# Computing Abductive Explanations for Boosted Regression Trees

**Gilles Audemard**[1], **Steve Bellart**[1], **Jean-Marie Lagniez**[1] and **Pierre Marquis**[1,2]

[1] Univ. Artois, CNRS, Centre de Recherche en Informatique de Lens (CRIL), F-62300 Lens, France
[2]Institut Universitaire de France
{audemard, bellart, lagniez, marquis}@cril.fr

## Abstract

We present two algorithms for generating (resp. evaluating) abductive explanations for boosted regression trees. Given an instance $\boldsymbol{x}$ and an interval $I$ containing its value $F(\boldsymbol{x})$ for the boosted regression tree $F$ at hand, the generation algorithm returns a (most general) term $t$ over the Boolean conditions in $F$ such that every instance $\boldsymbol{x}'$ satisfying $t$ is such that $F(\boldsymbol{x}') \in I$. The evaluation algorithm tackles the corresponding inverse problem: given $F$, $\boldsymbol{x}$ and a term $t$ over the Boolean conditions in $F$ such that $t$ covers $\boldsymbol{x}$, find the least interval $I_t$ such that for every instance $\boldsymbol{x}'$ covered by $t$ we have $F(\boldsymbol{x}') \in I_t$. Experiments on various datasets show that the two algorithms are practical enough to be used for generating (resp. evaluating) abductive explanations for boosted regression trees based on a large number of Boolean conditions.

## 1 Introduction

The past few years have witnessed the quick development of a new field, called eXplainable AI (XAI), aroused by the large spectrum of applications leveraging the stunning predictive power of machine learning (ML) models, their opacity, and the tremendous need for gaining trust in such models, especially when they are used in safety-critical applications (see for instance [Doshi-Velez and Kim, 2017; Adadi and Berrada, 2018; Lipton, 2018; Molnar, 2019; Xu *et al.*, 2019; Arrieta *et al.*, 2020; Caruana *et al.*, 2020; Rudin *et al.*, 2021]).

So far, most works about XAI have been concerned with explanation and verification issues about *classification functions*, i.e., mappings from the set $\boldsymbol{X}$ of instances to a finite set $\mathcal{C}$ of classes. In contrast, in this paper, we consider the generation and the evaluation of explanations for *regression functions*, i.e., mappings from $\boldsymbol{X}$ to $\mathbb{R}$. The explanations sought are abductive ones, i.e., they are intended to explain why the instances $\boldsymbol{x} \in \boldsymbol{X}$ that are considered have been mapped to their corresponding values $f(\boldsymbol{x})$ by the regression function $f$. In the following, we focus on regression functions given by *boosted regression trees*, that are combinatorial and non-differentiable in essence, and for which the generation of explanations is stimulating.

A major difference between a classification task and a regression one is that in the latter case the exact value taken by $f(\boldsymbol{x})$ does not really matter. Would this value be $f(\boldsymbol{x}) \pm \epsilon$ for a sufficiently small real number $\epsilon$ instead of $f(\boldsymbol{x})$, this would not be a big deal. Mathematically speaking, this means that what does matter is that the value of $f(\boldsymbol{x})$ belongs to some interval $I$. Of course, in the classification setting, things are much different since two distinct values of $f(\boldsymbol{x})$ actually indicate two distinct classes, and in the general case, there is no notion of distance or similarity between classes that would make sense ($\mathcal{C}$ is discrete and typically even not ordered). Such an interval allows for flexibility in the generation of abductive explanations since some imprecision about the value of the instances is tolerated.

The contribution of this paper mainly consists of the design and the assessment of two algorithms **G** and **E** for generating (resp. evaluating) abductive explanations for regression functions $f$ represented by boosted regression trees $F$. Such abductive explanations are represented by terms $t$, i.e., conjunctions of literals, over the Boolean conditions occurring in the boosted regression trees $F$ that are used to represent regression functions.

Thus, **G** returns a most general term $t$ over the Boolean conditions in $F$ such that every instance $\boldsymbol{x}'$ satisfying $t$ is such that $F(\boldsymbol{x}') \in I$. When dealing with the classification task and $I = [F(\boldsymbol{x}), F(\boldsymbol{x})]$, such explanations are referred to as PI-explanations [Shih *et al.*, 2018], sufficient reasons [Darwiche and Hirth, 2020], and also as (subset-minimal) abductive explanations [Ignatiev *et al.*, 2019a]. The terms derived by our algorithm are the subset-minimal terms (hence, the logically weakest ones) covering $\boldsymbol{x}$ and such that all the instances covered by them have $F$-predicted values in the preset interval $I$. We call this last condition the *coverage condition induced by the interval* $I$.

The evaluation algorithm **E** concerns the inverse problem, that can be defined as follows: given $F$, $\boldsymbol{x}$, a term $t$ over the Boolean conditions in $F$ such that $t$ covers $\boldsymbol{x}$, the goal is to determine the least interval $I_t$ that contains the regression values reached by every instance $\boldsymbol{x}'$ covered by $t$, i.e., we have $F(\boldsymbol{x}') \in I_t$. **E** can be used to *measure the extent to which* $t$ *is imprecise* when $t$ is viewed as an explanation of the value taken by $F$ for $\boldsymbol{x}$. The imprecision of $t$ simply is the length of $I_t$ (so the smaller the interval the better the precision). Interestingly, a monotonic relationship exists between the general-

ity of the abductive explanations $t$ covering $\boldsymbol{x}$ and the lengths of the corresponding intervals $I_t$: the more general the explanations, the larger the intervals.

In the following, we show that the problem of determining whether a term $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ and the problem of determining whether every instance covered by a term $t$ has an $F$-value belonging to a given interval $I$ are coNP-complete. As a direct consequence, there is little hope that the generation and evaluation problems considered in the paper can be solved in (deterministic) polynomial time. Indeed, for the generation problem, one tries to maximize the generality of the explanation that is produced, while for the evaluation problem, one tries to minimize the length of the interval that is reported. Accordingly, experiments must be achieved to figure out to which extent the two algorithms presented in the paper scale up. To this aim, **G** and **E** have been evaluated on various datasets. The experiments made show that they are rather efficient in terms of run time for being practical enough. Indeed, most of the time, the algorithms can be used to generate (resp. evaluate) in a few seconds abductive explanations for boosted regression trees based on a large number of Boolean conditions (up to 800). A valuable observation is that most of the time the (subset-minimal) abductive explanations that are generated are significantly smaller than the initial descriptions of the instances in terms of Boolean attributes.

The proofs of the propositions presented in the paper are given in a final appendix. Additional empirical results and the code used in our experiments are provided as a supplementary material, available from http://www.cril.fr/expekctation/index.html.

## 2 Formal Preliminaries

We consider a finite set $\mathcal{A} = \{A_1, \ldots, A_n\}$ 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* ($D_i$ is a set of values that are not specifically ordered, e.g., $D_i = \{$*"employed"*, *"unemployed"*, *"self-employed"*$\}$), or *Boolean* ($D_i = \{0, 1\}$). Thus, $\mathcal{A}$ is the union of three pairwise disjoint subsets $\mathcal{A}_N$, $\mathcal{A}_C$, $\mathcal{A}_B$ containing respectively the numerical, categorical, Boolean attributes. We suppose that the size of any element of $D_i$ for any $A_i$ is upper bounded by a preset constant. An *instance* $\boldsymbol{x}$ is a vector $(v_1, \ldots, v_n)$ where each $v_i$ ($i \in [n]$) is an element of $D_i$. Each pair $A_i = v_i$ is called a *characteristic* of the instance $\boldsymbol{x}$. $\boldsymbol{X}$ denotes the set of all instances.

A *regression tree* over $\mathcal{A}$ is a binary tree $T$, each of its internal nodes being labeled with a Boolean condition on an attribute from $\mathcal{A}$, and leaves are labeled by real numbers. The conditions are of the form $A_i > v_j^i$ with $v_j^i$ a number when $A_i$ is a numerical attribute, $A_i = v_j^i$ when $A_i$ is a categorical attribute, and $A_i$ (or equivalently $A_i = 1$) when $A_i$ is a Boolean attribute. The *value* $T(\boldsymbol{x}) \in \mathbb{R}$ of $T$ for an input instance $\boldsymbol{x} \in \boldsymbol{X}$ is given by the real number labelling the leaf reached from the root as follows: at each internal node go to the left or right child depending on whether or not the condition labelling the node is satisfied by $\boldsymbol{x}$. The *size* $|T|$ of $T$ is the sum of the sizes of its nodes, where the size of a node is the number of bits required to encode the corresponding condition (this size varies depending on the type of the attribute used in the condition). $min(T)$ (resp. $max(T)$) denotes the minimum (resp. maximum) number labelling a leaf of $T$.

