explanation even for the restricted class of decision trees. In this work, we focus on the restricted class of decision trees. Indeed, a SAT encoding has been proposed by [13] to derive a δ-probable reason for $\boldsymbol{x}$ given T. Our main motivation for using a greedy algorithm to derive a probabilistic explanation stems from the fact that the time required to obtain results using SAT encoding is high in many cases [14].

4.1 Greedy Algorithm

In the following, we propose a greedy algorithm to derive a set S based on a set of features $E \subseteq \boldsymbol{x}$ of exactly size $k \leq |E|$ (or at most size k) and a confidence parameter $\delta \in (0,1]$ to be determined. For a decision tree $T \in \texttt{DT}_n$, let $\phi = \texttt{DNF}(T)$, we exploit the fact that this DNF is orthogonal [15] to perform calculations in the least costly manner. The orthogonality of ϕ allows for counting the models of T in linear time. Thus, verifying the inequality $h(S) = \frac{w(\phi \wedge t_s)}{2^{n-|s|}} \geq \delta$ can be performed in linear time. Below, we explicitly detail our greedy algorithm.

Algorithm 1 is an adapted version of the GA algorithm proposed in [14]. This algorithm aims to find a probabilistic explanation of size k (or at most k) that minimizes the classification error (maximizes the value of δ) [14].

Algorithm 1

Input: a tree T, a set $E \subseteq \boldsymbol{x}$, and $k \leq |E|$
Output: a δ^*-probable reason
$E \leftarrow Lit(\boldsymbol{x})$, $S \leftarrow \emptyset$, $\phi \leftarrow \texttt{DNF}(T)$; `/*we define` $h(S) = \frac{w(\phi \wedge t_s)}{2^{n-|s|}}$`, E could be the instance` $\boldsymbol{x}$`, or` $p_{\boldsymbol{x}}^T$ `(direct reason), or a subset` $I \subseteq \boldsymbol{x}$ `*/`

for $l \in \{1, ..., k\}$ **do**
  $e^* \longleftarrow \underset{c \in E}{\text{argmax}}\ h(S \cup \{e\})$
  $S \longleftarrow S \cup \{e^*\}$
  $E \longleftarrow E - \{e^*\}$
$\delta^* = \frac{w(\phi \wedge t_S)}{2^{n-k}}$
return S, δ^*

Proposition 2. *For a decision tree $T \in \texttt{DT}_n$ and an instance $\boldsymbol{x}$, algorithm 1 runs in $O(k \cdot n^2|T|)$ time.*

Example 4. For the tree in figure 1. Let $\boldsymbol{x} = (1, 1, 1, 1)$. We search a probable reason of size $k = 2$. The steps of algorithm 1: $x_1 = \underset{e \in \{x_1, x_2, x_3, x_4\}}{\text{argmax}}\ h(\{e\})$[1], then $S = \{x_1\}$ and $x_4 = \underset{e \in \{x_2, x_3, x_4\}}{\text{argmax}}\ h(\{x_1\} \cup \{e\})$. Finally, we obtain $S = \{x_1, x_4\}$ and $\delta^* = \frac{w(\phi \wedge x_1 \wedge x_4)}{2^{4-2}} = 1$. Thus, algorithm 1 has captured a minimum-size sufficient reason.

4.2 Deriving a δ-probable reason

Finally, since algorithm 1 runs in **linear time** and typically captures a probabilistic explanation with a reliable δ parameter (maximum value for a limit size k) in practice. In the following, we will slightly modify algorithm 1 to derive a probabilistic explanation for the instance $\boldsymbol{x}$ given a decision tree T and a confidence parameter $\delta \in (0, 1]$. It is also possible to compute a specific probabilistic explanation based on a subset of features $E \subseteq \boldsymbol{x}$. To achieve this, we adjust the input of algorithm 2 using the subset E as the input to algorithm 2.

