Approximating Probabilistic Explanations via Supermodular Minimization

Louenas Bounia[1] **Frederic Koriche**[1]

[1]CRIL UMR CNRS 8188, Université d'Artois, France

Abstract

Explaining in accurate and intelligible terms the predictions made by classifiers is a key challenge of eXplainable Artificial Intelligence (XAI). To this end, an *abductive explanation* for the predicted label of some data instance is a subset-minimal collection of features such that the restriction of the instance to these features is sufficient to determine the prediction. However, due to cognitive limitations, abductive explanations are often too large to be interpretable. In those cases, we need to reduce the size of abductive explanations, while still determining the predicted label with high probability. In this paper, we show that finding such *probabilistic explanations* is NP-hard, even for decision trees. In order to circumvent this issue, we investigate the *approximability* of probabilistic explanations through the lens of supermodularity. We examine both greedy descent and greedy ascent approaches for supermodular minimization, whose approximation guarantees depend on the curvature of the "unnormalized" error function that evaluates the precision of the explanation. Based on various experiments for explaining decision tree predictions, we show that our greedy algorithms provide an efficient alternative to the state-of-the-art constraint optimization method.

1 INTRODUCTION

Basically, the *classification* problem is to extrapolate from a set of labeled data instances a hypothesis, or *classifier*, that accurately predicts the labels of new, incoming data instances. Decision trees, random forests, support vector machines, and neural nets, are common examples of classifiers for which theoretical properties have been extensively studied in the machine learning literature (see e.g. Sayed [2022] for a recent survey). The spectrum of applications for these classifiers is wide, ranging from document and image classification, to customer profiling and medical diagnosis. However, with the increasing deployment of data-driven learning models in our society comes the issue of *explaining* predictions in human intelligible terms. As a key topic of eXplainable Artificial Intelligence (XAI), this issue is exacerbated in sensitive domains, such as cybersecurity and healthcare, where explanations are crucial for building trust and confidence in the classifier [Guidotti et al., 2019, Miller, 2019, Samek et al., 2019, Molnar, 2020].

Among the various types of explanations proposed in the XAI literature, *formal* explanations are particularly interesting, since their soundness can be mathematically validated [Marques-Silva and Ignatiev, 2022]. Notably, when the classifier h is a Boolean function, a common explanation for predicting the output $h(\boldsymbol{x})$ of some data instance $\boldsymbol{x}$ is a subset-minimal collection of features I such that the restriction $\boldsymbol{x}_I$ of $\boldsymbol{x}$ to I determines $h(\boldsymbol{x})$. Such an *abductive* explanation [Ignatiev et al., 2019], also called *sufficient reason* [Darwiche and Hirth, 2020], is logically sound, because $\boldsymbol{x}_I$ can be viewed as a prime implicant of the hypothesis h that covers the instance $\boldsymbol{x}$ [Shih et al., 2018]. Although finding an abductive explanation is NP-hard in general, tractable cases have been identified for various hypothesis classes [Marques-Silva et al., 2020, Audemard et al., 2021, Huang et al., 2021, Cooper and Marques-Silva, 2023].

However, the soundness of explanations is not the only criterion for clarifying in intelligible terms the predictions made by classifiers. The *conciseness* property is also important, since an abductive explanation involving too many features cannot be understood by human users. Indeed, in cognitive psychology, it has long been recognized that there is an upper limit on our ability to reason about simultaneously interacting elements. As conjectured by Miller [1956], this limit is seven plus or minus two elements and, since then, it has been confirmed by many experiments in cognitive science. Thus, restricting the size of explanations appears as a *constraint* for ensuring their intelligibility.

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

Based on these considerations, how can we reduce the size of explanations while retaining much of their soundness? This is where *probabilistic explanations* [Wäldchen et al., 2021] come into the equation. Namely, suppose we are given an abductive explanation I for a classifier h and some instance $\boldsymbol{x}$, together with a size limit $k \leq |I|$. For any candidate subset S of I, let $\epsilon_{h,\boldsymbol{x}}(S)$ denote the probability that a random instance $\boldsymbol{y}$ covered by $\boldsymbol{x}_S$ is classified differently from $\boldsymbol{x}$ by h. In other words, $\epsilon_{h,\boldsymbol{x}}(S)$ is the probability of making an "explanation mistake" for inferring $h(\boldsymbol{x})$, using $\boldsymbol{x}_S$ instead of $\boldsymbol{x}$. With this notion in hand, the main problem considered in this study is to find a probabilistic explanation $S \subseteq I$ of size at most k such that $\epsilon_{h,\boldsymbol{x}}(S)$ is minimized.

Unfortunately, this optimization task is very expensive from a computational viewpoint. Indeed, the problem of finding a minimizer S of $\epsilon_{h,\boldsymbol{x}}(\cdot)$ subject to some cardinality constraint $|S| \leq k$ is $\mathrm{NP}^{\mathrm{PP}}$-hard for general classifiers [Wäldchen et al., 2021], and NP-hard for decision trees [Arenas et al., 2022]. As shown in the present study, this problem remains NP-hard for decision trees even in the restricted case where S is a subset of some given abductive explanation I.

In order to overcome such a computational barrier, this paper investigates the *approximability* of probabilistic explanations through the lens of supermodularity. As $\epsilon_{h,\boldsymbol{x}}(S)$ can be viewed as the number $\mu_{h,\boldsymbol{x}}(S)$ of mistakes induced from the choice of S, averaged over the number of instances covered by $\boldsymbol{x}_S$, our results exploit two key properties: (i) the unnormalized error function $\mu_{h,\boldsymbol{x}}(\cdot)$ is *supermodular* and *non-increasing*, and (ii) the normalization factor is *constant* for all subsets S with the same size. Thus, even if $\epsilon_{h,\boldsymbol{x}}(\cdot)$ is not supermodular, we can still use approximation algorithms for supermodular minimization, by coupling them with a level-wise selection method, in order to derive probabilistic explanations endowed with approximation guarantees.

To this point, it is well-known that the task of maximizing a non-decreasing submodular function subject to a cardinality constraint is $(1-\frac{1}{e})$-approximable [Nemhauser and Wolsey, 1978]. The situation is however different for minimizing non-increasing supermodular functions: the problem is not approximable to within a constant, unless $\mathrm{P} = \mathrm{NP}$ [Mittal and Schulz, 2013]. Still, approximation factors can be provided by taking into account the *curvature* of the objective function [Il'ev, 2001, Sviridenko et al., 2017].

In this paper, we present two conceptually simple and easy-to-implement algorithms, whose approximation factors depend on the curvature c of the function $\mu_{h,\boldsymbol{x}}(\cdot)$. The first algorithm is a *greedy descent* method that achieves a $\frac{e^p-1}{p}$-approximation, where $p = \frac{c}{1-c}$, and the second algorithm is a *greedy ascent* method that achieves a $\frac{1}{1-c}$-approximation. The sizes of the greedy descent and greedy ascent solutions are bounded by k and $k \ln\left(\frac{2e}{c}\right)$, respectively.

Both algorithms are empirically compared with the constraint-based approach suggested in [Arenas et al., 2022], which aims at inferring *optimal* probabilistic explanations for decision tree predictions. Experimental results indicate that our greedy algorithms can efficiently find accurate explanations and, unlike the constraint-based approach, they are able to scale up on high-dimensional explanation tasks.

This paper is organized as follows. The main concepts relating to probabilistic explanations and supermodular minimization are introduced in Section 2 and Section 3, respectively. Our approximation algorithms are theoretically analyzed in Section 4, and empirically validated in Section 5. Finally, the related work and some perspectives of further research are discussed in Section 6.

2 PROBABILISTIC EXPLANATIONS

In this section, we start with some background about probabilistic explanations, and then, we examine some computational aspects related to their evaluation and optimization.

