From Contrastive to Abductive Explanations and Back Again

Alexey Ignatiev, Nina Narodytska, Nicholas Asher, Joao Marques-Silva

From Contrastive to Abductive Explanations and Back Again

Alexey Ignatiev[1], Nina Narodytska[2(✉)], Nicholas Asher[3], and Joao Marques-Silva[3]

[1] Monash University, Melbourne, Australia
alexey.ignatiev@monash.edu
[2] VMware Research, Palo Alto, CA, USA
nnarodytska@vmware.com
[3] IRIT, CNRS, Toulouse, France
{nicholas.asher,joao.marques-silva}@irit.fr

Abstract. Explanations of Machine Learning (ML) models often address a 'Why?' question. Such explanations can be related with selecting feature-value pairs which are sufficient for the prediction. Recent work has investigated explanations that address a 'Why Not?' question, i.e. finding a change of feature values that guarantee a change of prediction. Given their goals, these two forms of explaining predictions of ML models appear to be mostly unrelated. However, this paper demonstrates otherwise, and establishes a rigorous formal relationship between 'Why?' and 'Why Not?' explanations. Concretely, the paper proves that, for any given instance, 'Why?' explanations are minimal hitting sets of 'Why Not?' explanations and vice-versa. Furthermore, the paper devises novel algorithms for extracting and enumerating both forms of explanations.

1 Introduction

The importance of devising mechanisms for computing explanations of Machine Learning (ML) models cannot be overstated, as illustrated by the fast-growing body of work in this area. A glimpse of the importance of explainable AI (XAI) is offered by a growing number of recent surveys and overviews [2,3,5,10,19,30–34,45,59–62,71,72,79].

Past work on computing explanations has mostly addressed *local* (or instance-dependent) explanations [15,16,38,51,69,70,75,76]. Exceptions include for example approaches that distill ML models, e.g. the case of NNs [26] among many others [69], or recent work on relating explanations with adversarial examples [39], both of which can be seen as seeking *global* (or instance-independent) explanations. Prior research has also mostly considered model-agnostic explanations [51,69,70]. Recent work on model-based explanations, e.g. [38,75], refers to local (or global) model-agnostic explanations as *heuristic*, given that these approaches offer no *formal* guarantees with respect to the underlying ML model. A taxonomy of ML model explanations is summarized in Table 1. Examples

https://doi.org/10.1007/978-3-030-77091-4_21

of heuristic approaches include [51,69,70], among many others[1]. In contrast, local (or global) model-based explanations are referred to as *rigorous*, since these offer the strongest formal guarantees with respect to the underlying ML model. Concrete examples of such rigorous approaches include [15,16,35,38–41,43,52,64,75–77].

Most work on computing explanations aims to answer a 'Why prediction π?' question. Some work proposes approximating the ML model's behavior with a linear model [51,69]. Most other work seeks to find a (often minimal) set of feature value pairs which is sufficient for the prediction, i.e. as long as those features take the specified values, the prediction does not change. For rigorous approaches, the answer to a 'Why prediction π?' question has been referred to as PI-explanations [75,76], abductive explanations [38], but also as (minimal) sufficient reasons [15,16]. (Hereinafter, we use the term *abductive explanation* because of the other forms of explanations studied in the paper.)

Another dimension of explanations, studied in recent work [60], is the difference between explanations for 'Why prediction π?' questions, e.g., 'Why did I get the loan?', and for 'Why prediction π and not δ?' questions, e.g., 'Why didn't I get the loan?'. Explanations for 'Why Not?' questions, labelled by [60] *contrastive* explanations, isolate a pragmatic component of explanations that *abductive explanations* lack. Concretely, an abductive explanation identifies a set of feature values which are sufficient for the model to make a prediction π and thus provides an answer to the question 'why π?' A contrastive explanation sets up a counterfactual link between what was a (possibly) *desired* outcome of a certain set of features and what was the observed outcome [1,13]. Thus, a contrastive explanation answers a 'Why π and not δ?' question [18,58,61].

