On the Explanatory Power of Boolean Decision Trees

Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche,
Jean-Marie Lagniez, Pierre Marquis

On the Explanatory Power of Boolean Decision Trees*

Gilles Audemard[1], Steve Bellart[1], Louenas Bounia[1], Frédéric Koriche[1], Jean-Marie Lagniez[1] and Pierre Marquis[1,2]

1. Univ. Artois, CNRS, CRIL, F-62300 Lens, France
2. Institut Universitaire de France

Abstract

Decision trees have long been recognized as models of choice in sensitive applications where interpretability is of paramount importance. In this paper, we examine the computational ability of Boolean decision trees for the explanation purpose. We focus on both abductive explanations (suited to explaining why a given instance has been classified as such by the decision tree at hand) and on contrastive explanations (suited to explaining why a given instance has not been classified by the decision tree as it was expected). More precisely, we are interested in deriving, minimizing, and counting abductive explanations and contrastive explanations. We prove that the set of all irredundant abductive explanations (also known as PI-explanations or sufficient reasons) for an instance given a decision tree can be exponentially larger than the size of the input (the instance and the decision tree). Therefore, generating the full set of sufficient reasons for an instance can be out of reach. In addition, deriving a single sufficient reason, though computationally easy when dealing with decision trees, does not prove enough in general; indeed, two sufficient reasons for the same instance may differ on many features. To deal with this issue and generate synthetic views of the set of all sufficient reasons, we define notions of relevant features and of necessary features that characterize the (possibly negated) features appearing in at least one or in every sufficient reason for an instance, and we show that they can be computed in polynomial time. We also introduce the notion of explanatory importance, that indicates how frequent each (possibly negated) feature is in the set of all sufficient reasons. We show how the explanatory importance of a (possibly negated) feature and the number of sufficient reasons for an instance can be obtained via a model counting operation, which turns out to be practical in many cases. We also explain how to enumerate minimum-size sufficient reasons. We finally show that, unlike sufficient reasons, the set of all contrastive explanations for an instance given a decision tree can be derived, minimized and counted in polynomial time.

*This is an extended and revised version of a selected paper (in French) from EGC'22.

1 Introduction

In essence, explaining a decision is to give the details or *reasons* that help the person who asked for an explanation (known as the explainee) (1) understand why the decision has been made. The explanation issue is of tremendous significance, especially when decisions are predictions made by Machine Learning (ML) classifiers. For such AI systems, with any data instance $\boldsymbol{x}$ considered at input, the ML model f outputs a decision that is a predicted class $f(\boldsymbol{x})$. When dealing with binary classifiers, which is what we do in this paper, two classes are possible: 1 for the instances classified as positive, and 0 for the negative ones.

Unsurprisingly, with the growing number of applications that rely on ML techniques, researches on eXplainable AI (XAI) have become increasingly important (see for instance (1; 2; 3; 4; 5; 6; 7; 8; 9; 10)). Actually, ML models with high prediction performance are often considered as poorly explainable (11; 12; 13; 14), and when the ability of delivering explanations is critical, a trade-off between the accuracy of the model and its explainability must be looked for (12; 15). Especially, most approaches to XAI that deal with black-box classifiers are of heuristic nature. They typically focus on a surrogate model instead of the black-box classifier at hand. As a consequence, the explanations of the predictions that are generated can be at the same time relevant to the surrogate model and irrelevant to the black-box classifier itself (see e.g., (16; 17)). Such approaches cannot be used in any high-risk context since they deliver model-agnostic explanations that can be *unsound* (18). This means that one can find *counterexamples* $\boldsymbol{x}'$ for such explanations. More precisely, given an instance $\boldsymbol{x}$ and an explanation for the prediction made $f(\boldsymbol{x})$, there may exist other instances $\boldsymbol{x}'$ sharing the same explanation as the one for $\boldsymbol{x}$ but such that $f(\boldsymbol{x}') \neq f(\boldsymbol{x})$ (19). When this happens, the "explanation" that is computed can hardly be considered as explaining the prediction $f(\boldsymbol{x})$ since the same explanation would also work for a distinct prediction $f(\boldsymbol{x}')$. In particular, the explainee cannot take advantage of such explanations to *derive sound conclusions* about the predictions made by the classifier. Unfortunately, such a scenario is not unfrequent. Indeed, it has been shown in (20) that the amount of counterexamples can be high when using some of the most popular approaches for computing model-agnostic explanations, namely LIME (21), Anchors (22), and SHAP (8).

Whatever the way $\boldsymbol{x}$ has been classified (positive or negative), an explainee may seek for explanations from two distinct types (1). On the one hand, abductive explanations (23)[1] for $\boldsymbol{x}$ are intended to explain why $\boldsymbol{x}$ has been classified in the way it has been classified by the ML model (thus, addressing the "Why?" question); on the other hand, the purpose of contrastive (also known as counterfactual) explanations for $\boldsymbol{x}$ is to explain why $\boldsymbol{x}$ has not been classified by the ML model as the explainee expected it (thus, addressing the "Why not?" question) (24). In both cases, explanations that are as simple as possible are often preferred (where simplicity is modeled as irredundancy, or even as size minimality). Clearly, every instance has an abductive explanation and a

[1]Unlike (24), we do not require abductive explanations to be always minimal w.r.t. set inclusion. The notion of abductive explanations considered here correspond to so-called weak abductive explanations in (25).

contrastive explanation[2] given a binary classifier. In particular, every ML model associates with any given instance $\boldsymbol{x}$ an abductive explanation, referred as the *direct reason* for $\boldsymbol{x}$, that is induced from the way the model classifies $\boldsymbol{x}$. This direct reason may coincide with $\boldsymbol{x}$ (in that case, it is not so helpful), but it may also be simpler than $\boldsymbol{x}$, as it is the case for decision trees.

Although there is no consensual view of what of *interpretability* means (26), *decision trees* (27; 28) are arguably among the most interpretable ML models for classification problems. Because of their interpretability, decision trees are often considered as target models for distilling black-box models into more comprehensible ones (29; 2). Furthermore, decision trees are often the components of choice for building (less interpretable, but potentially more accurate) ensemble classifiers, such as random forests (30) and gradient boosted trees (31).

The interpretability of decision trees is mainly endowed with two key characteristics. On the one hand, decision trees are *transparent*: each node in a decision tree has some meaning, and the principles used for generating all nodes can be explained. On the other hand, decision trees are *locally explainable*: by construction of a decision tree T, any input instance $\boldsymbol{x}$ is mapped to a unique root-to-leaf path that yields to a decision label. The subset of (positive and negative) features $t_{\boldsymbol{x}}^T$ occurring in the path used to find the right label 1 or 0 (in the binary classification case) for $\boldsymbol{x}$ in the decision tree T is called the path-restricted explanation for $\boldsymbol{x}$ (32), and it is the direct reason for classifying $\boldsymbol{x}$ as a positive instance or as a negative instance given T. Notably, $t_{\boldsymbol{x}}^T$ is an abductive explanation for $\boldsymbol{x}$ given T, which explains why $\boldsymbol{x}$ has been classified by T as it has been classified. Indeed, every instance $\boldsymbol{x}'$ that coincides with $\boldsymbol{x}$ on $t_{\boldsymbol{x}}^T$ is classified by T in the same way as $\boldsymbol{x}$. However, such direct reasons can contain arbitrarily many redundant features (32). This motivates to take account for other types of abductive explanations in the case of decision trees, namely, sufficient reasons (33) (also known as prime implicant explanations (34)), that are irredundant abductive explanations, and among them, minimum-size sufficient reasons.

Beyond these characteristics, the interpretability of decision trees can be assessed in a more formal way by focusing on a set of XAI queries of interest. In such a setting, an ML model is said to offer a given XAI query when there exists a polynomial-time algorithm for answering the query given the model. The more XAI queries offered, the more interpretable the ML model. Following this line of research, (35) points out a number of XAI queries (including both explanation and verification queries). As to explanation queries, the authors consider the issue of generating a sufficient reason and the issue of generating a minimum-size contrastive explanation. They identify a number of conditions about the representation of the Boolean classifier f at hand that prove sufficient to ensure that the XAI queries are tractable. Then (36) shows that the decision tree model satisfy those conditions, so that, as a consequence, it offers the XAI queries considered in (35). Notably, the authors also show that many other ML models (including decision lists, random forests, and boosted trees) do not offer any of the queries. Altogether, this shows on a formal basis that decision trees can be considered as a challenging ML model whenever interpretability is a crucial requirement.

In this paper, we focus on explanation queries for Boolean decision trees in a for-

[2]Unless f is a constant function, mapping every instance $\boldsymbol{x}$ to the same label.

mal XAI perspective. The abductive explanations and the contrastive explanations we consider are thus sound. We examine the computational ability of Boolean decision trees in deriving, minimizing and counting sufficient reasons and contrastive explanations. Each of those tasks can be viewed as an additional XAI query. We prove that the set of all sufficient reasons for an instance given a decision tree can be exponentially larger than the size of the input. When this is the case, generating the full set of sufficient reasons (i.e., the complete reason for the instance (33)) is typically out of reach. Furthermore, when computing this set is feasible but the set is large enough, the explainability issue is not dealt with since it would not really make sense to report hundreds explanations to the explainee (they would not be able to take advantage of them as a whole because of their cognitive limitations). However, computing a single sufficient reason, though tractable for decision trees, does not prove enough in general; indeed, two sufficient reasons for the same instance may differ on every feature. To deal with this issue and generate synthetic views of the set of all sufficient reasons, we define notions of relevant features and of necessary features that characterize the (possibly negated) features appearing in at least one or in every sufficient reason, and we show that they can be computed in polynomial time. We also introduce the notion of explanatory importance, that indicates how frequent each (possibly negated) feature is in the set of explanations. Though deriving the explanatory importance of a (possibly negated) feature in the set of sufficient reasons and determining the cardinality of this set are two computationally demanding tasks, we show how they can be achieved thanks to a model counting operation, which turns out to be practical in many cases. We also explain how to enumerate minimum-size sufficient reasons, which is a way to count them when they are not too numerous. We finally show that, from a computational standpoint, contrastive explanations highly depart from sufficient reasons. Indeed, the set of all contrastive explanations for an instance given a decision tree can be computed in polynomial time. As a consequence, such explanations can also be minimized and counted in polynomial time.

The rest of the paper is organized as follows. Preliminaries about Boolean functions, decision trees, abductive reasons, and contrastive explanations are given in Section 2. The computation of all sufficient reasons is considered in Section 3. Necessary and relevant features are presented in this section, as well as the approach for assessing the explanatory importance of a feature w.r.t. sufficient reasons and for counting the number of sufficient reasons. We also explain there how minimum-size sufficient reasons can be enumerated. An algorithm for computing all the contrastive explanations for the instance given the decision tree is presented in Section 4. Experimental results are reported in Section 5. Section 6 concludes the paper. For the sake of readability, all the proofs are reported in a final appendix.

2 Formal Preliminaries

2.1 Boolean Functions

For an integer n, let $[n]$ be the set $\{1, \ldots, n\}$. By $\mathcal{F}_n$ we denote the class of all Boolean functions from $\{0,1\}^n$ to $\{0,1\}$, and we use $X_n = \{x_1, \ldots, x_n\}$ to denote the set

of input Boolean variables, corresponding to the features under consideration. Any assignment $\boldsymbol{x} \in \{0,1\}^n$ is called an *instance*. If $f(\boldsymbol{x}) = 1$ for some $f \in \mathcal{F}_n$, then $\boldsymbol{x}$ is called a *model* of f. $\boldsymbol{x}$ is a *positive instance* when $f(\boldsymbol{x}) = 1$ and a *negative instance* when $f(\boldsymbol{x}) = 0$.