2.1 NOTATION AND PROBLEM FORMULATION

For a positive integer d, we use $[d]$ to denote the set $\{1, \cdots, d\}$. The classifiers under consideration in this study are hypotheses of the form $h : \{0,1\}^d \to \{0,1\}$. Thus, any input of h is d-dimensional Boolean vector $\boldsymbol{x}$, called *instance*, and the output of $h(\boldsymbol{x})$ is a Boolean value, classifying $\boldsymbol{x}$ as a negative example or a positive one. A *partial instance* is a vector $\boldsymbol{z} \in \{0,1,*\}^d$, where $z_i = *$ indicates that the ith feature of $\boldsymbol{z}$ is left undefined. An instance $\boldsymbol{x}$ is *covered* by $\boldsymbol{z}$, if $x_i = z_i$ for all features $i \in [d]$ such that $z_i \neq *$. For a subset $S \subseteq [d]$, the *restriction* of $\boldsymbol{x}$ to S, denoted $\boldsymbol{x}_S$, is the partial instance in $\{0,1,*\}^d$ such that, for each $i \in [d]$, $(x_S)_i = x_i$ if $i \in S$, and $(x_S)_i = *$ otherwise. Clearly, any instance $\boldsymbol{y} \in \{0,1\}^d$ is covered by $\boldsymbol{x}_S$ if and only if $\boldsymbol{y}_S = \boldsymbol{x}_S$.

Given a classifier h, and an instance $\boldsymbol{x}$ for which the prediction $h(\boldsymbol{x})$ must be explained, let $\epsilon_{h,\boldsymbol{x}} : 2^{[d]} \to \mathbb{R}$ denote the *error function* given by

$$\epsilon_{h,\boldsymbol{x}}(S) = \frac{|\{\boldsymbol{y} \in \{0,1\}^d : h(\boldsymbol{y}) \neq h(\boldsymbol{x}), \boldsymbol{y}_S = \boldsymbol{x}_S\}|}{|\{\boldsymbol{y} \in \{0,1\}^d : \boldsymbol{y}_S = \boldsymbol{x}_S\}|} \quad (1)$$

As indicated above, $\epsilon_{h,\boldsymbol{x}}(S)$ can be thought as the probability of making an "explanation mistake" when using the partial instance $\boldsymbol{x}_S$ instead of the complete instance $\boldsymbol{x}$. Given a precision parameter $\varepsilon \in [0,1]$, an explanation S is called $(1-\varepsilon)$-*probable* if $\epsilon_{h,\boldsymbol{x}}(S) \leq \varepsilon$. We say that S is *abductive* if $\epsilon_{h,\boldsymbol{x}}(S) = 0$, and $\epsilon_{h,\boldsymbol{x}}(S') > 0$ for every proper subset S' of S. Note that (1) can be rewritten as

$$\epsilon_{h,\boldsymbol{x}}(S) = \frac{\mu_{h,\boldsymbol{x}}(S)}{2^{d-|S|}} \quad (2)$$

where $\mu_{h,\boldsymbol{x}}(S)$ is the number of mistakes induced from the

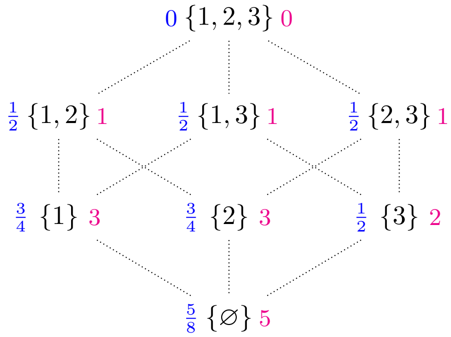

Figure 1: The error $\epsilon_{h,\boldsymbol{x}}(S)$ (in blue) and the number of mistakes $\mu_{h,\boldsymbol{x}}(S)$ (in magenta) for each $S \subseteq [3]$, using the classifier h given by (4) and the instance $\boldsymbol{x} = (1, 1, 1)$.

choice of S, that is,

$$\mu_{h,\boldsymbol{x}}(S) = |\{\boldsymbol{y} \in \{0,1\}^d : h(\boldsymbol{y}) \neq h(\boldsymbol{x}), \boldsymbol{y}_S = \boldsymbol{x}_S\}| \quad (3)$$

Example 1. Consider the classifier $h : \{0,1\}^3 \to \{0,1\}$ specified by the polynomial threshold function:

$$h(\boldsymbol{x}) = 1 \Leftrightarrow x_1x_2x_3 + x_1x_2 - x_1 - x_2 \geq 0 \qquad (4)$$

Given the instance $\boldsymbol{x} = (1, 1, 1)$ for which we need to explain $h(\boldsymbol{x}) = 1$, and using the Hasse diagram in Figure 1, it follows that $\{1, 2, 3\}$ is the only abductive explanation for h and $\boldsymbol{x}$. Yet, $\{1, 2\}$ and $\{3\}$ are both subset-minimal $\frac{1}{2}$-probable explanations for h and $\boldsymbol{x}$.

With these notions in hand, we are now in position to formulate the main problem considered in this study.

Problem 1. Given a classifier $h : \{0,1\}^d \to \{0,1\}$, an instance $\boldsymbol{x} \in \{0,1\}^d$, a set $I \subseteq [d]$ of features, a size limit $k \leq |I|$, find a subset $S \subseteq I$ of size at most k such that $\epsilon_{h,\boldsymbol{x}}(S)$ is minimized.

2.2 EVALUATING EXPLANATION ERRORS

It is easy to see that the problem of evaluating $\epsilon_{h,\boldsymbol{x}}(S)$ is #P-hard in general. However, Izza et al. [2022a] have shown that $\epsilon_{h,\boldsymbol{x}}(S)$ can be computed in polynomial time, when h is described by a decision tree. For completeness, we show here that $\epsilon_{h,\boldsymbol{x}}(S)$ can be evaluated in linear time for decision trees, using the orthogonality of decision trees, and the fact this property is closed under conditioning.

To this end, recall that a *(Boolean) decision tree* is a binary tree $\mathcal{T}$, each of whose internal nodes is labeled with one of d Boolean variables from $X_d = \{x_1, \cdots, x_d\}$, and whose leaves are labeled 0 or 1. The value $h(\boldsymbol{x}) \in \{0, 1\}$ of a hypothesis h described by $\mathcal{T}$ on an instance $\boldsymbol{x}$ is given by the label of the leaf reached from the root of $\mathcal{T}$ 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 $\mathcal{T}$, denoted $|\mathcal{T}|$, is given by the number of nodes in $\mathcal{T}$. For illustration, a decision tree representing the classifier (4) is given in Figure 2.

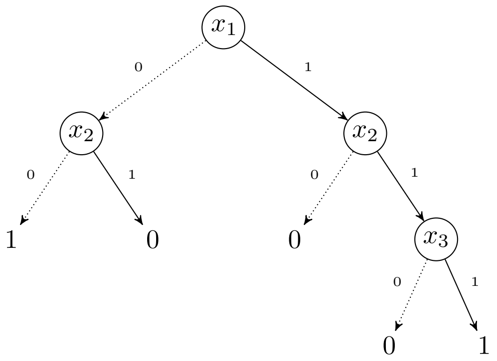

Figure 2: A decision tree representation of (4).

As usual, a *literal* over X_d is a variable x_i, or its negation $\overline{x}_i$. The negation of a literal l is given by $\neg l = x_i$ if $l = \overline{x}_i$, and $\neg l = \overline{x}_i$ if $l = x_i$. A a *term* is a conjunction of literals, and a *Disjunctive Normal Form (DNF)* formula is a disjunction of terms. Here, DNF formulas are viewed as sets of terms, and terms are viewed as sets of literals. A term is *inconsistent* if includes a pair $\{l, \neg l\}$ of opposite literals. A DNF formula $F = \{t_1, \cdots, t_m\}$ is *orthogonal* if $t_i \cup t_j$ is inconsistent for all pairs $i, j \in [m]$ such that $i \neq j$. The *conditioning* [Darwiche, 1999] of F by a term t, denoted $F \mid t$, is the formula obtained by removing from $\{t_1 \cup t, \cdots, t_m \cup t\}$ any term that is inconsistent.