5 Experiments

We conducted several experiments to evaluate the performance of our two algorithms. Our objectives are as follows:

– Measure the exactitude of the results from algorithm 1 in computing a probable reason of a certain size k and of size at most k. In particular, we compared the value of δ^* associated with the reason found by algorithm 1 to the optimal value δ_{opt} obtained through binary search of the SAT encoding [13].

[1] We recall that h is defined as $h(S) = \frac{w(\phi \wedge t_s)}{2^{n-|s|}}$

Algorithm 2 Derivation of a δ-probable reason

Input: a decision tree T, $\delta \in (0, 1]$, a subset $E \subseteq \boldsymbol{x}$
Output: a δ-probable reason
$\phi \leftarrow \texttt{DNF}(T)$, $S \leftarrow \emptyset$; /*we recall $h(S) = \frac{w(\phi \wedge t_s)}{2^{n-|s|}}$*/
for $l \in \{1, ..., |E|\}$ **do**
  $e^* \longleftarrow \underset{c \in E}{\text{argmax}}\ h(S \cup \{e\})$
  $S \longleftarrow S \cup \{e^*\}$
  if $\frac{h(S)}{2^{n-l}} \geq \delta$ **then**
    break
    return S
  $E \longleftarrow E - \{e^*\}$
$\delta^* = \frac{w(\phi \wedge t_S)}{2^{n-|S|}}$
return S, δ^*

– Evaluate the gain in intelligibility resulting from the emphasis on the size of probabilistic explanations (computed using algorithm 2) compared to the size of direct (P_x^T), sufficient (SR) (and minimum-size (MR)) reasons for an instance $\boldsymbol{x}$ given a decision tree T. We found that algorithm 1 for computing δ-probable reasons is more efficient in terms of computation time than the SAT method and can handle larger-scale problems where the SAT method becomes inefficient in terms of computation time.

5.1 Experimental Protocol

We considered 32 datasets, which are standard benchmarks from the well-known Kaggle[2], OpenML[3], and UCI[4] websites. Notably, mnist38 and mnist49 are subsets of the mnist dataset. Categorical features were treated as arbitrary numbers. As for numerical features, they were binarized using the decision tree learning algorithm employed. Classification performances for T_b were measured as the average accuracy achieved on a test set of over 150 instances. For decision tree learning, we used the CART algorithm, specifically its implementation provided by the Scikit-Learn library [21]. All hyperparameters were set to default.

For each dataset b, each decision tree T_b, and each instance $\boldsymbol{x}$ from the corresponding test set, to assess the reliability of our algorithm, we computed the δ^* corresponding to a probable reason of size at most k. To do so, we started by using the instance $E = t_{\boldsymbol{x}}$, then the direct reason $E = p_{\boldsymbol{x}}^T$, and finally a sufficient reason $E = SR(\boldsymbol{x})$, which we compared to δ_{opt} (see Table 1).

To compute a δ_{opt} corresponding to a probable reason of exact size k, we performed a binary search by extending the SAT encoding proposed by [13] to include the CNF encoding of the cardinality constraint $(\sum_{i \in E} x_i) = k$. Regarding

