Computing Abductive Explanations for Boosted Trees

Gilles Audemard¹

Jean-Marie Lagniez¹

Pierre Marquis^{1,2}

Nicolas Szczepanski¹

¹CRIL, Université d'Artois & CNRS, France ²Institut Universitaire de France, France

{audemard, lagniez, marquis, szczepanski}@cril.fr

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, rigorous explanations of 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 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 [12, 13, 14, 16, 19, 24, 25, 28, 30, 40]). Models with high prediction performance are usually considered as poorly intelligible [30, 1, 26, 6, 36]. Among them is the family of *boosted trees* [11], which is among the state-of-the-art ML models when dealing with tabular data [4].

Related Work. The design of efficient methods for interpreting ML models and explaining their decisions is acknowledged as an issue of the utmost importance when ML models are to be used in safety-critical applications [27]. 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", i.e., instances that are covered by the same explanation but are nevertheless classified differently by the model [21]. Especially, [17] shows that the amount of counterexamples can be high when using some of the most popular approaches for computing model-agnostic explanations, namely LIME [34], Anchors [35], and SHAP [25].

In order to avoid the generation of unsound explanations, a number of alternative approaches, falling under the *formal XAI* umbrella [27], 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 [32, 37, 39]). Thanks to such mappings, XAI queries about classifiers, including the generation of explanations, can be delegated to the corresponding circuits (see for instance [9, 3, 33]).

Ensemble methods (bagging, boosting, stacking, etc.) have been considered in such a perspective. Thus, [8, 23, 2] show how to derive abductive explanations for random forests [5]. 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, subset-minimal abductive explanations (alias sufficient reasons [9]) are often targeted. As to boosted trees, [21] 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). [18] 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. Showing how to enlarge the set of sufficient reasons that can be computed in practice for large datasets is the main goal of this paper. To reach this objective, we introduce the notion of tree-specific explanation for a boosted tree. We show that, unlike sufficient reasons for boosted trees, tree-specific explanations are abductive explanations that can be computed in polynomial time. We also show that, while tree-specific explanations are not subset-minimal in the general case, they turn out to be close to sufficient reasons in practice. Furthermore, because sufficient reasons can be derived from tree-specific explanations, computing tree-specific explanations can be exploited as a preprocessing step in the derivation of sufficient reasons. Experiments on various datasets show that leveraging tree-specific explanations for generating sufficient reasons is a valuable approach.

The proofs of the propositions presented in the paper are reported in a final appendix. A description of the datasets and the code used in our experiments are available from www.cril.fr/expekctation/.

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 x is a vector (v_1, \ldots, v_n) where each v_i ($i \in [n]$) is an element of D_i . x is also viewed as a term, i.e., a conjunctively-interpreted set of propositional atoms $t_x = \{(A_i = v_i) : i \in [n]\}$, 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. 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 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 $w(T,x) \in \mathbb{R}$ of T for an input instance $x \in X$ 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 x. A decision tree over $\{A_1,\ldots,A_n\}$ is a regression tree over $\{A_1,\ldots,A_n\}$ where leaves are labeled in $\{0,1\}$.

A forest over $\{A_1,\ldots,A_n\}$ associated with a class $j\in[m]$ is an ensemble of trees $F^j=\{T^j_1,\cdots,T^j_{p_j}\}$, where each T^j_k $(k\in[p_j])$ 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.

In a 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 $\{A_1,\ldots,A_n\}$. 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 a 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.

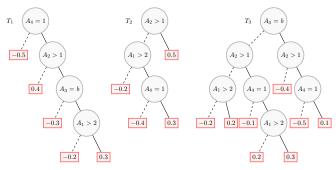


Figure 1: A boosted tree $BT = \{F\}$ consisting of a single forest $F = \{T_1, T_2, T_3\}$.