In this paper we focus on the relationship between *local* abductive and contrastive explanations[2]. One of our contributions is to show how recent approaches for computing rigorous abductive explanations [15,16,38,75,76] can also be exploited for computing contrastive explanations. To our knowledge, this is new. In addition, we demonstrate that rigorous (model-based) local abductive and contrastive explanations are related by a minimal hitting set relationship[3], which builds on the seminal work of Reiter in the 80s [68]. Crucially, this novel hitting set relationship reveals a wealth of algorithms for computing and for enumerating contrastive and abductive explanations. We emphasize that it allows designing the first algorithm to *enumerate* abductive explanations. Finally, we demonstrate feasibility of our approach experimentally. Furthermore, our experiments show that there is a strong correlation between contrastive explanations and explanations produced by the commonly used SHAP explainer.

---

[1] There is also a recent XAI service offered by Google: https://cloud.google.com/explainable-ai/, inspired on similar ideas [28].

[2] In contrast with recent work [39], which studies the relationship between *global* model-based (abductive) explanations and adversarial examples.

[3] A local abductive (resp. contrastive) explanation is a minimal hitting set of the set of all local contrastive (resp. abductive) explanations.

Table 1. Taxonomy of ML model explanations used in the paper.

<table>
<tr><td colspan="2"></td><td colspan="3">Instance-</td></tr>
<tr><td colspan="2"></td><td colspan="2">Dependent</td><td>Independent</td></tr>
<tr><td rowspan="4">ML model-</td><td>Agnostic</td><td colspan="2">Heuristic local explanation for π
Examples: SHAP, LIME, Anchor, etc.</td><td>Heuristic global explanation for π.
Examples: SHAP, LIME (e.g. submodular pick)</td></tr>
<tr><td rowspan="3">Based</td><td colspan="2">Rigorous local explanation for π
Examples:</td><td rowspan="3">Rigorous global explanation for π
Examples: absolute/global AXps</td></tr>
<tr><td>‘Why π?’</td><td>‘Why not $\neg\pi$?’</td></tr>
<tr><td>PI- (abductive) explanations (AXps)</td><td>contrastive (CXps) (our work)</td></tr>
</table>

2 Preliminaries

Explainability in Machine Learning. The paper assumes an ML model $\mathbb{M}$, which is represented by a finite set of first-order logic (FOL) sentences $\mathcal{M}$. (When applicable, simpler alternative representations for $\mathcal{M}$ can be considered, e.g. (decidable) fragments of FOL, (mixed-)integer linear programming, constraint language(s), etc.)[4] A set of features $\mathcal{F} = \{f_1, \ldots, f_L\}$ is assumed. Each feature f_i is categorical (or ordinal), with values taken from some set D_i. An *instance* is an assignment of values to features. The space of instances, also referred to as *feature* (or *instance*) *space*, is defined by $\mathbb{F} = D_1 \times D_2 \times \ldots \times D_L$. For real-valued features, we require that a suitable interval discretization is applied first as a preprocessing step, e.g. if we consider an income feature for a person, we can split an interval of possible values into a set of intervals and treat each interval as a feature value. Therefore, our approach is applicable to any data that can be represented using a set of feature, e.g. tabular data, images, text, etc.

A (feature) literal λ_i is of the form $(f_i = v_i)$, with $v_i \in D_i$. In what follows, a literal will be viewed as an atom, i.e. it can take value *true* or *false*. As a result, an instance can be viewed as a set of L literals, denoting the L distinct features, i.e. an instance contains a single occurrence of a literal defined on any given feature. A set of literals is consistent if it contains at most one literal defined on each feature. A consistent set of literals can be interpreted as a conjunction or as a disjunction of literals; this will be clear from the context. When interpreted as a conjunction, the set of literals denotes a *cube* in instance space, where the unspecified features can take any possible value of their domain. When interpreted as a disjunction, the set of literals denotes a *clause* in instance space. As before, the unspecified features can take any possible value of their domain.

The remainder of the paper assumes a classification problem with a set of classes $\mathbb{K} = \{\kappa_1, \ldots, \kappa_M\}$. A prediction $\pi \in \mathbb{K}$ is associated with each instance

[4] $\mathcal{M}$ is referred to as the (formal) model of the ML model $\mathbb{M}$. The use of FOL is not restrictive, with fragments of FOL being used in recent years for modeling ML models in different settings. These include NNs [38] and Bayesian Network Classifiers [76], among others.