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

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

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

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

Let us illustrate the previous notions on the following example:

Example 1. *The* DNF *formula* $f_1 = (x_1 \wedge \overline{x}_2) \vee (x_2 \wedge x_3)$ *and the* CNF *formula* $f_2 = (x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \overline{x}_3)$ *represent two Boolean functions from* $\mathcal{F}_3$*. We have* $Var(f_1) = Var(f_2) = \{x_1, x_2, x_3\}$*.* f *and* g *are consistent since* $\boldsymbol{x} = (1,1,1)$ *is a model of* f_1 *and a model of* f_2*.* f_2 *is monotone while* f_1 *is not since* f_1 *contains an occurrence of the positive literal* x_2 *and an occurrence of the negative literal* $\overline{x}_2$ *(which is complementary to* x_2*). We can see that* f_1 *implies* f_2*, but* f_1 *and* f_2 *are not equivalent since* f_2 *does not imply* f_1*. Indeed,* $(0,1,0)$ *is a model of* f_2 *but not a model of* f_1*. The term* $x_1 \wedge x_3$ *is an implicant of* f_1 *that covers* $\boldsymbol{x}$ *since the set of literals* $\{x_1, x_2, x_3\}$ *corresponding to* $t_{\boldsymbol{x}}$ *is a superset of the set of literals* $\{x_1, x_3\}$*. More specifically,* $x_1 \wedge x_3$ *is a prime implicant of* f_1 *since neither* x_1 *nor* x_3 *is an implicant of* f_1*.* f_1 *has three prime implicants:* $x_1 \wedge \overline{x}_2$*,* $x_2 \wedge x_3$*, and* $x_1 \wedge x_3$*. The*

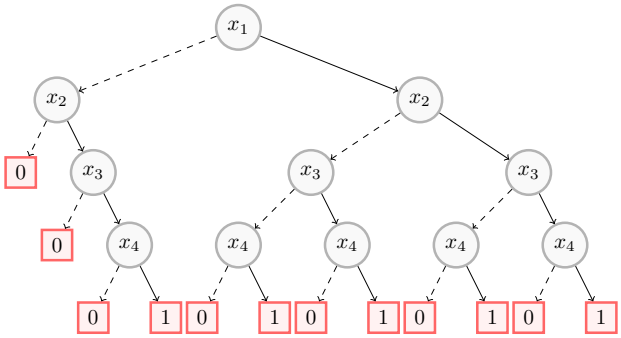


Figure 1: A decision tree T for recognizing *Cattleya* orchids. The left (resp. right) child of any decision node labelled by x_i corresponds to the assignment of x_i to 0 (resp. 1).

clause $x_1 \vee x_2 \vee \overline{x}_3$ is an implicate of f_2, but not a prime one, since the clause $x_1 \vee x_2$ is an implicate of f_2. In fact, $x_1 \vee x_2$ is the unique prime implicate of f_2.

2.2 Decision Trees, Abductive and Contrastive Explanations

With these basic notions in hand, we now focus on the following representation class of Boolean functions:

Definition 1 (Decision Tree)**.** *A (Boolean)* decision tree *over X_n is a binary tree T, each of whose internal nodes is labeled with one of n input Boolean variables from X_n, and whose leaves are labeled 0 or 1. Every variable is assumed (without loss of generality) to appear at most once on any root-to-leaf path (read-once property). The value $T(\boldsymbol{x}) \in \{0,1\}$ of T on an input instance $\boldsymbol{x}$ is given by the label of the leaf reached from the root as follows: at each node, go to the left or right child depending on whether the input value of the corresponding variable is 0 or 1, respectively. The size of T, denoted $|T|$, is given by the number of its nodes.*

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

For illustration, the following toy example will be used throughout the paper as a running example:

Example 2. *The decision tree in Figure 1 separates Cattleya orchids from other orchids using the following features: x_1: "has fragrant flowers", x_2: "has one or two leaves", x_3: "has large flowers", and x_4: "is sympodial".*

A DNF *representation of* T *is given by*

$$\texttt{DNF}(T) = \{\overline{x}_1 \wedge x_2 \wedge x_3 \wedge x_4, x_1 \wedge \overline{x}_2 \wedge \overline{x}_3 \wedge x_4, x_1 \wedge \overline{x}_2 \wedge x_3 \wedge x_4, \\ x_1 \wedge x_2 \wedge \overline{x}_3 \wedge x_4, x_1 \wedge x_2 \wedge x_3 \wedge x_4\}. \quad (1)$$

Dually, a CNF *representation of* T *is given by*

$$\texttt{CNF}(T) = \{x_1 \vee x_2, x_1 \vee \overline{x}_2 \vee x_3, x_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee x_4, \overline{x}_1 \vee x_2 \vee x_3 \vee x_4, \\ \overline{x}_1 \vee x_2 \vee \overline{x}_3 \vee x_4, \overline{x}_1 \vee \overline{x}_2 \vee x_3 \vee x_4, \overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee x_4\}. \quad (2)$$

As a salient characteristic, decision trees convey a single explicit abductive explanation for classifying any input instance: its direct reason. In general, this reason differs from the instance itself (but may nevertheless coincide with it in some cases).

Definition 2 (Direct Reason)**.** *Let* $T \in \texttt{DT}_n$ *and* $\boldsymbol{x} \in \{0,1\}^n$*. The* direct reason *for* $\boldsymbol{x}$ *given* T *is the term, denoted* $t^T_{\boldsymbol{x}}$*, corresponding to the unique root-to-leaf path of* T *that is compatible with* $\boldsymbol{x}$*, i.e., the path-restricted explanation for* $\boldsymbol{x}$ *given* T *(32).*

Another important notion of abductive explanations corresponds to the following concept of *sufficient reason* (33).

Definition 3 (Sufficient Reason)**.** *Let* $f \in \mathcal{F}_n$ *and* $\boldsymbol{x} \in \{0,1\}^n$ *such that* $f(\boldsymbol{x}) = 1$ *(resp.* $f(\boldsymbol{x}) = 0$*). A* sufficient reason *for* $\boldsymbol{x}$ *given* f *is a prime implicant* t *of* f *(resp.* $\neg f$*) that covers* $\boldsymbol{x}$*.* $sr(\boldsymbol{x}, f)$ *denotes the set of sufficient reasons for* $\boldsymbol{x}$ *given* f*.*

Thus, a sufficient reason (33) (also known and introduced as prime implicant explanation (34)) for an instance $\boldsymbol{x}$ given a class described by a Boolean function f is a subset t of the characteristics of $\boldsymbol{x}$ that is minimal with respect to set inclusion, and such that any instance $\boldsymbol{x}'$ sharing this set t of characteristics is classified by f as $\boldsymbol{x}$ is. Thus, when $f(\boldsymbol{x}) = 1$, t is a sufficient reason for $\boldsymbol{x}$ given f if and only if t is a prime implicant of f such that t covers $\boldsymbol{x}$, and when $f(\boldsymbol{x}) = 0$, t is a sufficient reason for $\boldsymbol{x}$ given f if and only if t is a prime implicant of $\neg f$ such that t covers $\boldsymbol{x}$. Accordingly, sufficient reasons are suited to explain why the instance at hand $\boldsymbol{x}$ has been classified by f as it has been classified.

Unlike sufficient reasons that are subset-minimal abductive explanations, direct reasons may contain arbitrarily many redundant features, even in the case f is represented by a decision tree (32). This explanation redundancy can be arbitrarily large and it is observed frequently in practice, even when optimal (size-minimal) decision trees are considered (37).

When considering the sufficient reasons of the input instance, one may be interested in focusing on the shortest ones, referred to as minimum-size sufficient reasons. Those reasons are valuable since conciseness is often a desirable property of explanations (Occam's razor). Formally:

Definition 4 (Minimum-size Sufficient Reason)**.** *Let* $f \in \mathcal{F}_n$ *and* $\boldsymbol{x} \in \{0,1\}^n$*. A* minimum-size sufficient reason *for* $\boldsymbol{x}$ *given* f *is a sufficient reason for* $\boldsymbol{x}$ *given* f *that contains a minimal number of literals.*

Finally, unlike direct and (possibly minimum-size) sufficient reasons that aim to explain the classification of the instance $\boldsymbol{x}$ under consideration as achieved by the classifier f, contrastive explanations are valuable when $\boldsymbol{x}$ has not been classified by f as expected by the explainee (1). In this case, one looks for minimal subsets of the features (w.r.t. set inclusion) that when switched in $\boldsymbol{x}$ are enough to get instances that are classified positively (resp. negatively) by f if $\boldsymbol{x}$ is classified negatively (resp. positively) by f. Note that computing contrastive explanations can also be useful when $\boldsymbol{x}$ has been classified by f as expected by the explainee. Indeed, whatever the expectations of the explainee, contrastive explanations make precise how to minimally change instances to get a different prediction. As such, they somehow indicate how much robust the prediction achieved can be considered (38; 35).

Formally, a *contrastive explanation* for $\boldsymbol{x}$ given f (24) is a subset t of the characteristics of $\boldsymbol{x}$ that is minimum w.r.t. set inclusion among those such that at least one instance $\boldsymbol{x}'$ that coincides with $\boldsymbol{x}$ except on the characteristics from t is not classified by f as $\boldsymbol{x}$ is.

Definition 5 (Contrastive Explanation)**.** *Let $f \in \mathcal{F}_n$ and $\boldsymbol{x} \in \{0,1\}^n$ such that $f(\boldsymbol{x}) = 1$ (resp. $f(\boldsymbol{x}) = 0$). A* contrastive explanation *for $\boldsymbol{x}$ given f is a term t over X_n such that $t \subseteq t_{\boldsymbol{x}}$, $t_{\boldsymbol{x}} \setminus t$ is not an implicant of f (resp. $\neg f$), and for every $\ell \in t$, $t \setminus \{\ell\}$ does not satisfy this last condition.*

Just like for sufficient reasons, one can be interested in focusing on the shortest contrastive explanations:

Definition 6 (Minimum-size Contrastive Explanation)**.** *Let $f \in \mathcal{F}_n$ and $\boldsymbol{x} \in \{0,1\}^n$. A* minimum-size contrastive explanation *for $\boldsymbol{x}$ given f is a contrastive explanation for $\boldsymbol{x}$ given f that contains a minimal number of literals.*

Example 3. *Based on our running example, we can observe that $T(\boldsymbol{x}) = 1$ for the instance $\boldsymbol{x} = (1,1,1,1)$ and that $T(\boldsymbol{x}') = 0$ for the instance $\boldsymbol{x}' = (0,0,0,0)$. The direct reason for $\boldsymbol{x}$ given T is the term $t^T_{\boldsymbol{x}} = x_1 \wedge x_2 \wedge x_3 \wedge x_4$. It coincides with the term $t_{\boldsymbol{x}}$. Contrastingly, the direct reason for $\boldsymbol{x}'$ given T is the term $t^T_{\boldsymbol{x}'} = \overline{x}_1 \wedge \overline{x}_2$, that does not coincide with $t_{\boldsymbol{x}'} = \overline{x}_1 \wedge \overline{x}_2 \wedge \overline{x}_3 \wedge \overline{x}_4$. $x_1 \wedge x_4$ and $x_2 \wedge x_3 \wedge x_4$ are the sufficient reasons for $\boldsymbol{x}$ given T. $x_1 \wedge x_4$ is the unique minimum-size sufficient reason for $\boldsymbol{x}$ given T. $\overline{x}_4$, $\overline{x}_1 \wedge \overline{x}_2$, and $\overline{x}_1 \wedge \overline{x}_3$ are the sufficient reasons for $\boldsymbol{x}'$ given T. $\overline{x}_4$ is the unique minimum-size sufficient reason for $\boldsymbol{x}'$ given T. x_4, $x_1 \wedge x_2$, and $x_1 \wedge x_3$ are the contrastive explanations for $\boldsymbol{x}$ given T. Thus, the instance $(1,1,1,0)$ that differs with $\boldsymbol{x}$ only on x_4 is not classified by T as $\boldsymbol{x}$ is ($(1,1,1,0)$ is classified as a negative instance). x_4 is the unique minimum-size contrastive explanations for $\boldsymbol{x}$ given T. $\overline{x}_1 \wedge \overline{x}_4$ and $\overline{x}_2 \wedge \overline{x}_3 \wedge \overline{x}_4$ are the contrastive explanations for $\boldsymbol{x}'$ given T. Thus, the instance $(1,0,0,1)$ that differs with $\boldsymbol{x}'$ only on x_1 and x_4 is not classified by T as $\boldsymbol{x}'$ is ($(1,0,0,1)$ is classified as a positive instance). $\overline{x}_1 \wedge \overline{x}_4$ is the unique minimum-size contrastive explanations for $\boldsymbol{x}'$ given T.*

We mention in passing that when dealing with decision trees T, we could have focused only on explanations for the *positive* instances $\boldsymbol{x}$ given T. This comes from the fact that $\mathtt{DT}_n$ is closed under negation, in the sense that for any $T \in \mathtt{DT}_n$, a decision tree

equivalent to $\neg T$ can be obtained by just replacing from T the label of each leaf with its complement. So, for any instance $\boldsymbol{x} \in \{0,1\}^n$, a direct reason (resp. sufficient reason, minimum-size sufficient reason, contrastive explanation) explaining why $T(\boldsymbol{x}) = 0$ is precisely the same as a direct reason (resp. sufficient reason, minimum-size sufficient reason, contrastive explanation) explaining why $(\neg T)(\boldsymbol{x}) = 1$. Considering T or its negation $\neg T$ has no computational impact since $\neg T$ can be computed in time linear in the size of T.

