Computing Abductive Explanations for Boosted Trees

Gilles Audemard, Jean-Marie Lagniez, Pierre Marquis, Nicolas Szczepanski

Computing Abductive Explanations for Boosted Trees

Gilles Audemard Univ. Artois, CNRS, CRIL
Jean-Marie Lagniez Univ. Artois, CNRS, CRIL
Pierre Marquis Univ. Artois, CNRS, CRIL, IUF
Nicolas Szczepanski Univ. Artois, CNRS, CRIL

Abstract

Boosted trees is a dominant ML model, exhibiting high accuracy. However, boosted trees are hardly intelligible, and this is a problem whenever they are used in safety-critical applications. Indeed, in such a context, provably sound explanations for the predictions made are expected. Recent work have shown how subset-minimal abductive explanations can be derived for boosted trees, using automated reasoning techniques. However, the generation of such well-founded explanations is intractable in the general case. To improve the scalability of their generation, we introduce the notion of tree-specific explanation for a boosted tree. We show that tree-specific explanations are provably sound abductive explanations that can be computed in polynomial time. We also explain how to derive a subset-minimal abductive explanation from a tree-specific explanation. Experiments on various datasets show the computational benefits of leveraging tree-specific explanations for deriving subset-minimal abductive explanations.

1 INTRODUCTION

The deployment of ML models in a large spectrum of applications has triggered the fast-growing development of eXplainable AI (XAI) (see for instance (Frosst and Hinton, 2017; Wang and Rudin, 2015; Guidotti et al., 2019; Hooker et al., 2019; Huysmans et al., 2011; Ignatiev et al., 2019a; Kim et al., 2018; Lundberg and Lee, 2017; Miller, 2019; Molnar, 2019; Shih et al., 2019b)). Models with high prediction performance are usually considered as poorly intelligible (Molnar, 2019; Arrieta et al., 2020; Lundberg et al., 2020; Caruana et al., 2020; Rudin et al., 2021). Among them is the family of *boosted trees* (Friedman, 2001), which is among the state-of-the-art ML models when dealing with tabular data (Borisov et al., 2021).

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

Motivations. The design of efficient methods for interpreting ML models and explaining their decisions (i.e., deriving local explanations) is acknowledged as an issue of the utmost importance when ML models are to be used in safety-critical applications (Marques-Silva and Ignatiev, 2022). Since most existing approaches to explaining ML models deliver model-agnostic explanations, they cannot be used in any high-risk context because the explanations that are generated are *unsound*: one can find "counterexamples" for them, i.e., pairs of instances that have the same explanation but are nevertheless classified differently by the model (Ignatiev et al., 2019b). In particular, (Ignatiev, 2020) shows that the amount of "counterexamples" can be high when using some of the most popular approaches for computing model-agnostic explanations, namely LIME (Ribeiro et al., 2016), Anchors (Ribeiro et al., 2018), and SHAP (Lundberg and Lee, 2017).

In order to avoid the generation of unsound explanations, a number of alternative approaches, falling under the *formal XAI* umbrella (Marques-Silva and Ignatiev, 2022), have shown how ML models of various types (including "black" boxes) can be associated with Boolean circuits (alias transparent or "white" boxes), exhibiting the same input-output behaviours (see among others (Narodytska et al., 2018; Shih et al., 2018b, 2019a)). Thanks to such mappings, XAI queries about classifiers, including the generation of explanations, can be delegated to the corresponding circuits (see for instance (Darwiche and Hirth, 2020; Barceló et al., 2020; Parmentier and Vidal, 2021)). The price to be paid for ensuring that the explanations that are generated are sound is a certain lack of scalability (the derivation of explanations is often NP-hard in the broad sense).

Ensemble methods (bagging, boosting, stacking, etc.) have already been considered in such a perspective. Unlike what happens for decision trees, the derivation of *abductive explanations* is not a computationally easy task when tree ensembles are considered. An abductive explanation for an instance given a classifier is a subset of the characteristics of the instance that is enough to justify how the instance has been classified. In order to avoid the presence of useless characteristics in explanations and consider the instance itself as a valuable explanation, subset-minimal abductive explanations (alias sufficient reasons (Darwiche

and Hirth, 2020)) are often targeted. (Choi et al., 2020; Izza and Marques-Silva, 2021; Audemard et al., 2022a) show how to derive abductive explanations for random forests (Breiman, 2001). As to boosted trees, (Ignatiev et al., 2019b) provides an SMT (satisfiability modulo theory) encoding scheme for boosted trees and shows how to use an SMT solver to compute sufficient reasons based on the encoding scheme. The corresponding XAI tool is called `XPlainer` (https://github.com/alexeyignatiev/xplainer). (Ignatiev et al., 2022) presents another encoding scheme, based on MaxSAT (maximum satisfiability), and indicates how to exploit a MaxSAT solver to compute sufficient reasons based on it. The associated tool is called `XReason` (https://github.com/alexeyignatiev/xreason). Deciding whether a given explanation is sound is intractable for boosted trees. Accordingly, though `XReason` typically exhibits better performances than `XPlainer`, its scalability is still an issue.

Contributions. Considering boosted trees, the main contribution of this paper is *a new approach to derive sufficient reasons, that is more efficient in practice than SOTA methods, as identified in the literature*. To reach the objective, we introduce the notion of *tree-specific explanation* (TS-explanation for short) given a boosted tree.

TS-explanations elaborate on the notion of majoritary reasons for random forests, introduced in (Audemard et al., 2022a). Both are (possibly redundant) abductive explanations that can be computed efficiently thanks to a greedy algorithm. The main differences between the present work and the one described in (Audemard et al., 2022a) are as follows. In (Audemard et al., 2022a), the computation of majoritary reasons is not used as a preprocessing to the computation of sufficient reasons, unlike what is done here. More importantly, majoritary reasons are about random forests, not gradient boosted trees (which is usually acknowledged as a more accurate, yet more opaque, model). Finally, the work presented in (Audemard et al., 2022a) concerns only binary classification and Boolean features, while in the present work, none of those restrictions is required.

Notably, our approach applies to boosted trees *as they have been learned*, and whatever the learning algorithm used to generate them. Thus, many boosted tree learning algorithms can be used upstream. Those algorithms may differ in a number of aspects, including the method used to compute gradients (standard gradient algorithm vs. Newton-Raphson), the use of subsampling, the use of a regularization function (lasso or ridge), the handling of categorical features, etc. The combination of techniques achieves several trade-offs in terms of accuracy of the resulting trees and the computation time required to generate them. In particular, XGBoost (Chen and Guestrin, 2016), LightGBM (Ke et al., 2017) and CatBoost (Prokhorenkova et al., 2018) that implement some of the combinations above generate boosted trees that can be used as inputs for our approach for computing explanations. This is also the case of AdaBoost[1] (Freund and Schapire, 1997; Schapire and Freund, 2014).

To be more precise, in the following we show that TS-explanations are abductive explanations that can be computed *in polynomial time*. This heavily contrasts with sufficient reasons, which cannot be derived in polynomial time (unless P = NP) when boosted trees are considered. TS-explanations are provably *sound*: no "counterexamples" for them may exist. While TS-explanations are not subset-minimal abductive explanations in the general case, we show that they are *close to sufficient reasons in practice*. Furthermore, because sufficient reasons can be derived from TS-explanations, computing TS-explanations can be exploited as a *preprocessing step* in the derivation of sufficient reasons. Experiments on various datasets show that leveraging TS-explanations for generating sufficient reasons is a valuable approach.

The proofs of the propositions given in the paper are reported to a final appendix. A description of the datasets and the code used in our experiments are provided as a supplementary material, available at www.cril.fr/expekctation/aistats23.zip.

2 PRELIMINARIES

For an integer n, let $[n] = \{1, \cdots, n\}$. We consider a finite set $\{A_1, \ldots, A_n\}$ of *attributes* (aka *features*) where each attribute A_i ($i \in [n]$) takes its value in a domain D_i. Three types of attributes are taken into account: *numerical* (the domain D_i is a totally ordered set of numbers, typically real numbers $\mathbb{R}$, or integers $\mathbb{Z}$), *categorical* (the domain is a set of values that are not specifically ordered, e.g., $D_i = \{b(lue), w(hite), r(ed)\}$), or *Boolean* (the domain D_i is $\mathbb{B} = \{0, 1\}$). An *instance* $\boldsymbol{x}$ is a vector $(v_1, \ldots, v_n)$ where each v_i ($i \in [n]$) is an element of D_i. $\boldsymbol{x}$ is also viewed as a term, i.e., a conjunctively-interpreted set of literals $t_{\boldsymbol{x}} = \{(A_i = v_i) : i \in [n]\}$, stating that each attribute A_i takes the corresponding value v_i. Each pair $A_i = v_i$ is called a *characteristic* of the instance. $\boldsymbol{X}$ denotes the set of all instances.