[2] `www.kaggle.com`
[3] `www.openml.org`
[4] `archive.ics.uci.edu/ml/`

dataset			decision tree			|Reason|			$\lvert\delta_{opt} - \delta^*_{algo_1}\rvert$				SAT
name	#F	#I	%A	$\lvert T\rvert$	Depth	$\lvert p_x^T\rvert$	$\lvert SR\rvert$	$\lvert MR\rvert$	k	t_x	p_x^T	SR	Time (s)
horse	29	299	84.44	34	13	6.0	6.6	5.2	5	0.0085	0.0004	0.0004	20,45
hungarian	13	294	68.54	62	12	6.0	5.8	4.9	5	0.008	0.002	0.002	5.53
primary. t	23	399	87.25	53	14	6.0	7.1	4.8	5	0.0294	0.0004	0.0004	17.25
mushroom	17	8124	100.0	20	7	5.0	4.7	4.4	5	0.0002	0.0002	0.0002	2.53
cars	21	406	97.54	30	10	4.4	5.5	4	4	0.0506	0.0026	0.0026	9.46
glass	31	214	83.08	36	11	6.9	8.4	6.4	5	0.009	0.0047	0.0051	3.82
placement	18	215	95.38	19	10	4.0	4.4	3.2	3	0.0001	0.0001	0.0001	0.72
spect	19	265	78.75	42	15	6.3	6.3	4.7	5	0.04	0.002	0.002	3.71
colic	40	368	80.18	55	13	8.0	10.2	7.5	6	0.0003	0.0002	0.0004	19.23
biomed	15	209	98.41	21	11	4.6	4.1	3.7	4	0.0004	0.0001	0.0001	0.67
student-por	30	649	91.79	33	9	5.0	6.3	4.9	4	0.0037	0.0007	0.0007	41.67
tic-tac-toe	9	958	97.92	83	9	5.8	4.8	4.4	4	0.0062	0.0005	0.0001	1.22
schizo	33	340	93.14	34	11	5.9	5.3	4.7	5	0.0019	0.0002	0.0002	4.53
vehicle	23	846	96.06	31	12	5.7	6.6	5.2	5	0.0005	0.0007	0.0006	3.34
balance	17	625	86.7	77	12	5.6	5.8	4.7	5	0.0048	0.0005	0.0005	13.17
compas	40	6172	66.14	570	20	11.1	9.4	6.6	7	0.006	0.006	0.006	1482.32
employee	63	4653	82.45	653	20	10.4	12.0	8.1	7	0.13	0.08	0.06	1314.84
fetal. h	93	2127	93.42	110	19	12.2	17.3	10.8	7	0.18	0.12	0.14	1125.83

Table 1: Statistics on the reliability of probabilistic explanations generated by algorithm 1 and comparison with the SAT method

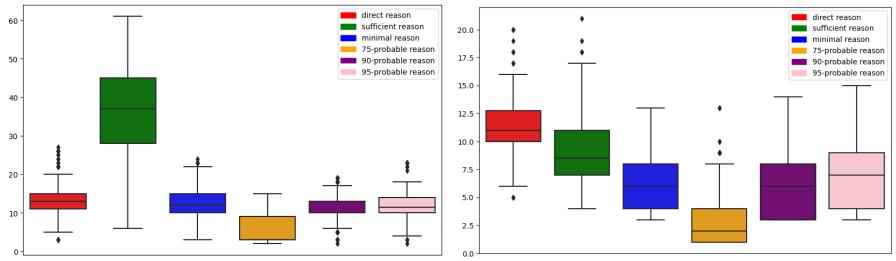

Fig. 2: Boxplots for "spambase" (left) and "compas" (right), representing the sizes of direct reasons, sufficient reasons, minimum-size sufficient reasons, and {75%, 90%, 95%}-probable reasons.

the δ_{opt} corresponding to a probable reason of size at most k, the SAT encoding is also extended [13] by adding the clause $C_E = \{x_i : (x_E)_i = 1\} \vee \{\overline{x_i} : (x_E)_i = 0\}$ to the CNF of the original encoding. The binary search was performed with an error precision of 10^{-3} and a time limit of $1800(s)$ by instance, which amounts to 10 SAT calls. Regarding Table 2, it is noteworthy that the SAT encoding does not scale due to memory consumption explosion. Therefore, for comparison (for $k = 7$), we limited ourselves to starting from the direct reason. For the computation of δ_{opt}, we used an exact algorithm consisting of testing all combinations $\frac{h(S)}{2^{n-7}}$ of subsets S of p_x^T of exact size 7, and taking the maximum value.