3 Computing All Sufficient Reasons

Sufficient reasons can be exponentially numerous When switching from the direct reason for an instance (that is unique but, in general, not redundancy-free) to its sufficient reasons, a main obstacle to be dealt with lies in the number of reasons to be considered. Indeed, even for the restricted class of decision trees with logarithmic depth, an input instance can have exponentially many sufficient reasons:

Proposition 1. *There is a decision tree $T \in \mathtt{DT}_n$ of depth $\log_2(n+1)$ such that for any $\boldsymbol{x} \in \{0,1\}^n$, the number of sufficient reasons for $\boldsymbol{x}$ given T is at least $\lfloor \frac{3}{2}^{\frac{n+1}{2}} \rfloor$.*

In many practical cases, the number of sufficient reasons for an instance given a decision tree can be very large. Thus, Figure 3 shows an `mnist49` instance (the leftmost subfigure) that has been considered in our experiments. A decision tree with high accuracy (see Section 5) has been learned for this dataset, and the instance has 569,351,040 sufficient reasons given this decision tree.

Several algorithms for generating the set of all sufficient reasons for an input instance given a decision tree have been designed so far (see (37) for a brief survey). Notably, based on the evidence that the set $sr(\boldsymbol{x}, T)$ of all sufficient reasons for $\boldsymbol{x}$ given T coincides with the set of prime implicants of the quantified Boolean formula

$$\forall \{\ell : \ell \in t_{\boldsymbol{x}}\} \cdot \mathtt{CNF}(T)$$

where every literal of $t_{\boldsymbol{x}}$ is universally quantified in $\mathtt{CNF}(T)$, together with the fact that a monotone `CNF` formula equivalent to $\mathtt{CNF}(T)$ can be obtained by removing from every clause c of $\mathtt{CNF}(T)$ every literal not belonging to $t_{\boldsymbol{x}}$ (see Proposition 21 and Theorem 10 from (39)), one can take advantage of the quasi-polynomial time algorithm of Gurvich and Khachiyan (40) for an incremental enumeration of the prime implicants of a monotone `CNF` formula in order to derive $sr(\boldsymbol{x}, T)$ with the same computational guarantees. Note however that the existence of an enumeration algorithm for sufficient reasons satisfying strong computational requirements about the enumeration process (e.g., incremental polynomial time, or even output polynomial time) is unlikely (41).

Another noteworthy observation is that, in the worst case, the sufficient reasons for a given instance may differ on every feature. For instance, consider the Boolean function f given by the decision tree T reported on Figure 2. f is equivalent to $x_1 \vee x_2$. The direct reason for $\boldsymbol{x} = (1,1)$ given T is x_1, and it is a sufficient reason for $\boldsymbol{x}$. However, $\boldsymbol{x}$ also has another sufficient reason given T, namely x_2. Such a diversity of the sufficient reasons for the same instance has also been pointed out in (37).

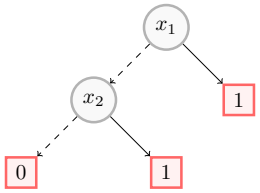


Figure 2: A decision tree T equivalent to $x_1 \vee x_2$.

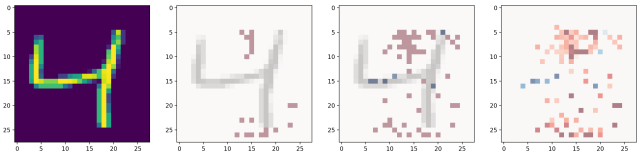


Figure 3: A `mnist49` instance (on the left), then (from left to right) two sufficient reasons for it, including a minimum-size one (the leftmost one out of the two) and an explanatory heat map for the instance (the rightmost picture).

Thus, considering only a single sufficient reason for the input instance is not satisfying; indeed, the sufficient reason that is returned to the explainee may be considered as bad by her/him, even though a much better reason for the explainee might exist.

In practice, instances may have very dissimilar sufficient reasons. As an illustration, the two central subfigures of Figure 3 present two sufficient reasons for the same instance, and one can observe that they differ on many features (blue (resp. red) dots correspond to pixels on (resp. off)).

To sum up, on the one hand, computing the set of all the sufficient reasons for a given instance is not always feasible in a reasonable amount of time. Furthermore, if the computation succeeds but the cardinality of the set is huge, the (disjunctively interpreted) set of sufficient reasons, equivalent to the *complete reason* for the instance (33), can hardly be considered as intelligible by the explainee. On the other hand, because of the diversity that can be observed in the set of sufficient reasons, deriving one sufficient reason, only, is not guaranteed to be informative enough. Thus, one needs to design approaches to synthesizing the set of sufficient reasons while avoiding the two pitfalls (the computational one and the informational one).

Synthesizing the set of sufficient reasons In this objective, the following notions of *necessary/(ir)relevant features* appear useful. These notions of necessity and relevance echo the ones that have been considered in (42) for logic-based abduction.

Definition 7 (Explanatory Features)**.** *Let $f \in \mathcal{F}_n$, and $\boldsymbol{x} \in \{0,1\}^n$ be an instance. Let e be an explanation type.*[3]

- *A literal ℓ over X_n is a* necessary feature *for the family e of explanations for $\boldsymbol{x}$ given f if and only if ℓ belongs to every explanation t for $\boldsymbol{x}$ given f such that t*

[3]For instance, e can be s when the sufficient reasons for $\boldsymbol{x}$ given f are targeted or c when the contrastive explanations for $\boldsymbol{x}$ given f are targeted.

is of type e. $Nec_e(\boldsymbol{x}, f)$ denotes the set of all necessary features for the family e of explanations for $\boldsymbol{x}$ given f.

- *A literal ℓ over X_n is a* relevant feature *for the family e of explanations for $\boldsymbol{x}$ given f if and only if ℓ belongs to at least one explanation t for $\boldsymbol{x}$ given f such that t is of type e. $Rel_e(\boldsymbol{x}, f)$ denotes the set of all relevant features for the family e of explanations for $\boldsymbol{x}$ given f. $Irr_e(\boldsymbol{x}, f)$, which is the complement of $Rel_e(\boldsymbol{x}, f)$ in the set of all literals over X_n, denotes the set of all irrelevant features for the family e of explanations for $\boldsymbol{x}$ given f.*

The necessary (resp. irrelevant) features for the family s of sufficient reasons for $\boldsymbol{x}$ given f are the most (resp. less) important features for explaining the classification of $\boldsymbol{x}$ by f, since they belong to every (resp. no) sufficient reason for $\boldsymbol{x}$ given f.

When a single sufficient reason t for $\boldsymbol{x}$ given f has been computed, the cardinality of t deprived from the features of $Nec_s(\boldsymbol{x}, f)$ is small, and the cardinality of the symmetric difference between t and $Rel_s(\boldsymbol{x}, f)$ is small as well, t can be viewed as a good representative of the complete reason for $\boldsymbol{x}$ given f in the sense that a sufficient reason t' for $\boldsymbol{x}$ given f that differs a lot from t cannot exist.

In the case when f is a decision tree T, though the set of all sufficient reasons for $\boldsymbol{x}$ given T cannot be generated when it is too large, $Nec_s(\boldsymbol{x}, f)$, $Rel_s(\boldsymbol{x}, f)$, and $Irr_s(\boldsymbol{x}, f)$ can be derived efficiently:

Proposition 2. *Let $T \in \mathtt{DT}_n$, and $\boldsymbol{x} \in \{0,1\}^n$. Computing $Nec_s(\boldsymbol{x}, T)$, $Rel_s(\boldsymbol{x}, T)$, and $Irr_s(\boldsymbol{x}, T)$ can be done in $\mathcal{O}((n + |T|) \cdot |T|)$ time.*

We mention in passing that the task of deciding whether a given feature belongs to a sufficient reason for an input instance is also referred to as the *feature membership problem* (43). Though this problem is NP-hard for arbitrary Boolean classifiers, it remains tractable for decision trees (43). This tractability result clearly coheres with Proposition 2.

Going a step further consists in evaluating the explanatory importance of every (positive or negative) feature:

Definition 8 (Explanatory Importance)**.** *Let $f \in \mathcal{F}_n$, and $\boldsymbol{x} \in \{0,1\}^n$ be an instance. Let e be an explanation type, and $E_e(\boldsymbol{x}, f)$ the set of all explanations for $\boldsymbol{x}$ given f that are of type e. The* explanatory importance *of a literal ℓ over X_n for $\boldsymbol{x}$ given f w.r.t. e is given by*

$$Imp_e(\ell, \boldsymbol{x}, f) = \frac{\#(\{t \in E_e(\boldsymbol{x}, f) : \ell \in t\})}{\#(E_e(\boldsymbol{x}, f))}.$$

Example 4. *Based on our running example, $Nec_s(\boldsymbol{x}, T) = \{x_4\}$ and $Rel_s(\boldsymbol{x}, T) = \{x_1, x_2, x_3, x_4\}$. We also have $Imp_s(x_4, \boldsymbol{x}, T) = 1$, $Imp_s(x_1, \boldsymbol{x}, T) = Imp_s(x_2, \boldsymbol{x}, T) = Imp_s(x_3, \boldsymbol{x}, T) = \frac{1}{2}$, and $Imp_s(\ell, \boldsymbol{x}, T) = 0$ for every other literal ℓ (the negative ones over $\{x_1, x_2, x_3, x_4\}$).*[4] *As to $\boldsymbol{x}'$, we have $Nec_s(\boldsymbol{x}', T) = \varnothing$ and $Rel_s(\boldsymbol{x}', T) = \{\overline{x}_1, \overline{x}_2, \overline{x}_3, \overline{x}_4\}$. We also have $Imp_s(\overline{x}_1, \boldsymbol{x}', T) = \frac{2}{3}$, $Imp_s(\overline{x}_2, \boldsymbol{x}', T) = Imp_s(\overline{x}_3, \boldsymbol{x}', T) =$*

[4] Note that by construction the explanatory importance $Imp_s(\ell, \boldsymbol{x}, f)$ of ℓ for $\boldsymbol{x}$ given a Boolean classifier f is equal to 0 when ℓ is not a literal of $t_{\boldsymbol{x}}$ since the sufficient reasons for $\boldsymbol{x}$ given f are terms containing only literals from $t_{\boldsymbol{x}}$.

$Imp_s(\overline{x}_4, \boldsymbol{x}', T) = \frac{1}{3}$, and $Imp_s(\ell, \boldsymbol{x}', T) = 0$ for every other literal ℓ (the positive ones over $\{x_1, x_2, x_3, x_4\}$).