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

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

Abductive Explanations. Explaining the classification achieved by a classifier f on an instance x consists in identifying a subset of the characteristics of x that is enough to get the class returned by f. Formally, an abductive explanation t for an instance $x \in X$ given a classifier f (that is binary or not) is a subset $t \in t_x$ such that every instance $x' \in X$ such that $t \subseteq t_{x'}$ is classified by f in the same way as x: f(x') = f(x). The size |t| of an abductive explanation t is the number of characteristics in it. A sufficient reason t for t for t given t is an abductive explanation for t given t such that no proper subset t' of t is an abductive explanation for t given t sufficient reasons t for t given t such that no proper subset t' of t is an abductive explanation for t given t sufficient reasons t for t given t such that no proper subset t' of t is an abductive explanation for t given t such that no proper subset t' of t is an abductive explanation for t given t such that no proper subset t' of t is an abductive explanation for t given t such that no proper subset t' of t is an abductive explanation for t given t such that t such t

Example 2. Considering again our running example, $t = \{(A_1 = 4), (A_4 = 1)\}$ is a sufficient reason for $\mathbf{x} = (A_1 = 4, A_2 = 3, A_3 = b, A_4 = 1)$ given $BT = \{F\}$. Indeed, all the instances

¹Unlike [20], we do not require abductive explanations to be minimal w.r.t. set inclusion, in order to keep the concept distinct (and actually more general) than the one of sufficient reasons.

²Sufficient reasons are also known as prime-implicant explanations [38].

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

1 t \leftarrow t_{\boldsymbol{x}}

2 foreach c_i \in t_{\boldsymbol{x}} do

3 \sqsubseteq if implicant(t \setminus \{c_i\}, \boldsymbol{x}, f) then t \leftarrow t \setminus \{c_i\}