To assess the **improvement in intelligibility** resulting from the emphasis on the size of probabilistic explanations, we reported the sizes of a direct reason, a sufficient reason for $\boldsymbol{x}$ given T_b, and a minimum-size sufficient reason for $\boldsymbol{x}$ given T_b calculated using the **PyXAI** tool [22]. We then reported the computation

times required to report δ_{opt} using binary search of the **SAT** encoding. For this, we used the *Pysat* library [5], which provides the implementation of the solver Glucose 4 (we set a time limit of 1800 seconds). And to calculate minimum-size sufficient reasons, we used the Pysat library, which allows using the Partial MaxSAT solver *RC*2. This solver was run using the parameters corresponding to the "Glucose" configuration.

All experiments were conducted on a computer equipped with an Intel(R) Core(TM) $i9-9900$ processor running at 3.10 GHz and 64 GiB of memory.

5.2 Results

The table 1 presents an excerpt of our results for 18 datasets. The first column gives the name of the dataset b. F represents the number of binary features, I the number of instances, $\%A$ the accuracy of tree T_b, $|T|$ its size, and $Depth$ represents the depth of the tree. The column $|Reason|$ indicates the average size of different calculated reasons: P_x^T, $|SR|$, $|MR|$ respectively represent the size of the direct reason, the size of the sufficient reason, and the size of the minimum-size sufficient reason. $|\delta_{opt} - \delta_{algo_1}|$ indicates the average error of δ corresponding to the reason returned by algorithm 1 for a size at most k, starting respectively from the complete instance $|t_x|$, from the direct reason $|p_x^T|$, and from the sufficient reason $|SR|$. **SAT** indicates the average computation times of the dichotomous search starting from the direct reason. The datasets in magenta indicate that the 1800 seconds time limit was reached at least once. We notice that the average error $|\delta_{opt} - \delta_{algo_1}|$ is generally of the order of 10^{-3}, especially when the input to the algorithm is p_x^T and SR. However, the accuracy slightly decreases when the input is the complete instance $\boldsymbol{x}$. This shows that our greedy algorithm generally finds a probable reason of size at most k with the highest possible confidence parameter δ. We also note that the error $|\delta_{opt} - \delta_{algo_1}|$ is high for datasets where the time limit is exceeded (in magenta). This is because the dichotomous search stopped before reaching a precision of $\epsilon = 10^{-3}$. In order to improve the intelligibility of our explanations, we calculated, using algorithm 2 (figure 2), δ-probable reasons for $\delta = 0.75$, $\delta = 0.9$, and $\delta = 0.95$. Our results show that probabilistic explanations are generally more concise than abductive explanations, including those of minimum-size ("minimum-size sufficient reasons").

Regarding the computation time required by the dichotomous search of the SAT encoding, we noticed that this time was very high, reaching up to 25 minutes in some cases, especially when the size of the tree $|T|$ is large (as in the case of "Compas" and "Employee") or when the number of binary features is high (as in the case of "Fetal. h"). We did not include the computation times of our greedy algorithm, as the average time per instance required for all our experiments does not exceed 0.5 second, demonstrating the computational efficiency of our algorithm compared to the SAT encoding. It is interesting to note that probable reasons of a certain size k (or at most k) offer users the possibility to control the

[5] `https://pysathq.github.io/`

size of the explanation based on a subset of variables of their choice. It is also essential to emphasize that these reasons are much more relevant to the user than a randomly selected reason.

The results of our experiments in Table 2 highlight the difficulty of computing probabilistic reasons using the SAT encoding, with the 1800-second time limit consistently reached. We included datasets of large dimensions, for which the SAT encoding fails to scale. We used the direct reason p_x^T as input to algorithm 1 and set the output size to $k = 7$. We notice that algorithm 1 is very efficient on these large datasets, and the error does not exceed the order of 10^{-3}. This confirms the reliability of our results, as presented in Table 1. It is also worth noting that this gives an advantage to our approach, especially when the SAT encoding fails to scale.