Importantly, the concept of explanatory importance must not be confused with the notion of feature importance (which can be defined and assessed in many different ways): the former is local (i.e., relative to an instance) and not global, it concerns literals and not variables (polarity matters), and it is about the explanation task, not the prediction one.

In order to compute the explanatory importance of a literal, a straightforward approach consists in enumerating the explanations of $E_e(\boldsymbol{x}, f)$. This is feasible when this set is not too large, but, as already mentioned, this is not always the case for sufficient reasons even when f is a decision tree T. Thus, for dealing with the remaining case, an alternative approach must be looked for.

Focusing on sufficient reasons, we designed such an approach for computing the explanatory importance $Imp_s(\ell, \boldsymbol{x}, T)$ w.r.t. sufficient reasons of a literal ℓ for an instance $\boldsymbol{x}$ given a decision tree T. This approach is given by Algorithm 1. As explained previously, we know that $sr(\boldsymbol{x}, T)$ is by construction the set of prime implicants of the CNF formula $C = \{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}$, which can be computed in time polynomial in the size of T and the size of $\boldsymbol{x}$. Then one can exploit the translation process presented in (44) showing how to associate in polynomial time with a given CNF formula (here, C) another CNF formula, say PIC, such that the models of PIC are in one-to-one correspondence with the prime implicants of C. The translation leverages Tseitin transformation (45), and requires auxiliary variables to be introduced. Every auxiliary variable that is introduced is definable from the initial variables (46), so that the number of models of the resulting CNF formula PIC is the same as the number of prime implicants of C. In our case, the translation can be simplified because C is a monotone CNF formula. Finally, we take advantage of the compiler D4 (47) to compile PIC into a d−DNNF circuit (48) $dDNNF$, and this enables us to compute both the number of sufficient reasons for $\boldsymbol{x}$ given T (it is given by $\#(dDNNF)$, the number of models of $dDNNF$) and the explanatory importance of every literal $\ell \in t_{\boldsymbol{x}}$ (it is the ratio of the number of sufficient reasons $\#(\{t \in E_s(\boldsymbol{x}, T) : \ell \in t\})$ for $\boldsymbol{x}$ given T that contains ℓ – this number coincides with the number of models $\#(dDNNF|\ell)$ of $dDNNF$ when conditioned by ℓ – divided by the number $\#(dDNNF)$ of sufficient reasons for $\boldsymbol{x}$ given T). The compilation phase into d−DNNF is computationally expensive in the general case, but the last step can be achieved in time polynomial in the size of $dDNNF$. Indeed, the d−DNNF language supports in polynomial time the model counting query and the conditioning transformation (49).

Example 5. *Let us illustrate Algorithm 1 on our running example focusing on instance $\boldsymbol{x} = (1, 1, 1, 1)$. Based on the CNF representation of T given in (2), we get*

$$C = (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_2 \vee x_3 \vee x_4) \\ \wedge (x_2 \vee x_4) \wedge (x_3 \vee x_4) \wedge x_4,$$

which can be simplified into the equivalent formula $(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge x_4$ using unit propagation. Then PIC is obtained by conjoining C with the following implications (one formula is generated per literal occurring in C): $x_1 \rightarrow (\overline{x}_2 \vee \overline{x}_3)$,

Algorithm 1: Computing the explanatory importance of literals w.r.t. sufficient reasons

Require: a decision tree $T \in \mathtt{DT}_n$ and an instance $\boldsymbol{x} \in \{0,1\}^n$
Ensure: the explanatory importance Imp_s of each literal from $t_{\boldsymbol{x}}$
$C = \{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}$
▷ *compute a* CNF *formula* C *such that the prime implicants of* C *are the sufficient reasons for* $\boldsymbol{x}$ *given* T
$PIC = CNF\_PI(C)$
▷ *translate* C *into a* CNF *formula* PIC *with models representing the prime implicants of* C *(44)*
$dDNNF = \mathtt{D4}(PIC)$ ▷ *compile* PIC *into a* d-DNNF *circuit* $dDNNF$ *(48; 47)*
for $\ell \in L$ **do**
$Imp_s(\ell, \boldsymbol{x}, T) = \frac{\#(dDNNF|\ell)}{\#(dDNNF)}$ ▷ *compute the explanatory importance of literals from* $t_{\boldsymbol{x}}$
end for
return (Imp_s)

$x_2 \rightarrow \overline{x}_1$, *and* $x_3 \rightarrow \overline{x}_1$. *Each implication makes precise conditions under which the corresponding literal* ℓ *(the condition of the implication) participates in a prime implicant of* C, *i.e., there exists a clause of* C *that is satisfied by* ℓ *but the clause is not satisfied any longer if* ℓ *is removed. In the general case, the conclusion part of each implication is a* DNF *formula (equivalent to the negation of all clauses of* C *deprived from* ℓ*), so that the implication itself cannot be read directly as a* CNF *formula: new variables must be introduced using Tseitin transformation to turn each implication into a* CNF *formula that has the same consequences over the initial set of variables as the corresponding implication. The resulting* CNF *formula is* PIC. *On the example, it turns out that introducing new variables is useless since every implication can be considered as a clause. Thus, one gets the* CNF *formula* $PIC = (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge x_4 \wedge (\overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee \overline{x}_1) \wedge (\overline{x}_3 \vee \overline{x}_1)$. *Finally,* PIC *can be compiled into the* d-DNNF *circuit* $dDNNF$ *given at Figure 4. This circuit* $dDNNF$ *has two models over* $\{x_1, x_2, x_3, x_4\}$, *namely* $(0,1,1,1)$ *and* $(1,0,0,1)$. *Each of them corresponds (in a one-to-one way) with a sufficient reason for* $\boldsymbol{x}$ *given* T, *given by the the literals of* $t_{\boldsymbol{x}}$ *occurring in it. Thus,* $(0,1,1,1)$ *is associated with* $x_2 \wedge x_3 \wedge x_4$, *and* $(1,0,0,1)$ *is associated with* $x_1 \wedge x_4$. *The two sufficient reasons for* $\boldsymbol{x}$ *given* T *are recovered, as expected. From* $dDNNF$, *we can easily compute the explanatory importance of the literals* ℓ *occurring in* $t_{\boldsymbol{x}}$ *by computing for each of them the ratio* $\frac{\#(dDNNF|\ell)}{\#(dDNNF)}$. *As expected, we get* $Imp_s(x_1, \boldsymbol{x}, T) = \frac{1}{2}$, $Imp_s(x_2, \boldsymbol{x}, T) = \frac{1}{2}$, $Imp_s(x_3, \boldsymbol{x}, T) = \frac{1}{2}$, *and* $Imp_s(x_4, \boldsymbol{x}, T) = 1$.

We will show in Section 5 that, despite a high complexity in the worst case (the size of $dDNNF$ can be exponential in $|T|$), this approach based on knowledge compilation proves quite efficient in practice.

Clearly enough, when $Imp_s(\ell, \boldsymbol{x}, T)$ has been computed for every $\ell \in t_{\boldsymbol{x}}$, one can easily generate explanatory heat maps. The rightmost subfigure of Figure 3 gives the explanatory heat map computed for the mnist instance corresponding to the leftmost subfigure. This instance has 12 necessary literals and 105 relevant literals. Blue (resp. red) pixels correspond to positive (resp. negative) literals in the instance, and the intensity of the color aims to reflect the explanatory importance of the corresponding literal.

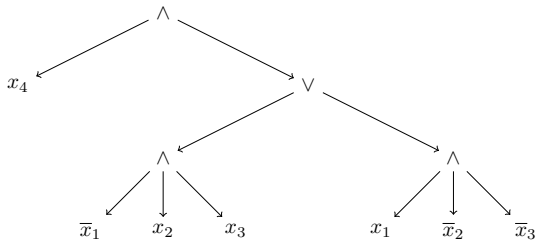


Figure 4: A d−DNNF circuit representing the sufficient reasons for $\boldsymbol{x}$ given T.

This explanatory heat map gives a synthetic explanation as to why the corresponding instance is recognized as a "4", and not as a "9". It suggests that the instance is a "4" because pixels "closing the loop" of a "9" and those forming the "tail" of a "9" are off.

Enumerating minimum-size sufficient reasons An approach to synthesizing the set of sufficient reasons consists in focusing on the minimum-size ones. Indeed, though the set of minimum-size sufficient reasons for an instance given a decision tree can also be exponentially large (60), the number of minimum-size sufficient reasons cannot exceed the number of sufficient reasons, and it can be significantly lower in practice.

It is known that a sufficient reason for an instance $\boldsymbol{x}$ given a decision tree T can be generated in time polynomial in the size of the input ($\boldsymbol{x}$ and T) using a greedy algorithm (see e.g., (32)). Adding the minimum-size condition makes the problem intractable:

Proposition 3. *(50) Let $T \in \mathtt{DT}_n$ and $\boldsymbol{x} \in \{0,1\}^n$. Computing a minimum-size sufficient reason for $\boldsymbol{x}$ given T is* NP*-hard.*

Despite this intractability result, minimum-size sufficient reasons can be generated in many practical cases. A common approach for handling NP-optimization problems is to rely on modern constraint solvers. One follows this direction here and casts the task of finding minimal sufficient reasons as a Boolean constraint optimization problem. We first need to recall that a PARTIAL MAXSAT problem consists of a pair $(C_{\text{soft}}, C_{\text{hard}})$ where C_{soft} and C_{hard} are (finite) set of clauses. The goal is to find a Boolean assignment that maximizes the number of clauses c in C_{soft} that are satisfied, while satisfying all clauses in C_{hard}.

Proposition 4. *Let T be a decision tree in $\mathtt{DT}_n$ and $\boldsymbol{x} \in \{0,1\}^n$ be an instance such that $T(\boldsymbol{x}) = 1$. Let $(C_{\text{soft}}, C_{\text{hard}})$ be an instance of the* PARTIAL MAXSAT *problem such that:*

$$C_{\text{soft}} = \{\overline{x}_i : x_i \in t_{\boldsymbol{x}}\} \cup \{x_i : \overline{x}_i \in t_{\boldsymbol{x}}\} \text{ and } C_{\text{hard}} = \{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}.$$

The intersection of $t_{\boldsymbol{x}}$ with $t_{\boldsymbol{x}^*}$ where $\boldsymbol{x}^*$ is an optimal solution of $(C_{\text{hard}}, C_{\text{soft}})$, is a minimum-size sufficient reason for $\boldsymbol{x}$ given T.

Clearly enough, if $\boldsymbol{x}$ is such that $T(\boldsymbol{x}) = 0$, then it is enough to consider the same instance of PARTIAL MAXSAT as above, except that $C_{\text{hard}} = \{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(\neg T)\}$.

Interestingly, one can take advantage of the PARTIAL MAXSAT characterization above for generating a preset number of minimum-size sufficient reasons via the use of blocking clauses. Basically, the approach is as follows: one generates a first reason t, then one adds to $C_{\rm hard}$ the negation of t as a clause and we resume until the bound is reached or no solution exists or the size of the last solution that has been generated is strictly larger than the size of the first reason t that has been computed. To avoid the latter test, after the generation of the first reason t, one can alternatively add to $C_{\rm hard}$ a $\mathtt{CNF}$ encoding of a cardinality constraint ensuring that the next reasons to be generated have the same size as the one of t.