In the binary case, a *classifier* f is defined as a mapping from $\boldsymbol{X}$ to $\{1, 0\}$. When $f(\boldsymbol{x}) = 1$, $\boldsymbol{x}$ is said to be a *positive instance*, otherwise it is a *negative instance*. The set of all positive instances forms a target concept, and the set of all negative instances is the complementary concept. More generally, in the multi-class case, more than one concept (together with the complementary concept) is considered. A *classifier* f is then defined as a mapping from $\boldsymbol{X}$ to $[m]$ with $m > 1$. Each integer from $[m]$ identifies a class and

[1] In this case, in order to recover the same format of boosted trees as the one used by the other boosted tree learning algorithms, the coefficients associated with the trees generated by AdaBoost must be propagated to the leaves of those trees using a simple product.

when $f(\boldsymbol{x}) = j$ with $j \in [m]$, the instance $\boldsymbol{x}$ is said to be classified as an element of class j.

Trees and Forests. A *regression tree* over $\{A_1, \ldots, A_n\}$ is a binary tree T, each of its internal nodes being labeled with a Boolean condition on an attribute from $\{A_1, \ldots, A_n\}$, and leaves are labeled by real numbers. The conditions are typically of the form $A_i > v_j$ with v_j a number when A_i is a numerical attribute, $A_i = v_j$ when A_i is a categorical attribute, and A_i (or equivalently $A_i = 1$) when A_i is a Boolean attribute. The weight $w(T, \boldsymbol{x}) \in \mathbb{R}$ of T for an input instance $\boldsymbol{x} \in \boldsymbol{X}$ 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 or not the condition labelling the node is satisfied by $\boldsymbol{x}$. $w(T, \boldsymbol{x})$ can also be viewed as the "output" of T on $\boldsymbol{x}$. In the binary classification case, a *decision tree* over $\{A_1, \ldots, A_n\}$ is a regression tree over $\{A_1, \ldots, A_n\}$ where leaves are labeled in $\{0, 1\}$ (Breiman et al., 1984; Quinlan, 1986).

A *forest* over $\{A_1, \ldots, A_n\}$ associated with a class $j \in [m]$ is an ensemble of trees $F^j = \{T_1^j, \cdots, T_{p_j}^j\}$, where each T_k^j ($k \in [p_j]$) is a regression tree over $\{A_1, \ldots, A_n\}$, and such that the weight $w(F^j, \boldsymbol{x}) \in \mathbb{R}$ of F^j for an input instance $\boldsymbol{x} \in \boldsymbol{X}$ is given by

$$w(F^j, \boldsymbol{x}) = \sum_{k=1}^{p_j} w(T_k^j, \boldsymbol{x}).$$

A *random forest* over $\{A_1, \ldots, A_n\}$ is a forest over $\{A_1, \ldots, A_n\}$ that consists only of decision trees (Breiman, 2001).

In the binary classification case, a *boosted tree* BT over $\{A_1, \ldots, A_n\}$ is a forest $F = \{T_1, \cdots, T_p\}$. In a multi-class context, a *boosted tree* BT over $\{A_1, \ldots, A_n\}$ is a collection of m forests $BT = \{F^1, \ldots, F^m\}$ over $\{A_1, \ldots, A_n\}$ (Freund and Schapire, 1997; Schapire and Freund, 2014; Friedman, 2001). The size of a forest F^j is given by $|F^j| = \sum_{k=1}^{p_j} |T_k^j|$, where $|T_k^j|$ is the number of nodes occurring in T_k^j. The size of a boosted tree BT is given by $|BT| = \sum_{j=1}^{m} |F^j|$.

In the binary classification case, an instance $\boldsymbol{x}$ is considered as a positive instance when $w(F, \boldsymbol{x}) > 0$ and as a negative instance otherwise. We note $BT(\boldsymbol{x}) = 1$ in the first case and $BT(\boldsymbol{x}) = 0$ in the second case. In a multi-class context, an instance $\boldsymbol{x}$ is classified as an element of class $j \in [m]$, noted $BT(\boldsymbol{x}) = j$, if and only if $w(F^j, \boldsymbol{x}) > w(F^i, \boldsymbol{x})$ for every $i \in [m] \setminus \{j\}$. If $w(F^j, \boldsymbol{x}) = w(F^i, \boldsymbol{x})$ for every $i, j \in [m]$, then $BT(\boldsymbol{x})$ is defined as a preset element of $[m]$ (e.g., a most frequent class in the dataset used to learn BT). Whatever the case (binary or multi-class), computing $BT(\boldsymbol{x})$ can be achieved in polynomial time in $|BT| + n$.

Example 1. *As an example of binary classification, consider four attributes: A_1, A_2 are numerical, A_3 is categorical, and A_4 is Boolean. The boosted tree $BT = \{F\}$ in*

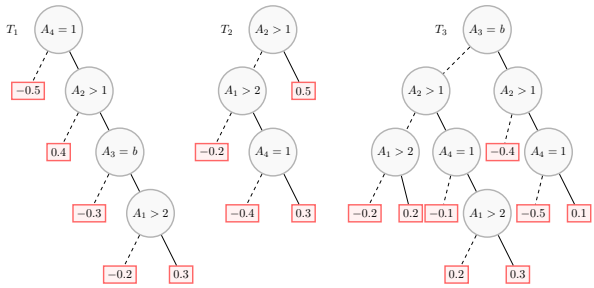


Figure 1: A boosted tree $BT = \{F\}$ consisting of a single forest $F = \{T_1, T_2, T_3\}$. In each tree, the left (dashed) arc (resp. the right (plain) arc) outgoing from any node labelled by a condition c corresponds to the case c is false (resp. true).

Figure 1 is composed of a single forest F, which consists of three regression trees T_1, T_2, T_3.

Consider $\boldsymbol{x} = (A_1 = 4, A_2 = 3, A_3 = b, A_4 = 1)$. We have $w(T_1, \boldsymbol{x}) = 0.3$, $w(T_2, \boldsymbol{x}) = 0.5$, and $w(T_3, \boldsymbol{x}) = 0.1$. So $w(F, \boldsymbol{x}) = 0.9$, and $\boldsymbol{x}$ is classified as a positive instance by F, thus it is classified as such by BT: $BT(\boldsymbol{x}) = 1$.

Abductive Explanations & Sufficient Reasons. Explaining the classification achieved by a classifier f on an instance $\boldsymbol{x}$ consists in identifying a subset of the characteristics of $\boldsymbol{x}$ that is enough to get the class returned by f. Formally, an *abductive explanation* (Ignatiev et al., 2019c) (also called weak abductive explanation (Huang et al., 2021)) t for an instance $\boldsymbol{x} \in \boldsymbol{X}$ given a classifier f (that is binary or not) is a (conjunctively-interpreted) subset $t \subseteq t_{\boldsymbol{x}}$ such that every instance $\boldsymbol{x}' \in \boldsymbol{X}$ covered by t, i.e., satisfying $t \subseteq t_{\boldsymbol{x}'}$, is classified by f in the same way as $\boldsymbol{x}$: $f(\boldsymbol{x}') = f(\boldsymbol{x})$.[2] The size $|t|$ of an abductive explanation t is the number of characteristics in it. A *sufficient reason* t for $\boldsymbol{x} \in \boldsymbol{X}$ given f is an abductive explanation for $\boldsymbol{x}$ given f such that no proper subset t' of t is an abductive explanation for $\boldsymbol{x}$ given f. Stated otherwise, the sufficient reasons for $\boldsymbol{x}$ given f are the subset-minimal abductive explanations for $\boldsymbol{x}$ given f.

Example 2. *For our running example, $t = \{(A_1 = 4), (A_4 = 1)\}$ is a sufficient reason for $\boldsymbol{x} = (A_1 = 4, A_2 = 3, A_3 = b, A_4 = 1)$ given $BT = \{F\}$. Indeed, all the instances $\boldsymbol{x}'$ extending t can be gathered into four categories, obtained by considering the truth values of the Boolean conditions over the two remaining attributes (A_2 and A_3) as encountered in the trees of BT. In every case, we have $w(F, \boldsymbol{x}') > 0$ (see Table 1), showing that $BT(\boldsymbol{x}') = 1$. Since $BT(\boldsymbol{x}) = 1$, t is an abductive explanation for $\boldsymbol{x}$ given BT. Since no proper subset of t satisfies this property, t actually is a sufficient reason for $\boldsymbol{x}$ given BT. Similarly, we can show that $t' = \{(A_2 = 3), (A_4 = 1)\}$ is sufficient*

[2] In subsequent papers, including (Ignatiev et al., 2020), abductive explanations have been required to be minimal w.r.t. set inclusion - under this requirement, they coincide with sufficient reasons (Darwiche and Hirth, 2020), also called prime-implicant explanations (Shih et al., 2018a).

Algorithm 1: SR($\boldsymbol{x}, f$)

$t \leftarrow t_{\boldsymbol{x}}$
foreach $c_i \in t_{\boldsymbol{x}}$ **do**
  if *implicant*$(t \setminus \{c_i\}, \boldsymbol{x}, f)$ **then** $t \leftarrow t \setminus \{c_i\}$
return t

reason for $\boldsymbol{x}$ given BT.

Sufficient reasons are usually preferred to other abductive explanations since they are more simple: they do not contain any characteristics of the instance at hand that are not useful to explain the prediction made by f.

Computing Sufficient Reasons. In order to compute a sufficient reason for an input instance $\boldsymbol{x}$ given a classifier f, one can take advantage of a simple greedy algorithm (see Algorithm 1). Starting with $t = t_{\boldsymbol{x}}$, this algorithm considers all the characteristics $c_i = (A_i = v_i)$ of $\boldsymbol{x}$ in a specific order and, at each step, tests whether t deprived of c_i is still an abductive explanation for $\boldsymbol{x}$ given f. If the test is positive, c_i is removed from t, otherwise it is kept. Once all the characteristics c_i of $\boldsymbol{x}$ have been considered, the resulting term t is by construction a sufficient reason for $\boldsymbol{x}$ given f.