$X \in \mathbb{F}$. Throughout this paper, an ML model $\mathbb{M}$ will be associated with some logical representation (or encoding), whose consistency depends on the (input) instance and (output) prediction. Thus, we define a predicate $\mathcal{M} \subseteq \mathbb{F} \times \mathbb{K}$, such that $\mathcal{M}(X, \pi)$ is true iff the input X is consistent with prediction π given the ML model $\mathbb{M}$[5]. We further simplify the notation by using $\mathcal{M}_\pi(X)$ to denote a predicate $\mathcal{M}(X, \pi)$ for a concrete prediction π.

Moreover, we will compute *prime implicants* of $\mathcal{M}_\pi$. These predicates defined on $\mathbb{F}$ and represented as consistent conjunctions (or alternatively as sets) of feature literals. Concretely, a consistent conjunction of feature literals τ is an implicant of $\mathcal{M}_\pi$ if the following FOL statement is true:

$$\forall (X \in \mathbb{F}).\tau(X) \rightarrow \mathcal{M}(X, \pi) \tag{1}$$

The notation $\tau \vDash \mathcal{M}_\pi$ is used to denote that τ an implicant of $\mathcal{M}_\pi$. Similarly, a consistent set of feature literals ν is the negation of an implicate of $\mathcal{M}_\pi$ if the following FOL statement is true:

$$\forall (X \in \mathbb{F}).\nu(X) \rightarrow (\vee_{\rho \neq \pi} \mathcal{M}(X, \rho)) \tag{2}$$

$\mathcal{M}_\pi \vDash \neg\nu$, or alternatively $(\nu \vDash \neg\mathcal{M}_\pi) \equiv (\nu \vDash \vee_{\rho \neq \pi} \mathcal{M}_\rho)$. An implicant τ (resp. implicate ν) is called *prime* if none of its proper subsets $\tau' \subsetneq \tau$ (resp. $\nu' \subsetneq \nu$) is an implicant (resp. implicate).

Abductive explanations represent prime implicants of the decision function associated with some predicted class π[6].

Analysis of Inconsistent Formulas. Throughout the paper, we will be interested in formulas $\mathcal{F}$ that are *inconsistent* (or *unsatisfiable*), i.e. $\mathcal{F} \vDash \bot$, represented as conjunctions of clauses. Some clauses in $\mathcal{F}$ can be *relaxed* (i.e. allowed not to be satisfied) to restore consistency, whereas others cannot. Thus, we assume that $\mathcal{F}$ is partitioned into two first-order subformulas $\mathcal{F} = \mathcal{B} \cup \mathcal{R}$, where $\mathcal{R}$ contains the *relaxable* clauses, and $\mathcal{B}$ contains the *non-relaxable* clauses. $\mathcal{B}$ can be viewed as (consistent) background knowledge, which must always be satisfied.

Given an inconsistent formula $\mathcal{F}$, represented as a set of first-order clauses, we identify the clauses that are responsible for unsatisfiability among those that can be relaxed, as defined next[7].

Definition 1 (Minimal Unsatisfiable Subset (MUS)). *Let $\mathcal{F} = \mathcal{B} \cup \mathcal{R}$ denote an inconsistent set of clauses ($\mathcal{F} \vDash \bot$). $\mathcal{U} \subseteq \mathcal{R}$ is a* Minimal Unsatisfiable Subset *(MUS) iff $\mathcal{B} \cup \mathcal{U} \vDash \bot$ and $\forall_{\mathcal{U}' \subsetneq \mathcal{U}}$, $\mathcal{B} \cup \mathcal{U}' \nvDash \bot$.*

[5] This alternative notation is used for simplicity and clarity with respect to earlier work [38,39,75]. Furthermore, defining $\mathcal{M}$ as a predicate allows for multiple predictions for the same point in feature space. Nevertheless, such cases are not considered in this paper.

[6] By definition of prime implicant, abductive explanations are sufficient reasons for the prediction. Hence the names used in recent work: abductive explanations [38], PI-explanations [75,76] and sufficient reasons [15,16].

[7] The definitions in this section are often presented for the propositional case, but the extension to the first-order case is straightforward.