A *boosted regression tree* over $\mathcal{A}$ is an ensemble of trees (alias a forest) $F = \{T_1, \cdots, T_m\}$, where each $T_i$ ($i \in [m]$) is a regression tree over $\mathcal{A}$, and such that the value $F(\boldsymbol{x}) \in \mathbb{R}$ of $F$ for an input instance $\boldsymbol{x} \in \boldsymbol{X}$ is given by $F(\boldsymbol{x}) = \oplus_{i=1}^{m} T_i(\boldsymbol{x})$. In the following, $\oplus$ is the *sum* operator but other operators strictly monotonic in each argument could be used instead. The *size* $|F|$ of $F$ is the sum of the size of its trees. $F(\boldsymbol{x})$ can be computed in time linear in $|F|$ and $|\boldsymbol{x}|$.

Let $\mathcal{B}$ denote the set of all Boolean conditions used in $F$. When $|\mathcal{B}| = p$, a boosted regression tree $F$ over $\mathcal{A}$ can be viewed alternatively as a mapping from $\{0,1\}^p$ to $\mathbb{R}$. Every $A_i \in \mathcal{A}$ corresponds to a set of Boolean conditions in $\mathcal{B}$, noted $\tau(A_i)$, so that $\bigcup_{A_i \in \mathcal{A}} \tau(A_i) = \mathcal{B}$. The definition of $\tau(A_i)$ depends on the type of $A_i$. Thus, if $A_i$ is a numerical attribute and $D_i^f = \{v_1^i, \ldots, v_{k_i}^i\}$ is the set of values ordered in ascending way such that the Boolean condition $(A_i > v_j^i)$ ($j \in [k_i]$) occurs in at least one tree of $F$, then $\tau(A_i) = \{(A_i > v_j^i) : j \in [k_i]\}$. For the sake of simplicity, $(A_i > v_j^i)$ is also noted $B_j^i$. If $A_i$ is a categorical attribute and $D_i^f = \{v_1^i, \ldots, v_{k_i}^i\}$ is the set of values such that the Boolean condition $(A_i = v_j^i)$ ($j \in [k_i]$) occurs in at least one tree of $F$, then $\tau(A_i) = \{(A_i = v_j^i) : j \in [k_i]\}$. Each $(A_i = v_j^i)$ is also noted $B_j^i$. Finally, if $A_i$ is a Boolean attribute, then $\tau(A_i) = \{(A_i = 1)\}$. $(A_i = 1)$ is also noted $B^i$.

An important observation is that the Boolean conditions used in $F$ are not necessarily *independent*. For instance, it can be the case that the two Boolean conditions $(A_1 > 1000)$ and $(A_1 > 2000)$ occur in $F$ when $A_1 \in \mathcal{A}$ is a numerical attribute, and/or that the two Boolean conditions $(A_2 =$ *"self-employed"*$)$ and $(A_2 =$ *"unemployed"*$)$ occur in $F$ when $A_2 \in \mathcal{A}$ is a categorical attribute. However, no instance of $\boldsymbol{X}$ can render $(A_1 > 2000)$ true and $(A_1 > 1000)$ false, or $(A_1 =$ *"self-employed"*$)$ true and $(A_2 =$ *"unemployed"*$)$ true. Thus, some *constraints* $\Sigma$ over $\mathcal{B}$ must be exploited to characterize the truth assignments over $\mathcal{B}$ that actually correspond to instances from $\boldsymbol{X}$ [Gorji and Rubin, 2022]. Especially, if $A_i \in \mathcal{A}$ is a numerical attribute then $\Sigma$ contains the clauses $(A_i > v_j^i) \vee \neg(A_i > v_{j+1}^i)$ for $j \in [k_i - 1]$. If $A_i \in \mathcal{A}$ is a categorical attribute then $\Sigma$ contains the clauses $\neg(A_i = v_j^i) \vee \neg(A_i = v_l^i)$ for $j \in [k_i]$ and $l \in [k_i] \setminus [j]$. The size of $\Sigma$ is at most quadratic in the size of $\mathcal{B}$, so at most quadratic in the size of $F$.

**Example 1.** *Let us consider a loan application scenario that will be used as a running example. The goal is to predict the amount of money that can be granted to an applicant described using three attributes ($\mathcal{A} = \{A_1, A_2, A_3\}$). $A_1$ is a numerical attribute giving the income per month of the applicant, $A_2$ is a categorical feature giving its employment status as "employed", "unemployed" or "self-employed", and $A_3$ is a Boolean feature set to true if the customer is married, false otherwise. We suppose that the boosted regres-*

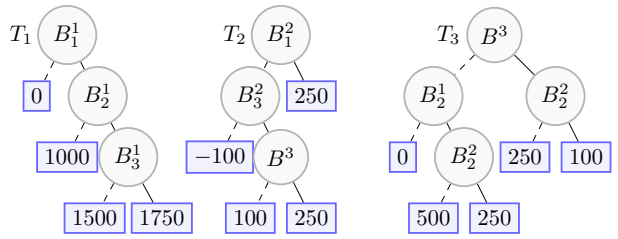


Figure 1: A boosted regression tree.

*sion tree $F$ over $\mathcal{A}$ depicted in Figure 1 has been learned. $F$ is built upon Boolean conditions: $\mathcal{B} = \{B_1^1, B_2^1, B_3^1, B_1^2, B_2^2, B_3^2, B^3\}$. $B_1^1$, $B_2^1$, and $B_3^1$ represent respectively the conditions "$A_1 > 1000\$$", "$A_1 > 2000\$$" and "$A_1 > 3000\$$". Similarly, $B_1^2$, $B_2^2$ and $B_3^2$ represent respectively the conditions $A_2$ = "employed", $A_2$ = "unemployed" and $A_2$ = "self-employed". Finally, $B^3$ represents the condition $(A_3 = 1)$ ("the applicant is married"). By construction, $\Sigma = (B_1^1 \vee \neg B_2^1) \wedge (B_2^1 \vee \neg B_3^1) \wedge (\neg B_1^2 \vee \neg B_2^2) \wedge (\neg B_1^2 \vee \neg B_3^2) \wedge (\neg B_2^2 \vee \neg B_3^2)$. Suppose that the applicant is described by $\boldsymbol{x}_{ex} = (2200\$, \text{"self-employed"}, 1)$. Then, $F(\boldsymbol{x}_{ex}) = 1500 + 250 + 250 = 2000\$$.*

A term $t$ over $\mathcal{B} = \{B_1, \ldots, B_p\}$ is a conjunctively-interpreted set of literals over $\mathcal{B}$. $\top$ denotes the term associated with the empty set of literals. $t$ is *canonical* when it contains one literal per Boolean condition in $\mathcal{B}$. Such a term thus corresponds to a truth assignment over $\mathcal{B}$. Every instance $\boldsymbol{x} \in \boldsymbol{X}$ can be associated with a canonical term $t_{\boldsymbol{x}}$ over $\mathcal{B}$ such that for every $i \in [p]$, $B_i$ (resp. $\neg B_i$) belongs to $t_{\boldsymbol{x}}$ if and only if $\boldsymbol{x}$ satisfies (resp. does not satisfy) the condition $B_i$. A term $t$ over $\mathcal{B}$ *covers* an instance $\boldsymbol{x} \in \boldsymbol{X}$ whenever $t \subseteq t_{\boldsymbol{x}}$. Deciding whether $t$ covers $\boldsymbol{x}$ can be achieved in time linear in $|t|$ and $|\boldsymbol{x}|$.

Every term $t$ over $\mathcal{B}$ that is consistent (i.e., $t$ does not contain both an element of $\mathcal{B}$ and its negation) can be *simplified* using $\Sigma$. Thus, each time $t$ contains two distinct literals $\ell$ and $\ell'$ such that $\ell'$ is entailed by $\ell$ given $\Sigma$, $\ell'$ can be removed from $t$. The specific nature of $\Sigma$ ensures that such a simplification process is *confluent*, i.e., the term obtained at the end of the simplification process (called the *simplification* of $t$ given $\Sigma$) is uniquely defined whatever the ordering according to which the literals are considered (this would not be ensured if $\Sigma$ was any formula over $\mathcal{B}$). Thus, when simplified, $t$ cannot contain more than one positive (resp. negative) literal issued from the same numerical attribute $A_i$ and more than one positive literal issued from the same categorical attribute $A_i$ (and if such a positive literal exists, $t$ does not contain any negative literal issued from $A_i$). By construction, the simplification of $t$ given $\Sigma$ is equivalent to $t$ under $\Sigma$. Furthermore, the simplification of $t$ given $\Sigma$ can be computed in time $\mathcal{O}(|\Sigma| \cdot |t|^2)$.