Proposition 1. Given a classifier $h : \{0,1\}^d \to \{0,1\}$ represented by some decision tree $\mathcal{T}$, an instance $\boldsymbol{x} \in \{0,1\}^d$, and a set of features $S \subseteq [d]$, evaluating $\epsilon_{h,\boldsymbol{x}}(S)$ can be done in $\mathcal{O}(|S| \cdot |\mathcal{T}|)$ time.

Proof. It is well-known that $\mathcal{T}$ can be transformed in linear time into an equivalent orthogonal DNF formula, denoted $\text{DNF}(\mathcal{T})$, where each term corresponds to a path from the root to a leaf labeled with 1. Given an instance $\boldsymbol{x} \in \{0,1\}^d$ and a set S of features, let $t_{\boldsymbol{x}_S}$ be the term associated with the partial instance $\boldsymbol{x}_S$, that is,

$$t_{\boldsymbol{x}_S} = \bigcup_{i=1}^{d} \{x_i : (x_S)_i = 1\} \cup \{\overline{x}_i : (x_S)_i = 0\}$$

By construction, $\text{DNF}(\mathcal{T}) \mid t_{\boldsymbol{x}_S}$ is orthogonal and hence, for decision trees, (3) can simply be rewritten as:

$$\mu_{h,\boldsymbol{x}}(S) = \begin{cases} \sum_{t \in \text{DNF}(\mathcal{T}) | t_{\boldsymbol{x}_S}} 2^{d-|t|} & \text{if } h(\boldsymbol{x}) = 1 \\ 2^d - \sum_{t \in \text{DNF}(\mathcal{T}) | t_{\boldsymbol{x}_S}} 2^{d-|t|} & \text{if } h(\boldsymbol{x}) = 0 \end{cases}$$

The result follows from (2), together with the fact that $\text{DNF}(\mathcal{T}) \mid t_{\boldsymbol{x}_S}$ can be derived in $O(|S| \cdot |\mathcal{T}|)$ time. □

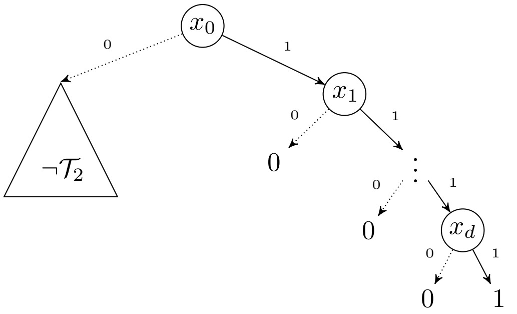

Figure 3: The decision tree $\mathcal{T}_3$ in the proof of Proposition 2.

2.3 MINIMIZING EXPLANATION ERRORS

The next result shows that the decision version of Problem 1 is generally hard to solve for decision trees, even when the set I of candidate features is an abductive explanation.

Proposition 2. Given a classifier h represented by some decision tree $\mathcal{T}$, an instance $\boldsymbol{x} \in \{0,1\}^d$, an abductive explanation $I \subseteq [d]$ for h and $\boldsymbol{x}$, an integer $k < |I|$, and a threshold $\varepsilon \in (0, \frac{1}{2})$, the problem of finding a subset $S \subseteq I$ of size at most k satisfying $\epsilon_{h,\boldsymbol{x}}(S) \leq \varepsilon$ is NP-hard.

Proof. We consider three problems $\boldsymbol{P}_1, \boldsymbol{P}_2$ and $\boldsymbol{P}_3$, each taking as input a decision tree representation $\mathcal{T}$ of some classifier h, an instance $\boldsymbol{x} \in \{0,1\}^d$, and two parameters k and ε. The third problem is also given an abductive explanation I for h and $\boldsymbol{x}$. For $\boldsymbol{P}_1$, $k \leq [d]$ and $\varepsilon \in (0, \frac{1}{2})$, for $\boldsymbol{P}_2$, $k \leq [d]$ and $\varepsilon \in (\frac{1}{2}, 1)$, and for $\boldsymbol{P}_3$, $k < |I|$ and $\varepsilon \in (0, \frac{1}{2})$. The corresponding tasks are given as follows:

$\boldsymbol{P}_1$: Find $S \subseteq [d]$ such that $|S| \leq k$ and $\epsilon_{h,\boldsymbol{x}}(S) \leq \varepsilon$;

$\boldsymbol{P}_2$: Find $S \subseteq [d]$ such that $|S| \leq k$ and $\epsilon_{h,\boldsymbol{x}}(S) \geq \varepsilon$;

$\boldsymbol{P}_3$: Find $S \subseteq I$ such that $|S| \leq k$ and $\epsilon_{h,\boldsymbol{x}}(S) \leq \varepsilon$.

By Theorem 2 in [Arenas et al., 2022], $\boldsymbol{P}_1$ is NP-hard. Based on this result, we give here a chain of polynomial-time reductions $\boldsymbol{P}_1 \preceq_p \boldsymbol{P}_2 \preceq_p \boldsymbol{P}_3$.

Given an instance $(\mathcal{T}_1, \boldsymbol{x}_1, k_1, \varepsilon_1)$ of $\boldsymbol{P}_1$, we build an instance $(\mathcal{T}_2, \boldsymbol{x}_2, k_2, \varepsilon_2)$ of $\boldsymbol{P}_2$, where $\mathcal{T}_2$ is the negation of $\mathcal{T}_1$, $\boldsymbol{x}_2 = \boldsymbol{x}_1$, $k_2 = k_1$ and $\varepsilon_2 = 1 - \varepsilon_1$. Note that $\mathcal{T}_2$ can be constructed in polynomial time by simply switching the label of each leaf in $\mathcal{T}_1$. Let h_1 and h_2 denote the hypotheses associated with $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. Since by construction, $\epsilon_{h_2,\boldsymbol{x}}(S) = 1 - \epsilon_{h_1,\boldsymbol{x}}(S)$, we have $\boldsymbol{P}_1 \preceq_p \boldsymbol{P}_2$, and hence, $\boldsymbol{P}_2$ is NP-hard.

Now, consider an instance $(\mathcal{T}_2, \boldsymbol{x}_2, k_2, \varepsilon_2)$ of $\boldsymbol{P}_2$. Without loss of generality, we assume here that $\boldsymbol{x}_2$ is the d-dimensional all-ones vector $\mathbf{1}$, and $h_2(\boldsymbol{x}) = 1$ where h_2 is the hypothesis associated with $\mathcal{T}_2$. We construct an instance $(\mathcal{T}_3, \boldsymbol{x}_3, I, k_3, \varepsilon_3)$ of $\boldsymbol{P}_3$ in the following way. Let $\mathcal{T}_3$ be the decision tree defined according to Figure 3. The root node of $\mathcal{T}_3$ is labeled with x_0, the subtree rooted at the left child of x_0 is the negation of $\mathcal{T}_2$, and the subtree rooted at the right child of x_0 is the caterpillar encoding the conjunction $x_1 \wedge \cdots \wedge x_d$. Let $\boldsymbol{x}_3$ be the all-ones vector $\mathbf{1}$ over $\{0, 1, \cdots, d\}$, and $I = \{0, 1, \cdots, d\}$. The remaining parameters are set to $k_3 = k_2$ and $\varepsilon_3 = 1 - \varepsilon_2$.