The computationally demanding step in this greedy algorithm is the call to function `implicant` that tests whether t deprived of c_i is still an abductive explanation for $\boldsymbol{x}$ given f, i.e., any instance covered by $t \setminus \{c_i\}$ is classified in the same way as $\boldsymbol{x}$ by f. Though this test can be achieved in polynomial time for some families of classifiers f (including decision trees) (Izza et al., 2020; Huang et al., 2021), it is intractable in general. Indeed, it is coNP-hard when f is a random forest (Audemard et al., 2022a). Similarly, when f is a boosted tree BT, we easily get that:

Proposition 1. *Let BT be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$. Let $t \subseteq t_{\boldsymbol{x}}$. Deciding whether t is an abductive explanation for $\boldsymbol{x}$ given BT is* coNP*-complete.* coNP*-hardness still holds in the restricted case every A_i ($i \in [n]$) is Boolean and BT consists of a single forest.*

In order to achieve the `implicant` test when f is a boosted tree BT, several approaches can be followed. (Ignatiev et al., 2019b) took advantage of an SMT (SAT modulo theory) encoding of the boosted tree and then on an SMT solver to compute sufficient reasons. More recently, (Ignatiev et al., 2022) pointed out a more sophisticated encoding based on MaxSAT and exploited a MaxSAT solver to compute sufficient reasons. Though this latter approach exhibited better performances in practice, its scalability is still an issue (the datasets considered in the experiments presented in (Ignatiev et al., 2022) contain at most 60 attributes).

3 COMPUTING TS-EXPLANATIONS

Worst / Best Instances. As explained before, when the classifier at hand is a regression tree, a forest, or (more generally) a boosted tree BT, the classification of an instance $\boldsymbol{x} \in \boldsymbol{X}$ depends on the weights of the tree(s) of the classifier for the instance. Because of this weight-based mechanism, the notion of abductive explanation t for $\boldsymbol{x}$ can be characterized via the notion of *worst /best instance* extending t. Let us start with the binary case:

Definition 1. *Let $BT = \{F\}$ be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$. Let $t \subseteq t_{\boldsymbol{x}}$.*

- *A* worst instance *extending t given F is an instance $\boldsymbol{x}' \in \boldsymbol{X}$ such that $t \subseteq t_{\boldsymbol{x}'}$ and $\boldsymbol{x}' = argmin_{\boldsymbol{x}'' \in \boldsymbol{X} : t \subseteq t_{\boldsymbol{x}''}}(\{w(F, \boldsymbol{x}'')\})$.*
- *A* best instance *extending t given F is an instance $\boldsymbol{x}' \in \boldsymbol{X}$ such that $t \subseteq t_{\boldsymbol{x}'}$ and $\boldsymbol{x}' = argmax_{\boldsymbol{x}'' \in \boldsymbol{X} : t \subseteq t_{\boldsymbol{x}''}}(\{w(F, \boldsymbol{x}'')\})$.*

In this definition, the condition $t \subseteq t_{\boldsymbol{x}''}$ is used to characterize the set of instances $\boldsymbol{x}''$ of interest, i.e., those covered by t. $W(t, F)$ (resp. $B(t, F)$) denotes the set of worst (resp. best) instances extending t given F, and $w_{\downarrow}(t, F)$ (resp. $w_{\uparrow}(t, F)$) denotes the weight of any worst (resp. best) instance covered by t given F.

Intuitively, a worst/best instance extending a given term t is one of minimal/maximal weight. If such a weight is positive/negative, then so is the weight of any instance covered by t. Accordingly, we have:

Proposition 2. *In the binary case, let $BT = \{F\}$ be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$. Let $t \subseteq t_{\boldsymbol{x}}$.*

- *If $BT(\boldsymbol{x}) = 1$, then t is an abductive explanation for $\boldsymbol{x}$ given BT if and only if any $\boldsymbol{x}' \in W(t, F)$ is such that $BT(\boldsymbol{x}') = 1$.*
- *If $BT(\boldsymbol{x}) = 0$, then t is an abductive explanation for $\boldsymbol{x}$ given BT if and only if any $\boldsymbol{x}' \in B(t, F)$ is such that $BT(\boldsymbol{x}') = 0$.*

Example 3. *For our running example, $t = \{(A_1 = 4), (A_4 = 1)\}$ is an abductive explanation for $\boldsymbol{x} = (A_1 = 4, A_2 = 3, A_3 = b, A_4 = 1)$ given $BT = \{F\}$ because any worst instance covered by t, i.e., any $\boldsymbol{x}'$ satisfying $(A_1 = 4) \wedge (A_2 \leq 1) \wedge (A_3 = b) \wedge (A_4 = 1)$ is such that $w(F, \boldsymbol{x}') = 0.3$ (hence $w(F, \boldsymbol{x}') > 0$) (see Table 1).*

In the multi-class case, a similar notion of worst instance can be stated:[3]

Definition 2. *Let $BT = \{F^1, \ldots, F^m\}$ be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$ such that $BT(\boldsymbol{x}) = i$. Let $t \subseteq t_{\boldsymbol{x}}$. Given BT and $\boldsymbol{x}$, a* worst instance *extending t is an instance $\boldsymbol{x}' \in \boldsymbol{X}$ such that*

[3] A notion of best instance could also be defined but it is useless for our purpose.

$A_1 = 4$	$\mathbf{A_2 > 1}$	$\mathbf{A_3 = b}$	$A_4 = 1$	$w(T_1, \boldsymbol{x}')$	$w(T_2, \boldsymbol{x}')$	$w(T_3, \boldsymbol{x}')$	$w(F, \boldsymbol{x}')$
1	**0**	**0**	1	0.4	0.3	0.2	0.9
1	**0**	**1**	1	0.4	0.3	-0.4	0.3
1	**1**	**0**	1	-0.3	0.5	0.3	0.5
1	**1**	**1**	1	0.3	0.5	0.1	0.9

Table 1: Weights of BT for instances $\boldsymbol{x}'$ extending t.

$t \subseteq t_{\boldsymbol{x}'}$ *and* $\boldsymbol{x}' = argmin_{\boldsymbol{x}'' \in \boldsymbol{X} : t \subseteq t_{\boldsymbol{x}''}}(\{w(F^i, \boldsymbol{x}'') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}'')\})$.

Then we have:

Proposition 3. *Let $BT = \{F^1, \ldots, F^m\}$ be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$ such that $BT(\boldsymbol{x}) = i$. Let $t \subseteq t_{\boldsymbol{x}}$. t is an abductive explanation for $\boldsymbol{x}$ given BT if and only if for any worst instance $\boldsymbol{x}'$ extending t given BT and $\boldsymbol{x}$, $w(F^i, \boldsymbol{x}') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}') > 0$ holds.*

Propositions 1 and 2 (or 3) show together that identifying a worst (resp. best) instance $\boldsymbol{x}' \in \boldsymbol{X}$ extending a term $t \subseteq t_{\boldsymbol{x}}$ given a boosted tree BT is intractable. Indeed, if it were not the case, we could check in polynomial time whether t is an abductive explanation for $\boldsymbol{x}$ given BT by testing whether $\boldsymbol{x}'$ is classified by BT in the same way as $\boldsymbol{x}$.

Computing Worst/Best Instances for Trees. Interestingly, when the classifier consists of a regression tree T, identifying an element of $W(t, T)$ (resp. $B(t, T)$) is easy: there exists a simple, linear-time, algorithm to compute $w_{\downarrow}(t, T)$ and $w_{\uparrow}(t, T)$, and as a by-product, to derive a worst instance and a best instance extending t given T. Basically, the algorithm consists of freezing in T every arc corresponding to a condition not satisfied by t, which can be done in time linear in the size of the input. A valid root-to-leaf path in the resulting tree is a root-to-leaf path of T not containing any frozen arc. The weight $w_{\downarrow}(t, T)$ of T for a worst (resp. best) instance extending t simply is the minimal (resp. maximal) weight labelling a leaf of a valid root-to-leaf path in the resulting tree, and it can be determined in time linear in the size of the input. Any $\boldsymbol{x}'$ satisfying the conditions associated with a valid root-to-leaf path leading to a minimal (resp. maximal) weight leaf and satisfying $t \subseteq t_{\boldsymbol{x}'}$ is a worst (resp. best) instance extending t given T.