## 3 On Abductive Explanations for Boosted Regression Trees

Let $F$ be a boosted tree over $\mathcal{A}$. Suppose that $t$ is a term over $\mathcal{B}$ that covers $\boldsymbol{x} \in \boldsymbol{X}$. In the *classification* case, $t$ is viewed as an abductive explanation for $\boldsymbol{x}$ given $F$ when every instance $\boldsymbol{x}'$ covered by $t$ is classified as $\boldsymbol{x}$ by $F$: $F(\boldsymbol{x}') = F(\boldsymbol{x})$. Thus, any $t$ is (or is not) an abductive explanation for $\boldsymbol{x}$ given $F$ depending on the evaluation of this condition. In the *regression* case, every $t$ is *more or less* an abductive explanation for $\boldsymbol{x}$ given $F$, i.e., the $F$-value of every instance $\boldsymbol{x}'$ covered by $t$ is more or less distant to $F(\boldsymbol{x})$. Accordingly, a notion of imprecision of $t$ can be defined.

**Definition 1.** *Let $F$ be a boosted regression tree over $\mathcal{A}$ and $t$ a term over $\mathcal{B}$. The* imprecision *of $t$ is defined as the length $L_t = M_t - m_t$ of the interval $I_t = [m_t, M_t]$ induced by $t$ and defined by $m_t = min(\{F(\boldsymbol{x}) : \boldsymbol{x} \in \boldsymbol{X}, t \text{ covers } \boldsymbol{x}\})$ and $M_t = max(\{F(\boldsymbol{x}) : \boldsymbol{x} \in \boldsymbol{X}, t \text{ covers } \boldsymbol{x}\})$.*

A monotonic relationship exists between the generality of the terms $t$ over $\mathcal{B}$ that can serve as abductive explanations and the lengths of the corresponding intervals $I_t$:

**Proposition 1.** *Let $F$ be a boosted regression tree over $\mathcal{A}$, and $t$, $t'$ two terms over $\mathcal{B}$. If $t \subseteq t'$ holds then $I_t \supseteq I_{t'}$ holds, so that $L_t \geq L_{t'}$.*

**Generating abductive explanations** We are interested in explaining the values predicted by boosted regression trees using abductive explanations defined as follows:

**Definition 2.** *Let $F$ be a boosted regression tree over $\mathcal{A}$, $\boldsymbol{x} \in \boldsymbol{X}$ an instance, and $I$ an interval over the reals. A term $t$ over $\mathcal{B}$ is*

- *an* abductive explanation *for $\boldsymbol{x}$ given $F$ and $I$ if and only if $t$ covers $\boldsymbol{x}$ and for every instance $\boldsymbol{x}' \in \boldsymbol{X}$ that is covered by $t$, we have $F(\boldsymbol{x}') \in I$.*
- *a* subset-minimal abductive explanation *for $\boldsymbol{x}$ given $F$ and $I$ if and only if $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ and no proper subset of $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$.*

Among the abductive explanations for $\boldsymbol{x}$ is its *direct reason* $t_{\boldsymbol{x}}^F$, defined as the union over the trees $T_i \in F$ of the sets of literals (one per tree $T_i$) containing the literals encountered in the unique path of $T_i$ that is compatible with $t_{\boldsymbol{x}}$. By construction, $t_{\boldsymbol{x}}^F \subseteq t_{\boldsymbol{x}}$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and *any* $I$ containing $F(\boldsymbol{x})$. As every abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ (whatever $I$), $t_{\boldsymbol{x}}^F$ is consistent with $\Sigma$ (otherwise, it would not cover $\boldsymbol{x}$). However, $t_{\boldsymbol{x}}^F$ can be highly redundant (i.e., in general, it is not a subset-minimal abductive explanation for $\boldsymbol{x}$ given $F$ and $I$).

**Example 2.** $t_{\boldsymbol{x}_{ex}} = \{B_1^1, B_2^1, \overline{B_3^1}, \overline{B_1^2}, \overline{B_2^2}, B_3^2, B^3\}$.
*The simplification of $t_{\boldsymbol{x}_{ex}}^F = t_{\boldsymbol{x}_{ex}}$ is $\{B_2^1, \overline{B_3^1}, B_3^2, B^3\}$.*

A standard approach to the generation of a subset-minimal abductive explanation consists of taking advantage of a greedy algorithm. Our greedy algorithm **G** to compute a subset-minimal abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ considers at start the canonical term $t_{\boldsymbol{x}}$ over $\mathcal{B}$ (actually, any term $t$ over $\mathcal{B}$ that covers $\boldsymbol{x}$ can do the job, thus we could start with $t_{\boldsymbol{x}}^F$ or its simplification instead). **G** proceeds as follows. It tries to eliminate successively literals $\ell$ from $t$. Thus, each time $t \setminus \{\ell\}$ still satisfies the coverage condition induced by $I$, $\ell$ is removed from $t$, otherwise it is kept. Accordingly, implementing this approach mainly amounts to deciding the coverage condition. However, this is a computationally hard problem:

**Proposition 2.** *Let $F$ be a boosted regression tree over $\mathcal{A}$, $\boldsymbol{x} \in \boldsymbol{X}$ an instance, and $I$ an interval over the reals. Let $t$ be a term over $\mathcal{B}$. Deciding whether $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ is* coNP*-complete.*

Therefore, **G** uses first an incomplete, yet polynomial-time approximate coverage test, using a method close to the one used for generating (subset-minimal) abductive explanations given a boosted *classification* tree, as described in [Audemard *et al.*, 2023]. When the approximate coverage test succeeds, $\ell$ can be removed from $t$ for sure. When the test fails, $\ell$ is kept even but this does not imply that $t \setminus \{\ell\}$ necessarily violates the exact coverage test. What makes those approximate coverage tests appealing is that they can be achieved very efficiently and that, empirically, they often lead to the removal of many literals from the term $t$ one started with. Then, a second pass of **G** over the remaining literals, using this time expensive, but exact, coverage tests, ensure that the resulting term is a subset-minimal abductive explanation for $\boldsymbol{x}$ given $F$ and $I$. Each exact coverage test is achieved using a constraint-based encoding of the coverage condition and the use of a solver on the encoding. Each of the two passes takes account for the constraint $\Sigma$ about the attributes from $\mathcal{B}$ so that any truth assignment over $\mathcal{B}$ that violates $\Sigma$ is discarded, as in [Gorji and Rubin, 2022]. Notably, **G** exhibits an *anytime behaviour*. At each step after the initialization, the current term $t$ provably is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$. When more time is allocated to the algorithm, the generality of $t$ may only increase and if the algorithm is not interrupted, a subset-minimal abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ is returned. Furthermore, it is guaranteed that the subset-minimal abductive explanation produced by **G** whenever it terminates normally is simplified.

**A constraint-based model for the exact coverage test** Let us now explain how to build a constraint-based model $\mathcal{M}_g$ that can be used to achieve an exact coverage test, i.e., to decide whether a given term $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$. $\mathcal{M}_g$ contains MILP constraints and indicator constraints (which are supported by the solver used, namely CPLEX [Cplex, 2009]). We start by defining a set of MILP constraints over $\mathcal{B}$ encoding the corresponding domain theory $\Sigma$:

$$\begin{array}{l} \forall A_i \in \mathcal{A}_N, \forall j \in [k_i - 1], B^i_j - B^i_{j+1} \geq 0 \\ \forall A_i \in \mathcal{A}_C, \forall B^i_j, B^i_k \in \tau(A_i), j \neq k, B^i_j + B^i_k \leq 1 \end{array} \quad (1)$$

$t$ is represented by the following MILP constraints:

$$\begin{array}{l} \forall B^i_j \in t, B^i_j = 1 \\ \forall \overline{B^i_j} \in t, B^i_j = 0 \end{array} \quad (2)$$