Let h_3 be the hypothesis associated with $\mathcal{T}_3$. For any $i \in \{0, 1, \cdots, d\}$, let $\boldsymbol{y}_{i \leftarrow 0}$ be the instance in $\{0,1\}^{d+1}$ obtained by flipping the value $x_{3,i}$ and leaving all other values of $\boldsymbol{x}_3$ unchanged. Since $h_3(\boldsymbol{x}_3) = 1$, and $h_3(\boldsymbol{y}_{i \leftarrow 0}) = 0$ for all $i \in \{0, 1, \cdots, d\}$, it follows that I is an abductive explanation for h_3 and $\boldsymbol{x}_3$. Moreover, for any proper subset S of I that *includes* the feature 0, we have

$$\epsilon_{h_3,\boldsymbol{x}_3}(S) = 1 - \frac{1}{2^{d+1-|S|}} \geq \frac{1}{2} > \varepsilon_3$$

So, in order to solve $\boldsymbol{P}_3$, we need to identify a subset $S \subseteq I$ of size at most k_3 that *excludes* the feature 0, while satisfying $\epsilon_{h_3,\boldsymbol{x}_3}(S) \leq \varepsilon_3$. But we can see that for any $S \subseteq I \backslash \{0\}$,

$$\begin{aligned}\epsilon_{h_3,\boldsymbol{x}_3}(S) &= \frac{|\{\boldsymbol{y} \in \{0,1\}^d : h_2(\boldsymbol{y}) = h_2(\boldsymbol{x}_2), \boldsymbol{y}_S = (\boldsymbol{x}_2)_S\}|}{|\{\boldsymbol{y} \in \{0,1\}^d : \boldsymbol{y}_S = (\boldsymbol{x}_2)_S\}|} \\ &= 1 - \epsilon_{h_2,\boldsymbol{x}_2}(S)\end{aligned}$$

Therefore, $\epsilon_{h_3,\boldsymbol{x}_3}(S) \leq \varepsilon_3$ if and only if $\epsilon_{h_2,\boldsymbol{x}_2}(S) \geq \varepsilon_2$. It follows that, $\boldsymbol{P}_2 \preceq_p \boldsymbol{P}_3$, and hence, $\boldsymbol{P}_3$ is NP-hard. □

3 SUPERMODULAR MINIMIZATION

The main idea of this study is to relax the requirement of finding an optimal solution to Problem 1, and instead settle for a solution that is "good enough", using supermodular minimization algorithms. In this section, we start with some basic notions about supermodularity, and then, we examine some useful properties of the mistake function.

3.1 SUPERMODULAR FUNCTIONS

Given a real-valued set function $f : 2^{[d]} \to \mathbb{R}$, the quantities

$$\begin{aligned}\mathrm{L}_f(i \mid S) &= f(S \setminus \{i\}) - f(S), \text{ and} \\ \mathrm{G}_f(i \mid S) &= f(S \cup \{i\}) - f(S)\end{aligned}$$

are respectively capturing the *marginal loss* of removing an element i from a set S, and *marginal gain* of adding an element i to a set S. A set function f is *non-increasing* if $\mathrm{L}_f(i \mid S) \geq 0$ for all $S \subseteq [d]$ and $i \in S$, and f is *non-decreasing* if $\mathrm{G}_f(i \mid S) \geq 0$ for all $S \subseteq [d]$ and $i \in [d] \setminus S$. A set function f is *supermodular* if it satisfies the diminishing loss condition

$$\mathrm{L}_f(i \mid S) \geq \mathrm{L}_f(i \mid T)$$

for all $S \subseteq T \subseteq [d]$ and $i \in S$. Dually, f is *submodular* if it satisfies the diminishing gain condition

$$\mathrm{G}_f(i \mid S) \geq \mathrm{G}_f(i \mid T)$$

for all $S \subseteq T \subseteq [d]$ and $i \in [d] \setminus T$. Based on the fact that $\mathrm{L}_f(i \mid S) = -\mathrm{G}_f(i \mid S \setminus \{i\})$ for all $S \subseteq [d]$ and $i \in S$, f is supermodular if and only if $-f$ is submodular. Finally, f is *modular* if it is both submodular and supermodular.

For a non-negative set function f and a nonempty subset $I \subseteq [d]$, the *curvature* of f over 2^I is given by

$$c = 1 - \min_{i \in I} \frac{\mathrm{L}_f(i \mid I)}{\mathrm{L}_f(i \mid \{i\})} = 1 - \min_{i \in I} \frac{\mathrm{G}_f(i \mid I \setminus \{i\})}{\mathrm{G}_f(i \mid \varnothing)} \quad (5)$$

Clearly, $c \in [0, 1]$ whenever f is non-decreasing and submodular, or non-increasing and supermodular. Note that the curvature coincides with the notion of "steepness" defined in [Il'ev, 2001]. When $I = [d]$, c is called the *total curvature* of f [Conforti and Cornuéjols, 1984]. Notably, in the case where f is non-increasing and supermodular, the condition $c < 1$ is sufficient for ensuring that the task of minimizing f subject to a cardinality constraint is approximable to within a constant [Il'ev, 2001, Sviridenko et al., 2017].

3.2 MINIMIZING EXPLANATION MISTAKES

In light of Figure 1, we can see that the error function $\epsilon_{h,\boldsymbol{x}}(\cdot)$ is generally *not* supermodular or submodular, and *not* non-increasing or non-decreasing. However, if we instead focus on the unnormalized version $\mu_{h,\boldsymbol{x}}(\cdot)$ given in (3), then the following properties can be derived.

Proposition 3. Let $h : \{0,1\}^d \to \{0,1\}$ be a classifier, $\boldsymbol{x} \in \{0,1\}^d$ be an instance, and $I \subseteq [d]$ be any nonempty set of features. Then, $\mu_{h,\boldsymbol{x}}(\cdot)$ is supermodular and non-increasing. Furthermore, if I is an abductive explanation for h and $\boldsymbol{x}$, then the curvature c of $\mu_{h,\boldsymbol{x}}(\cdot)$ over 2^I satisfies $c < 1$.

Proof. Let f be the function $\mu_{h,\boldsymbol{x}}(\cdot)$, and N be the set of instances $\boldsymbol{y} \in \{0,1\}^d$ such that $h(\boldsymbol{x}) \neq h(\boldsymbol{y})$. For any subset $S \subseteq [d]$, let $C(\boldsymbol{x}_S)$ denote the set of instances $\boldsymbol{y} \in \{0,1\}^d$ covered by $\boldsymbol{x}_S$, and for any feature $i \in S$, let $\overline{C}(\boldsymbol{x}_{S\setminus\{i\}})$ denote the set $C(\boldsymbol{x}_{S\setminus\{i\}}) \setminus C(\boldsymbol{x}_S)$.

The fact that f is non-increasing directly follows from the observation that $\mathrm{L}_f(i \mid S) = |\overline{C}(\boldsymbol{x}_{S\setminus\{i\}}) \cap N| \geq 0$ for any $S \subseteq [d]$ and any $i \in S$. Now, given any superset T of S, we have $\overline{C}(\boldsymbol{x}_{T\setminus\{i\}}) \subseteq \overline{C}(\boldsymbol{x}_{S\setminus\{i\}})$. It follows that $\overline{C}(\boldsymbol{x}_{T\setminus\{i\}}) \cap N \subseteq \overline{C}(\boldsymbol{x}_{S\setminus\{i\}}) \cap N$, and hence, $\mathrm{L}_f(i \mid T) \leq \mathrm{L}_f(i \mid S)$. Therefore, f is supermodular.