4 return t
```

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 the four cases, we have w(F, x') > 0 (see Table 1), showing that BT(x') = 1. Since BT(x) = 1, t is an abductive explanation for x given BT. Since no proper subset of t satisfies this property, t actually is a sufficient reason for x given BT.

$A_1 = 4$	$A_2 > 1$	$A_3 = b$	$A_4 = 1$	$w(T_1, \boldsymbol{x}')$	$w(T_2, \boldsymbol{x}')$	$w(T_3, \boldsymbol{x}')$	$w(F, \boldsymbol{x}')$
1	0	0	1	0.4	$0.3 \\ 0.3$	0.2	
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 x' extending t.

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.

3 Computing Sufficient Reasons

In order to compute a sufficient reason for an input instance x given a classifier f, one can take advantage of a simple greedy algorithm (see Algorithm 1). Starting with $t = t_x$, this algorithm considers all the characteristics $c_i = (A_i = v_i)$ of x in a specific order and, at each step, tests whether t deprived of c_i is still an abductive explanation for 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 x have been considered, the resulting term t is by construction a sufficient reason for 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 x given f, i.e., any instance covered by $t \setminus \{c_i\}$ is classified in the same way as x by f. Though this test can be achieved in polynomial time for some families of classifiers f (including decision trees) [22, 15], it is intractable in general. Especially, it is CONP-hard when f is a random forest [2]. Similarly, when f is a boosted tree BT, we have:

Proposition 1. Let BT be a boosted tree over $\{A_1, \ldots, A_n\}$ and $x \in X$. Let t be a subset of t_x . Deciding whether t is an abductive explanation for 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. [21] 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, [18] 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 [18] contain at most 60 attributes).

4 Computing Tree-Specific 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 $x \in X$ 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 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 $\mathbf{x} \in \mathbf{X}$. Let t be a subset of $t_{\mathbf{x}}$.

- A worst instance extending t given F is an instance $\mathbf{x}' \in \mathbf{X}$ such that $t \subseteq t_{\mathbf{x}'}$ and $w(F, \mathbf{x}') = min(\{w(F, \mathbf{x}'') : \mathbf{x}'' \in \mathbf{X} \text{ and } t \subseteq t_{\mathbf{x}''}\}).$
- A best instance extending t given F is an instance $x' \in X$ such that $t \subseteq t_{x'}$ and $w(F, x') = max(\{w(F, x'') : x'' \in X \text{ and } t \subseteq t_{x''}\}).$

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 extending t given F. On this ground, we have:

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

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

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

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

Definition 2. Let $BT = \{F^1, \dots, F^m\}$ be a boosted tree over $\{A_1, \dots, A_n\}$ and $\mathbf{x} \in \mathbf{X}$ such that $BT(\mathbf{x}) = i$. Let t be a subset of $t_{\mathbf{x}}$. Given BT and \mathbf{x} , a worst instance extending t is an instance $\mathbf{x}' \in \mathbf{X}$ such that $t \subseteq t_{\mathbf{x}'}$ and \mathbf{x}' minimizes $w(F^i, \cdot) - \max_{i \in [m] \setminus \{i\}} w(F^j, \cdot)$.

Then we have:

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

Propositions 1 and 2 (or 3) show together that identifying a worst (resp. best) instance $x' \in X$ extending a term $t \subseteq t_x$ given a boosted tree BT is intractable.⁴ Indeed, if it were not the case, we could check in polynomial time whether t is an abductive explanation for x given BT by testing whether x' is classified by BT in the same way as 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 x' satisfying the conditions associated with a valid root-to-leaf path leading to a minimal (resp. maximal) weight leaf and satisfying $t \subseteq t_{x'}$ 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 that:

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

⁴Intractability is still the case in the more specific case the classifier is a random forest.

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

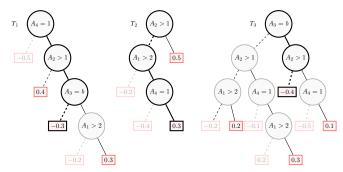


Figure 2: Worst instances and the corresponding weights for the regression trees used in BT.

Tree-Specific Explanations. We are now in position to define the notion of *tree-specific explanation t* for an instance x given a boosted tree BT. We start with the binary classification case, i.e., when BT consists of a single forest F:

Definition 3. Let $F = \{T_1, \dots, T_p\}$ be a forest over $\{A_1, \dots, A_n\}$ and $\mathbf{x} \in \mathbf{X}$.

- If F(x) = 1, then t is a tree-specific explanation for x given F if and only if t is a subset of t_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(x) = 0, then t is a tree-specific explanation for x given F if and only if t is a subset of t_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, tree-specific explanations can be defined as follows:

Definition 4. Let $BT = \{F^1, \dots, F^m\}$ be a boosted tree over $\{A_1, \dots, A_n\}$ where each F^j $(j \in [m])$ contains p_j trees, and $\mathbf{x} \in \mathbf{X}$ such that $BT(\mathbf{x}) = i$. t is a tree-specific explanation for \mathbf{x} given BT if and only if t is a subset of $t_{\mathbf{x}}$ 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 tree-specific explanations valuable is that they are abductive explanations:

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

Especially, 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 tree-specific 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 tree-specific explanation t for x given BT is not a sufficient reason for x given BT: t may contain characteristics of 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 $\mathbf{x} = (A_1 = 4, A_2 = 3, A_3 = b, A_4 = 1)$ given $BT = \{F\}$ is not a tree-specific explanation for \mathbf{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 $w(F, \mathbf{x}) = 0.9 > 0$. Contrastingly, $t' = \{(A_2 = 3), (A_4 = 1)\}$ is a tree-specific explanation for \mathbf{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 $w_{\downarrow}(t', T_1) + w_{\downarrow}(t', T_2) + w_{\downarrow}(t', T_3) = 0.3 > 0$. It can be verified that t' also is a sufficient reason for \mathbf{x} given BT.

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

Though subset-minimality is required in both cases, the discrepancy between tree-specific explanations and sufficient reasons can be easily explained by the fact that tree-specific 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 $A_2 \le 1$.

In the worst case, the number of useless characteristics in a tree-specific 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 $\mathbf{x} \in \mathbf{X}$. It can be the case that the unique tree-specific explanation for \mathbf{x} given BT consists of $t_{\mathbf{x}}$ itself, while \varnothing is the unique sufficient reason for \mathbf{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 tree-specific explanations valuable is that they can be computed efficiently. Thus, the greedy algorithm TS given by Algorithm 2 can be used to derive in $\mathcal{O}(n|BT|)$ a tree-specific explanation for \boldsymbol{x} given BT in the multi-class case.

Proposition 6. Let BT be a boosted tree over $\{A_1, \ldots, A_n\}$ and $\mathbf{x} \in \mathbf{X}$. $TS(\mathbf{x}, BT)$ returns a tree-specific explanation for \mathbf{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 the instance is positive, and the sign of $\sum_{k=1}^p w_{\uparrow}(t, T_k)$ when the instance is negative).

Interestingly, when dealing with boosted trees, the greedy algorithm SR (Algorithm 1) for deriving sufficient reasons can be exploited to remove useless characteristics in tree-specific explanations, i.e., to generate sufficient reasons from tree-specific explanations. Viewed from a different angle, the computation of a tree-specific 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 $SR(TS(t_{\boldsymbol{x}},BT),BT)$.

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 5 will show it, TS may remove in a very efficient way many useless characteristics of x, thus avoiding many calls to the computationally expensive function implicant.

5 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. 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 [7] that learns gradient boosted trees. All parameters have been set to their default values (especially, 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 counter-productive 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 4097when 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 x, TS has been called 1000 times: at each run, an elimination ordering of the characteristics of x (as considered at line 3. of Algorithm 2) has been picked up uniformly at random, and a shortest tree-specific 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.

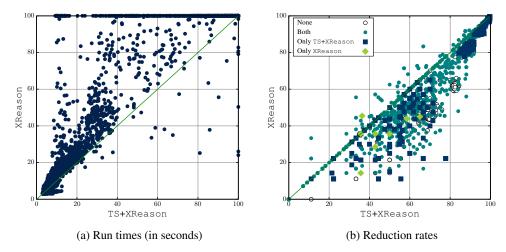


Figure 3: Comparing TS+XReason to XReason.

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 this scatter plot, 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. Indeed, size is one of the criteria to be considered when evaluating the intelligibility of an explanation: everything else being equal, shortest explanations are easier to understand than longer explanations. The reduction rate achieved by an abductive explanation t for an instance $\mathbf{x} = (v_1, \dots, v_n)$ is given by $1 - \frac{|t|}{n}$. For each dot corresponding to an instance $\mathbf{x} = (v_1, \dots, v_n)$, the x-coordinate of the dot is the reduction rate achieved by the abductive explanation generated by TS+XReason for \mathbf{x} , while the y-coordinate is the reduction rate achieved by the abductive explanation generated by XReason for \mathbf{x} . We can observe on this figure that TS+XReason produces in general much smaller abductive explanations than those generated by XReason.

On Figure 3 (b), we used different dot representations for instances, 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. One can 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, and for 895 abductive explanations that have been generated, there are no subset-minimality guarantees for any of the programs used. Thus, a significant amount of 621 sufficient reasons have been gained by taking advantage of TS as a preprocessing to XReason.