Informally, an MUS provides the minimal information that needs to be added to the background knowledge $\mathcal{B}$ to obtain an inconsistency; it explains the causes for this inconsistency. Alternatively, one might be interested in correcting the formula, removing some clauses in $\mathcal{R}$ to achieve consistency.

Definition 2 (Minimal Correction Subset (MCS)). *Let $\mathcal{F} = \mathcal{B} \cup \mathcal{R}$ denote an inconsistent set of clauses ($\mathcal{F} \models \bot$). $\mathcal{T} \subseteq \mathcal{R}$ is a* Minimal Correction Subset *(MCS) iff $\mathcal{B} \cup \mathcal{R} \setminus \mathcal{T} \not\models \bot$ and $\forall_{\mathcal{T}' \subsetneq \mathcal{T}}$, $\mathcal{B} \cup \mathcal{R} \setminus \mathcal{T}' \models \bot$.*

A fundamental result in reasoning about inconsistent clause sets is the minimal hitting set (MHS) duality relationship between MUSes and MCSes [11,68]: *MCSes are MHSes of MUSes and vice-versa.* This result has been extensively used in the development of algorithms for MUSes and MCSes [8,48,49], and also applied in a number of different settings. Recent years have witnessed the proposal of a large number of novel algorithms for the extraction and enumeration of MUSes and MCSes [7,9,29,48]. Although most work addresses propositional theories, these algorithms can easily be generalized to any other setting where entailment is monotonic, e.g. SMT [17].

Running Example. The following example will be used to illustrate the main ideas.

Example 1. We consider a textbook example [66] [Figure 7.1, page 289] addressing the classification of a user's preferences regarding whether to read or to skip a given book. For this dataset, the set of features is:

$$\{\ \mathsf{A(uthor)}, \mathsf{T(hread)}, \mathsf{L(ength)}, \mathsf{W(hereRead)}\ \}$$

All features take one of two values, respectively $\{\mathsf{known}, \mathsf{unknown}\}$, $\{\mathsf{new}, \mathsf{followUp}\}$, $\{\mathsf{long}, \mathsf{short}\}$, and $\{\mathsf{home}, \mathsf{work}\}$. An example instance is: $\{(\mathsf{A} = \mathsf{known}), (\mathsf{T} = \mathsf{new}), (\mathsf{L} = \mathsf{long}), (\mathsf{W} = \mathsf{home})\}$. This instance is identified as e_1 [66] with prediction skips. Figure 1a shows a possible decision tree for this example [66][8]. The decision tree can be represented as a set of rules as shown in Fig. 1b[9].

Our goal is to reason about the ML model, i.e. to implement model-based reasoning, so we need to propose a logical representation for the ML model.

Example 2. For implementing model-based reasoning, we need to develop an encoding in some suitable fragment of FOL[10]. 0-place predicates[11] are used for

[8] The choice of a decision tree aims only at keeping the example(s) presented in the paper as simple as possible. The ideas proposed in the paper apply to *any* ML model that can be represented with FOL. This encompasses *any* existing ML model, with minor adaptations in case the ML model keeps state.

[9] The abbreviations used relate with the names in the decision tree, and serve for saving space.

[10] Depending on the ML problem, more expressive fragments of FOL logic could be considered [47]. Well-known examples include real, integer and integer-real arithmetic, but also nonlinear arithmetic [47].

[11] Which in this case are used as propositional variables.