Finally, $\mathrm{L}_f(i \mid I) > 0$ whenever I is a (non-empty) abductive explanation for h and $\boldsymbol{x}$. This, together with the fact that, by supermodularity, $\mathrm{L}_f(i \mid \{i\}) \geq \mathrm{L}_f(i \mid I)$ for any $i \in I$, implies that $c \in [0, 1)$. □

4 APPROXIMATION ALGORITHMS

After providing an overview of probabilistic explanations and supermodular minimization, we now present two greedy approximation algorithms for Problem 1.

4.1 GREEDY DESCENT

A natural approach for minimizing a supermodular and non-increasing function f subject to a cardinality constraint $|S| \leq k$ is to start from the input set I of candidate features, and to iteratively remove from the current solution S any feature i that minimizes the marginal loss $\mathrm{L}_f(i \mid S)$, until the desired size $|S| = k$ is reached. As shown by Il'ev [2001], this greedy method achieves a $\frac{e^p - 1}{p}$-approximation, where $p = \frac{c}{1-c}$, and c is the curvature of f over 2^I.

In the setting of our study, the error function $\epsilon_{h,\boldsymbol{x}}(\cdot)$ in (2) is a normalized version of $\mu_{h,\boldsymbol{x}}(\cdot)$, which is supermodular and non-increasing. Furthermore, the normalization factor $2^{d-|S|}$ is *constant* for all subsets S with the same size. Based on these properties, we can combine the above greedy descent approach for $f = \mu_{h,\boldsymbol{x}}(\cdot)$, with a level-wise selection method that stores the subsets $S_0, S_1, \cdots, S_k$ obtained for each size $j \in \{0, 1, \cdots, k\}$, and that returns from this sequence the best subset S_j with respect to $\epsilon_{h,\boldsymbol{x}}(\cdot)$. A formal description is given in Algorithm 1.

Proposition 4. Let S^* be an optimal solution to Problem 1, let c be the curvature of $\mu_{h,\boldsymbol{x}}(\cdot)$ over 2^I, and assume that I is an abductive explanation for h and $\boldsymbol{x}$. Then, the solution S_{GD} returned by Greedy Descent (GD) satisfies:

$$\epsilon_{h,\boldsymbol{x}}(S_{\text{GD}}) \leq \left(\frac{e^p - 1}{p}\right) \epsilon_{h,\boldsymbol{x}}(S^*) \text{ where } p = \frac{c}{1-c} < 1$$

Proof. The fact that $p < 1$ follows from Proposition 3. Let j^* be the size of S^*, and let S_{j^*} be the solution computed by GD at the end of the step $j = n - j^* + 1$. Note that $|S^*| = |S_{j^*}|$. So, by application of Corollary 4 in [Il'ev, 2001], we must have

$$\mu_{h,\boldsymbol{x}}(S_{j^*}) \leq \left(\frac{e^p - 1}{p}\right) \mu_{h,\boldsymbol{x}}(S^*)$$

Since GD is returning a minimizer of $\epsilon_{h,\boldsymbol{x}}(\cdot)$ over the sequence $S_0, \cdots, S_{j^*}, \cdots, S_k$, it follows that

$$\begin{aligned} \epsilon_{h,\boldsymbol{x}}(S_{\text{GD}}) &\leq \epsilon_{h,\boldsymbol{x}}(S_{j^*}) = \frac{\mu_{h,\boldsymbol{x}}(S_{j^*})}{2^{d-j^*}} \\ &\leq \left(\frac{e^p - 1}{p}\right) \frac{\mu_{h,\boldsymbol{x}}(S^*)}{2^{d-j^*}} = \left(\frac{e^p - 1}{p}\right) \epsilon_{h,\boldsymbol{x}}(S^*) \end{aligned}$$

□

Algorithm 1: Greedy Descent (GD)

```
Input: classifier $h$, instance $\boldsymbol{x}$, feature set $I$, integer $k$
Set $S_n = I$, where $n = |I|$
For $j = n$ downto $1$ do
    Let $i^* \in \mathrm{Argmin}_{i \in S_j}\, \mu_{h,\boldsymbol{x}}(S_j \setminus \{i\})$
    Set $S_{j-1} = S_j \setminus \{i^*\}$
Let $S_{\text{GD}} \in \mathrm{Argmin}_{S \in \{S_0, S_1, \cdots, S_k\}}\, \epsilon_{h,\boldsymbol{x}}(S)$
Return $S_{\text{GD}}$
```

4.2 GREEDY ASCENT

An alternative approach is to consider the objective function $f = -\mu_{h,\boldsymbol{x}}(\cdot)$, which is submodular and non-decreasing. Based on the well-known greedy method for submodular maximization [Nemhauser and Wolsey, 1978], we could start from $S_0 = \varnothing$, and iteratively add to the current solution S_{j-1} any maximizer $i \in I \setminus S_{j-1}$ of the marginal gain $\mathrm{G}_f(i \mid S_{j-1})$ until $|S_j| = k$. Unfortunately, such a method would fail here because f is *non-positive*. Yet, as observed by Liberty and Sviridenko [2017], this issue can be alleviated by slightly increasing the size limit k. More precisely, given a parameter $\gamma \in (0, 1)$, the greedy method achieves a $\frac{1}{1-\gamma}$-approximation, whenever it is allowed to improve its solution S_{j-1} until $|S_j| = k\lceil \ln\left(f(\varnothing)/\gamma f(S_{j-1})\right)\rceil$. By coupling this idea with the level-wise selection method suggested above, we get a greedy ascent algorithm for minimizing $\epsilon_{h,\boldsymbol{x}}(\cdot)$, detailed in Algorithm 2.

Proposition 5. Under the conditions of Proposition 4, the solution S_{GA} returned by Greedy Ascent (GA) satisfies:

$$\epsilon_{h,\boldsymbol{x}}(S_{\text{GA}}) \leq \left(\frac{1}{1-c}\right) \epsilon_{h,\boldsymbol{x}}(S^*), \text{ and}$$

$$|S_{\text{GA}}| \leq k \left(1 + \left\lceil \ln \frac{\mu_{h,\boldsymbol{x}}(\varnothing)}{\mu_{h,\boldsymbol{x}}(S^*)} \right\rceil\right) \leq k \left\lceil \ln \frac{2e}{c} \right\rceil$$

Proof. The upper bound on $\epsilon_{h,\boldsymbol{x}}(S_{\text{GA}})$ can be derived from the following chain of inequalities:

$$\begin{aligned}\epsilon_{h,\boldsymbol{x}}(S_{\text{GA}}) \leq \epsilon_{h,\boldsymbol{x}}(S_j) = \frac{\mu_{h,\boldsymbol{x}}(S_j)}{2^{d-j}} &\leq \left(\frac{1}{1-c}\right) \frac{\mu_{h,\boldsymbol{x}}(S^*)}{2^{d-j}} \\ &\leq \left(\frac{1}{1-c}\right) \frac{\mu_{h,\boldsymbol{x}}(S^*)}{2^{d-|S^*|}} = \left(\frac{1}{1-c}\right) \epsilon_{h,\boldsymbol{x}}(S^*)\end{aligned}$$

where the first inequality uses the fact that S_{GA} is a minimizer of $\epsilon_{h,\boldsymbol{x}}(\cdot)$ over $\{S_0, \cdots, S_j\}$, the second inequality follows from [Liberty and Sviridenko, 2017, Theorem 5] and $c \leq \gamma$, and the last inequality follows from $|S^*| \leq j$.