Finally, when dealing with instances $\boldsymbol{x}$ for which computing a minimum-size sufficient reason t is too demanding in practice (i.e., the PARTIAL MAXSAT solver used to compute it does not terminate in due time), one can relax the size minimality condition about explanations and computes an abductive explanation h for $\boldsymbol{x}$ given T in polynomial time, while ensuring some guarantees about the size of h. Indeed, the derivation of a minimum-size reason t for an instance $\boldsymbol{x}$ given a decision tree T amounts to the computation of a minimal hitting set t of the hypergraph with $\{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}$ as set of hyperedges. The point is that a simple greedy algorithm running in $\mathcal{O}(n \cdot |T|)$ time can be exploited to derive a hitting set h of such a hypergraph: this algorithm consists in choosing a literal ℓ of $t_{\boldsymbol{x}}$ that belongs to a maximum number of clauses of $\{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}$, then deleting ℓ from $t_{\boldsymbol{x}}$ and all the clauses containing ℓ from $\{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}$, and repeating this process until the resulting set of clauses is empty. h is the set of literals ℓ of $t_{\boldsymbol{x}}$ that have been picked up in the process. It turns out that the size of h is guaranteed to be lower than the size of any minimum-size sufficient reason t up to a factor $\ln m - \ln\ln m + 0.78$ (51), where m is the number of clauses in $\mathtt{CNF}(T)$ (or, equivalently, the number of branches in T).

4 Computing All Contrastive Explanations

Interestingly, it has been shown that sufficient reasons and contrastive explanations are connected by a minimal hitting set duality (24). This duality can be leveraged to derive one of the two sets of explanations from the other one using algorithms for computing minimal hitting sets (52; 53).

However, in the case of decision trees, a more direct and much more efficient approach to derive all the contrastive explanations for $\boldsymbol{x} \in \{0,1\}^n$ given $T \in \mathtt{DT}_n$ can be designed. Indeed, unlike what happens for sufficient reasons (see Section 3), the set of *all* contrastive explanations for $\boldsymbol{x} \in \{0,1\}^n$ given a decision tree $T \in \mathtt{DT}_n$ can be computed in polynomial time from $\boldsymbol{x}$ and T. Note that this result has also been obtained in parallel and independently of us (see (43)).

Proposition 5. *The set of all contrastive explanations for $\boldsymbol{x} \in \{0,1\}^n$ given a decision tree $T \in \mathtt{DT}_n$ can be computed in time polynomial in $n + |T|$ as*

$$min(\{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}, \subseteq)$$

when $T(\boldsymbol{x}) = 1$, and as

$$min(\{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(\neg T)\}, \subseteq)$$

when $T(\boldsymbol{x}) = 0$.

Example 6. *On our running example, using again (2) for the* CNF *representation of* T *together with* $\boldsymbol{x} = (1, 1, 1, 1)$*, we have* $min(\{c \cap t_{\boldsymbol{x}} : c \in \mathtt{CNF}(T)\}, \subseteq) = \{x_1 \vee x_2, x_1 \vee x_3, x_4\}$*, which corresponds to the contrastive explanations* $x_1 \wedge x_2$*,* $x_1 \wedge x_3$*, and* x_4 *for* $\boldsymbol{x}$ *given* T *(viewing clauses and terms as sets of literals). Similarly, we have*

$$\mathtt{CNF}(\neg T) = \{x_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4, \overline{x}_1 \vee x_2 \vee x_3 \vee \overline{x}_4, \overline{x}_1 \vee x_2 \vee \overline{x}_3 \vee \overline{x}_4, \\ \overline{x}_1 \vee \overline{x}_2 \vee x_3 \vee \overline{x}_4, \overline{x}_1 \vee \overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4\}.$$

Thus, with $\boldsymbol{x}' = (0, 0, 0, 0)$*, we have* $min(\{c \cap t_{\boldsymbol{x}'} : c \in \mathtt{CNF}(\neg T)\}, \subseteq) = \{\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_4, \overline{x}_1 \vee \overline{x}_4\}$*; which corresponds to the contrastive explanations* $\overline{x}_2 \wedge \overline{x}_3 \wedge \overline{x}_4$ *and* $\overline{x}_1 \wedge \overline{x}_4$ *for* $\boldsymbol{x}'$ *given* T.

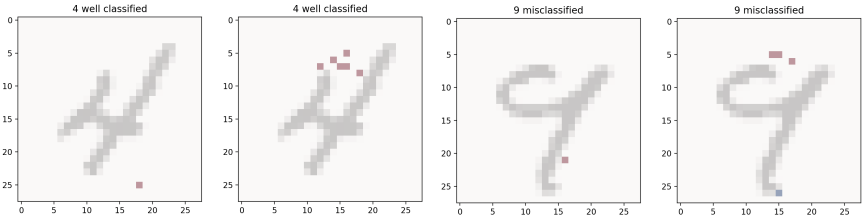


Figure 5: Two instances from `mnist49`: a "4" correctly identified as a "4" by the decision tree, and a "9" misclassified as a "4" by the decision tree. For each instance, two contrastive explanations are reported (the leftmost ones correspond to the "4" that is correctly classified and the rightmost ones correspond to the "9" that is misclassified.

As a matter of illustration, Figure 5 presents two instances from `mnist49` that have been considered in our experiments: the first instance is a "4" correctly identified as a "4" by the decision tree, and the second instance is a "9" misclassified as a "4" by the decision tree. One can observe that the "4" really looks as a "4", while the "9" could also be seen as a "4". The "4" correctly classified has 42 contrastive explanations given the decision tree: 10 of size 1, 11 of size 2, 11 of size 3, 5 of size 4, 3 of size 5, and 2 of size 6. The misclassified "9" has 21 contrastive explanations given the decision tree: 3 of size 1, 5 of size 2, 10 of size 3, and 3 of size 4. For those two instances, the size of the contrastive explanations is rather small (with a maximal size of 6 for the first instance), given that 784 features are used to describe the instances (the pictures consist of 28×28 pixels). For each instance, two contrastive explanations are reported on the figure. The leftmost picture shows a contrastive explanation of size 1 for the "4" instance at hand. This first explanation is not very satisfying: one can hardly see why turning on the red pixel would change the "4" into a "9". This shows that our predictor is not as robust as one would expect it to be. The next picture (from left to right) presents another contrastive explanation (of size 6) for the same "4" instance. This explanation is intuitively better than the previous one (roughly, it tells that "closing the loop" by turning on the six red pixels given by the explanation would be enough

Table 1: Description of the 20 datasets for which empirical results are provided.

Dataset	#I	#F	#C	source
recidivism	26020	11	2	Kaggle
adult	48842	14	2	UCI
bank marketing	45211	17	2	UCI
bank	41188	21	2	Kaggle
lending loan	9578	13	2	Kaggle
contraceptive	1473	9	3	UCI
compas	5278	14	2	OpenML
christine	5418	1637	2	OpenML
farm-ads	4143	54877	2	UCI
mnist49	13782	784	2	-
spambase	4601	57	2	UCI
mnist38	13966	784	2	-
madelon	4400	500	2	UCI
gisette	7000	5000	2	OpenML
gina	3153	970	2	OpenML
price phone	2000	20	4	Kaggle
letter	20000	16	2	UCI
titanic	623	5	2	Kaggle
yeast	2417	117	2	OpenML
soybean	683	35	2	UCI

to change the prediction from "4" to "9"). Similarly, the last two pictures give two contrastive instances for the "9" that is misclassified. The first one has size 1 and the second one (corresponding to the rightmost picture) has size 4. The second explanation looks slightly better than the first one, that is clearly not satisfying.

As illustrated by Figure 5, Proposition 5 has two direct, yet noteworthy, consequences: on the one hand, the number of contrastive explanations for any instance given a decision tree cannot exceed the number of branches in the tree; on the other hand, the size of any contrastive explanation for any instance given a decision tree is upper bounded by the depth of the tree. Those two properties offered by decision trees are valuable from the intelligibility perspective. However, the second property can also be viewed as a testimony of the *intrinsic limitation of the robustness of decision trees* as an ML model: *changing a few features (no more than the depth of the tree) in the input instance (whatever it is) is enough to change the prediction made*. For a fixed depth, this holds *whatever the accuracy of the tree*. Thus, though the average accuracy of the decision tree used to classify the two instances at Figure 5 exceeds 95% on the test set (see Section 5), turning on a single pixel is enough for the predictor to recognize a "9" instead of a "4" for the first instance, and a "4" instead of a "9" for the second instance.

Other straightforward consequences of Proposition 5 are that computing necessary / relevant features and computing the explanatory importance of features w.r.t. contrastive explanations can be achieved in time polynomial in $n + |T|$. Because they can be enumerated efficiently, contrastive explanations can also be easily minimized and counted.

5 Experiments

5.1 Empirical setting

We have considered 90 datasets, which are standard benchmarks from the well-known repositories Kaggle (www.kaggle.com), OpenML (www.openml.org), and UCI (archive.ics.uci.edu/ml/). `mnist38` and `mnist49` are subsets of the `mnist` dataset, restricted

Table 2: Empirical results for the 20 datasets.

	Decision Tree				| Sufficient |		| Min.-Sized |		#Nec. Features		#Rel. Features	
Dataset	%A	#N	#B	#D	med	max	med	max	med	max	med	max
recidivism	63.4	13828.8	147.6	27.8	14	22	13	22	6	19	60	98
adult	81.4	12934.0	2974.8	47.0	16	36	16	36	7	22	263	543
bank marketing	87.4	6656.4	1432.6	30.7	14	21	14	21	3	16	247	398
bank	89.0	5523.6	977.8	29.6	13	24	13	24	4	15	200	330
lending loan	73.5	2610.4	1131.4	33.3	16	31	16	31	8	25	226	442
contraceptive	50.4	1252.2	88.6	22.0	11	20	11	20	8	17	25	47
compas	66.0	1230.0	46.2	19.1	6	14	6	14	3	12	16	33
christine	63.4	853.2	426.0	36.6	12	47	12	47	8	41	92	202
farm-ads	86.8	544.8	264.6	85.5	20	99	20	99	16	92	73	192
mnist49	95.5	539.6	267.9	28.8	22	30	22	30	9	19	91	166
spambase	92.0	536.4	264.8	30.1	15	29	15	29	9	24	68	146
mnist38	96.0	506.6	251.4	27.2	19	28	19	28	8	20	93	157
madelon	75.2	357.8	178.2	16.7	10	20	10	20	7	18	38	103
gisette	93.3	347.8	173.2	36.0	27	39	27	39	19	34	64	113
gina	85.8	337.0	173.0	24.1	12	26	12	26	7	19	54	108
price phone	82.2	335.6	161.5	13.2	8	14	8	14	4	11	27	76
letter	99.3	317.0	95.9	15.8	6	15	6	15	1	12	49	81
titanic	75.9	274.0	116.3	17.1	7	17	7	17	4	14	22	58
yeast	97.3	68.8	33.9	18.7	15	20	15	20	7	16	26	38
soybean	96.9	46.2	19.3	11.6	2	9	2	9	1	8	8	19

	#Sufficient		#Contrastive		| Contrastive |		#Min.-Sized	
Dataset	med	max	med	max	med	max	med	max
recidivism	10387	9734080	54	145	3	16	2	144
adult	-	$\geq$ 157383572260730000000000	201	470	4	16	3	256
bank marketing	-	$\geq$ 74603752134843500000000	189	337	4	13	8	432
bank	-	$\geq$ 7433951979018500000	150	277	4	13	4	168
lending loan	459258918095775	943243242816203000000000000000	157	311	3	12	3	192
contraceptive	20,50	4272	21	52	2	11	2	48
compas	16	444	13	33	2	11	2	21
christine	63108	2167735434744	71	151	3	8	2	4096
farm-ads	1177,50	921895392	59	166	2	10	-	$\geq$ 10000
mnist49	7392384	715892613696000	61	106	2	12	-	$\geq$ 10000
spambase	15712	2535069312	50	107	2	11	4	384
mnist38	14849376	16922386736640	62	107	3	11	32	3072
madelon	106	3221020	30	72	2	9	2	32
gisette	3905	234593712	50	81	2	10	-	$\geq$ 10000
gina	4544	432967680	38	77	2	8	4	6144
price phone	109	363828	19	50	2	7	2	32
letter	9342	28391526	32	56	3	9	4	256
titanic	44	49920	16	38	2	9	2	96
yeast	128	24576	18	23	2	5	8	4608
soybean	3	60	5	15	2	6	1	20

to the instances of 3 and 8 (resp. 4 and 9) digits. Because some datasets are suited to the multi-label classification task, we used the standard "one versus all" policy to deal with them: all the classes but the target one are considered as the complementary class of the target. Categorical features have been treated as arbitrary numbers (the scale is nominal). As to numeric features, no data preprocessing has taken place: these features have been binarized on-the-fly by the decision tree learning algorithm that has been used.