-					Run time					Reduction rate			
Dataset	#Cls.	#Feat.	#Inst.	Acc.			KReason		XReasc		_	+ XReason	XReason
					TS	XReason	TS+XReason	#TO	XReason	#TO	TS	TS+XReason	Arreason
gina_agnostic	2	971	3468	95.13	1.80	95.92	97.72	66	100	100	87.51	88.56	81.66
malware	2	1085	6248	99.46	0.22	5.79	6.01	0	8.34	0	98.74	98.85	98.64
ad_data	2	1559	3279	97.78	0.32	13.88	14.20	0	71.41	47	98.95	99.18	98.26
christine	2	1637	5418	73.81	6.13	94.42	100	100	100	100	82.34	82.42	62.44
cnae	9	857	1079	91.48	5.65	49.57	55.23	3	89	50	94.17	94.90	92.61
gisette	2	5001	7000	97.83	2.95	90.59	93.54	23	100		97.05	97.23	95.46
arcene	2	10001	200	80.50	0.20	5.70	5.89	0	4.49	0	99.62	99.63	99.59
dexter	2	20001	600	91.83	0.30	6.75	7.05	0	11.25	0	99.88	99.90	99.85
allBooks	8	8267	590	87.12	7.62	51.01	58.62	3	100	100	99.12	99.23	98.54
farm-ads	2	54877	4143	90.27	2.86	97.91	99.99	93	100	100	99.62	99.64	99.55
dorothea	2	100001	1150	93.91	0.68	11.50	12.17	0	17.28	0	99.91	99.92	99.90

Table 2: Performances of TS+XReason and XReason in terms of run times and reduction rates for 10 datasets.

To complete the results furnished by the scatter plots, Table 2 presents some details concerning 10 datasets among the 50 datasets used. The columns give, from left to right, the name of the dataset, the number of classes, features and instances in it, the mean accuracy, and results (over 100 instances) about TS+XReason and XReason in terms of mean run times (in seconds)⁶, number of TOs, and then in terms of mean reduction rates. For TS+XReason, we report the mean run times and mean reduction rates achieved by each component of it (i.e., by TS as a preprocessing step, and then by XReason run on the abductive explanation generated by TS). In light of those results, TS appears as valuable: TS is computationally efficient (the cumulated run times over 1000 runs are bounded by a few seconds), and TS achieves important reduction rates (close to those achieved by TS+XReason). Notably, the reduction rates achieved by TS are often higher than those achieved by XReason. This explains the good performance of the pipeline TS+XReason.

6 Conclusion

We have introduced a new notion of abductive explanation for boosted trees, called tree-specific explanations. In the worst case, tree-specific explanations can be arbitrarily larger than sufficient reasons, thus they may contain many useless characteristics. However, we have shown that, unlike sufficient reasons, their generation is tractable. We have presented a polynomial-time algorithm TS for computing tree-specific explanations, and we have proved its correctness. Because a sufficient reason can be extracted from a tree-specific explanation, TS can be used as a preprocessing step for

⁵In 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 [10, 31], since intelligibility typically depends on the explainee (i.e., the person who asked for an explanation) [29].

⁶Whenever a TO occurs, we considered a computation time equal to 100s.

greedy algorithms (like XReason) that derive sufficient reasons from boosted trees. Empirically, we have shown that the combination 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 can also be useful alone, as a generator of abductive explanations.

Many research perspectives can be envisioned. 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, incorporating domain knowledge about the characteristics. It would also be useful to investigate whether the notions of worst/best instances for a forest could be exploited for improving the generation of counterfactual explanation for boosted trees.

Acknowledgments

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

- [1] 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.
- [2] 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, 2022.
- [3] P. Barceló, M. Monet, J. Pérez, and B. Subercaseaux. Model interpretability through the lens of computational complexity. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.*
- [4] 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.
- [5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [6] R. Caruana, S. M. Lundberg, M. Túlio Ribeiro, H. Nori, and S. Jenkins. Intelligible and explainable machine learning: Best practices and practical challenges. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 3511–3512. ACM, 2020.
- [7] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proc. of KDD'16*, page 785–794, 2016.
- [8] 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.*
- [9] A. Darwiche and A. Hirth. On the reasons behind decisions. In *Proc. of ECAI'20*, pages 712–720, 2020.
- [10] F. Doshi-Velez and B. Kim. A roadmap for a rigorous science of interpretability. *CoRR*, abs/1702.08608, 2017.
- [11] J. H. Friedman. Greedy function approximation. *The Annals of Statistics: A Gradient Boosted Machine*, 29(5):1189–1232, 2001.
- [12] N. Frosst and G. E. Hinton. Distilling a neural network into a soft decision tree. In *Proc. of the First International Workshop on Comprehensibility and Explanation in AI and ML*, volume 2071 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.