The first upper bound on $|S_{\text{GA}}|$ simply follows from [Liberty and Sviridenko, 2017, Theorem 5], and the fact that $\gamma \geq \frac{1}{e}$. For the last bound on $|S_{\text{GA}}|$, we know that $\frac{1}{c} = \mu_{h,\boldsymbol{x}}(\varnothing) - \min_{i \in I} \mu_{h,\boldsymbol{x}}(\{i\})$, whenever I is an abductive explanation. This, together with the fact that $\min_{i \in I} \mu_{h,\boldsymbol{x}}(\{i\}) \leq \frac{1}{2}\mu_{h,\boldsymbol{x}}(\varnothing)$, yields $\mu_{h,\boldsymbol{x}}(\varnothing) \leq \frac{2}{c}$. □

Algorithm 2: Greedy Ascent (GA)

```
Input: classifier $h$, instance $\boldsymbol{x}$, feature set $I$, integer $k$
Let $c$ be the curvature of $\mu_{h,\boldsymbol{x}}(\cdot)$ over $2^I$
Set $j = 0$, $S_0 = \varnothing$ and $\gamma = \max\{\frac{1}{e}, c\}$
Repeat
    Let $i^* \in \mathrm{Argmin}_{i \in I \setminus S_{j-1}}\, \mu_{h,\boldsymbol{x}}(S_{j-1} \cup \{i\})$
    Set $S_j = S_{j-1} \cup \{i^*\}$
Until $j = k\left\lceil \ln\left(\frac{\mu_{h,\boldsymbol{x}}(\varnothing)}{\gamma \cdot \mu_{h,\boldsymbol{x}}(S_j)}\right)\right\rceil$
Let $S_{\text{GA}} \in \mathrm{Argmin}_{S \in \{S_0, S_1, \cdots, S_j\}}\, \epsilon_{h,\boldsymbol{x}}(S)$
Return $S_{\text{GA}}$
```

4.3 APPLICATION TO DECISION TREES

The approximation bounds derived in Propositions 4 and 5 hold for *any* (Boolean) hypothesis class. However, in order to ensure that GD and GA are computationally efficient, each call to the value oracle $\mu_{h,\boldsymbol{x}}(\cdot)$ should run in polynomial time. As emphasized in Section 2.2, this is the case for decision trees. Namely, if the input classifier h of GD is represented by a decision tree $\mathcal{T}$, then by Proposition 1 and the fact that the number of calls to the value oracle is quadratic in $n = |I|$, implies that GD runs in $\mathcal{O}(n^3|\mathcal{T}|)$ time. For GA, the number of calls to the value oracle is bounded by $jn + n + 2$, where j is the number of iterations of the main loop, and $n + 2$ is the number of calls required to compute c. So, GA runs in $\mathcal{O}(kn^2(1 + \ln 2/c)|\mathcal{T}|)$ time.

5 EXPERIMENTS

In order to validate the effectiveness of our algorithms, we have considered various instances of Problem 1, where the input classifier is described by a decision tree. The code was written using the `Python` language. All the experiments have been conducted on a computer equipped with a 3.1 GHz Intel(R) Core i9-9900 CPU and 64 GiB of RAM.

5.1 EXPERIMENTAL SETUP

In our experiments, we have considered $B = 50$ datasets, or *benchmarks*, from the standard repositories *Kaggle*, *OpenML* and *UCI*. Notably, *mnist38* and *mnist49* are subsets of the dataset *mnist*. Except for *cnae*, all datasets are binary classification tasks with a number of attributes ranging from 10^1 to 10^5. The multi-label classification task *cnae* was transformed into a binary classification task by considering the dominant label versus all other labels.

Benchmark				$\epsilon_{h,\boldsymbol{x}}(S)$			$\lvert S \rvert$			Time (s)
name	*acc*	*d*	$\lvert I \rvert$	GA	GD	SAT	GA	GD	SAT	SAT
meta-data	87.42	44	5.09	0.08 (±0.11)	0.08 (±0.11)	0.08 (±0.11)	3.10	3.10	3.10	12.14
glass	78.46	31	5.38	0.26 (±0.11)	0.26 (±0.11)	0.26 (±0.11)	2.14	2.14	2.14	2.36
student perf.	91.79	30	5.41	0.26 (±0.11)	0.26 (±0.11)	0.26 (±0.11)	2.00	2.00	2.00	2.16
primary tumor	84.31	23	6.23	0.09 (±0.09)	0.09 (±0.09)	0.09 (±0.08)	4.22	4.22	4.22	3.58
liver disorders	75.96	58	6.38	0.18 (±0.09)	0.18 (±0.08)	0.18 (±0.08)	4.00	4.00	4.00	27.33
schizophrenia	80.39	33	6.39	0.37 (±0.24)	0.37 (±0.24)	0.37 (±0.24)	1.27	1.27	1.27	4.79
hungarian	62.92	13	6.65	0.12 (±0.12)	0.12 (±0.12)	0.11 (±0.10)	3.58	3.56	3.56	1.68
horse colic	75.68	40	6.73	0.14 (±0.07)	0.13 (±0.07)	0.13 (±0.07)	4.03	4.06	4.06	11.56
indian liver	64.57	84	8.21	0.10 (±0.09)	0.10 (±0.09)	0.16 (±0.12)	5.08	4.89	6.12	176.28
pima indians	75.32	97	8.30	0.15 (±0.14)	0.15 (±0.14)	0.16 (±0.12)	5.85	5.84	6.58	484.6
loan eligibility	74.31	68	8.47	0.19 (±0.13)	0.18 (±0.13)	0.20 (±0.14)	5.60	5.70	6.82	42.87
patient treat.	66.01	10	8.92	0.05 (±0.09)	0.03 (±0.06)	0.03 (±0.08)	5.63	5.94	5.94	24.08
wine	69.58	11	9.03	0.09 (±0.10)	0.09 (±0.09)	0.09 (±0.12)	5.59	5.64	5.62	36.32
employee attr.	82.45	63	10.56	0.06 (±0.09)	0.06 (±0.09)	0.20 (±0.11)	6.41	6.39	6.98	1017.24
contraceptive	51.36	90	10.84	0.06 (±0.08)	0.06 (±0.08)	0.39 (±0.17)	4.27	4.26	5.95	1096.07
compas	67.60	40	10.95	0.03 (±0.07)	0.04 (±0.08)	0.05 (±0.09)	5.68	5.83	6.78	1082.32
fetal health	91.85	93	11.33	0.12 (±0.06)	0.12 (±0.06)	0.23 (±0.11)	5.59	5.59	6.00	930.61
dorothea	91.88	10^5	12.90	0.25 (±0.10)	0.25 (±0.10)	–	6.70	6.70	–	–
bank market.	89.49	882	13.11	0.29 (±0.08)	0.29 (±0.07)	–	6.99	6.99	–	–
mnist49	95.99	784	15.57	0.37 (±0.14)	0.37 (±0.14)	–	6.97	6.89	–	–
spambase	92.11	236	16.09	0.24 (±0.11)	0.23 (±0.09)	–	6.87	6.87	–	–
mnist38	96.42	784	17.89	0.37 (±0.13)	0.38 (±0.14)	–	6.93	6.93	–	–
cnae	92.59	856	19.07	0.32 (±0.25)	0.32 (±0.25)	–	5.97	5.97	–	–
gisette	94.10	5000	21.42	0.32 (±0.11)	0.32 (±0.11)	–	6.88	6.88	–	–
farm ads	80.78	54877	23.15	0.13 (±0.17)	0.13 (±0.17)	–	6.31	6.31	–	–

Table 1: Experimental results on 25 benchmarks for decision tree explanations, using $k = 7$.