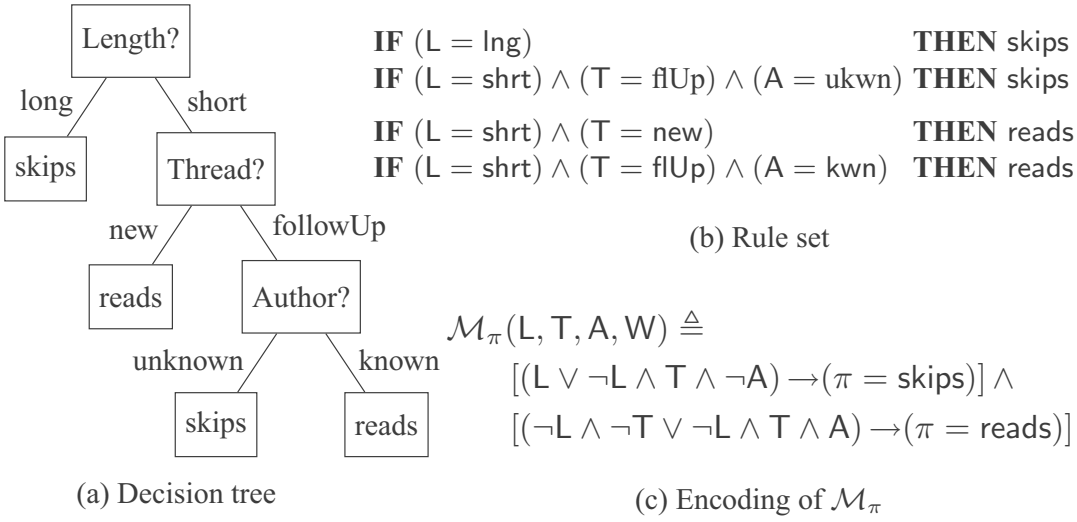


Fig. 1. Running example [66]

L, T, A and W, as follows. We will associate $(\mathsf{L} = \mathsf{long})$ with L and $(\mathsf{L} = \mathsf{short})$ with $\neg\mathsf{L}$. Similarly, we associate $(\mathsf{T} = \mathsf{new})$ with $\neg\mathsf{T}$, and $(\mathsf{T} = \mathsf{followUp})$ with T. We associate $(\mathsf{A} = \mathsf{known})$ with A and $(\mathsf{A} = \mathsf{unknown})$ with $\neg\mathsf{A}$. Furthermore, we associate $(\mathsf{W} = \mathsf{home})$ with $\neg\mathsf{W}$ and $(\mathsf{W} = \mathsf{work})$ with W. An example encoding is shown in Fig. 1c. The explicit values of π are optional (i.e. propositional values could be used) and serve to illustrate how non-propositional valued could be modeled.

3 Contrastive vs. Abductive Explanations

Recent work [15,38,75,76] proposed to relate model-based explanations with prime implicants. All these approaches compute a set of feature values which, if unchanged, are sufficient for the prediction. Thus, one can view such explanations as answering a 'Why?' question: *the prediction is the one given, as long as some selected set of feature values is the one given.* In this paper, such explanations will be referred to as *abductive explanations*, motivated by one of the approaches used for their computation [38].

3.1 Defining Abductive Explanations (AXps)

As indicated earlier in the paper, we focus on *local model-based* explanations.

Definition 3 (Abductive Explanation [38]**).** *Given an instance τ, with a prediction π, and an ML model represented with a predicate $\mathcal{M}_\pi$, i.e. $\tau \models \mathcal{M}_\pi$, an* abductive explanation *is a minimal subset of literals of τ, $\sigma \subseteq \tau$, such that $\sigma \models \mathcal{M}_\pi$.*

Example 3. With respect to Example 1, let us consider the instance $(\mathsf{A} = \mathsf{known}, \mathsf{T} = \mathsf{new}, \mathsf{L} = \mathsf{short}, \mathsf{W} = \mathsf{work})$, which we represent instead as

$(\mathsf{A}, \neg\mathsf{T}, \neg\mathsf{L}, \mathsf{W})$, corresponding to prediction $\pi = \mathsf{reads}$. By inspection of the decision tree (see Fig. 1a), a possible answer to the 'Why pred. reads?' question is: $\{\neg\mathsf{L}, \neg\mathsf{T}\}$. In this concrete case we can conclude that this is the only abductive explanation by inspection of the decision tree.

3.2 Defining Contrastive Explanations (CXps)

As [60] notes, contrastive explanations are,

> *"sought in response to particular counterfactual cases... That is, people do not ask why event P happened, but rather why event P happened instead of some event Q."*

As a result, we are interested in providing an answer to the question 'Why π and not δ?', where π is the prediction given some instance τ, and δ is some other (desired) prediction.

Example 4. We consider again Example 1, but with the instance specified in Example 3. A possible answer to the question 'Why pred. reads and not pred. skips??' is $\{\mathsf{L}\}$. Indeed, given the input instance $(\mathsf{A}, \neg\mathsf{T}, \neg\mathsf{L}, \mathsf{W})$, if the value of feature L changes from short to long, and the value of the other features remains unchanged, then the prediction will change from reads to skips.