- [13] 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.

- [14] 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.
- [15] X. Huang, Y. Izza, A. Ignatiev, and J. Marques-Silva. On efficiently explaining graph-based classifiers. In Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR 2021, Online event, November 3-12, 2021, pages 356–367, 2021.
- [16] 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.
- [17] A. Ignatiev. Towards trustable explainable AI. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, pages 5154–5158. ijcai.org, 2020.
- [18] A. Ignatiev, Y. Izza, P.J. Stuckey, and J. Marques-Silva. Using MaxSAT for efficient explanations of tree ensembles. In *Proc. of AAAI* '22, 2022.
- [19] A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-based explanations for machine learning models. In *Proc. of AAAI'19*, pages 1511–1519, 2019.
- [20] A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-based explanations for machine learning models. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI'19)*, pages 1511–1519, 2019.
- [21] A. Ignatiev, N. Narodytska, and J. Marques-Silva. On validating, repairing and refining heuristic ML explanations. *CoRR*, abs/1907.02509, 2019.
- [22] Y. Izza, A. Ignatiev, and J. Marques-Silva. On explaining decision trees. *CoRR*, abs/2010.11034, 2020.
- [23] Y. Izza and J. Marques-Silva. On explaining random forests with SAT. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI'21)*, 2021.
- [24] 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.
- [25] S. Lundberg and S-I. Lee. A unified approach to interpreting model predictions. In *Proc. of NIPS'17*, pages 4765–4774, 2017.
- [26] 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.
- [27] J. Marques-Silva and A. Ignatiev. Delivering trustworthy AI through formal XAI. In *Proc. of AAAI*'22, 2022.
- [28] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.
- [29] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.
- [30] Ch. Molnar. Interpretable Machine Learning A Guide for Making Black Box Models Explainable. Leanpub, 2019.
- [31] 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.
- [32] 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.
- [33] A. Parmentier and T. Vidal. Optimal counterfactual explanations in tree ensembles. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event,* volume 139 of *Proceedings of Machine Learning Research*, pages 8422–8431. PMLR, 2021.
- [34] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

- [35] M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-precision model-agnostic explanations. In *Proc. of AAAI'18*, pages 1527–1535, 2018.
- [36] 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.
- [37] A. Shih, A. Choi, and A. Darwiche. Formal verification of Bayesian network classifiers. In *Proc. of PGM'18*, pages 427–438, 2018.
- [38] A. Shih, A. Choi, and A. Darwiche. A symbolic approach to explaining bayesian network classifiers. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI'18)*, pages 5103–5111, 2018.
- [39] A. Shih, A. Choi, and A. Darwiche. Compiling Bayesian networks into decision graphs. In *Proc. of AAAI'19*, pages 7966–7974, 2019.
- [40] A. Shih, A. Darwiche, and A. Choi. Verifying binarized neural networks by Angluin-style learning. In *Proc. of SAT'19*, pages 354–370, 2019.

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 [2] (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 $BT(\boldsymbol{x}) = 1$. 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 $\boldsymbol{x} \in \mathbb{R}$, 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

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

as expected.

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

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

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

Proof of Proposition 4

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

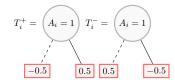
Since t is a tree-specific explanation for x given BT, t is a subset of t_x such that for every $k \in [m] \setminus \{i\}$, we have $\sum_{l=1}^{p_i} w_{\downarrow}(t, T_l^i) > \sum_{l=1}^{p_k} w_{\uparrow}(t, T_l^k)$. In particular, for k = j, we have $\sum_{l=1}^{p_i} w_{\downarrow}(t, T_l^i) > \sum_{l=1}^{p_j}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

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

Finally, we get that $w(F^j, \mathbf{x}') = \sum_{l=1}^{p_j} w(T^j_l, \mathbf{x}') \leq \sum_{l=1}^{p_j} w_{\uparrow}(t, T^j_l) < \sum_{l=1}^{p_i} w_{\downarrow}(t, T^i_l) \leq \sum_{l=1}^{p_i} w(T^i_l, \mathbf{x}') = w(F^i, \mathbf{x}')$. 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 $\boldsymbol{x}=(0,\dots,0)$. 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 $w_{\uparrow}(t_{\boldsymbol{x}}, T_k^j) = w(T_k^j, \boldsymbol{x})$. Accordingly, $\sum_{k=1}^{p_i} w_{\downarrow}(t_{\boldsymbol{x}}, T_k^i) > \sum_{k=1}^{p_j} w_{\uparrow}(t_{\boldsymbol{x}}, T_k^j)$ 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 $t \in t' \subseteq t_x$, 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 $t \in t'$, 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 $k \in [p_i]$.