For each benchmark $b \in [B]$, an explanation task consists in a tuple $(\mathcal{T}, \boldsymbol{x}, I, k)$ described as follows. $\mathcal{T}$ is the decision tree representation of some classifier h, which is learned from the training part of b. In our experiments, we have used a `Scikit-Learn` implementation of the CART algorithm for generating $\mathcal{T}$. The predictive accuracy of h is measured on the test part of b. By interpreting each internal node of $\mathcal{T}$ as a Boolean feature, the instance $\boldsymbol{x}$ to be explained is taken from the test part of b, and binarized according to the d features occurring in $\mathcal{T}$. The set I is given by the collection of features occurring in the (single) root-to-leaf path in $\mathcal{T}$ that is consistent with the instance $\boldsymbol{x}$. Here, I is often referred to as a *path-explanation* [Izza et al., 2022b], or *direct reason* [Audemard et al., 2022a]. As observed in [Izza et al., 2022b], I is not necessarily minimal with respect to set inclusion. Finally, we have used $k \ = 7 \pm 2$ for the size limit. The performance of explanation algorithms on a benchmark b is measured by drawing uniformly at random m instances $\boldsymbol{x}$ from the test set of b, and averaging the resulting error $\epsilon_{h,\boldsymbol{x}}(S)$ and size $|S|$ of the output $S \subseteq I$. In our experiments, m was set to $\min\{s, 150\}$, where s is the size of the test set of b.

To compare the performance of GD and GA with an exact solver, we have chosen the SAT-based approach in [Arenas et al., 2022]. Namely, a SAT encoding was provided for the following task: given as input a decision tree $\mathcal{T}$ for some classifier h, an instance $\boldsymbol{x}$, and two parameters $k \leq d$ and $\varepsilon \in [0, 1)$, return as output "yes" if there is a set of features S satisfying both $|S| \leq k$ and $\epsilon_{h,\boldsymbol{x}}(S) \leq \varepsilon$, and "no" otherwise. In the setting of our experimental setup, S is a subset of the path-explanation I for $\boldsymbol{x}$. So, the above SAT encoding was extended to the decision version of Problem 1, by adding the clause $\bigvee\{x_i : (x_I)_i = 1\} \vee \{\overline{x}_i : (x_I)_i = 0\}$. For the original version of Problem 1, a binary search over the interval $(0, 1]$ was performed in order to find a minimizer S of $\epsilon_{h,\boldsymbol{x}}(\cdot)$ with precision of 10^{-3}, which requires at most 10 calls to the SAT solver. We used a `Pysat` implementation of GLUCOSE 4 for the solver, with a timeout of 30 minutes per explanation task.[1]

5.2 EXPERIMENTAL RESULTS

In Table 1 is reported an overview of our results on 25 of 50 benchmarks, for $k = 7$. The leftmost column gives the name of the dataset b. The columns *acc* and d are respectively giving the accuracy and the number of features of the decision tree. The rows are sorted according to the average size $|I|$ of the path-explanation. The fifth, sixth, and seventh columns are reporting the results for the average error $\epsilon_{h,\boldsymbol{x}}(S)$ of

[1] We mention in passing that an SMT-based approach was recently proposed in [Izza et al., 2022a], but the code was not available at the time of writing this paper.

the explanation S returned by GA, GD, and the SAT-based approach, respectively. The next three columns are reporting the average size of S for these algorithms. Finally, the last column gives the average run-times (in seconds) of the SAT-based approach. Notably, for the 7 datasets in blue, the SAT solver occasionally reaches the timeout before the end of binary search, which results in a degradation of precision. For the 8 datasets in magenta, the solver could not perform a single run of binary search before reaching the timeout. We have not reported the run-times of GA and GD, because they could always find a solution in less than 0.1 seconds.

In light of these results, we can observe that the performance of greedy algorithms for minimizing $\epsilon_{h,\boldsymbol{x}}(\cdot)$ is remarkable, especially in comparison with the performance of the SAT-based approach. For the benchmarks where the SAT solver could return an optimal solution S^*, the differences $\epsilon_{h,\boldsymbol{x}}(S_{\text{GD}}) - \epsilon_{h,\boldsymbol{x}}(S^*)$ and $\epsilon_{h,\boldsymbol{x}}(S_{\text{GA}}) - \epsilon_{h,\boldsymbol{x}}(S^*)$ are most often negligible. Moreover, for high-dimensional datasets such as *dorothea*, *gisette* and *farm ads*, both GA and GD remain stable by providing explanations with comparable errors in a few tenths of a millisecond. Regarding the conciseness of explanations, we can see that $|S_{\text{GD}}|$ is on average smaller than $|S^*|$. Interestingly, $|S_{\text{GA}}|$ is on average smaller than the size limit $k = 7$, which indicates that the upper bound on $|S_{\text{GA}}|$ in Proposition 5 is rarely attained in practice. Finally, GA and GD could efficiently reduce path-explanations I which are not always abductive. In other words, both algorithms are, in practice, robust enough to handle some explanation tasks for which the curvature c of the unnormalized error function is close to or equal to 1.

6 DISCUSSION

Related Work. Clarifying in a comprehensible way the prediction $h(\boldsymbol{x})$ made by some classifier h on an input data instance $\boldsymbol{x}$ often takes the form of a set I of features which in conjunction determine $h(\boldsymbol{x})$ [Ribeiro et al., 2018]. Such an explanation is abductive [Ignatiev et al., 2019], or sufficient [Darwiche and Hirth, 2020], precisely when I is minimal with respect to inclusion. The problem of finding abductive explanations has been a subject of extensive research, recently surveyed in [Marques-Silva and Ignatiev, 2022]. The hypothesis classes which are tractable for computing abductive explanations include, among others, decision trees [Audemard et al., 2021, Huang et al., 2021, Izza et al., 2022b], Naive Bayes classifiers [Marques-Silva et al., 2020], monotone threshold functions [Cooper and Marques-Silva, 2023], and Boolean functions compiled into deterministic Decomposable Negation Normal Form (dDNNF) [Audemard et al., 2020, Huang et al., 2022]. Actually, even when the problem of finding an abductive explanation is NP-hard, empirical results indicate that it can often be solved in practice using SAT-based approaches [Ignatiev and Silva, 2021, Izza and Marques-Silva, 2021, Ignatiev et al., 2022]. However, due to cognitive limitations, a major weakness of abductive explanations is their uncontrollable size. In order to circumvent this issue, a common approach is to seek for abductive explanations of minimum size. Unfortunately, the corresponding optimization problem is NP-hard for decision trees [Barceló et al., 2020], and Σ_2^p-hard in general [Audemard et al., 2022b]. Furthermore, even if shortest abductive explanations could be found in a reasonable amount of time, their size remains uncontrollable.

By capturing a natural trade-off between conciseness and precision, probabilistic explanations have been a subject of growing research in the past two years [Blanc et al., 2021, Izza et al., 2021, Wäldchen et al., 2021, Wang et al., 2021, Arenas et al., 2022, Izza et al., 2022a, Wäldchen, 2022]. Recall that a size-k $(1-\varepsilon)$-probable explanation for a classifier h and an instance $\boldsymbol{x}$ is a subset $S \subseteq [d]$ such that $|S| \leq k$ and $\epsilon_{h,\boldsymbol{x}}(S) \leq \varepsilon$. Finding such explanations is NP^{PP}-hard in general [Wäldchen et al., 2021, Wäldchen, 2022], and NP-hard for decision trees [Arenas et al., 2022]. In the present study, we have shown that this problem remains NP-hard for decision trees, even under the assumption that S is a subset of some given abductive explanation I.

Heuristic approaches to probabilistic explanations have been considered in [Izza et al., 2021, 2022a]. The optimization task is symmetric to that of Problem 1: given a hypothesis h, an instance $\boldsymbol{x}$, a set of features I and an error parameter ε, the goal is to find a $(1-\varepsilon)$-probable explanation $S \subseteq I$ that minimizes $|S|$. For this task, the authors have proposed a greedy algorithm that runs in polynomial time, when h is described by a decision tree $\mathcal{T}$, and I is a path-explanation for $\boldsymbol{x}$ and $\mathcal{T}$. However, this algorithm does not provide any approximation guarantee with respect to the optimal size.