For every benchmark b, a 10-fold cross validation process has been achieved. Namely, a set of ten decision trees T_b have been computed and evaluated from the labelled instances of b, partitioned into 10 parts. One part was used as the test set and the remaining 9 parts as the training set for generating a decision tree. Each T_b is thus in one-to-one correspondence with the test set chosen within the whole dataset b. The classification performance for b was measured as the mean accuracy obtained over the 10 decision trees generated from b. The CART algorithm, and more specifically its implementation provided by the Scikit-Learn library (54) has been used to learn decision trees. All hyper-parameters of the learning algorithm have been set to their default value. Notably, decision trees have been learned using the Gini criterion, and without any maximal depth or any other manual limitation.

For each benchmark b, each decision tree T_b, and a subset of at most 100 instances $\boldsymbol{x}$ picked up at random in the test set following a uniform distribution, we computed sufficient reasons for $\boldsymbol{x}$ given T_b (using the standard greedy algorithm run on the direct reason $t_{\boldsymbol{x}}^{T_b}$), and minimum-size sufficient reasons for $\boldsymbol{x}$ given T_b using the PARTIAL MAXSAT encoding presented in Proposition 4. This enabled us to draw some statistics (median, maximum) about the sizes of the reasons that have been generated. Using the algorithm presented in the proof of Proposition 2, we also derived the necessary and relevant explanatory features for each $\boldsymbol{x}$, and again drew some statistics about them. Exploiting the model counter `D4`, we computed the number of sufficient reasons for $\boldsymbol{x}$ given T_b, as well as the explanatory importance of every feature of $\boldsymbol{x}$. Taking advantage of the algorithm given in Proposition 4, we computed the number of contrastive explanations for $\boldsymbol{x}$ given T_b, and drew some statistics about those numbers and about the sizes of the contrastive explanations. Finally, using the approach described in Section 3, we enumerated all the minimum-size sufficient reasons for $\boldsymbol{x}$ given T_b up to a limit of 10 000, and again drew some statistics about the numbers of minimum-size sufficient reasons. Of course, for each computation, we measured the corresponding runtimes since this is fundamental to determine the extent to which the algorithms are practical.

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. `D4` (47) was run with its default parameters. For computing minimum-size reasons, we used the Pysat library (55), which provides the implementation of the RC2 PARTIAL MAXSAT solver. This solver was run using the parameters corresponding to the "Glucose" setting. A time-out of 100s per instance was set for `D4`.

5.2 Results

For space reasons, we report an excerpt of our results, focusing on 20 benchmarks out of 90. The selected datasets are among those containing many instances and/or many features. They are described in Table 1. The leftmost column of this table gives the name of the dataset. Columns $\#I$, $\#F$, $\#C$, and source give, respectively, the number of instances in the dataset, the number of features used to describe instances, the number of classes, and the repository the dataset comes from.

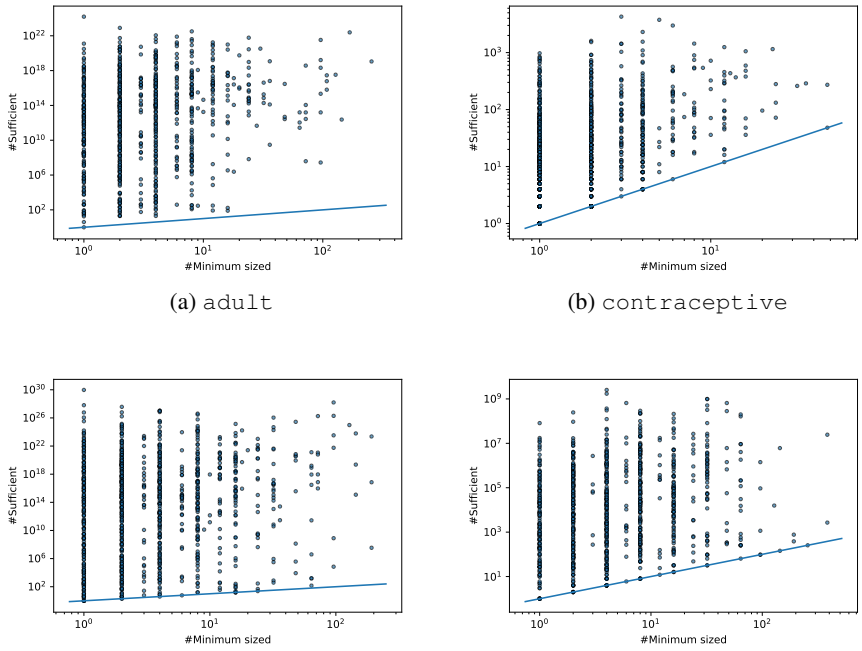
Table 2 (top and bottom) presents the results obtained for the 20 datasets given in Table 1. The leftmost column of Table 2 gives the name of the dataset b. Columns $\%A$, $\#N$, $\#B$, and $\#D$ give, respectively, the mean accuracy over the ten decision trees, the average number of nodes in those trees, the average number of binary features they are based on, and the average depth of their branches. The next columns give statistics (median, maximum) about, respectively, the size of the first sufficient reason (| Sufficient |) and of the first minimum-size sufficient reason (| Min.-Sized |) that have been computed for the instance at hand. Then one can find the numbers of necessary (#Nec. Features) and relevant (#Rel. Features) features that appear in the set of sufficient reasons for the instance. Table 2 (bottom) gives statistics (median, maximum) about, respectively, the numbers of sufficient reasons (#Sufficient) that have been computed, the numbers of contrastive explanations (#Contrastive) and their sizes (| Contrastive |), and finally the numbers of minimum-size sufficient reasons (#Min-Sized).

Figure 6 gives scatter plots for comparing the numbers of sufficient reasons with the numbers of minimum-size sufficient reasons for instances from four datasets, namely `adult`, `contraceptive`, `lending loan`, and `spambase`. Each dot corresponds to an instance (out of 1000) of the corresponding dataset for which no time out has been reached. The x-coordinate of an instance is the number of minimum-size sufficient reasons of the instance, and its y-coordinate is the number of sufficient reasons. The blue line in each scatter plot gathers the dots (x, y) such that $y = x$ (observe that different scales are used on the two axes for the sake of readability).

Figure 7 gives box plots for comparing the sizes of sufficient reasons with the sizes of minimum-size sufficient reasons for instances from four datasets (the same ones as those considered in Figure 6). For each dataset, ten instances have been picked up uniformly at random. For each of them, a minimum-size sufficient reason has been computed, as well as a number of sufficient reasons up to a maximum of 10 000. The distribution of the sizes of those reasons is reported in a box plot (the minimal value corresponds to the size of a minimum-size sufficient reason). One can observe significant differences in the resulting pictures: for some instances (e.g., instance 9 of `adult`), the obtained distribution is spread out (the minimum-size sufficient reasons are more than seven times smaller than some sufficient reasons); for other instances (e.g., instance 4 of `adult`), the distribution is quite narrow.

Table 2 also shows that, empirically, the number of contrastive explanations for an instance is typically far smaller than its number of sufficient reasons. It also shows that the sizes of contrastive explanations are in general very small. This coheres with the result stated by Proposition 5.

As to the computation times, it turns out that all the algorithms described in the previous sections proved as efficient in practice. This is not surprising for those al-



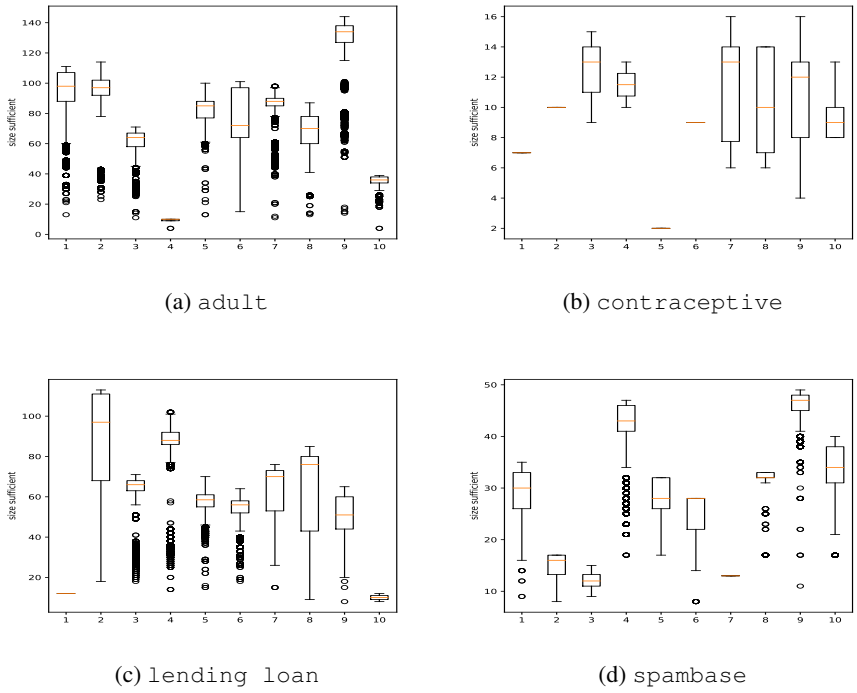
(a) `adult` (b) `contraceptive`

(c) `lending loan` (d) `spambase`

Figure 6: Comparing the numbers of sufficient reasons with the number of minimum-size sufficient reasons for instances from four datasets.

gorithms having a polynomial-time worst-case complexity (the greedy algorithm for computing a sufficient reason, the one for deriving explanatory features, and the one for computing all the contrastive explanations). It was less obvious at first sight for the algorithms used for counting the number of sufficient reasons and for computing the explanatory importance of features. However, all the computations that have been run have terminated in due time, except for instances from 3 datasets out of 90, namely `adult`, `bank_marketing`, and `bank`. For these datasets, the time limit of 100s has been reached for, respectively, 203, 150, and 336 instances out of 1000 (in this case, the median number of sufficient reasons has not been reported). Notably, for all the 90 datasets but those 3, the median time required for counting the number of sufficient reasons and computing the explanatory importance of features never exceeded 1s.

Computing a minimum-size sufficient reason, and more generally all such reasons looked challenging as well, due to both the intrinsic complexity of computing a minimum-size sufficient reason and to their number. Nevertheless, our enumeration algorithm succeeded in deriving *all the minimum-size sufficient reasons* for every dataset except 3 out of 90, namely `farm-ads`, `mnist49`, and `gisette`. For these datasets, the limit of 10 000 reasons has been reached for, respectively, 5, 16, and 3 instances out of 1000. Interestingly, the median time needed to derive all the minimum-size sufficient



(a) `adult` (b) `contraceptive`

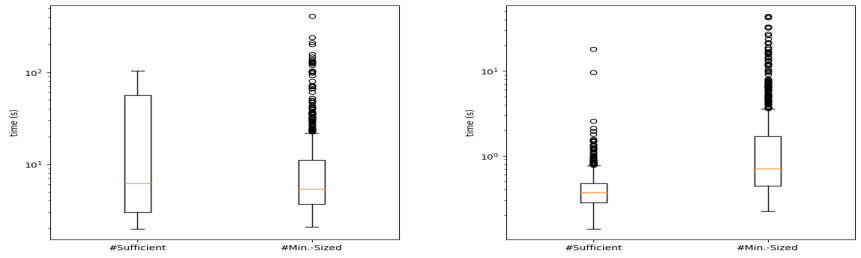
(c) `lending loan` (d) `spambase`

Figure 7: Comparing the sizes of sufficient reasons with the sizes of minimum-size sufficient reasons for instances from four datasets.

reasons for the instances for which the computation has been successful exceeded 1s only for 2 datasets (`adult` and `bank_marketing`).