Example 4. *Considering our running example again, let us identify worst instances extending $t = \{(A_1 = 4), (A_4 = 1)\}$ for each of the trees T_1, T_2, and T_3. On Figure 2, every frozen arc (and the corresponding subtree) is watermark displayed; the minimal weight leaves are bold, and the arcs of the corresponding root-to-leaf paths are bold. We have:*

- *Every $\boldsymbol{x}' \in \boldsymbol{X}$ satisfying $(A_1 = 4) \wedge (A_2 > 1) \wedge (A_3 \neq b) \wedge (A_4 = 1)$ is an element of $W(t, T_1)$,*
- *Every $\boldsymbol{x}' \in \boldsymbol{X}$ satisfying $(A_1 = 4) \wedge (A_2 \leq 1) \wedge (A_4 = 1)$ is an element of $W(t, T_2)$,*
- *Every $\boldsymbol{x}' \in \boldsymbol{X}$ satisfying $(A_1 = 4) \wedge (A_2 \leq 1) \wedge (A_3 = b) \wedge (A_4 = 1)$ is an element of $W(t, T_3)$.*

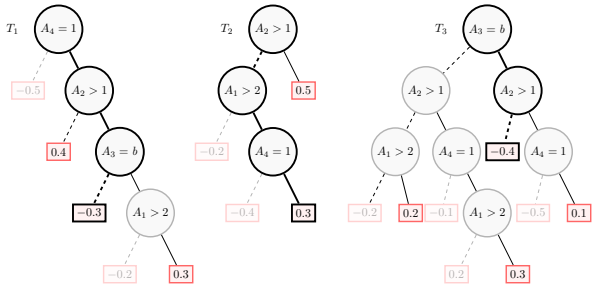


Figure 2: Worst instances and the corresponding weights for the regression trees used in BT.

Tree-Specific Explanations. We are now ready to define the notion of *TS-explanation* t for an instance $\boldsymbol{x}$ given a boosted tree BT. We start with the binary case, i.e., when BT consists of a single forest F:

Definition 3. *Let $F = \{T_1, \cdots, T_p\}$ be a forest over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$.*

- *If $F(\boldsymbol{x}) = 1$, then t is a* tree-specific explanation *for $\boldsymbol{x}$ given F if and only if t is a subset of $t_{\boldsymbol{x}}$ such that $\sum_{k=1}^{p} w_{\downarrow}(t, T_k) > 0$ and no proper subset of t satisfies the latter condition.*
- *If $F(\boldsymbol{x}) = 0$, then t is a* tree-specific explanation *for $\boldsymbol{x}$ given F if and only if t is a subset of $t_{\boldsymbol{x}}$ such that $\sum_{k=1}^{p} w_{\uparrow}(t, T_k) \leq 0$ and no proper subset of t satisfies the latter condition.*

More generally, in the multi-class setting, TS-explanations can be defined as follows:

Definition 4. *Let $BT = \{F^1, \cdots, F^m\}$ be a boosted tree over $\{A_1, \ldots, A_n\}$ where each F^j ($j \in [m]$) contains p_j trees, and $\boldsymbol{x} \in \boldsymbol{X}$ such that $BT(\boldsymbol{x}) = i$. t is a* tree-specific explanation *for $\boldsymbol{x}$ given BT if and only if t is a subset of $t_{\boldsymbol{x}}$ such that for every $j \in [m] \setminus \{i\}$, we have $\sum_{k=1}^{p_i} w_{\downarrow}(t, T_k^i) > \sum_{k=1}^{p_j} w_{\uparrow}(t, T_k^j)$, and no proper subset of t satisfies the latter condition.*

A first key property that makes TS-explanations valuable is that they are abductive explanations:

Proposition 4. *Let BT be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$. If t is a TS-explanation for $\boldsymbol{x}$ given BT, then t is an abductive explanation for $\boldsymbol{x}$ given BT.*

Accordingly, each time the test $\forall j \in [m] \setminus \{i\}, \sum_{k=1}^{p_i} w_\downarrow(t, T_k^i) > \sum_{k=1}^{p_j} w_\uparrow(t, T_k^j)$ succeeds, it is ensured that t is an abductive explanation for $\boldsymbol{x}$ given BT. However, the condition is only sufficient: when the test fails, it can be the case that t is an abductive explanation for $\boldsymbol{x}$ given BT nevertheless. Testing the condition $\forall j \in [m] \setminus \{i\}, \sum_{k=1}^{p_i} w_\downarrow(t, T_k^i, \boldsymbol{x}) > \sum_{k=1}^{p_j} w_\uparrow(t, T_k^j, \boldsymbol{x})$ thus amounts to making an *incomplete implicant test*.

It is easy to check that TS-explanations coincide with sufficient reasons for regression trees. Unsurprisingly, given the complexity shift pointed out in Proposition 1, this equivalence does not hold for forests or boosted trees. Thus, in the general case, a TS-explanation t for $\boldsymbol{x}$ given BT is not a sufficient reason for $\boldsymbol{x}$ given BT: t may contain characteristics of $\boldsymbol{x}$ that could be removed without questioning the classification achieved by BT.

Example 5. *Considering our running example again, the sufficient reason $t = \{(A_1 = 4), (A_4 = 1)\}$ for $\boldsymbol{x} = (A_1 = 4, A_2 = 3, A_3 = b, A_4 = 1)$ given $BT = \{F\}$ is not a TS-explanation for $\boldsymbol{x}$ given BT. Indeed, we have $w_\downarrow(t, T_1) = -0.3$, $w_\downarrow(t, T_2) = 0.3$, and $w_\downarrow(t, T_3) = -0.4$, hence $w_\downarrow(t', T_1) + w_\downarrow(t', T_2) + w_\downarrow(t', T_3) = -0.4 < 0$ while $w(F, \boldsymbol{x}) = 0.9 > 0$. Contrastingly, the sufficient reason $t' = \{(A_2 = 3), (A_4 = 1)\}$ for $\boldsymbol{x}$ given BT also is a TS-explanation for $\boldsymbol{x}$ given BT. We have $w_\downarrow(t', T_1) = -0.3$, $w_\downarrow(t', T_2) = 0.5$, and $w_\downarrow(t', T_3) = 0.1$, hence $w_\downarrow(t', T_1) + w_\downarrow(t', T_2) + w_\downarrow(t', T_3) = 0.3 > 0$.*

Though subset-minimality is required in both cases, the fact that TS-explanations and sufficient reasons do not coincide (in general) can be easily explained by the fact that TS-explanations consider the trees *separately*: it can be easily the case that two distinct trees T_k^j and T_l^j belonging to the same forest F^j do not share any worst instance extending a given term t. In symbols, we may have $W(t, T_k^j) \cap W(t, T_l^j) = \varnothing$.

Example 6. *For our running example, no worst instance extending $t = \{(A_1 = 4), (A_4 = 1)\}$ given T_1 is also a worst instance extending $t = \{(A_1 = 4), (A_4 = 1)\}$ given T_2 or given T_3. Indeed, every worst instance extending $t = \{(A_1 = 4), (A_4 = 1)\}$ given T_1 must satisfy $A_2 > 1$, while every worst instance extending $t = \{(A_1 = 4), (A_4 = 1)\}$ given T_2 or T_3 must satisfy the complementary condition $A_2 \leq 1$.*

In the worst case, the number of useless characteristics in a TS-explanation can be equal to the number n of attributes:

Proposition 5. *Let BT be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$. It can be the case that the unique TS-explanation for $\boldsymbol{x}$ given BT consists of $t_{\boldsymbol{x}}$ itself, while $\varnothing$ is the unique sufficient reason for $\boldsymbol{x}$ given BT. This holds even in the restricted case BT consists of a single forest and every attribute is Boolean.*

A second key property that makes TS-explanations valuable is that they can be computed efficiently. Indeed, the greedy algorithm TS given by Algorithm 2 can be used to derive in time $\mathcal{O}(n|BT|)$ a TS-explanation for $\boldsymbol{x}$ given BT in the multi-class case.

Algorithm 2: $\texttt{TS}(\boldsymbol{x}, BT)$

```
t ← t_x
j ← BT(x)
foreach c_i ∈ t_x do
    if ∄k ∈ [m] \ {j} s.t. Σ_{l=1}^{p_j} w↓(t \ {c_i}, T_l^j) ≤
      Σ_{l=1}^{p_k} w↑(t \ {c_i}, T_l^k) then
        t ← t \ {c_i}
return t
```

Proposition 6. *Let BT be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\boldsymbol{x} \in \boldsymbol{X}$. $\texttt{TS}(\boldsymbol{x}, BT)$ returns a TS-explanation for $\boldsymbol{x}$ given BT.*

Clearly enough, an algorithm closely similar to TS can be designed to handle the binary classification case (in that case, at each iteration, one just needs to test the sign of $\sum_{k=1}^{p} w_\downarrow(t, T_k)$ when $\boldsymbol{x}$ is positive, and the sign of $\sum_{k=1}^{p} w_\uparrow(t, T_k)$ when $\boldsymbol{x}$ is negative). The same performance guarantees as in the multi-class case are ensured.

Interestingly, when dealing with boosted trees, the greedy algorithm SR (Algorithm 1) for deriving sufficient reasons can be exploited to remove useless characteristics in TS-explanations, i.e., to generate sufficient reasons from TS-explanations. Viewed from a different angle, the computation of a TS-explanation for an instance $\boldsymbol{x}$ given a boosted tree BT can be exploited as a *preprocessing* step in SR. This combination is given by the *pipeline*

$$\texttt{SR}(\texttt{TS}(t_{\boldsymbol{x}}, BT), BT).$$

The rationale for this preprocessing step is the fact that TS is a polynomial-time algorithm, while implicant is not. As the experiments reported in Section 4 will show it, TS may remove in a very efficient way many useless characteristics of $\boldsymbol{x}$, thus avoiding many calls to the computationally expensive function implicant.

