

Proofs

Proof of Proposition 1

Proof.

- **Membership:** Suppose that $f(x) = 1$. x has a k -anchored abductive explanation t given f and Σ if and only if one can guess a term t over X and check in (deterministic) polynomial time in the size of the input that (1) $t \subseteq t_x$, (2) there exist at least k instances $x' \in R_C^+$ such that $t \subseteq t_{x'}$, (3) there is no instance $x' \in R_C^-$ such that $t \subseteq t_{x'}$, and finally (4) check that t is an implicant of $\Sigma \Rightarrow f$. In order to achieve the latter test, one call to an NP oracle is required in the general case. Indeed, t is not an implicant of $\Sigma \Rightarrow f$ if and only if there exists $x' \in X$ such that $t \subseteq t_{x'}$ and $f(x') = 0$. Once a n -uple x' has been guessed, one just need to test in (deterministic) polynomial time that x' satisfies Σ , that $t \subseteq t_{x'}$ and finally that $f(x') = 0$. The case when $f(x) = 0$ is similar (in the guess & check algorithm above, replace R_C^+ by R_C^- and vice-versa, and replace f by $\neg f$).
- **Hardness:** By reduction from the problem of deciding whether an instance $x \in X$ has an abductive explanation of size $\leq s$ for a binary classifier f (s is a non-negative integer $< n$). This problem has been shown to be Σ_2^P -hard even in the restricted case when f is a random forest over a set $X = \{x_1, \dots, x_n\}$ of logically independent variables (i.e., $\Sigma = \top$), see Proposition 5 from [Audemard *et al.*, 2022]. Thus, let us assume that $\Sigma = \top$. Suppose also that $f(x) = 1$. With f and x let us associate in polynomial time $R_C^- = \emptyset$ and $R_C^+ = \{x' \in X : d_H(x, x') = 1\}$ the set of n instances x' being at Hamming distance 1 from x . The point is that $x \in X$ has an abductive explanation given f and Σ , that is of size $\leq s$ with $s < n$ if and only if $x \in X$ has a k -anchored abductive explanation given f and Σ , with $k = n - s$. Indeed, $x \in X$ has a $n - s$ -anchored abductive explanation given f and Σ iff there exists a term t that is an implicant of $\Sigma \Rightarrow f$ (or, equivalently, an implicant of f since $\Sigma = \top$) covering at least $n - s$ instances x' from R_C^+ iff there exists a term t that is an implicant of f and a subset of the intersection of $t_{x'}$ for at least $n - s$ instances x' from R_C^+ . Since the elements of R_C^+ are all at Hamming distance 1 from x , the intersection of $t_{x'}$ for at least $n - s$ instances x' from R_C^+ contains at most s literals. Furthermore, every implicant of f that contains at most s literals covers at least $n - s$ instances x' from R_C^+ . Altogether, $x \in X$ has a $n - s$ -anchored abductive explanation given f and Σ iff there exists a term t that is an implicant of f and that contains at most s literals, which completes the proof. \square

The complexity of deciding whether an instance x has a k -anchored abductive explanation given f and Σ can be lowered by considering additional assumptions about the language used to represent f and the domain theory Σ under consideration. A propositional formula (or a Boolean circuit)

Σ is said to be *tractable* for clausal entailment, i.e., there exists a polynomial-time algorithm that takes as inputs Σ and any clause c over X and returns 1 when c is a logical consequence of Σ , and returns 0 otherwise. Now, let \mathcal{L} be a propositional language of representations φ of binary classifiers over X . \mathcal{L} is said to satisfy the *constrained implicant query* if and only if there exists a polynomial-time algorithm that takes as inputs a term t over X , a propositional formula (or a Boolean circuit) Σ over X that is tractable for clausal entailment, a representation φ in \mathcal{L} and a Boolean value b , and that returns 1 if t is an implicant of $\Sigma \Rightarrow \varphi^b$ and 0 otherwise, where $\varphi^b = \varphi$ when $b = 1$ and $\varphi^b = \neg\varphi$ when $b = 0$.

It turns out that the language \mathcal{L} of decision trees over X satisfies the constrained implicant query. This comes from the fact that when f is given as a decision tree, one can turn f and $\neg f$ in linear time into equivalent CNF formulae: $f \equiv \bigwedge_{i=1}^p c_i$ and $\neg f \equiv \bigwedge_{i=1}^q c'_i$, where each c_i ($i \in [p]$) and each c'_i ($i \in [q]$) is a clause over X . Indeed, it is well-known that any decision tree f can be encoded in linear time into an equivalent disjunction of terms, where each term used coincides with a 1-path of f (i.e., a path from the root to a leaf labeled with 1), but also as a conjunction of clauses, where each clause used is the negation of a term describing a 0-path of f . Furthermore, every decision tree f can be negated in linear time (replacing every 1-leaf of f by a 0-leaf and every 0-leaf of f by a 1-leaf leads to a decision tree equivalent to $\neg f$). Then t is an implicant of $\Sigma \Rightarrow f$ (resp. $\Sigma \Rightarrow \neg f$) if and only if each of the p (resp. q) clauses $\neg t \vee c_i$ (resp. $\neg t \vee c'_i$) is a logical consequence of Σ , which can be tested in polynomial time when Σ is tractable. Interestingly, the domain theory issued from the Boolean encoding of numerical attributes, as used in tree-based ML models, are tractable for clausal entailment (they consist of conjunctions of binary clauses).

This leads to the following proposition:

Proposition 2. *Given a domain theory about X (represented by a propositional formula or a Boolean circuit Σ), a binary classifier f over X (represented by a propositional formula or a Boolean circuit from a propositional language \mathcal{L} satisfying the constrained implicant query), an instance $x \in X$, a set $R_C \subseteq X$ of reference instances and an integer $k > 0$, the problem of deciding whether an instance $x \in X$ has a k -anchored abductive explanation given f and Σ is NP-complete.*

Proof of Proposition 2

Proof.

- **Membership:** Consider again the guess & check algorithm given in the proof of Proposition 1. The three check steps (1), (2), and (3) can be achieved in (deterministic) polynomial time, and this is also the case of step (4) when the language \mathcal{L} into which f is represented offers a polynomial-time constrained implicant test.
- **Hardness:** When \mathcal{L} satisfies the constrained implicant query, NP-hardness is the case even if $\Sigma = \top$ and f is represented by a decision tree. The result comes from the same reduction as pointed out in the proof of Proposition 1, but assuming now that f is a decision tree over

880 X . Indeed, deciding whether an instance $x \in X$ has an
881 abductive explanation of size $\leq s$ for a decision tree f
882 over X is \mathbf{NP} -complete (see Proposition 6 from [Barceló
883 *et al.*, 2020]).

884

□