Each tree $T_i$ of $F$ is represented by its set of terms $\{t^i_1, \ldots, t^i_{p_i}\}$, where each term gathers the literals representing the conditions that are true in a root-to-leaf path of $T_i$. Each $t^i_j$ ($j \in [p_i]$) characterizes a unique path of $T_i$ and $w^i_j$ is the real number labelling the leaf of the $j^{th}$ path. With each tree $T_i$ ($i \in [m]$) we associate a set of Boolean variables $\mathcal{L}_i = \{L^i_{t^i_1}, \ldots, L^i_{t^i_{p_i}}\}$, such that $L^i_{t^i_j}$ ($j \in [p_i]$) is true when the conditions given by $t^i_j$ are met. For all $i \in [m]$, the following set of MILP constraints indicates how each $L^i_{t^i_j}$ is connected to the Boolean variables of $\mathcal{B}$:

$$\forall t^i_j \in T_i, \sum_{B^i_j \in t^i_j} B^i_j + \sum_{\overline{B^i_j} \in t^i_j} (1 - B^i_j) - L^i_{t^i_j} \leq |t^i_j| - 1 \quad (3)$$

Because $T_i$ is a decision tree, the terms in $\{t^i_1, \ldots, t^i_{p_i}\}$ are orthogonal (i.e., pairwise inconsistent). This implies that exactly one $L^i_{t^i_j}$ must be set to true, which is ensured by the following set of MILP constraints:

$$\forall i \in [m], \sum_{t^i_j \in T_i} L^i_{t^i_j} = 1 \quad (4)$$

We also consider a set of continuous variables $\mathcal{W} = \{W_1, \ldots, W_m\}$ and some constraints that associate with each tree $T_i$ ($i \in [m]$) the real number $W_i$ that corresponds to the value of $T_i$ for a selected root-to-leaf path made precise by a truth assignment over $\mathcal{L}_i$. Each $W_i$ ($i \in [m]$) is defined by the following MILP constraints:

$$\forall i \in [m], \sum_{j \in [p_i]} L^i_{t^i_j} \times w^i_j = W_i \quad (5)$$

Let $FW$ be a continuous variable that represents the value of the regression tree for any truth assignment over $\mathcal{B}$. The following linear constraint computes $FW$:

$$\sum_{W_i \in \mathcal{W}} W_i = FW \quad (6)$$

Given a non-empty interval $I = (lb, ub)$, a term $t$ over $\mathcal{B}$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ when it is impossible to find a truth assignment over $\mathcal{B}$ that extends $t$ and such that such that $(FW \leq lb)$ or $(FW \geq ub)$ holds. This disjunction is represented using the following constraints, involving two binary variables $IL$ and $IU$ which serve as indicator variables:

$$(IL = 1) \rightarrow (FW \leq lb) \quad (7)$$

$$(IU = 1) \rightarrow (FW \geq ub) \quad (8)$$

$$IL + IU = 1 \quad (9)$$

By construction, $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ if and only if the model $\mathcal{M}_g$ gathering all the constraints above is inconsistent. Note that when $I = \varnothing$, no computation is required since no abductive explanation for $\boldsymbol{x}$ given $F$ and $I$ may exist. Finally, when $I$ is a singleton (e.g., $I = \{F(\boldsymbol{x})\}$), we can use $(F(\boldsymbol{x}) - \epsilon, F(\boldsymbol{x}) + \epsilon)$ as the initial interval, where $\epsilon > 0$ is a preset threshold that is as small as expected.

**Example 3.** *Consider $I_1 = [1750, 2250]$ and $I_2 = [1500, 2500]$, which contain the $F$-value (2000) of $\boldsymbol{x}_{ex}$ = (2200\$, "self-employed", 1).*
*$\boldsymbol{x}_{ex}$ has two subset-minimal abductive explanations given $F$ and $I_1$: $\{B^1_2, B^2_3, B^3\}$ and $\{B^1_2, \overline{B^1_3}, B^2_3\}$. This means that to get an amount of loan granted between 1750\$ and 2250\$ the applicant's incomes must exceed 2000\$, he/she has to be self-employed and married, or the applicant's incomes must*

*exceed 2000$ but not exceed 3000$ and he/she has to be self-employed.*
*$\boldsymbol{x}_{ex}$ has a unique subset-minimal abductive explanation given $F$ and $I_2$: $\{B_2^1\}$. This means that to get an amount of loan granted between 1500$ and 2500$ the applicant's incomes must exceed 2000$.*

**Evaluating abductive explanations** Once an abductive explanation $t$ for $\boldsymbol{x}$ given $F$ and an interval $I$ has been computed (or a candidate $t$ covering $\boldsymbol{x}$ and consistent with $\Sigma$ is pointed out by the human user who wants to get some explanations - aka the explainee [Miller, 2019]), it is interesting to be able to evaluate its imprecision. Indeed, it can be the case that $I \neq I_t$. To be more specific, if $t$ is an abductive explanation $t$ for $\boldsymbol{x}$ given $F$ and $I$, then only $I_t \subseteq I$ is ensured. Computing $I_t$ and its length is thus a valuable approach to determine to which extent the actual imprecision $L_t$ of the explanation $t$ differs from the admissible imprecision considered initially by the explainee (the length of $I$). However, identifying $I_t$ is computationally demanding in general, as a consequence of the following proposition, which is close to Proposition 2 but considers different inputs (no instance $\boldsymbol{x}$ is considered as an input in Proposition 3).

**Proposition 3.** *Let $F$ be a boosted regression tree over $\mathcal{A}$, and $I$ an interval over the reals. Let $t$ be a term over $\mathcal{B}$ such that $t \wedge \Sigma$ is consistent. Deciding whether every instance $\boldsymbol{x} \in \boldsymbol{X}$ covered by $t$ is such that $F(\boldsymbol{x}) \in I$ is* coNP*-complete.*

In our algorithm **E**, the problem of deriving $I_t$ is tackled using binary search (finding the two bounds $m_t$, $M_t$ of $I_t$). At each step, in order to determine whether a given value is an admissible lower bound of $m_t$ (or an upper bound of $M_t$), a constraint-based encoding of the condition is produced and a solver is used to address the corresponding decision problem. To ensure the termination of the search, a preset threshold $\epsilon$ is used when the distance between the two successive values that have been computed is lower than $\epsilon$. By construction, **E** also has an *anytime* behaviour: if **E** is stopped then the interval $I$ given by the current lower (resp. upper) bound of $m_t$ (resp. $M_t$) is such that $I_t \subseteq I$. $I$ can then be viewed as an upper approximation of $I_t$.

**A constraint-based model for finding bounds for $I_t$** Let us explain how to determine $m_t$ or a lower bound of it (identifying $M_t$ or an upper bound of it is similar). Let $\mathcal{M}_e$ be the constraint-based model containing every constraint from $\mathcal{M}_g$, but Equation (9). We consider two variables $lower$ and $lower_b$ such that at each step of the binary search, $m_t$ provably belongs to $[lower_b, lower]$. At start, $lower_b$ is set to $m_F = \sum_{i=1}^{n} min(T_i)$ and $lower$ is set to $F(\boldsymbol{x}_t)$ where $\boldsymbol{x}_t$ is any instance over $\mathcal{B}$ that satisfies $t \wedge \Sigma$. Note that $\boldsymbol{x}_t$ can be computed in linear time from $t$ and $\Sigma$ because $t \wedge \Sigma$ is a 2-CNF formula [Even *et al.*, 1976]. Let $mid = \frac{lower+lower_b}{2}$. If $\mathcal{M}_e \wedge (FW \leq mid)$ is inconsistent, then $mid$ is an acceptable lower bound of $m_t$, so that $lower_b$ can be set to $mid$. In the remaining case when $\mathcal{M}_e \wedge (FW \leq mid)$ is consistent, instead of setting $lower$ to $mid$, $lower$ can be safely set to $FW$, since $FW$ is an upper bound of $m_t$ that is at least as good as $mid$ given that $FW \leq mid$ holds. Then, the binary search resumes using the updated bounds. Clearly, using $FW$ instead of $mid$ in the case when $\mathcal{M}_e \wedge (FW \leq mid)$ is consistent leads to boost the binary search.

**Example 4.** *Here are some terms $t$ over $\mathcal{B}$ covering the instance $\boldsymbol{x}_{ex}$ considered in the running example and the corresponding least intervals $I_t$.*

$$
\begin{array}{ll}
t_1 = \{B_2^1, \overline{B_3^1}, B_3^2, B^3\} & I_{t_1} = [2000, 2000] \\
t_2 = \{\overline{B_3^1}, B_3^2, B^3\} & I_{t_2} = [500, 2000] \\
t_3 = \{B_2^1, B_3^2, B^3\} & I_{t_3} = [2000, 2250] \\
t_4 = \{B_2^1, \overline{B_3^1}, B^3\} & I_{t_4} = [1650, 1650] \\
t_5 = \{B_2^1, \overline{B_3^1}, B_3^2\} & I_{t_5} = [1850, 2250] \\
t_6 = \top & I_{t_6} = [-100, 2500]
\end{array}
$$

*$t_3$ and $t_5$ are the subset-minimal abductive explanations for $\boldsymbol{x}_{ex}$ given $F$ and $I = [1750, 2250]$. We have both $I_{t_3} \subset I$ and $I_{t_5} \subset I$.*

## 4 Empirical Evaluation

The generation algorithm **G** and the evaluation algorithm **E** have been assessed on several datasets in order to figure out the extent to which they are practical.