4 EXPERIMENTS

Empirical Protocol. The empirical protocol was as follows. We have considered 50 datasets, which are standard benchmarks (adult, farm-ads, ...) coming from the well-known repositories Kaggle (www.kaggle.com), OpenML (www.openml.org), and UCI (archive.ics.uci.edu/ml/). For these datasets, the number of classes varies from 2 to 9 classes, the number of attributes (features) from 10 to 100001, and the number of instances from 345 to 48842. Categorical features have been treated as numbers. This

choice has been motivated by the fact that most of the time the types of the attributes are not documented in the datasets. As to numerical features, no data preprocessing has taken place: these features have been binarized on-the-fly by the learning algorithm that has been used, namely XGBoost (Chen and Guestrin, 2016) that learns gradient boosted trees. XGBoost has been used without any tuning. Since our purpose is to be able to explain the classification achieved by the boosted trees as they have been learned, hyper-parameters have not been optimized but set to their default values (in particular, 100 trees per class have been considered and the maximum depth of each tree was set to 6).

For every dataset, a 10-fold cross validation process has been achieved. Ten boosted trees have been learned per dataset. The mean accuracy per dataset varies from 53.23% up to 100% (in average, it is equal to 88.5%). Ten instances have been picked up uniformly at random in the test set associated with the training set used to learn each boosted tree. This led to 100 instances per dataset, giving a total of 5000 instances for which explanations about the way they were classified have been sought for. To get such explanations, we ran implementations of the algorithms presented in the previous sections: `SR` implemented as `XReason` (with its default parameters), our own implementation of `TS`, and an implementation `TS+XReason` of the pipeline of the two. In order to implement this pipeline, we had to modify `XReason` in such a way that it can use as input any abductive explanation for the instance at hand, and not only the instance itself. By default, `XReason` starts by removing some useless characteristics of the input instance using a core-guided mechanism. Though beneficial when `XReason` is used alone, this step turns out to be counterproductive when `XReason` is combined with `TS`. Indeed, in our experiments, the number of instances (out of 5000) "solved" by `TS+XReason` with this treatment switched on is 4016, while it is equal to 4097 when the treatment was switched off. Hence, in our experiments, the treatment has been switched off when `TS` was run upstream to `XReason`, and switched on when `XReason` was used alone. We have also modified `XReason` to make it provide the abductive explanation that is available when the time limit is reached (in this case, the returned explanation is not guaranteed to be subset-minimal). In our experiments, for each instance $\boldsymbol{x}$, `TS` has been called 1000 times: at each run, an elimination ordering of the characteristics of $\boldsymbol{x}$ (as considered at line 3. of Algorithm 2) has been picked up uniformly at random, and a shortest TS-explanation among those generated for the 1000 runs was finally returned.

All the experiments have been conducted on a computer equipped with Intel(R) XEON E5-2637 CPU @ 3.5 GHz and 128 Gib of memory. For each algorithm, a timeout (TO) of 100 seconds per instance has been considered. This was deemed as a reasonable time bound for providing an explanation to a human user.

Dataset	#Cls.	#Feat.	#Feat. Used	#Inst.	Acc.
gina_agnostic	2	970	477.90(±10.46)	3468	95.13(±1.35)
malware	2	1084	88.60(±1.85)	6248	99.46(±0.26)
ad_data	2	1558	78.70(±4.24)	3279	97.78(±0.83)
christine	2	1636	1272.20(±8.06)	5418	73.81(±1.82)
cnae	9	856	132.30(±4.24)	1079	91.48(±2.64)
gisette	2	5000	783.60(±14.33)	7000	97.83(±0.56)
arcene	2	10000	143.50(±5.90)	200	80.50(±7.23)
dexter	2	20000	104.50(±4.54)	600	91.83(±3.69)
allBooks	8	8266	315.90(±8.70)	590	87.12(±3.73)
farm-ads	2	54876	714.10(±16.70)	4143	90.27(±1.45)
dorothea	2	100000	222.20(±10.85)	1150	93.91(±2.55)

Table 2: A focus on 10 datasets.

Results. A synthesis of the results we obtained is provided on Figure 3. On those two scatter plots, each dot corresponds to an instance among the 5000 instances tested.

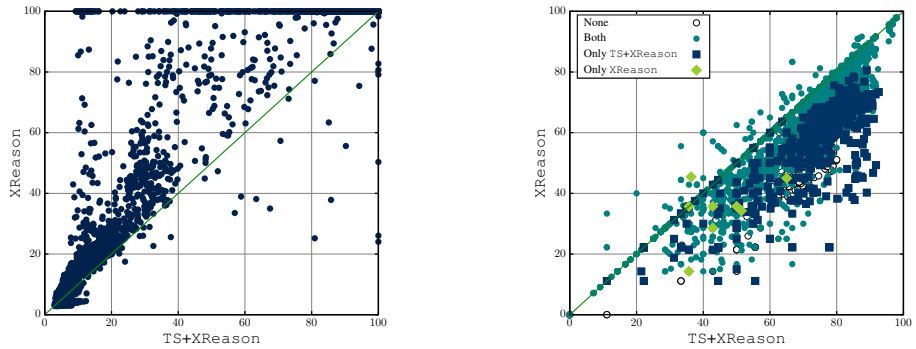
Figure 3 (a) is about computation times. The x-coordinate (resp. y-coordinate) of a dot is the time (in seconds) required by `TS+XReason` (resp. `XReason`) to compute an abductive explanation for the associated instance. By construction, this abductive explanation is a sufficient reason for the instance when the computation stops before the time limit.

In light of Figure 3 (a), two observations can be made. On the one hand, the run times of `TS+XReason` are significantly smaller than those of `XReason`. On the other hand, the time limit has been reached much more often by `XReason` than by `TS+XReason`.

Figure 3 (b) is about the size of the abductive explanations that are generated, and more precisely, about reduction rates, where the reduction rate achieved by an explanation t for an instance $\boldsymbol{x}$ over n attributes is given by $1 - \frac{|t|}{n-d}$ where d is the number of attributes that are dropped by XGBoost. For example, if $|t| = 10$, $n = 100$, and $d = 20$ (meaning that only $n - d = 80$ attributes are used in the boosted tree), the reduction rate is 87.5%. Indeed, size is one of the criteria to be considered when evaluating the intelligibility[4] of an explanation: everything else being equal, shortest explanations are easier to understand than longer explanations. For each dot corresponding to an instance $\boldsymbol{x}$, the x-coordinate (resp. y-coordinate) of the dot is the reduction rate of the explanation for $\boldsymbol{x}$ generated by `TS+XReason` (resp. by `XReason`). Different dot representations for instances have been used in this figure, depending on the fact that a sufficient reason for the instance at hand has been computed (or not) within the time limit by any of the two programs, or by both of them.

Figure 3 (b) shows that `TS+XReason` leads in general to much better reduction rates than those obtained by `XReason`, thus to much smaller explanations. One can

[4] In general, the intelligibility of an explanation does not reduce to its size and an accurate evaluation of it cannot be achieved in a context-independent way (Doshi-Velez and Kim, 2017; Narayanan et al., 2018), since intelligibility typically depends on the explainee (i.e., the person who asked for an explanation) (Miller, 2019).



(a) Run times (in seconds)

(b) Reduction rates

Figure 3: Comparing TS+XReason to XReason.

Dataset	Inst.	Run time						Reduction rate		
		TS+XReason				XReason			TS+XReason		XReason
		TS	TS+XReason	#TO	#SUF.	XReason	#TO	#SUF.	TS	TS+XReason	
gina_agnostic	SUF.	1.86(±0.18)	91.87(±6.18)		34				75.48(±1.48)	77.54(±1.50)	
	TO	1.77(±0.15)	100(±0.20)	66		100(±0.01)	100		74.21(±1.81)	76.38(±2.08)	62.78(±2.45)
malware	SUF.	0.22(±0.02)	6.01(±1.22)		100	8.34(±0.81)		100	84.63(±5.54)	85.90(±4.84)	83.40(±5.62)
ad_data	SUF.	0.32(±0.03)	14.2(±9.49)		100	46.03(±20.65)		53	79.27(±8.89)	83.97(±8.30)	72.49(±9.90)
	TO					100(±0.01)	47				57.78(±11.92)
christine	TO	6.13(±0.40)	100(±0.32)	100		100(±0.01)	100		77.29(±0.78)	77.39(±0.81)	51.70(±1.64)
cnae	SUF.	5.68(±0.60)	53.82(±21.71)		97	77.96(±13.29)		50	62.88(±9.95)	67.61(±10.38)	61.11(±11.01)
	TO	4.72(±0.09)	100(±0.10)	3		100(±0.05)	50		41.85(±9.09)	46.70(±9.27)	43.16(±8.84)
gisette	SUF.	2.91(±0.22)	91.49(±6.81)		77				81.34(±1.63)	82.51(±1.61)	
	TO	3.10(±0.18)	100(±0.33)	23		100(±0.02)	100		80.47(±1.1)	81.78(±1.16)	71.02(±2.29)
arcene	SUF.	0.20(±0.02)	5.89(±1.88)		100	4.49(±0.31)		100	73.71(±5.17)	74.41(±5.09)	71.36(±5.50)
dexter	SUF.	0.30(±0.04)	7.05(±1.23)		100	11.25(±1.32)		100	77.31(±7.63)	80.05(±7.05)	70.68(±5.70)
allBooks	SUF.	7.60(±0.97)	57.33(±14.91)		97				77.01(±7.23)	79.98(±7.34)	
	TO	8.11(±1.24)	100(±0.34)	3		100(±0.03)	100		69.52(±3.17)	73.41(±3.67)	61.70(±7.32)
farm_ads	SUF.	2.89(±0.21)	99.26(±1.01)		7				72.29(±1.44)	73.57(±1.40)	
	TO	2.86(±0.19)	100(±0.25)	93		100(±0.03)	100		70.47(±1.58)	72.27(±1.60)	65.37(±1.89)
dorothea	SUF.	0.68(±0.06)	12.17(±1.28)		100	17.28(±2.53)		100	59.48(±6.21)	62.12(±6.02)	52.84(±7.22)