Dataset	#I	#F	%A	$\lvert\delta_{opt} - \delta^*\rvert$	$\lvert P_x^T\rvert$
Gisette	5000	7000	98.56	0	21.42
Mnist38	13966	784	95.44	0.0003	17.89
Mnist49	13782	784	95.48	0	15.57
Christine	1636	5418	61.25	0.001	9.47
Bank	41188	882	89.49	0.0003	13
Dexter	20000	600	90.70	0	8.32
Gina-agnostic	970	48842	85.84	0.0001	9.84
Gina	970	3468	84.53	0.0001	9.69
Farm-ads	54877	1543	86.8	0.0003	23.15
Cnae	1080	856	92.59	0.0006	19.07
Dorothea	1150	10^5	91.8	0	12.9
Adult	48842	2974	81.16	0.001	16.43
Spambase	4601	236	92.11	0.0002	16.09
Ad-data	5000	1023	99.19	0	9.29

Table 2: Table of results for 14 datasets: number of instances I, number of binary features F, accuracy $\%A$, and average error of $|\delta_{opt} - \delta^*|$ for $k = 7$.

To illustrate the **gain in intelligibility** achieved when transitioning from abductive explanations (direct reasons and sufficient reasons) to probabilistic reasons (δ-probable reasons) for 150 instances, we created several box plots for two datasets: "compas" (on the right), which illustrates the transition from direct reasons to {75%, 90%, 95%}-probable reasons, and "spambase" (on the left), which illustrates the transition from sufficient reasons to {75%, 90%, 95%}-probable reasons. Figure 2 presents these box plots. We can observe a significant reduction in the number of features used in direct reasons during the transition to a 0.75-probable reason, as well as for sufficient reasons.

Finally, since the reduction in the size of reasons obtained by considering 75%-probable reasons compared to direct reasons and sufficient reasons seemed significant, we also conducted additional experiments to gain a clearer view of the reduction that can be achieved with variations in δ. We calculated the sizes

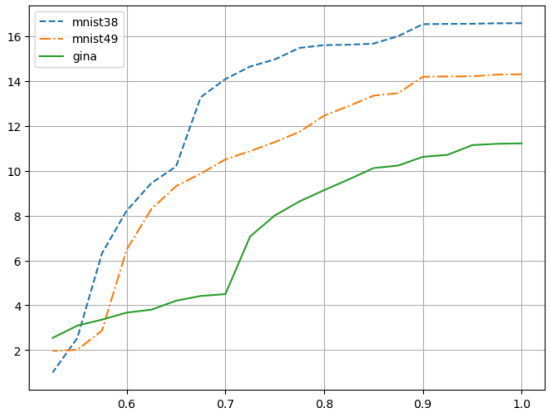

Fig. 3: Average size of δ-probable reasons as δ varies from 0.5 to 1 (sufficient reasons) for the datasets "mnist38", "mnist49", and "gina".

of δ-probable reasons for instances as δ varies from 0.5 to 1. The figure presents such plots for the datasets "mnist38", "mnist49", and "gina". As expected, we can observe that the average size of δ-probable reasons gradually increases as δ increases and stabilizes when the algorithm captures a sufficient reason. This clearly demonstrates the gain in intelligibility obtained.

6 Conclusion

In this paper, we leveraged recent work on approximating probabilistic explanations to enhance intelligibility, particularly in the context of decision trees. By nature, probabilistic explanations cannot be larger than sufficient reasons, but they prove to be valuable concepts for obtaining more understandable explanations for human users. All these reasons are smaller than the instances themselves. Although minimum-size sufficient reasons are the shortest abductive explanations possible. Our experiments showed that additional size reduction can be achieved with δ-probable reasons. Furthermore, we find that our greedy algorithm enables the derivation of reliable and concise probable reasons, with remarkably low computational cost compared to the SAT method. We can assert that deriving reliable probable reasons is considerably simplified using our greedy algorithm, making the intelligibility gain they provide almost cost-free. In our future work, we plan to investigate the approximation of probabilistic explanations on other types of classifiers, namely random forests, boosted Trees.