**Experimental setup** The empirical protocol we considered was as follows. We have focused on 10 datasets for regression, which are standard benchmarks found on the web sites kaggle (https://www.kaggle.com/), UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/index.php) or openML (https://www.openml.org/). These datasets are described in Table 1.

For each dataset, the algorithms `XGBoost` [Chen and Guestrin, 2016] and `LightGBM` [Ke *et al.*, 2017] have been used to learn boosted regression trees. Numerical attributes have been binarized on-the-fly by the boosted tree learning algorithms. Categorical attributes have been one-hot encoded. All the hyper-parameters of the two learning algorithms have been set to their default values (100 trees per forest, with a depth at most 6 for `XGBoost` and a number of leaves at most 31 for `LightGBM`). Thus, no tuning has been performed. Indeed, our purpose is to evaluate the performance of **G** and **E**, whatever the quality of the boosted regression trees we started with. Hence, our experiments have concerned both accurate predictors, and predictors exhibiting rather low accuracies. For each dataset, each boosted tree has been learned

| Name | #A | #N | #C | #B | #I |
|---|---|---|---|---|---|
| Winequality-red | 11 | 0 | 0 | 11 | 1599 |
| Winequality-white | 11 | 0 | 0 | 11 | 4898 |
| CreditcardFraudDet. | 29 | 0 | 0 | 29 | 284807 |
| l4d2-player-stats-final | 112 | 111 | 1 | 0 | 20830 |
| Houses-prices | 46 | 26 | 20 | 0 | 2919 |
| Steel ind. energy cons. | 9 | 6 | 3 | 0 | 35040 |
| Bike sharing: hour | 15 | 13 | 0 | 2 | 17379 |
| Bike sharing: daily | 13 | 11 | 0 | 2 | 731 |
| NASA airfoil self-noise | 5 | 5 | 0 | 0 | 1503 |
| abalone | 9 | 8 | 1 | 0 | 4177 |

Table 1: Description of the datasets used. #A is the number of attributes per instance in the considered dataset. #N, #C, and #B are respectively the number of **n**umerical, **c**ategorical and **B**oolean attributes. #I is the number of instances in the dataset.

| Dataset / Boosted Tree | | | Instance / Direct Reason | | $I^{0.5}_{F,\boldsymbol{x}}$ | | | | | | $I^{2.5}_{F,\boldsymbol{x}}$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Name | R2 | #Cond | $\text{Size}_I$ | $\text{Size}_D$ | $\text{Size}_G$ | $\text{TO}_G$ | $\text{Time}_G$ | %Red | $\text{TO}_E$ | $\text{Time}_E$ | $\text{Size}_G$ | $\text{TO}_G$ | $\text{Time}_G$ | %Red | $\text{TO}_E$ | $\text{Time}_E$ |
| Winequality-red | 0.42 | 666 | 21.5(±0.8) | 21.4(±0.9) | 21.2(±1.2) | 0 | 2.2(±0.3) | 20.9(±23.2) | 0 | 0.8(±0.0) | 19.9(±1.5) | 0 | 2.5(±0.3) | 18.6(±12.1) | 0 | 0.9(±0.1) |
| Winequality-white | 0.46 | 764 | 21.8(±0.6) | 21.8(±0.6) | 21.4(±0.8) | 0 | 2.2(±0.2) | 20.2(±16.8) | 0 | 0.8(±0.0) | 19.9(±1.2) | 0 | 2.5(±0.3) | 18.1(±12.1) | 0 | 0.9(±0.1) |
| CreditcardFraudDet. | 0.96 | 1506 | 56.0(±1.8) | 39.9(±2.0) | 25.7(±3.9) | 4 | 216(±244) | 8.9(±6.0) | 22 | 258(±301) | 18.8(±2.4) | 90 | 770(±207) | 25.1(±11.8) | 97 | 806(±97.6) |
| l4d2-player-stats-final | -2.62 | 1378 | 171(±18.9) | 95.4(±10.9) | 56.2(±7.1) | 98 | 4.8(±0.3) | 18.1(±7.6) | 96 | 172(±175) | 15.6(±2.6) | 100 | - | 52.2(±7.9) | 100 | - |
| Houses-prices | 0.88 | 897 | 61.8(±6.8) | 62.2(±8.0) | 49.5(±8.6) | 0 | 5.9(±1.0) | 5.6(±4.7) | 0 | 2.2(±0.7) | 33.9(±10.1) | 0 | 72.2(±132) | 2.6(±2.3) | 0 | 52.0(±98.3) |
| Steel ind. energy cons. | 0.99 | 533 | 13.5(±2.4) | 11.2(±1.5) | 7.0(±2.0) | 0 | 1.5(±0.2) | 36.8(±15.7) | 0 | 1.0(±0.2) | 5.0(±1.5) | 0 | 1.6(±0.3) | 44.6(±17.7) | 0 | 1.9(±0.5) |
| Bike sharing: hour | 0.99 | 603 | 20.0(±2.0) | 11.2(±2.1) | 3.7(±0.7) | 0 | 1.5(±0.3) | 42.6(±19.2) | 0 | 1.4(±0.2) | 3.2(±0.7) | 0 | 1.4(±0.2) | 33.0(±13.1) | 0 | 2.0(±0.3) |
| Bike sharing: daily | 0.99 | 618 | 19.3(±1.2) | 18.1(±1.4) | 10.9(±1.6) | 0 | 1.8(±0.5) | 16.9(±9.1) | 0 | 1.5(±0.8) | 4.0(±0.8) | 0 | 4.0(±3.5) | 30.7(±10.6) | 0 | 61.9(±65.1) |
| NASA airfoil self-noise | 0.92 | 149 | 8.8(±0.8) | 8.8(±0.8) | 8.5(±1.0) | 0 | 1.2(±0.3) | 68.5(±21.0) | 0 | 0.8(±0.0) | 7.3(±1.1) | 0 | 1.2(±0.3) | 48.1(±18.8) | 0 | 1.0(±0.2) |
| abalone | 0.99 | 456 | 17.0(±0.3) | 16.8(±0.7) | 2.1(±0.6) | 0 | 1.6(±0.4) | 90.1(±8.3) | 0 | 0.9(±0.0) | 2.0(±0.1) | 0 | 1.5(±0.5) | 97.5(±8.1) | 0 | 1.1(±0.2) |

Table 2: Statistics about the computations of our algorithms on boosted regression trees learned using `LightGBM`.

from a training set containing 80% of the dataset, and its accuracy was measured as its mean $R2$ score [Ling and Kenny, 1981] over the remaining 20% of the dataset.

In order to evaluate **G**, we needed to consider intervals $I$. To this purpose, for each dataset and each boosted tree $F$, we have first estimated the range of values that can be reached by $F = \{T_1, \ldots, T_m\}$. This estimate is given by the interval $I_F = [m_F, M_F]$ where $M_F = \sum_{i=1}^{n} max(T_i)$ echoes $m_F = \sum_{i=1}^{n} min(T_i)$. $I_F$ can be computed in time linear in $|F|$. Note that while the image $F(\boldsymbol{X})$ and its superset given by the interval $I_\top$ are guaranteed to be included in $I_F$,[1] interval $I_F$ does not coincide with $I_\top$ in general[2] (the minimal or maximal leaves over the trees of $F$ may easily correspond to incompatible paths). The length $L_F = M_F - m_F$ of $I_F$ gave us a range of values that can be used to define intervals $I$ of various lengths containing $F(\boldsymbol{x})$ (whatever $\boldsymbol{x}$) and reflecting various imprecision levels about the $F$-predicted value of $\boldsymbol{x}$. Thus, for any $\boldsymbol{x} \in \boldsymbol{X}$ and any $r \in [0, 100]$, we defined

$$I^r_{F,\boldsymbol{x}} = [F(\boldsymbol{x}) - (\frac{r}{100} \cdot L_F), F(\boldsymbol{x}) + (\frac{r}{100} \cdot L_F)].$$

Centered intervals $I^r_{F,\boldsymbol{x}}$ have been considered for the ease of empirical protocol, only.[3] Indeed, our algorithm **G** can take any interval $I$ as input, provided that $F(\boldsymbol{x}) \in I$. Then, for each dataset, each boosted tree $F$ and a pool of 100 instances $\boldsymbol{x} \in \boldsymbol{X}$ drawn uniformly at random from the test set, we have computed the simplification of the direct reason $t^F_{\boldsymbol{x}}$ and we have run **G** in order to derive a subset-minimal abductive explanation $t$ for $\boldsymbol{x}$ given $F$ and $I^r_{F,\boldsymbol{x}}$ where $r \in \{0.5, 1, 2.5, 5, 10\}$. We took advantage of the CPLEX solver [Cplex, 2009] in this computation.