The following definition of a (local model-based) contrastive explanation captures the intuitive notion of the contrastive explanation discussed in the example above.

Definition 4 (Contrastive Explanation). *Given an instance τ, with a prediction π, and an ML model represented by a predicate $\mathcal{M}_\pi$, i.e. $\tau \models \mathcal{M}_\pi$, a* contrastive explanation *is a minimal subset of literals of τ, $\rho \subseteq \tau$, such that $\tau \setminus \rho \not\models \mathcal{M}_\pi$.*

This definition means that, there is an assignment to the features with literals in ρ, such that the prediction differs from π. Observe that a CXp is defined to answer the following (more specific) question 'Why (pred. π and) not $\neg\pi$?'. The more general case of answering the question 'Why (pred. π and) not δ?' will be analyzed later. Also, we point out a connection between notions of CXp and robustness defined in [74]. In [74], the local robustness for a given instance τ is defined as the minimum Hamming distance from τ to an perturbed input τ' s.t. $\tau' \models \neg\mathcal{M}_\pi$. Note that given such a perturbed sample τ', we can obtain a minimum size CXp. This CXp contains all perturbed features of τ. Furthermore, links between robustness and counterfactual explainability (and so contrastive explanations) have been studied in recent work [6].

3.3 Relating Abductive and Contrastive Explanations

The previous section proposed a rigorous, model-based, definition of contrastive explanation. Given this definition, one can think of developing dedicated algo-

rithms that compute CXps using a decision procedure for the logic used for representing the ML model. Instead, we adopt a simpler approach. We build on a fundamental result from model-based diagnosis on the hitting set relationship between diagnoses and conflicts first investigated by Reiter [68] (and more generally for reasoning about inconsistency [8,11]) and demonstrate a similar relationship between AXps and CXps. In turn, this result reveals a variety of novel algorithms for computing CXps, but also offers ways for enumerating both CXps and AXps.

Local Abductive Explanations (AXps). Consider a set of feature values τ, s.t. the prediction is π, for which the notation $\tau \models \mathcal{M}_\pi$ is used. We will use the equivalent statement, $\tau \land \neg\mathcal{M}_\pi \models \bot$. Thus,

$$\tau \land \neg\mathcal{M}_\pi \tag{3}$$

is inconsistent, with the background knowledge being $\mathcal{B} \triangleq \neg\mathcal{M}_\pi$ and the relaxable clauses being $\mathcal{R} \triangleq \tau$. As proposed in [38,75], a (local abductive) explanation is a subset-minimal set σ of the literals in τ, such that, $\sigma \land \neg\mathcal{M}_\pi \models \bot$. Thus, σ denotes a subset of the example's input features which, no matter the other feature values, ensure that the ML model predicts π. Thus, any MUS of Eq. 3 is a (local abductive) explanation for $\mathbb{M}$ to predict π given τ.

Proposition 1. *Local model-based abductive explanations are MUSes of the pair* $(\mathcal{B}, \mathcal{R})$*,* $\tau \land \neg\mathcal{M}_\pi$*, where* $\mathcal{R} \triangleq \tau$ *and* $\mathcal{B} \triangleq \neg\mathcal{M}_\pi$*.*

Example 5. Consider the ML model from Example 1, the encoding from Example 2, and the instance $\{\mathsf{A}, \neg\mathsf{T}, \mathsf{L}, \neg\mathsf{W}\}$, with prediction $\pi = \mathsf{skips}$ (wrt Fig. 1, we replace $\mathsf{skips} = \mathsf{skips}$ with **true** and $\mathsf{skips} = \mathsf{reads}$ with **false**). We can thus confirm that $\tau \models \mathcal{M}_\pi$. We observe that the following holds:

$$\mathsf{A} \land \neg\mathsf{T} \land \mathsf{L} \land \neg\mathsf{W} \models \begin{bmatrix} (\mathsf{L} \lor \neg\mathsf{L} \land \mathsf{T} \land \neg\mathsf{A}) \rightarrow \mathbf{true} \\ \land \\ (\neg\mathsf{L} \land \neg\mathsf{T} \lor \neg\mathsf{L} \land \mathsf{T} \land \mathsf{A}) \rightarrow \mathbf{false} \end{bmatrix} \tag{4}$$