Figure 8 gives box plots depicting the distributions of the computation times spent to count the sufficient reasons and the minimum-size sufficient reasons for 1000 instances from two datasets, `adult` and `lending loan`. The approach presented in Section 3 and based on the model counter `D4` has been used for solving the counting problem about sufficient reasons, and the enumeration method with blocking clauses described in the same section has been exploited to count minimum-size sufficient reasons. `D4` succeeded in counting in due time (100s) the sufficient reasons for 797 instances out of 1000. When it failed, the corresponding computation time has been set to 100s for drawing the statistics. The enumeration method succeeded in computing all the minimum-size sufficient reasons for every instance. Figure 8 clearly illustrates that the time required for counting sufficient reasons or minimum-size sufficient reasons remains small enough most of the time, despite the possibly huge number of reasons.

Interestingly, our experiments have also highlighted that the greedy algorithm for deriving a sufficient reason computes in practice a minimum-size sufficient reason in many cases. This explains the discrepancy between the results reported in Table 2 and those depicted on Figure 7. Indeed, the size of a single sufficient reason per instance



(a) `adult` (b) `lending loan`

Figure 8: Distributions of the computation times spent to count sufficient reasons and minimum-size sufficient reasons for 1000 instances from two datasets.

(the one furnished by the greedy algorithm) has been considered to compute the results reported in Table 2, while all its sufficient reasons (up to a maximum of 10 000) have been taken into account to derive the values used to draw Figure 7.

When considering the full set of reasons, a considerable difference between the number of sufficient reasons and the number of minimum-size sufficient reasons can also be observed (see Table 2 and Figure 6). Focusing on minimum-size reasons leads often to drastically reduce the number of reasons, sometimes by several orders of magnitude. For some instances, the number of minimum-size sufficient reasons is small enough so that it is conceivable to provide each of them to the explainee for further examination. This is seldom the case for sufficient reasons.

Our experiments have also shown that the number of explanatory relevant features for an instance is typically much lower than the number of binary features used to describe it, and that the number of explanatory necessary features is also significantly lower than the number of explanatory relevant features. The gap between the two explains the possibly enormous number of sufficient reasons.

Finally, like minimum-size sufficient reasons, the number of contrastive explanations appears not very large in many cases. Hopefully, this coheres with the existence of the theoretical bound - the number of branches in the tree - as pointed out in Section 4.

6 Conclusion

In light of our results, it turns out that the explanatory power of decision trees goes far beyond its ability to generate efficiently direct reasons. From a decision tree, the explanatory importance of features and the minimum-size reasons for an instance can be computed efficiently in practice most of the time.

To be more precise, from our results, fully addressing the “Why not?” question for decision trees appears as easier than fully addressing the “Why?” question: computing the full set of sufficient reasons for the instance at hand is typically out of reach, while

computing its full set of contrastive explanations is tractable. Especially, the limited robustness of decision trees, i.e., the fact that the decision made about an instance can be questioned when a few features of the instance are changed, can be explained by the fact that the size of any contrastive explanation for any instance given a decision tree is upper bounded by the depth of the tree (but this size does not depend directly on the accuracy of the tree).

Nevertheless, our results show that the full set of sufficient reasons for an instance given a decision tree can be synthesized efficiently most of the time. Computing necessary features and relevant features w.r.t. sufficient reasons is tractable. Furthermore, the computation of the explanatory importance of features w.r.t. sufficient reasons turns out to be practical very often.

Thus, the language of decision trees appears not only as appealing for the learning purpose when dealing with instances that are not too noisy, but also as a good target when one needs to reason on the various forms of explanations (abductive and contrastive ones) associated with the predictions made. This coheres with (and completes) the results reported in (35; 36), showing that many other explanation and verification tasks are tractable for decision tree classifiers.

Several extensions of this work can be envisioned.

One of them consists in focusing on restricted classes of decision trees (e.g., the class of decision trees representing monotone Boolean functions), and to determine for such subclasses whether sufficient reasons could be enumerated efficiently, and if this is not the case, to determine whether the explanatory importance of features w.r.t. sufficient reasons and the minimum-size sufficient reasons could be computed in a tractable way.

To deal with the possibly exponentially large number of sufficient reasons for an instance, another perspective is to design and evaluate approaches that generate only a subset of the whole set of sufficient reasons, provided that this subset of reasons is diverse enough, i.e., the reasons that are delivered are as dissimilar as possible. While the diversity issue has already been considered for counterfactual model-agnostic explanations (56), it would be useful to tackle it for sufficient reasons by taking account for several notions of explanation similarity.

Exploring approaches to deriving better explanations would be useful as well. In this paper, a focus was made on minimum-size explanations. The significance of such explanations comes from Occam's razor: everything else being equal, it makes sense to prefer a shorter explanation to a longer one, since a shorter explanation can be considered as more intelligible than a longer one. However, size is only one of the criteria to be considered, and though there is no consensus about what is a good explanation (57; 26; 58; 59), it is clear that many other criteria that are not intrinsic to explanations but heavily depend on the explainee must be taken into account as well. Especially, user preferences, when available, can be used to select explanations of improved quality. It has been shown that the exploitation of user preferences may drastically reduce the number of abductive explanations (60). It would be interesting to evaluate the extent to which handling user preferences (of various kinds) impacts the explanatory importance of features when dealing with decision trees.

Acknowledgements

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

References

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

[2] N. Frosst, 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, Vol. 2071 of CEUR Workshop Proceedings, CEUR-WS.org, 2017.

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

[4] S. Hooker, D. Erhan, P.-J. Kindermans, B. Kim, A benchmark for interpretability methods in deep neural networks, in: Proc. of NeurIPS'19, 2019, pp. 9737–9748.

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

[6] A. Ignatiev, N. Narodytska, J. Marques-Silva, Abduction-based explanations for machine learning models, in: Proc. of AAAI'19, 2019, pp. 1511–1519.

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

[8] S. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proc. of NIPS'17, 2017, pp. 4765–4774.

[9] M. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?": Explaining the predictions of any classifier, in: Proc. of KDD'16, 2016, pp. 97–101.

[10] A. Shih, A. Darwiche, A. Choi, Verifying binarized neural networks by Angluin-style learning, in: Proc. of SAT'19, 2019, pp. 354–370.

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

[12] A. B. Arrieta, N. D. R., J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, F. Herrera, Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI, Inf. Fusion 58 (2020) 82–115. doi:10.1016/j.inffus.2019.12.012.
URL https://doi.org/10.1016/j.inffus.2019.12.012

[13] S. M. Lundberg, G. G. Erion, H. Chen, A. J. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, S. Lee, From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell. 2 (1) (2020) 56–67. doi:10.1038/s42256-019-0138-9.
URL https://doi.org/10.1038/s42256-019-0138-9

[14] R. Caruana, S. M. Lundberg, M. T. Ribeiro, H. Nori, S. Jenkins, Intelligible and explainable machine learning: Best practices and practical challenges, in: R. Gupta, Y. Liu, J. Tang, B. A. Prakash (Eds.), KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020, ACM, 2020, pp. 3511–3512. doi:10.1145/3394486.3406707.
URL https://doi.org/10.1145/3394486.3406707

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

[16] T. Laugel, M. Lesot, C. Marsala, X. Renard, M. Detyniecki, The dangers of post-hoc interpretability: Unjustified counterfactual explanations, in: Proc. of IJCAI'19, 2019, pp. 2801–2807.

[17] T. Laugel, M. Lesot, C. Marsala, X. Renard, M. Detyniecki, Unjustified classification regions and counterfactual explanations in machine learning, in: Proc. of ECML/PKDD'19, 2019, pp. 37–54.

[18] J. Marques-Silva, A. Ignatiev, Delivering trustworthy AI through formal XAI, in: Proc. of AAAI'22, 2022.

[19] A. Ignatiev, N. Narodytska, J. Marques-Silva, On validating, repairing and refining heuristic ML explanations, CoRR abs/1907.02509. arXiv:1907.02509.
URL http://arxiv.org/abs/1907.02509

[20] A. Ignatiev, Towards trustable explainable AI, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org, 2020, pp. 5154–5158. doi:10.24963/ijcai.2020/726.
URL https://doi.org/10.24963/ijcai.2020/726

[21] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: Proc. of SIGKDD'16, 2016, pp. 1135–1144.

[22] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: Proc. of AAAI'18, 2018, pp. 1527–1535.

[23] A. Ignatiev, N. Narodytska, J. Marques-Silva, Abduction-based explanations for machine learning models, in: Proc. of AAAI'19, 2019, pp. 1511–1519.

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

[25] X. Huang, Y. Izza, A. Ignatiev, M. C. Cooper, N. Asher, J. Marques-Silva, Efficient explanations for knowledge compilation languages, CoRR abs/2107.01654. arXiv:2107.01654.
URL https://arxiv.org/abs/2107.01654

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

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

[28] J. R. Quinlan, Induction of decision trees, Machine Learning 1 (1) (1986) 81–106.

[29] L. Breiman, N. Shang, Born again trees, Tech. rep., https://www.stat.berkeley.edu/ breiman/BAtrees.pdf (1996).

[30] L. Breiman, Random forests, Machine Learning 45 (1) (2001) 5–32.

[31] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proc. of KDD'16, 2016, p. 785–794.

[32] Y. Izza, A. Ignatiev, J. Marques-Silva, On explaining decision trees, CoRR abs/2010.11034.

[33] A. Darwiche, A. Hirth, On the reasons behind decisions, in: Proc. of ECAI'20, 2020, pp. 712–720.

[34] A. Shih, A. Choi, A. Darwiche, A symbolic approach to explaining Bayesian network classifiers, in: Proc. of IJCAI'18, 2018, pp. 5103–5111.

[35] G. Audemard, F. Koriche, P. Marquis, On tractable XAI queries based on compiled representations, in: Proc. of KR'20, 2020, pp. 838–849.

[36] G. Audemard, S. Bellart, L. Bounia, F. Koriche, J. Lagniez, P. Marquis, On the computational intelligibility of boolean classifiers, in: Proc. of KR'21, 2021, pp. 74–86. doi:10.24963/kr.2021/8.

[37] Y. Izza, A. Ignatiev, J. Marques-Silva, On tackling explanation redundancy in decision trees, CoRR abs/2205.09971. arXiv:2205.09971, doi:10.48550/arXiv.2205.09971.
URL https://doi.org/10.48550/arXiv.2205.09971

[38] A. Shih, A. Choi, A. Darwiche, Formal verification of Bayesian network classifiers, in: Proc. of PGM'18, 2018, pp. 427–438.

[39] A. Darwiche, P. Marquis, On quantifying literals in Boolean logic and its applications to explainable AI, J. Artif. Intell. Res. 72 (2021) 285–328. doi:10.1613/jair.1.12756.

[40] V. Gurvich, L. Khachiyan, On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions, Discret. Appl. Math. 96-97 (1999) 363–373.

[41] A. de Colnet, P. Marquis, On the complexity of enumerating prime implicants from decision-DNNF circuits, in: Proc. of IJCAI-ECAI'22, 2022.

[42] T. Eiter, G. Gottlob, The complexity of logic-based abduction, Journal of the Association for Computing Machinery 42 (1) (1995) 3–42.

[43] X. Huang, Y. Izza, A. Ignatiev, J. Marques-Silva, On efficiently explaining graph-based classifiers, in: Proc. of KR'21, 2021, pp. 356–367. doi:10.24963/kr.2021/34.
URL https://doi.org/10.24963/kr.2021/34

[44] S. Jabbour, J. Marques-Silva, L. Sais, Y. Salhi, Enumerating prime implicants of propositional formulae in conjunctive normal form, in: Proc. of JELIA'14, 2014, pp. 152–165.