Finally, in order to assess the performance of **E**, we started from the subset-minimal abductive explanations $t$ for $\boldsymbol{x}$ given $F$ and $I^r_{F,\boldsymbol{x}}$ (where $r \in \{0.5, 1, 2.5, 5, 10\}$) that have been computed by **G**, and derived from them using the evaluation algorithm the corresponding intervals $I_t$ and we measured their lengths. Using the terms $t$ computed by **G** in our experiments was a natural choice, but other alternatives were possible (the two algorithms **E** and **G** are completely independent and they can be used separately). The parameter $\epsilon$ used in **E** has been set to 0.01 in our experiments.

For each dataset, each boosted tree $F$, we counted the number of instances $\boldsymbol{x} \in \boldsymbol{X}$ (out of 100) for which a subset-minimal abductive explanation has been computed in due time given $F$ and each of the 5 intervals $I^r_{F,\boldsymbol{x}}$ that have been considered (a time-out (TO) of 900s has been considered per instance and interval). For each instance for which the computation has been successful, we measured the time needed to get the result and the size of the resulting subset-minimal abductive explanation $t$. Then we counted the number of resulting terms $t$ for which the evaluation algorithm terminates in due time (using a TO of 600s for each bound). We measured the time required to derive $I_t$, and the length $L_t$ of this interval in order to compare it with the length $\frac{2r}{100} \cdot L_F$ of the interval $I^r_{F,\boldsymbol{x}}$ that has been considered when computing $t$. Especially, we measured the reduction of the imprecision that is obtained. This reduction is defined by 0 when $L_F = 0$ and by $\frac{\frac{2r}{100} \cdot L_F - L_t}{\frac{2r}{100} \cdot L_F}$ in the remaining case. 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.

**Experimental results** Table 2 presents some empirical results obtained for the ten datasets that have been considered. The leftmost column of Table 2 gives the name of the dataset $b$. Columns R2, #Cond, $\text{Size}_I$ and $\text{Size}_D$ give, respectively, the $R2$ score of the boosted tree $F$ learned using `LightGBM`, the number of distinct Boolean conditions occurring in it, the mean and standard deviation of the lengths of the simplified instances over $\mathcal{B}$, and finally, the mean and standard deviation of the lengths of the associated simplified direct reasons. Then, for each interval $I^r_{F,\boldsymbol{x}}$ considered ($r = 0.5$ and $r = 2.5$ in Table 2), we report the mean and standard deviation of the sizes of the abductive explanations $t$ (that have been computed using **G**), and the mean and standard deviation of the reductions of the interval considered for deriving $t$ (the reductions are computed using **E**), the number of timeouts and the mean time and standard deviation of the computation times (in seconds) when the corresponding algorithm (**G** or **E**) terminated in due time.[4]

In light of the experiments, **G** and **E** appear as practical for boosted trees involving a significant number of Boolean

[1] Let us recall that $\top$ denotes the empty conjunction of literals.

[2] Note that, unlike what happens in the general case, we have $I_F = I_\top$ for the running example.

[3] Instead of $I^r_{F,\boldsymbol{x}}$, we could have considered intervals generated from prediction interval estimators (see Section 5).

[4] For space reasons, more detailed results including results about the boosted trees learned using `XGBoost` and/or concerning other values of $r$ are not reported here, but they can be found in the supplementary material.

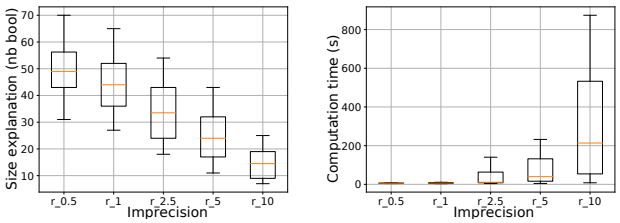


Figure 2: Empirical results about algorithm **G** on the *houses-prices* dataset.

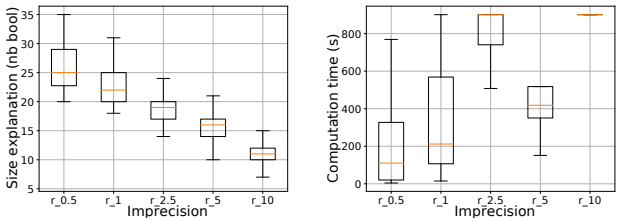


Figure 3: Empirical results about algorithm **G** on the *creditcard* dataset.

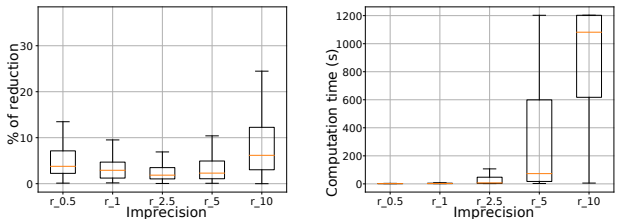


Figure 4: Empirical results about algorithm **E** on the *houses-prices* dataset.

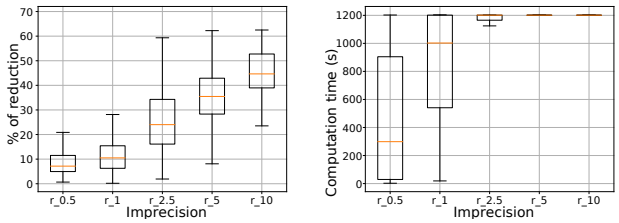


Figure 5: Empirical results about algorithm **E** on the *creditcard* dataset.

conditions (up to 800). This corresponds to the boosted trees computed using `LightGBM` on every dataset but the larger ones, namely *l4d2-player-stats-final*, *creditCardFraudDetection*, and *houses-prices*. Thus, for the seven other datasets, no timeout has occurred during the computations and the mean computation times for deriving a subset-minimal abductive explanation for any of the instance (out of 100) never exceeded 90 seconds (and most of the time, only a couple of seconds was required). A valuable observation is that both **G** and **E** provide useful outputs even when they are interrupted before a normal termination.

In practice, the sizes of the explanations generated using **G** can be much smaller than the sizes of the instances they explain, and much smaller than the sizes of the corresponding direct reasons. For instance, considering the line associated with the dataset *bike sharing: daily* and the columns $\text{Size}_I$, $\text{Size}_D$ and $\text{Size}_G$ (for $r = 0.5$) in Table 2, we can check that the subset-minimal abductive explanations that have been computed are of average size 3.7, while the corresponding instances are of average size 20 and their direct reasons are of average size 11.2. Table 2 also shows that the the reduction of the imprecision that is achieved by **E** is significant most of the time.

Our experiments have also permitted to assess on a qualitative, yet empirical basis the connections between the generality of the explanations (given by their sizes) and the imprecision considered at start for generating them using **G**. We illustrate them on two datasets: *houses-prices* and *creditcard*. Figure 2 (left) (resp. Figure 3 (left)) gives box plots synthesizing the distribution of sizes of the subset-minimal abductive explanations that have been generated for various values of $r$ for the boosted tree trained on *houses-prices* (resp. *creditcard*) using `LightGBM`. Figure 2 (right) (resp. Figure 3 (right)) is about the corresponding computation times. Figure 4 and Figure 5 report similar results about the reductions of the intervals considered as input by **G** (left) and the corresponding computation times (right).

We can observe on Figure 2 (left) that the sizes of the explanations significantly decrease when the admissible imprecision increases. Contrariwise, the computation times (see Figure 2 (right)) increase when the admissible imprecision increases (more literals must be removed by the greedy generation algorithm to get subset-minimal abductive explanations).