Table 3: Performances of TS+XReason and XReason in terms of run times and reduction rates on 10 datasets.

observe that the number of sufficient reasons that have been (provably) derived by TS+XReason in at most 100 seconds is significantly higher than the number of sufficient reasons that have been (provably) derived by XReason. More in detail, out of 5000 abductive explanations, 3476 sufficient reasons have been obtained in due time by the two programs, while 621 have been obtained in due time by TS+XReason alone, 8 have been obtained in due time by XReason alone. This gives a win rate of more than 98% for the pipeline. Overall, a significant amount of $621 - 8 = 613$ sufficient reasons have been gained by taking advantage of TS as a preprocessing to XReason. For 895 abductive explanations that have been generated, there are no subset-minimality guarantees whatever TS+XReason or XReason was used to derive them.

In order to evaluate the impact of picking a shortest TS-explanation out of 1000 instead of using only the one that has been computed first, we have considered a variant of TS that derives only one TS-explanation. We observed that the performances of the pipelines TS+XReason are similar in the two cases. Thus, when a single TS-explanation is considered, the reduction rate decreases a bit on average (-4.25%), leading to slightly longer explanations. On the contrary, computation time decreases a bit on average (-7.65%). Focusing on the harder instances (those for which the pipeline required more than $10s$ to terminate), the reduction rate decreases a bit more on average (-5.14%) but the computation time increases slightly on average ($+5.04\%$). Hence the ability to compute many TS-explanations almost "for free" and to keep a smallest one is valuable but not the main cause for the efficiency of the pipeline.

To complete the scatter plots, Tables 2 and 3 report some details about 10 datasets out of 50 (the ones based on the largest numbers of attributes). The columns of Table 2 give, from left to right, the name of the dataset, the number of classes and features in it, the mean number of features used in the boosted trees that have been generated, the number of instances in the dataset, the mean accuracy of the boosted trees. For each dataset, Table 3 presents the results obtained on 100 instances by TS+XReason and XReason in terms

of mean run times (given in seconds) number of TOs and number of sufficient reasons computed in due time, and then in terms of mean reduction rates. For TS+XReason, we report the mean run times and mean reduction rates achieved by TS alone, and then by the pipeline TS+XReason where XReason is run on the abductive explanation generated by TS. The 100 instances tested per dataset are divided into two subsets, each one corresponding to a line: those instances for which a TO occurs (in that case, the abductive explanation that is generated may be redundant) and those for which a sufficient reason has been computed within the time limit. Whenever every instance or no instance out of 100 has been solved by the pipeline TS+XReason and by XReason alone before the time limit has been reached, only one line has been kept.

For some instances, it can be observed that the pipeline TS+XReason does not succeed in computing a sufficient reason in due time (this happens e.g., for every instance of the dataset 'christine', see Table 3). Even if XReason does not do better for this dataset, this can be considered as a limitation of the approach. It is also worth noting that when computing a sufficient reason, deriving first a TS-explanation is sometimes a waste of time. In our experiments, this is reflected by the 8 instances for which XReason alone succeeded to compute a sufficient reason in due time, while TS+XReason failed.

However, the advantages offered by the pipeline are significant in practice most of the time (the two scatter plots and the table show clear benefits in terms of computation time and explanation size). Thus, TS+XReason appears as a better algorithm than XReason for the purpose of computing sufficient reasons. This is particularly salient in the multi-class case (both in terms of computation times and sufficient reasons found). TS also appears as a valuable algorithm for generating abductive explanations because of its efficiency (the cumulated run times over 1000 runs per instance are bounded by a few seconds). The main disadvantage of considering TS-explanations instead of sufficient reasons is that TS-explanations may contain redundant characteristics (so shorter reasons may exist in the worst case). However, our experiments show that the reduction rates achieved by TS are, in practice, close to those achieved by TS+XReason, and often significantly higher than those achieved by XReason,[5]. This holds both for instances for which a sufficient reason has been derived in due time and for instances for which the explanation that is derived may be redundant. This discrepancy can be explained by the fact that the implicant test performed by the TS algorithm is incomplete. As a consequence, a characteristics of the input instance can be kept while it would be removed if XReason was used instead. Removing such a characteristics during one of the first iterations of the greedy algorithm may have a strong impact on the search space and may prevent from removing many other characteristics later.

[5]Please keep in mind that an instance may have exponentially TS-explanations / sufficient reasons and that the two algorithms we consider compute only one of them.

5 OTHER RELATED WORK

Because of their intrinsic interpretability, trees have been considered in ML and data mining for a long time and for various purposes (including classification and regression, but also reinforcement learning (Silva et al., 2020)). Trees are first-class components (weak learners) of ensemble methods. Trees have been used in various ways to resolve the tension between generalization and interpretability of black box models, for instance by mimicking the input-output functions discovered by deep neural networks (Frosst and Hinton, 2017), or by designing deep models that are more interpretable (Wu et al., 2018, 2020).

From a formal XAI perspective, decision trees have been investigated in depth (Audemard et al., 2022c; Izza et al., 2022). Decision trees support in polynomial time a number of XAI queries that are intractable for other ML models, including random forests and boosted trees (Audemard et al., 2021). Among those queries is the generation of a sufficient reason for a given instance, that is the query we focused on in this paper.

6 CONCLUSION

We have introduced a new notion of abductive explanation for boosted trees, called tree-specific (TS) explanations. In the worst case, TS-explanations can be arbitrarily larger than sufficient reasons. However, unlike sufficient reasons, their generation is tractable. We have presented a polynomial-time algorithm TS for computing TS-explanations, and proved its correctness. Because a sufficient reason can be extracted from a TS-explanation, TS can be used as a preprocessing step for greedy algorithms deriving sufficient reasons. Empirically, we have shown that TS+XReason significantly improves the state-of-the-art. Finally, in practice, the abductive explanations computed by TS are often close to sufficient reasons. This shows that TS is also useful alone, as an efficient generator of provably sound, valuable abductive explanations.

Various perspectives for extending this work can be envisioned. Thus, instead of considering the characteristics of the input instance randomly (line 3 of TS), it would make sense to design heuristics for making more informed choices, exploiting the derivation process user preferences about the characteristics. This can be a way to derive *preferred* explanations (Audemard et al., 2022b), with the user in the loop. From a practical side, it would be useful to implement algorithms for deriving preferred explanations. We plan to add such algorithms to PyXAI (www.cril.fr/pyxai/), the Python library for XAI we develop.

Acknowledgements

Many thanks to the anonymous reviewers for their numerous comments and suggestions that helped to improve the paper. This work has benefited from the support of the AI Chair EXPEKCTATION (ANR-19-CHIA-0005-01) of the French National Research Agency. It was also partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

A. Barredo Arrieta, N. Díaz R., J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58:82–115, 2020.

G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, and P. Marquis. On the computational intelligibility of boolean classifiers. In *Proc. of KR'21*, pages 74–86, 2021. doi: 10.24963/kr.2021/8.

G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, and P. Marquis. Trading complexity for sparsity in random forest explanations. In *Proc. of AAAI'22*, pages 5461–5469, 2022a.

G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, and P. Marquis. On preferred abductive explanations for decision trees and random forests. In *Proc. of IJCAI'22*, pages 643–650, 2022b.

G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, and P. Marquis. On the explanatory power of Boolean decision trees. *Data Knowl. Eng.*, 142:102088, 2022c.

P. Barceló, M. Monet, J. Pérez, and B. Subercaseaux. Model interpretability through the lens of computational complexity. In *Proc. of NeurIPS'20*, 2020.

V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci. Deep neural networks and tabular data: A survey. *CoRR*, abs/2110.01889, 2021.

L. Breiman. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

R. Caruana, S. M. Lundberg, M. Túlio Ribeiro, H. Nori, and S. Jenkins. Intelligible and explainable machine learning: Best practices and practical challenges. In *Proc. of KDD'20*, pages 3511–3512. ACM, 2020.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proc. of KDD'16*, page 785–794, 2016.

A. Choi, A. Shih, A. Goyanka, and A. Darwiche. On symbolically encoding the behavior of random forests. In *Proc. of FoMLAS'20, 3rd Workshop on Formal Methods for ML-Enabled Autonomous Systems, Workshop at CAV'20*, 2020.

A. Darwiche and A. Hirth. On the reasons behind decisions. In *Proc. of ECAI'20*, pages 712–720, 2020.

F. Doshi-Velez and B. Kim. A roadmap for a rigorous science of interpretability. *CoRR*, abs/1702.08608, 2017. URL http://arxiv.org/abs/1702.08608.

Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

J. H. Friedman. Greedy function approximation: A gradient boosted machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

N. Frosst and G. E. Hinton. Distilling a neural network into a soft decision tree. In *Proc. of 1st International Workshop on Comprehensibility and Explanation in AI and ML*, volume 2071 of *CEUR Workshop Proceedings*, 2017.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):93:1–93:42, 2019.

S. Hooker, D. Erhan, P-J. Kindermans, and B. Kim. A benchmark for interpretability methods in deep neural networks. In *Proc. of NeurIPS'19*, pages 9737–9748, 2019.

X. Huang, Y. Izza, A. Ignatiev, and J. Marques-Silva. On efficiently explaining graph-based classifiers. In *Proc. of KR'21*, pages 356–367, 2021.

J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decis. Support Syst.*, 51(1):141–154, 2011.

A. Ignatiev. Towards trustable explainable AI. In *Proc. of IJCAI'20*, pages 5154–5158, 2020.

A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-based explanations for machine learning models. In *Proc. of AAAI'19*, pages 1511–1519, 2019a.

A. Ignatiev, N. Narodytska, and J. Marques-Silva. On validating, repairing and refining heuristic ML explanations. *CoRR*, abs/1907.02509, 2019b. URL http://arxiv.org/abs/1907.02509.

A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-based explanations for machine learning models. In *Proc. of AAAI'19*, pages 1511–1519, 2019c.

A. Ignatiev, N. Narodytska, N. Asher, and J. Marques-Silva. On relating 'why?' and 'why not?' explanations. *CoRR*, abs/2012.11067, 2020.

A. Ignatiev, Y. Izza, P.J. Stuckey, and J. Marques-Silva. Using MaxSAT for efficient explanations of tree ensembles. In *Proc. of AAAI'22*, pages 3776–3785, 2022.

Y. Izza and J. Marques-Silva. On explaining random forests with SAT. In *Proc. of IJCAI'21*, pages 2584–2591, 2021.

Y. Izza, A. Ignatiev, and J. Marques-Silva. On explaining decision trees. *CoRR*, abs/2010.11034, 2020.

Y. Izza, A. Ignatiev, and J. Marques-Silva. On tackling explanation redundancy in decision trees. *J. Artif. Intell. Res.*, 75:261–321, 2022.

G. Ke, Q. Meng, Th. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Proc. of NeurIPS'17*, pages 3146–3154, 2017.

B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). In *Proc. of ICML'18*, pages 2668–2677, 2018.

S. Lundberg and S-I. Lee. A unified approach to interpreting model predictions. In *Proc. of NIPS'17*, pages 4765–4774, 2017.

S. M. Lundberg, G. G. Erion, H. Chen, A. J. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.I. Lee. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.*, 2(1): 56–67, 2020.

J. Marques-Silva and A. Ignatiev. Delivering trustworthy AI through formal XAI. In *Proc. of AAAI'22*, pages 12342–12350, 2022.

T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

Ch. Molnar. *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. Leanpub, 2019.

M. Narayanan, E. Chen, J. He, B. Kim, S. Gershman, and F. Doshi-Velez. How do humans understand explanations from machine learning systems? an evaluation of the human-interpretability of explanation. *CoRR*, abs/1802.00682, 2018. URL http://arxiv.org/abs/1802.00682.

N. Narodytska, S. Prasad Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh. Verifying properties of binarized deep neural networks. In *Proc. of AAAI'18*, pages 6615–6624, 2018.

A. Parmentier and T. Vidal. Optimal counterfactual explanations in tree ensembles. In *Proc. of ICML'21*, volume 139 of *Proceedings of Machine Learning Research*, pages 8422–8431, 2021.

L. O. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: unbiased boosting with categorical features. In *Proc. of NeurIPS'18*, pages 6639–6649, 2018.

J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

M. T. Ribeiro, S. Singh, and C. Guestrin. "Why should I trust you?": Explaining the predictions of any classifier. In *Proc. of KDD'16*, pages 1135–1144, 2016.

M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proc. of AAAI'18*, pages 1527–1535, 2018.

C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *CoRR*, abs/2103.11251, 2021.

R.E. Schapire and Y. Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2014.

A. Shih, A. Choi, and A. Darwiche. A symbolic approach to explaining bayesian network classifiers. In *Proc. of IJCAI'18*, pages 5103–5111, 2018a.

A. Shih, A. Choi, and A. Darwiche. Formal verification of Bayesian network classifiers. In *Proc. of PGM'18*, pages 427–438, 2018b.

A. Shih, A. Choi, and A. Darwiche. Compiling Bayesian networks into decision graphs. In *Proc. of AAAI'19*, pages 7966–7974, 2019a.

A. Shih, A. Darwiche, and A. Choi. Verifying binarized neural networks by Angluin-style learning. In *Proc. of SAT'19*, pages 354–370, 2019b.

A. Silva, M. C. Gombolay, T. W. Killian, I. Dario Jimenez Jimenez, and S.-H. Son. Optimization methods for interpretable differentiable decision trees applied to reinforcement learning. In *Proc. of AISTATS'20*, pages 1855–1865, 2020.

F. Wang and C. Rudin. Falling rule lists. In *Proc. of AISTATS'15*, 2015.

M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez. Beyond sparsity: Tree regularization of deep models for interpretability. In *Proc. of AAAI'18*, pages 1670–1678, 2018.

M. Wu, S. Parbhoo, M. C. Hughes, R. Kindle, L. A. Celi, M. Zazzi, V. Roth, and F. Doshi-Velez. Regional tree regularization for interpretability in deep neural networks. In *Proc. of AAAI'20*, pages 6413–6421, 2020.

A MISSING PROOFS

Proof of Proposition 1

Proof.

- Membership to coNP: we consider the complementary problem and show that it belongs to NP. In order to determine whether t is not an abductive explanation for $\boldsymbol{x}$ given BT, it is enough to guess an instance $\boldsymbol{x}' \in \boldsymbol{X}$ such that $t \subseteq t_{\boldsymbol{x}'}$ and to check that $BT(\boldsymbol{x}') \neq BT(\boldsymbol{x})$. Since the class associated by BT to any input instance can be computed in time polynomial in the size of BT and the size of the instance, the conclusion follows.
- coNP-hardness: it has been shown in (Audemard et al., 2022a) (Proposition 3) that deciding whether t is an abductive explanation for $\boldsymbol{x}$ given a random forest RF over Boolean attributes is coNP-complete. Thus, it is enough to show that we can associate in polynomial time any random forest $RF = \{T_1, \ldots, T_p\}$ over Boolean attributes $A_1, \ldots, A_n$ to a boosted tree $BT = \{F\}$ with $F = \{T'_1, \ldots, T'_p\}$ such that for any $\boldsymbol{x} \in \boldsymbol{X}$, we have $RF(\boldsymbol{x}) = 1$ if and only if $BT(\boldsymbol{x}) = 1$.

  The reduction is easy: each T'_i ($i \in [p]$) is obtained in linear time from T_i by replacing every 0-leaf (resp. 1-leaf) of T_i by a leaf labelled by $-w$ (resp. w) where w is a (fixed) positive number (e.g., $w = 0.5$). By construction, we have $RF(\boldsymbol{x}) = 1$ if and only if $\sum_{j=1}^{p} T_i(\boldsymbol{x}) > \frac{p}{2}$ if and only if $\sum_{j=1}^{p} T'_i(\boldsymbol{x}) > 0$ if and only if $BT(\boldsymbol{x}) = 1$.

□

Proof of Proposition 2

Proof. Suppose that $BT(\boldsymbol{x}) = 1$, i.e., $w(F, \boldsymbol{x}) > 0$. By definition, t is an abductive explanation for $\boldsymbol{x}$ given BT if and only if any $\boldsymbol{x}' \in \boldsymbol{X}$ such that $t \subseteq t_{\boldsymbol{x}'}$ satisfies $BT(\boldsymbol{x}') = 1$. Since any $\boldsymbol{x}'' \in W(t, F)$ satisfies $t \subseteq t_{\boldsymbol{x}''}$, we must have $BT(\boldsymbol{x}'') = 1$. Conversely, suppose that for any $\boldsymbol{x}'' \in W(t, F)$ we have $BT(\boldsymbol{x}'') = 1$. Then we have $w(F, \boldsymbol{x}'') > 0$. By definition of $W(t, F)$, for any $\boldsymbol{x}' \in \boldsymbol{X}$ such that $t \subseteq t_{\boldsymbol{x}'}$, we have $w(F, \boldsymbol{x}') \geq w(F, \boldsymbol{x}'')$. Since $BT(\boldsymbol{x}'') = 1$, we have $w(F, \boldsymbol{x}'') > 0$, hence by transitivity of $>$, we get that $w(F, \boldsymbol{x}') > 0$, or equivalently that $BT(\boldsymbol{x}') = 1$.