To the best of our knowledge, approximation approaches to probabilistic explanations have only been investigated in [Blanc et al., 2021]. Again, the problem under consideration is to find a $(1-\varepsilon)$-probable explanation S that minimizes $|S|$. Based on some results on implicit learning, the authors gave a PAC-style polynomial-time algorithm that takes as input a classifier h, an instance $\boldsymbol{x}$, a confidence parameter δ, and a precision parameter ε, and that returns as output a set $S \subseteq [d]$ with the following guarantees: (i) $|S|$ is polynomial in d, $1/\delta$ and $1/\varepsilon$, and (ii) if $\boldsymbol{x}$ is drawn uniformly at random over $\{0,1\}^d$, then $\epsilon_{h,\boldsymbol{x}} \leq \varepsilon$ with probability at least $(1-\delta)$. However, this algorithm is mainly of theoretical interest, since $|S|$ is in $\mathcal{O}\left((1/\delta)^9(1/\epsilon)^{12}\right)$ and, more importantly, the instances to be explained in practical applications are rarely picked at random according to the uniform distribution.

Perspectives. In our study, probabilistic explanations have been examined through the prism of supermodular minimization. Inspired from results in [Il'ev, 2001, Liberty and Sviridenko, 2017], we have proposed two greedy approximation algorithms for minimizing explanation errors subject to a cardinality constraint, whose performance essentially depends on the curvature c of the unnormalized error function $\mu_{h,\boldsymbol{x}}(\cdot)$. Importantly, our approximation results hold for any (Boolean) hypothesis class, and hence, our greedy algorithms are computationally efficient whenever $\mu_{h,\boldsymbol{x}}(\cdot)$ can be evaluated in polynomial time. Beyond decision trees, which have been examined in this paper, (ordered) binary decision diagrams [Hu et al., 2022] and dDNNF representations [Huang et al., 2022] are examples of classifiers satisfying this condition.

This work leaves open several questions. Notably, (i) what is the optimal approximation factor for minimizing the error of probabilistic reasons under a cardinality constraint? A partial answer might come from Sviridenko et al. [2017], who gave a near-optimal algorithm for minimizing a non-increasing supermodular function subject to a matroid constraint. But this method is mainly of theoretical interest, as its computational complexity is prohibitive. So, (ii) can we find alternative, near-optimal approximation algorithms which are computationally efficient? Finally, (iii) using sampling methods, can we extend approximation algorithms to hypothesis classes for which the problem of evaluating $\mu_{h,\boldsymbol{x}}(\cdot)$ is intractable?

Acknowledgements

Many thanks to the 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

Marcelo Arenas, Pablo Barceló, Miguel A. Romero Orth, and Bernardo Subercaseaux. On computing probabilistic explanations for decision trees. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022.

Gilles Audemard, Frederic Koriche, and Pierre Marquis. On tractable XAI queries based on compiled representations. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 838–849, 2020.

Gilles Audemard, Steve Bellart, Louenas Bounia, Frederic Koriche, Jean-Marie Lagniez, and Pierre Marquis. On the computational intelligibility of boolean classifiers. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 74–86, 2021.

Gilles Audemard, Steve Bellart, Louenas Bounia, Frederic Koriche, Jean-Marie Lagniez, and Pierre Marquis. On the explanatory power of boolean decision trees. *Data & Knowledge Engineering*, 142:102088, 2022a.

Gilles Audemard, Steve Bellart, Louenas Bounia, Frederic Koriche, Jean-Marie Lagniez, and Pierre Marquis. Trading complexity for sparsity in random forest explanations. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5461–5469, 2022b.

Pablo Barceló, Mikaël Monet, Jorge Pérez, and Bernardo Subercaseaux. Model interpretability through the lens of computational complexity. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

Guy Blanc, Jane Lange, and Li-Yang Tan. Provably efficient, succinct, and precise explanations. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, pages 6129–6141, 2021.

Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete Applied Mathematics*, 7(3): 251–274, 1984.

Martin Cooper and João Marques-Silva. Tractability of explaining classifier decisions. *Artificial Intelligence*, 316:103841, 2023.

Adnan Darwiche. Compiling knowledge into decomposable negation normal form. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 284–289, 1999.

Adnan Darwiche and Auguste Hirth. On the reasons behind decisions. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, pages 712–720, 2020.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):93:1–93:42, 2019.

Hao Hu, Marie-José Huguet, and Mohamed Siala. Optimizing binary decision diagrams with MaxSAT for classification. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 3767–3775, 2022.

Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, and João Marques-Silva. On efficiently explaining graph-based classifiers. In *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 356–367, 2021.

Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, Martin Cooper, Nicholas Asher, and João Marques-Silva. Tractable explanations for d-dnnf classifiers. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5719–5728, 2022.

Alexey Ignatiev and João Marques Silva. SAT-based rigorous explanations for decision lists. In *Proceedings of the 24th International Conference on Theory and Applications of Satisfiability Testing (SAT)*, pages 251–269, 2021.

Alexey Ignatiev, Nina Narodytska, and João Marques-Silva. Abduction-based explanations for machine learning models. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1511–1519, 2019.

Alexey Ignatiev, Yacine Izza, Peter Stuckey, and João Marques-Silva. Using MaxSAT for efficient explanations of tree ensembles. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 3776–3785, 2022.

Victor P. Il'ev. An approximation guarantee of the greedy descent algorithm for minimizing a supermodular set function. *Discrete Applied Mathematics*, 114(1-3):131–146, 2001.

Yacine Izza and João Marques-Silva. On explaining random forests with SAT. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2584–2591, 2021.

Yacine Izza, Alexey Ignatiev, Nina Narodytska, Martin Cooper, and João Marques-Silva. Efficient explanations with relevant sets. *CoRR*, abs/2106.00546, 2021.

Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin Cooper, and João Marques-Silva. On computing probabilistic abductive explanations. *CoRR*, abs/2212.05990, 2022a.

Yacine Izza, Alexey Ignatiev, and João Marques-Silva. On tackling explanation redundancy in decision trees. *Journal of Artificial Intelligence Research*, 75:261–321, 2022b.

Edo Liberty and Maxim Sviridenko. Greedy minimization of weakly supermodular set functions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 19:1–19:11, 2017.

João Marques-Silva and Alexey Ignatiev. Delivering trustworthy AI through formal XAI. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 12342–12350, 2022.

João Marques-Silva, Thomas Gerspacher, Martin Cooper, Alexey Ignatiev, and Nina Narodytska. Explaining naive bayes and other linear classifiers with polynomial time and delay. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

Georges A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63(2):81–97, 1956.

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

Shashi Mittal and Andreas S. Schulz. An FPTAS for optimizing a class of low-rank functions over a polytope. *Mathematical Programming*, 141(1-2):103–120, 2013.

Christoph Molnar. *Interpretable Machine Learning*. Leanpub, 2020.

George L. Nemhauser and Laurence A. Wolsey. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research*, 3(3): 177–188, 1978.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1527–1535, 2018.

Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019.

Ali H. Sayed. *Inference and Learning from Data*. Cambridge University Press, 2022.

Andy Shih, Arthur Choi, and Adnan Darwiche. A symbolic approach to explaining Bayesian network classifiers. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5103–5111, 2018.

Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. *Mathematics of Operations Research*, 42(4):1197–1218, 2017.

Stephan Wäldchen. *Towards Explainable Artificial Intelligence – Interpreting Neural Network Classifiers with Probabilistic Prime Implicants*. PhD thesis, Technischen Universität Berlin, 2022.

Stephan Wäldchen, Jan MacDonald, Sascha Hauch, and Gitta Kutyniok. The computational complexity of understanding binary classifier decisions. *Journal of Artificial Intelligence Research*, 70:351–387, 2021.

Eric Wang, Pasha Khosravi, and Guy Van den Broeck. Probabilistic sufficient explanations. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence, (IJCAI)*, pages 3082–3088, 2021.