Figure 4 (left) and Figure 5 (left) show that the reduction of the imprecision that is achieved by **E** can be important, and Figure 4 (right) and Figure 5 (right) show that the computation times typically increase when the generality of the term considered at start increases (this can be easily explained by the fact that the number of instances covered by the term increases as well). For each of **G** and **E**, we observed that the standard deviation of the computation times is sometimes high (larger than the mean). Similar observations can be done for the other datasets.[5]

## 5 Other Related Work

Most existing work about XAI for regression exploits solutions developed for the classification task, reducing regression into (multi-class) classification. This is typically achieved through a partitioning of the reals into intervals, requiring the elicitation of decision boundaries that can be more or less arbitrary. Once done, popular XAI techniques for classification can be leveraged. Thus, [Strumbelj and Kononenko, 2011] presents an approach to explanation based on the notion of feature importance. [Kontokosta, 2019;

[5] Additional box plots are provided in the supplementary material.

Moore and Bell, 2022] use Shapley values and take advantage of them in the context of two applications (energy efficiency of buildings and prediction of myocardial infarction). What makes such XAI approaches appealing is that they are model-agnostic and scalable, so they can be used to explain predictions achieved by very powerful ML models (e.g., deep neural nets). The (heavy) price to be paid is that they do not offer any formal insurance of rigor [Ignatiev, 2020].

Contrariwise, our approach to XAI is specific to boosted regression trees. However, the (subset-minimal) abductive explanations that are generated by **G** and evaluated by **E** are provably correct. This correctness guarantee comes from the logical setting considered for representing boosted trees and the use of automated reasoning tools for deriving (and evaluating) explanations. This makes our work relevant to *formal XAI* [Marques-Silva and Ignatiev, 2022].

Prior work in formal XAI that have been concerned with tree ensembles have mainly focused on the computation of contrastive explanations (see, for instance, [Cui *et al.*, 2015; Kanamori *et al.*, 2020; Parmentier and Vidal, 2021; Hada and Carreira-Perpiñán, 2021]) and on the classification issue (see, for instance, [Choi *et al.*, 2020; Izza and Marques-Silva, 2021; Audemard *et al.*, 2022; Ignatiev *et al.*, 2019b; 2022]). Thus, they are significantly different from our own work, centered on abductive explanations for regression.

Estimating prediction intervals is a well-established topic in Machine Learning (or even in traditional statistics), and is widely used in practice. A number of approaches for estimating prediction intervals have been pointed out so far (e.g., via quantile regression [Koenker, 2005]) and they are based a variety of techniques (e.g., the jacknife method [Barber *et al.*, 2021]). The main goal is to measure the robustness of the predictor and to control the use of the predictor by exploiting statistical guarantees. Accordingly, those approaches are focused on instances, not on explanations for instances, while our algorithm **E** is about the evaluation of abductive explanations, with logic-based guarantees, which is quite a different perspective. Note nevertheless that the output of prediction interval estimators could be used to define intervals used as inputs by our algorithm **G**.

Finally, [Letzgus *et al.*, 2022] identifies important conditions about the regression problem that should be considered when developing dedicated XAI approaches, but are not always guaranteed by XAI approaches to classification. One of them states that explanations should be produced relative to some reference value. Our algorithm **G** takes such a reference value into account, represented by an interval $I$. As we have shown both in theory and in practice, the choice of $I$ may substantially affect the subset-minimal abductive explanations that are derived.

## 6 Conclusion

We have presented and assessed two anytime algorithms **G** and **E** for generating (resp. evaluating) abductive explanations for boosted regression trees. The datasets used for learning the boosted trees can be based on data of mixed type (including categorical and numerical attributes). This leads to boosted trees containing Boolean conditions that are not independent in general. In our approach, the underlying domain theory is used to ensure that the abductive explanations that are generated are not unnecessarily specific, but also to simplify the explanations. Most of the time, our algorithms can be used to generate (resp. evaluate) in a few seconds abductive explanations for boosted regression trees based on a large number of Boolean conditions (up to $800$). A valuable observation is that, in general, the (subset-minimal) abductive explanations that are generated using **G** are significantly smaller than the initial descriptions of the instances in terms of Boolean attributes. Furthermore, the reduction of the imprecision that is achieved by **E** can be very significant as well.

Notably, the correctness of **G** (resp. **E**) does not require any specific assumption on the way the ensemble of regression trees $F$ used as input has been learned. Thus, **G** and **E** are applicable to general regression forests, and not only to boosted regression trees, and provide the same guarantees for general regression forests as the ones offered for boosted regression trees. Especially, it does not matter if bagging has been used instead of boosting as an ensemble learning method, so that $F$ actually is a random forest. Note nevertheless that for random forests, the computation of a prediction interval that could be used as an input of **G** is much easier than for boosted trees, since trees in a random forest are independently generated.

As illustrated by our experiments, it can be the case that the size of the explanations produced by **G** is quite large. In such a case, pieces of knowledge (when available) can be leveraged to try to simplify the explanations further. However, it may happen that simplified explanations are still too large to be viewed as intelligible enough by the explainee. It is important to keep in mind that such a situation does not reflect a drawback of the XAI approach (which aims at explaining the behaviour of the predictor as it is, and not as it could be), but a feature of the predictor itself. Indeed, the fact that the explanations are large indicates that many attributes are needed to explain the regression value that has been computed. Knowing it, the explainee is free to decide what to do with the value and the predictor (trust in it or not).

This work calls for a number of perspectives. One of them is to focus on a specific application where the expertise of a human user can be exploited to assess the quality of the explanations that are generated. We are confident that the possibility of computing $I_t$ given $t$ and $F$ can be leveraged to design interaction protocols with an explainee, in the objective of providing explanations that achieve a good generality/precision trade-off and fitting the expectations of the explainee [Doshi-Velez and Kim, 2017; Narayanan *et al.*, 2018]. An example of such a process (with the explainee-in-the-loop) would be as follows: starting from $\boldsymbol{x}$ and an interval $I$ furnished by the explainee, one first computes a subset-minimal explanation $t$ for $\boldsymbol{x}$ and $I$ using **G** and then evaluates it using **E** by computing $I_t$. If the explainee finds $t$ too specific, then one asks her/him for a subset of literals $t'$ to be removed from $t$ in order to make $t$ more general and evaluate the term $t \setminus t'$. If the precision $L_{t \setminus t'}$ is fine with the explainee, we can stop the interaction and return $t \setminus t'$. If it is deemed too large, one can resume at the first step and look for another subset-minimal explanation for $\boldsymbol{x}$ and $I$.

## Acknowledgments

Many thanks to the anonymous reviewers for their comments and insights. 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

[Adadi and Berrada, 2018] A. Adadi and M. Berrada. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[Arrieta *et al.*, 2020] 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.

[Audemard *et al.*, 2022] 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, 2022.

[Audemard *et al.*, 2023] G. Audemard, J.-M. Lagniez, P. Marquis, and N. Szczepanski. Computing abductive explanations for boosted trees. In *Proc. of the 26th International Conference on Artificial Intelligence and Statistics, AISTATS'23*, volume 206 of *Proce. of Machine Learning Research*, pages 4699–4711, 2023.

[Barber *et al.*, 2021] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Predictive inference with the jackknife+. *The Annals of Statistics*, 49(1):486–507, 2021.

[Caruana *et al.*, 2020] 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.

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

[Choi *et al.*, 2020] 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.

[Cplex, 2009] IBM ILOG Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.

[Cui *et al.*, 2015] Z. Cui, W. Chen, Y. He, and Y. Chen. Optimal action extraction for random forests and boosted trees. In *Proc. of KDD'15*, pages 179–188, 2015.

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

[Doshi-Velez and Kim, 2017] F. Doshi-Velez and B. Kim. A roadmap for a rigorous science of interpretability. *CoRR*, abs/1702.08608, 2017.

[Even *et al.*, 1976] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multicommodity flow problems. *SIAM J. Comput.*, 5(4):691–703, 1976.

[Gorji and Rubin, 2022] N. Gorji and S. Rubin. Sufficient reasons for classifier decisions in the presence of domain constraints. In *Proc. of AAAI'22*, pages 5660–5667, 2022.

[Hada and Carreira-Perpiñán, 2021] S.S. Hada and M. Carreira-Perpiñán. Exploring counterfactual explanations for classification and regression trees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 489–504, 2021.

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

[Ignatiev *et al.*, 2019b] A. Ignatiev, N. Narodytska, and J. Marques-Silva. On validating, repairing and refining heuristic ML explanations. *CoRR*, abs/1907.02509, 2019.

[Ignatiev *et al.*, 2022] 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.