[45] G. Tseitin, On the complexity of derivation in propositional calculus, Steklov Mathematical Institute, 1968, Ch. Structures in Constructive Mathematics and Mathematical Logic, pp. 115–125.

[46] J. Lang, P. Marquis, On Propositional Definability, Artificial Intelligence 172:8-9 (2008) 991–1017.

[47] J.-M. Lagniez, P. Marquis, An Improved Decision-DNNF Compiler, in: Proc. of IJCAI'17, 2017, pp. 667–673.

[48] A. Darwiche, Decomposable negation normal form, Journal of the Association for Computing Machinery 48 (4) (2001) 608–647.

[49] A. Darwiche, P. Marquis, A knowledge compilation map, Journal of Artificial Intelligence Research 17 (2002) 229–264.

[50] P. Barceló, M. Monet, J. Pérez, B. Subercaseaux, Model interpretability through the lens of computational complexity, in: Proc. of NeurIPS'20, 2020.

[51] P. Slavík, A tight analysis of the greedy algorithm for set cover, Journal of Algorithms 25 (2) (1997) 237 – 254.

[52] R. Reiter, A theory of diagnosis from first principles, Artificial Intelligence 32 (1987) 57–95.

[53] F. Wotawa, A variant of Reiter's hitting-set algorithm, Inf. Process. Lett. 79 (1) (2001) 45–51.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[55] A. Ignatiev, A. Morgado, J. Marques-Silva, PySAT: A Python toolkit for prototyping with SAT oracles, in: Proc. of SAT'18, 2018, pp. 428–437.
URL https://github.com/pysathq/pysat

[56] R. K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: M. Hildebrandt, C. Castillo, L. E. Celis, S. Ruggieri, L. Taylor, G. Zanfir-Fortuna (Eds.), FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020, ACM, 2020, pp. 607–617. doi:10.1145/3351095.3372850.
URL https://doi.org/10.1145/3351095.3372850

[57] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, CoRR abs/1702.08608.
URL http://arxiv.org/abs/1702.08608

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

[59] R. Srinivasan, A. Chander, Explanation perspectives from the cognitive sciences - A survey, in: Proc. of IJCAI'20, 2020, pp. 4812–4818.

[60] G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, P. Marquis, On preferred abductive explanations for decision trees and random forests, in: Proc. of IJCAI'22, 2022.

[61] J. Lang, P. Liberatore, P. Marquis, Propositional independence: Formula-variable independence and forgetting, Journal of Artificial Intelligence Research 18 (2003) 391–443.

[62] P. Marquis, Consequence finding algorithms, Vol. 5 of Handbook on Defeasible Reasoning and Uncertainty Management Systems, Kluwer Academic Publisher, 2000, Ch. 2, pp. 41–145.

Proofs

Proof of Proposition 1

Proof. Let T be the complete binary tree of depth k, formed by $n = 2^k - 1$ internal nodes and 2^k leaves. We assume a breadth-first ordering of internal nodes, such that the root is labeled by x_1, the nodes of depth 1 are labeled by x_2 and x_3, and so on. Each internal node at depth $k - 1$ from the root of T has two children, one of it is

a 0-leaf and the other one is a 1-leaf. For an arbitrary instance $\boldsymbol{x} \in \{0,1\}^n$ and any complete subtree T' of T of depth d, let $s(\boldsymbol{x}, T')$ denote the set of sufficient reasons of $\boldsymbol{x}$ given T', and let $\sigma(\boldsymbol{x}, d) = |s(\boldsymbol{x}, T')|$ denote the number of those sufficient reasons. We show by induction on d that:

$$\sigma(\boldsymbol{x}, 1) = 1 \tag{3}$$

$$\sigma(\boldsymbol{x}, d+1) = \sigma(\boldsymbol{x}, d)(\sigma(\boldsymbol{x}, d) + 1) \tag{4}$$

For the base case (3), any complete subtree T' of T of depth $d = 1$ has a single internal node, say x_i, with two leaves labeled by 0 and 1, respectively. Therefore, the unique sufficient reason for $\boldsymbol{x}$ given T' is either x_i or $\overline{x}_i$, and hence, $\sigma(\boldsymbol{x}, 1) = 1$. Now, consider any complete subtree T' of T of depth $d+1$ rooted at a node x_i. Let $T'_l(x_i)$ and $T'_r(x_i)$ denote the subtrees of depth d, respectively rooted at the left child of x_i and the right child of x_i. Suppose without loss of generality that the unique path leading to $T'(\boldsymbol{x}) = 1$ includes the left child of x_i (i.e. $T'_l(\boldsymbol{x}) = 1$). By construction,

$$\begin{aligned} s(\boldsymbol{x}, T') = \{t_l \wedge t_r : t_l \in s(\boldsymbol{x}, T'_l), t_r \in s(\boldsymbol{x}, T'_r)\} \\ \cup \{l_i \wedge t_l : t_l \in s(\boldsymbol{x}, T'_l)\} \end{aligned}$$

where $l_i = \overline{x}_i$ if $x_i = 0$ in $\boldsymbol{x}$, and $l_i = x_i$ otherwise. Since by induction hypothesis $s(\boldsymbol{x}, T'_l) = s(\boldsymbol{x}, T'_r) = \sigma(\boldsymbol{x}, d)$, it follows that $\sigma(\boldsymbol{x}, d+1) = \sigma(\boldsymbol{x}, d)^2 + \sigma(\boldsymbol{x}, d)$. Finally, since the doubly exponential sequence[5] given by $a(1) = 1$ and $a(d+1) = a(d)^2 + a(d)$ satisfies $a(d) = \lfloor c^{2^{d-1}} \rfloor$, where $c \sim 1.59791$, it follows that $\sigma(\boldsymbol{x}, k) \geq \lfloor (3/2)^{2^{k-1}} \rfloor$. Using $2^{k-1} = (n+1)/2$, we get the desired result. □

Proof of Proposition 2

Proof. The algorithms to compute $Nec_s(\boldsymbol{x}, T)$, $Rel_s(\boldsymbol{x}, T)$, and $Irr_s(\boldsymbol{x}, T)$ are as follows: first compute $\texttt{CNF}(T)$ and then remove from this set of clauses every literal that does not belong to $t_{\boldsymbol{x}}$. This can be done in $\mathcal{O}(n \cdot |T|)$ time. By construction, the resulting CNF formula (say, f) is monotone: every literal in it occurs with the same polarity as the one it has in $t_{\boldsymbol{x}}$. Furthermore, the size of f cannot exceed the size of $\texttt{CNF}(T)$, thus the size of T.

Since f is a monotone CNF formula, its prime implicates can be computed by removing from f every clause that is a strict superset of another clause of f.This can be achieved in quadratic time in the size of f, thus in the size of T. Let g be the resulting formula in prime implicates form and equivalent to f. g is equivalent to the complete reason for $\boldsymbol{x}$ given T. Since it is in prime implicates form, g is Lit-dependent on every literal occurring in it (i.e., g is Lit-simplified, see Proposition 8 in (61) for details), hence so is the complete reason for $\boldsymbol{x}$ given T.

This means that for every literal ℓ occurring in g, there exists a sufficient reason for $\boldsymbol{x}$ given T that contains ℓ, so that $Rel_s(\boldsymbol{x}, T)$ is the set of literals occurring in g and $Irr_s(\boldsymbol{x}, T)$ is the complement of $Rel_s(\boldsymbol{x}, T)$ in the set of all literals over X_n. Finally, since by definition the literals of $Nec_s(\boldsymbol{x}, T)$ must belong to every sufficient reason for $\boldsymbol{x}$ given T, they are given by the unit clauses that belong to g. □

[5] See https://oeis.org/A007018.

Proof of Proposition 4

Proof. Let $\boldsymbol{x}^*$ be any solution of $(C_{\rm soft}, C_{\rm hard})$. Observe that the set of all hard clauses $c \cap t_{\boldsymbol{x}}$ (where c is a clause of $\texttt{CNF}(T)$) corresponds to a *monotone* $\texttt{CNF}$ formula. Therefore, in order to satisfy such a clause $c \cap t_{\boldsymbol{x}}$, $\boldsymbol{x}^*$ must set a literal ℓ of $t_{\boldsymbol{x}}$ to 1. Thus, $\boldsymbol{x}^*$ satisfies all the hard clauses of $C_{\rm hard}$ if and only if the term consisting of the literals that are shared by $t_{\boldsymbol{x}} = \bigwedge_{i=1}^{n} \ell_i$ and $t_{\boldsymbol{x}^*}$ is an implicant of T and is implied by $\boldsymbol{x}$.

The soft clauses of $C_{\rm soft}$ are used to select among the assignments that satisfy all the hard clauses, the ones that correspond to minimum-size sufficient reasons. Soft clauses are given by literals ℓ_i, which are precisely the complementary literals to those occurring in $t_{\boldsymbol{x}}$. Having a soft clause ℓ_i violated by $\boldsymbol{x}^*$ means that the literal $\overline{\ell}$ of $t_{\boldsymbol{x}}$ is necessary to get an implicant of T given the assignment of the other variables in $\boldsymbol{x}^*$. Whenever a soft clause ℓ_i is violated by $\boldsymbol{x}^*$ a penalty of 1 incurs. This ensures that the term consisting of the literals that are shared by $t_{\boldsymbol{x}} = \bigwedge_{i=1}^{n} \ell_i$ and $t_{\boldsymbol{x}^*}$ is a minimum-size sufficient reason for $\boldsymbol{x}$ given T. □

Proof of Proposition 5

Proof. Let $f \in \mathcal{F}_n$ and $\boldsymbol{x} \in \{0,1\}^n$ such that $f(\boldsymbol{x}) = 1$ (the case when $f(\boldsymbol{x}) = 0$ can be handled in the same way by considering $\neg f$ instead of f). By definition, the sufficient reasons t for $\boldsymbol{x}$ given f are the prime implicants of f that covers $\boldsymbol{x}$. Thus, they are precisely the prime implicants of the (conjunctively-interpreted) set of clauses $\{c \cap t_{\boldsymbol{x}} : c \in \texttt{CNF}(f)\}$ where $\texttt{CNF}(f)$ is any $\texttt{CNF}$ formula equivalent to f. Furthermore, the complete reason for $\boldsymbol{x}$ given f (equivalent to the disjunction of all the sufficient reasons for $\boldsymbol{x}$ given f (33)) is a monotone Boolean function because every sufficient reason covers $\boldsymbol{x}$ which assigns in a unique way every variable from X_n. The prime implicates of such a monotone function are precisely the minimal hitting sets of the prime implicants of the function. Because of the minimal hitting set duality between sufficient reasons and contrastive explanations for $\boldsymbol{x}$ given f (24), the contrastive explanations for $\boldsymbol{x}$ given f are thus the sets of literals corresponding to the prime implicates of $\{c \cap t_{\boldsymbol{x}} : c \in \texttt{CNF}(f)\}$. Now, since the (conjunctively-interpreted) set of clauses $\{c \cap t_{\boldsymbol{x}} : c \in \texttt{CNF}(f)\}$ is equivalent to the complete reason for $\boldsymbol{x}$ given f, it is a monotone function, and as a consequence, its prime implicates are its minimal elements w.r.t. $\subseteq$. This comes from the correctness of any resolution-based algorithm for generating prime implicates (see e.g., (62)). Finally, when f is a decision tree T, $\{c \cap t_{\boldsymbol{x}} : c \in \texttt{CNF}(T)\}$ can be computed in time polynomial in $n + |T|$ because $\texttt{CNF}(T)$ can be computed in time linear in $|T|$. Using an extra quadratic time in the size of this set $\{c \cap t_{\boldsymbol{x}} : c \in \texttt{CNF}(T)\}$, its minimal elements w.r.t. $\subseteq$ can be selected. The resulting set is by construction the set of all the contrastive explanations for $\boldsymbol{x}$ given T, and this set has been computed in time polynomial in $n + |T|$. □