Similarly, consider the case when $BT(\boldsymbol{x}) = 0$, i.e., $w(F, \boldsymbol{x}) \leq 0$. By definition, t is an abductive explanation for $\boldsymbol{x}$ given BT if and only if any $\boldsymbol{x}' \in \boldsymbol{X}$ such that $t \subseteq t_{\boldsymbol{x}'}$ satisfies $BT(\boldsymbol{x}') = 0$. Since any $\boldsymbol{x}'' \in B(t, F)$ satisfies $t \subseteq t_{\boldsymbol{x}''}$, we must have $BT(\boldsymbol{x}'') = 0$. Conversely, suppose that for any $\boldsymbol{x}'' \in B(t, F)$ we have $BT(\boldsymbol{x}'') = 0$. Then we have $w(F, \boldsymbol{x}'') \leq 0$. By definition of $B(t, F)$, for any $\boldsymbol{x}' \in \boldsymbol{X}$ such that $t \subseteq t_{\boldsymbol{x}'}$, we have $w(F, \boldsymbol{x}') \leq w(F, \boldsymbol{x}'')$. Since $BT(\boldsymbol{x}'') = 0$, we have $w(F, \boldsymbol{x}'') \leq 0$, hence by transitivity of $\leq$, we get that $w(F, \boldsymbol{x}') \leq 0$, or equivalently that $BT(\boldsymbol{x}') = 0$. □

Proof of Proposition 3

Proof. If t is an abductive explanation for $\boldsymbol{x}$ given BT , then for every $\boldsymbol{x}'$ extending t we must have $BT(\boldsymbol{x}') = i$, that is $w(F^i, \boldsymbol{x}') > w(F^j, \boldsymbol{x}')$ for every $j \in [m] \setminus \{i\}$. This is equivalent to state that $w(F^i, \boldsymbol{x}') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}') > 0$. Since any worst instance $\boldsymbol{x}'$ extending t given BT and $\boldsymbol{x}$ is an instance that extends t, we have

$$w(F^i, \boldsymbol{x}') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}') > 0,$$

as expected.

Conversely, suppose that for any worst instance $\boldsymbol{x}'$ extending t given BT and $\boldsymbol{x}$, we have $w(F^i, \boldsymbol{x}') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}') > 0$. By definition, if $\boldsymbol{x}'$ is a worst instance extending t given BT and $\boldsymbol{x}$, then for any $\boldsymbol{x}'' \in \boldsymbol{X}$ that extends t, we have

$$w(F^i, \boldsymbol{x}') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}') \leq w(F^i, \boldsymbol{x}'') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}'').$$

Hence, if $w(F^i, \boldsymbol{x}') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}') > 0$, we also have that

$$w(F^i, \boldsymbol{x}'') - max_{j \in [m] \setminus \{i\}} w(F^j, \boldsymbol{x}'') > 0,$$

showing that $BT(\boldsymbol{x}'') = i$. □

Proof of Proposition 4

Proof. Towards a contradiction, suppose that $BT(\boldsymbol{x}) = i \in [m]$ and there exists an instance $\boldsymbol{x}'$ extending t and such that $BT(\boldsymbol{x}') = j \in [m]$ with $j \neq i$. This implies that $w(F^j, \boldsymbol{x}') > w(F^k, \boldsymbol{x}')$ for every $k \in [m] \setminus \{j\}$. So, for $k = i$, we have $w(F^j, \boldsymbol{x}') > w(F^i, \boldsymbol{x}')$.

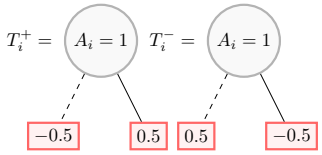
Since t is a tree-specific explanation for $\boldsymbol{x}$ given BT, t is a subset of $t_{\boldsymbol{x}}$ such that for every $k \in [m] \setminus \{i\}$, we have $\sum_{l=1}^{p_i} w_{\downarrow}(t, T_l^i) > \sum_{l=1}^{p_k} w_{\uparrow}(t, T_l^k)$. In particular, for $k = j$, we have $\sum_{l=1}^{p_i} w_{\downarrow}(t, T_l^i) > \sum_{l=1}^{p_j} w_{\uparrow}(t, T_l^j)$.

However, by definition of the utmost instances, for every $\boldsymbol{x}'$ extending t, we have $w(T_l^i, \boldsymbol{x}') \geq w_{\downarrow}(t, T_l^i)$ for every $T_l^i \in F^i$ and $w(T_l^k, \boldsymbol{x}') \leq w_{\uparrow}(t, T_l^k)$ for every $T_l^k \in F^k$ with $k \in [m] \setminus \{i\}$. In particular, we have $w(T_l^j, \boldsymbol{x}') \leq w_{\uparrow}(t, T_l^j)$ for every $T_l^j \in F^j$.

Finally, we get that $w(F^j, \boldsymbol{x}') = \sum_{l=1}^{p_j} w(T_l^j, \boldsymbol{x}') \leq \sum_{l=1}^{p_j} w_{\uparrow}(t, T_l^j) < \sum_{l=1}^{p_i} w_{\downarrow}(t, T_l^i) \leq \sum_{l=1}^{p_i} w(T_l^i, \boldsymbol{x}') = w(F^i, \boldsymbol{x}')$. A contradiction. □

Proof of Proposition 5

Proof. Consider $BT = \{F\}$ with $F = \{T_i^+, T_i^- : i \in [n]\}$ where for each $i \in [n]$,



Consider the instance $\boldsymbol{x} = (0, \ldots, 0)$. We have $w(F, \boldsymbol{x}) = 0$, hence $F(\boldsymbol{x}) = 0$. Consider any $i \in [n]$, let $\overline{(A_i = 1)} \in t_{\boldsymbol{x}}$ and $t = t_{\boldsymbol{x}} \setminus \{\overline{A_i = 1}\}$. For every $j \in [n] \setminus \{i\}$, we have $B(t, T_j^+) = B(t, T_j^-) = \{\boldsymbol{x}\}$. We also have $B(t, T_i^-) = \{\boldsymbol{x}\}$. Furthermore, $B(t, T_i^+) = \{\boldsymbol{x}'\}$ where $\boldsymbol{x}'$ is the instance that coincides with $\boldsymbol{x}$, except that $(A_i = 1) \in t_{\boldsymbol{x}'}$. Accordingly, $\sum_{j=1}^{n}(w_{\uparrow}(t, T_j^+) + w_{\uparrow}(t, T_j^-)) = 1 > 0$, showing that t is not a tree-specific explanation for $\boldsymbol{x}$ given F. Since the weights of the utmost instances extending a term t given $BT = \{F\}$ varies monotonically when t is deprived of some of its elements and since $\sum_{j=1}^{n}(w_{\uparrow}(t_{\boldsymbol{x}}, T_j^+) + w_{\uparrow}(t_{\boldsymbol{x}}, T_j^-)) = 0$, we can conclude that $t_{\boldsymbol{x}}$ is the unique tree-specific explanation for $\boldsymbol{x}$ given F. Contrastingly, since for every $\boldsymbol{x}' \in \boldsymbol{X}$, we have $F(\boldsymbol{x}') = 0$, $\varnothing$ is the (unique) sufficient reason for $\boldsymbol{x}$ given F. □

Proof of Proposition 6

Proof. The proof consists of two points. First, we check that for every $j \in [m] \setminus \{i\}$ (where $BT(\boldsymbol{x}) = i$), we have $\sum_{k=1}^{p_i} w_{\downarrow}(t_{\boldsymbol{x}}, T_k^i) > \sum_{k=1}^{p_j} w_{\uparrow}(t_{\boldsymbol{x}}, T_k^j)$ holds. Since $\boldsymbol{x}$ is the unique instance that extends $t_{\boldsymbol{x}}$, for any tree T_k^l of BT, $\boldsymbol{x}$ is also a worst and a best instance extending $t_{\boldsymbol{x}}$ given T_k^l. Thus, for each $k \in [p_i]$, we have $w_{\downarrow}(t_{\boldsymbol{x}}, T_k^i) = w(T_k^i, \boldsymbol{x})$ and for each $k \in [p_j]$, we have $w_{\uparrow}(t_{\boldsymbol{x}}, T_k^j) = w(T_k^j, \boldsymbol{x})$. Acccordingly, $\sum_{k=1}^{p_i} w_{\downarrow}(t_{\boldsymbol{x}}, T_k^i) > \sum_{k=1}^{p_j} w_{\uparrow}(t_{\boldsymbol{x}}, T_k^j)$ is equivalent to $\sum_{k=1}^{p_i} w(T_k^i, \boldsymbol{x}) > \sum_{k=1}^{p_j} w(T_k^j, \boldsymbol{x})$, which is equivalent to $w(F^i, \boldsymbol{x}) > w(F^j, \boldsymbol{x})$ and finally to $BT(\boldsymbol{x}) = i$, which holds.

The second point consists in verifying that if t, t' verify $t \subset t' \subseteq t_{\boldsymbol{x}}$, and for every $j \in [m] \setminus \{i\}$, we have $\sum_{k=1}^{p_i} w_{\downarrow}(t', T_k^i) \leq \sum_{k=1}^{p_j} w_{\uparrow}(t', T_k^j)$ holds, then $\sum_{k=1}^{p_i} w_{\downarrow}(t, T_k^i) \leq \sum_{k=1}^{p_j} w_{\uparrow}(t, T_k^j)$ holds as well. This comes directly from the fact that when $t \subset t'$, we have $w_{\downarrow}(t, T_k^i) \leq w_{\downarrow}(t', T_k^i)$ for each $k \in [p_i]$ and we have $w_{\uparrow}(t, T_k^j) \geq w_{\uparrow}(t', T_k^j)$ for each $k \in [p_j]$. □