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

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

[Kanamori *et al.*, 2020] K. Kanamori, T. Takagi, K. Kobayashi, and H. Arimura. DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In *Proc. of IJCAI'20*, pages 2855–2862, 2020.

[Ke *et al.*, 2017] 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.

[Koenker, 2005] R. Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

[Kontokosta, 2019] S. Papadopoulosand C. Kontokosta. Grading buildings on energy performance using city benchmarking data. *Applied Energy*, 233:244–253, 2019.

[Letzgus *et al.*, 2022] S. Letzgus, P. Wagner, J. Lederer, W. Samek, K.R. Müller, and G. Montavon. Toward explainable artificial intelligence for regression models: A methodological perspective. *IEEE Signal Process. Mag.*, 39(4):40–58, 2022.

[Ling and Kenny, 1981] R. Ling and D. Kenny. Correlation and causation. *Journal of the American Statistical Association*, 77:489, 1981.

[Lipton, 2018] Z. C. Lipton. The mythos of model interpretability. *Communications of the ACM*, 61(10):36–43, 2018.

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

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

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

[Moore and Bell, 2022] A. Moore and M. Bell. XGBoost, a novel explainable AI technique, in the prediction of myocardial infarction, a UK biobank cohort study. *medRxiv*, 2022.

[Narayanan *et al.*, 2018] 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.

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

[Rudin *et al.*, 2021] 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.

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

[Strumbelj and Kononenko, 2011] E. Strumbelj and I. Kononenko. A general method for visualizing and explaining black-box regression models. In *Adaptive and Natural Computing Algorithms ICANNGA*, volume 6594 of *LNCS*, pages 21–30, 2011.

[Xu *et al.*, 2019] F. Xu, H. Uszkoreit, Y. Du, W. Fan, D. Zhao, and J. Zhu. Explainable AI: A brief survey on history, research areas, approaches and challenges. In *Proc. of NLPCC'19*, pages 563–574, 2019.

## Proofs

### Proof of Proposition 1

*Proof.* The result comes directly from the fact that if $S$ and $S'$ are two subsets of $\boldsymbol{X}$ such that $S \subseteq S'$ (here $S$ is the set of instances of $\boldsymbol{X}$ covered by $t$ and $S'$ is the set of instances of $\boldsymbol{X}$ covered by $t'$) then $\{F(\boldsymbol{x}) : \boldsymbol{x} \in S\} \subseteq \{F(\boldsymbol{x}) : \boldsymbol{x} \in S'\}$. As a consequence,

$$min(\{F(\boldsymbol{x}) : \boldsymbol{x} \in S'\}) \leq min(\{F(\boldsymbol{x}) : \boldsymbol{x} \in S\})$$

and

$$max(\{F(\boldsymbol{x}) : \boldsymbol{x} \in S'\}) \geq max(\{F(\boldsymbol{x}) : \boldsymbol{x} \in S\}).$$

□

### Proof of Proposition 2

*Proof.* The problem we consider can be stated formally as the following decision problem $\mathrm{ABD}_r$:

- **Input:** A term $t$ over $\mathcal{B}$, a boosted tree $F$ over a set $\mathcal{A}$ of attributes, an instance $\boldsymbol{x}$ over $\mathcal{A}$ and an interval $I$ over the reals.
- **Question:** It $t$ an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$?
- Membership to coNP: we consider the complementary problem $\overline{ABD_r}$ and show that it belongs to NP. In order to determine whether or not $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$, we first test whether $t$ covers $\boldsymbol{x}$ in (deterministic) linear time. If not, we conclude that $t$ is not an abductive explanation for $\boldsymbol{x}$ given $F$ and $I$. Otherwise, we guess an instance $\boldsymbol{x}' \in \boldsymbol{X}$ and check that $t$ covers $\boldsymbol{x}'$ and that $F(\boldsymbol{x}') \notin I$. Since $F(\boldsymbol{x}')$ can be computed in time linear in the size of $F$ and the size of an instance, the conclusion follows.
- coNP-hardness: in the binary classification case, it has been shown in [Audemard *et al.*, 2023] (Proposition 1) that the following decision problem $\mathrm{ABD}_c$ is coNP-hard:
  - **Input:** A term $t$, a boosted tree $F$, and an instance $\boldsymbol{x}$ over a set $\mathcal{A}$ of Boolean attributes.
  - **Question:** It $t$ an abductive explanation for $\boldsymbol{x}$ given $F$?

  $\mathrm{ABD}_c$ is closely related to $\mathrm{ABD}_r$. Thus, when $\mathcal{A}$ contains only Boolean attributes, we can assume that $\mathcal{B} = \mathcal{A}$ so that every term $t$ over $\mathcal{A}$ also is a term over $\mathcal{B}$. By definition, $t$ is an abductive explanation for $\boldsymbol{x}$ given $F$ if and only if every instance $\boldsymbol{x}' \in \boldsymbol{X}$ covered by $t$ is such that $F(\boldsymbol{x}') > 0$ when $F(\boldsymbol{x}) > 0$ and $F(\boldsymbol{x}') \leq 0$ when $F(\boldsymbol{x}) \leq 0$. $F(\boldsymbol{x})$ can be computed in time linear in the size of $F$ and the size of an instance. The linear-time reduction from $\mathrm{ABD}_c$ to $\mathrm{ABD}_r$ that we consider associates with any instance $(t, F, \boldsymbol{x})$ of $\mathrm{ABD}_c$ the instance $(t, F, \boldsymbol{x}, I)$ of $\mathrm{ABD}_r$ where $I = (0, +\infty)$ when $F(\boldsymbol{x}) > 0$ and $I = (-\infty, 0]$ when $F(\boldsymbol{x}) \leq 0$. Thus, when $F(\boldsymbol{x}) > 0$, for any instance $\boldsymbol{x}' \in \boldsymbol{X}$, we have $F(\boldsymbol{x}') > 0$ if and only if $F(\boldsymbol{x}') \in (0, +\infty)$, while in the remaining case when $F(\boldsymbol{x}) \leq 0$, we have $F(\boldsymbol{x}') \leq 0$ if and only if $F(\boldsymbol{x}') \in (-\infty, 0]$. Accordingly, $(t, F, \boldsymbol{x})$ is a positive instance of $\mathrm{ABD}_c$ if and only if the corresponding instance $(t, F, \boldsymbol{x}, I)$ of $\mathrm{ABD}_r$ is positive as well. This concludes the proof.

□

### Proof of Proposition 3

*Proof.* The problem we consider can be stated formally as the following decision problem $\mathrm{EVA}_r$:

- **Input:** A term $t$ over $\mathcal{B}$, a boosted tree $F$ over a set $\mathcal{A}$ of attributes, and an interval $I$ over the reals.
- **Question:** Is every instance $\boldsymbol{x} \in \boldsymbol{X}$ covered by $t$ such that $F(\boldsymbol{x}) \in I$?
- Membership to coNP: we consider the complementary problem $\overline{EVA_r}$ and show that it belongs to NP. In order to show that there exists an instance $\boldsymbol{x} \in \boldsymbol{X}$ covered by $t$ such that $F(\boldsymbol{x}) \notin I$, it is enough to guess $\boldsymbol{x}$, to compute $F(\boldsymbol{x})$ in time linear in the size of $F$ and the size of an instance, and to verify that $F(\boldsymbol{x}) \notin I$.
- coNP-hardness: To prove that $\mathrm{EVA}_r$ is coNP-hard, we can use a polynomial-time reduction from $\mathrm{ABD}_c$ to $\mathrm{EVA}_r$ that is very similar to the one reported in the proof of Proposition 2. Consider an instance $(t, F, \boldsymbol{x})$ of $\mathrm{ABD}_c$ and let us associate with it in linear time the instance $(t, F, I)$ of $\mathrm{EVA}_r$ where $I = (0, +\infty)$ when $F(\boldsymbol{x}) > 0$ and $I = (-\infty, 0]$ when $F(\boldsymbol{x}) \leq 0$. When $F(\boldsymbol{x}) > 0$, for any instance $\boldsymbol{x}' \in \boldsymbol{X}$, we have $F(\boldsymbol{x}') > 0$ if and only if $F(\boldsymbol{x}') \in (0, +\infty)$, while in the remaining case when $F(\boldsymbol{x}) \leq 0$, we have $F(\boldsymbol{x}') \leq 0$ if and only if $F(\boldsymbol{x}') \in (-\infty, 0]$. Accordingly, $(t, F, \boldsymbol{x})$ is a positive instance of $\mathrm{ABD}_c$ if and only if the corresponding instance $(t, F, I)$ of $\mathrm{EVA}_r$ is positive as well.

□