Acknowledgements

Many thanks to the anonymous reviewers for their comments and insights. This work has benefited from the support of the AI Chair **SAFE IA** (funded by the UTC Foundation for Innovation).

References

1. Leo Breiman. "Random Forests." Machine Learning 45 (2001): 5-32.
2. Chen, Tianqi and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016): n. pag.
3. Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier." Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016): n. pag.
4. Lundberg, Scott M. and Su-In Lee. "A Unified Approach to Interpreting Model Predictions." Neural Information Processing Systems (2017).
5. Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. "Anchors: High-Precision Model-Agnostic Explanations." AAAI Conference on Artificial Intelligence (2018).
6. Andy Shih, Arthur Choi and Adnan Darwiche. "A Symbolic Approach to Explaining Bayesian Network Classifiers." ArXiv abs/1805.03364 (2018): n. pag.
7. Alexey Ignatiev, Nina Narodytska et Joao Marques-Silva. "Abduction-Based Explanations for Machine Learning Models." AAAI Conference on Artificial Intelligence (2018).
8. Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez and Pierre Marquis. "On the Computational Intelligibility of Boolean Classifiers." ArXiv abs/2104.06172 (2021): n. pag.
9. Miller, Tim. "Explanation in Artificial Intelligence: Insights from the Social Sciences." Artif. Intell. 267 (2017): 1-38.
10. Christoph Molnar, Giuseppe Casalicchio and Bernd Bischl. "Interpretable Machine Learning - A Brief History, State-of-the-Art and Challenges." PKDD/ECML Workshops (2020).
11. Miller, George A. "The magical number seven plus or minus two: some limits on our capacity for processing information." Psychological review 63 2 (1956): 81-97.
12. Stephan Waeldchen, Jan Macdonald, Sascha Hauch and Gitta Kutyniok. "The Computational Complexity of Understanding Binary Classifier Decisions." J. Artif. Intell. Res. 70 (2021): 351-387.
13. Marcelo Arenas, Pablo Barceló, Miguel Romero and Bernardo Subercaseaux. "On Computing Probabilistic Explanations for Decision Trees." ArXiv abs/2207.12213 (2022): n. pag.
14. Bounia, Louenas and Frédéric Koriche. "Approximating probabilistic explanations via supermodular minimization." Conference on Uncertainty in Artificial Intelligence (2023).
15. Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez and Pierre Marquis. "On the explanatory power of Boolean decision trees." Data Knowl. Eng. 142 (2022): 102088.
16. Yacine Izza, Alexey Ignatiev and Joao Marques-Silva. "On Explaining Decision Trees." ArXiv abs/2010.11034 (2020): n. pag.
17. Darwiche, Adnan and Auguste Hirth. "On The Reasons Behind Decisions." European Conference on Artificial Intelligence (2020).
18. Louenas Bounia, Matthieu Goliot and Anasse Chafik. "Impact of Weight Functions on Preferred Abductive Explanations for Decision Trees." ICCBR Workshops (2023).
19. Stephan Wäldchen, Jan Macdonald, Sascha Hauch and Gitta Kutyniok. "The Computational Complexity of Understanding Network Decisions." ArXiv abs/1905.09163 (2019): n. pag.

20. Darwiche, Adnan. “Compiling Knowledge into Decomposable Negation Normal Form.” International Joint Conference on Artificial Intelligence (1999).
21. Pedregosa, Fabian et al. “Scikit-learn: Machine Learning in Python.” ArXiv abs/1201.0490 (2011): n. pag.
22. Gilles Audemard, Steve Bellart, Louenas Bounia, Jean-Marie Lagniez, Pierre Marquis and Nicolas Szczepanski. “PyXAI : calculer en Python des explications pour des modèles d’apprentissage supervisé.” Extraction et la Gestion des Connaissances EGC (2023).