which can be rewritten as,

$$\mathsf{A} \land \neg\mathsf{T} \land \mathsf{L} \land \neg\mathsf{W} \land \begin{bmatrix} (\mathsf{L} \lor \neg\mathsf{L} \land \mathsf{T} \land \neg\mathsf{A}) \land \neg\mathbf{true} \\ \lor \\ (\neg\mathsf{L} \land \neg\mathsf{T} \lor \neg\mathsf{L} \land \mathsf{T} \land \mathsf{A}) \land \neg\mathbf{false} \end{bmatrix} \tag{5}$$

It is easy to conclude that Eq. 5 is inconsistent. Moreover, $\sigma = (\mathsf{L})$ denotes an MUS of Eq. 5 and denotes one abductive explanation for why the prediction is skips for the instance τ.

Local Contrastive Explanations (CXps). Suppose we compute instead an MCS ρ of Eq. 3, with $\rho \subseteq \tau$. As a result, $\bigwedge_{l \in \tau \setminus \rho}(l) \land \neg\mathcal{M}_\pi \not\models \bot$ holds. Hence, assigning feature values to the inputs of the ML model is consistent with a prediction that is *not* π, i.e. a prediction of some value other than π. Observe that ρ is a subset-minimal set of literals which causes $\tau \setminus \rho \land \neg\mathcal{M}_\pi$ to be satisfiable, with any satisfying assignment yielding a prediction that is not π.

Proposition 2. *Local model-based contrastive explanations are MCSes of the pair* $(\mathcal{B}, \mathcal{R})$*,* $\tau \land \neg\mathcal{M}_\pi$*, where* $\mathcal{R} \triangleq \tau$ *and* $\mathcal{B} \triangleq \neg\mathcal{M}_\pi$*.*

Example 6. From Eq. 3 and Eq. 5 we can also compute $\rho \subseteq \tau$ such that $\tau \setminus \rho \land \neg\mathcal{M}_\pi \not\models \bot$. For example $\rho = (\mathsf{L})$ is an MCS of Eq. 5[12]. Thus, from $\{\mathsf{A}, \neg\mathsf{T}, \neg\mathsf{W}\}$ we can get a prediction other than skips, by considering feature value $\neg\mathsf{L}$.

Duality Among Explanations. Given the results above, and the hitting set duality between MUSes and MCSes [11,68], we have the following.

Theorem 1. *AXps are MHSes of CXps and vice-versa.*

Proof. Immediate from Definition 3, Definition 4, Proposition 1, Proposition 2, and Theorem 4.4 and Corollary 4.5 of [68]. □

Proposition 1, Proposition 2, and Theorem 1 can now serve to exploit the vast body of work on the analysis of inconsistent formulas for computing both contrastive and abductive explanations and, arguably more importantly, to enumerate explanations. Existing algorithms for the extraction and enumeration of MUSes and MCSes require minor modifications to be applied in the setting of AXps and CXps. The resulting algorithms are briefly summarized in Sect. 4. Interestingly, a consequence of the duality is that computing an abductive explanation is *harder* than computing a contrastive explanation in terms of the number of calls to a decision procedure Sect. 4.

Discussion. As observed above, the contrastive explanations we are computing answer the question: 'Why (π and) not $\neg\pi$?'. A more general contrastive explanation would be 'Why (π and) not δ, with $\pi \neq \delta$?' [60]. Note that, since the prediction π is given, we are only interested in changing the prediction to either $\neg\pi$ or δ. We refer to answering the first question as a *basic* contrastive explanation, whereas answering the second question will be referred to as a *targeted* contrastive explanation, and written as CXp_δ. The duality result between AXps and CXps in Theorem 1 applies *only* to basic contrastive explanations. Nevertheless, the algorithms for MCS extraction for computing a basic CXp can also be adapted to computing targeted CXps, as follows. We want a pick of feature values such that the prediction is δ. We start by letting all features to take any value, and such that the resulting prediction is δ. We then iteratively attempt to fix feature values to those in the given instance, while the prediction remains δ. This way, the set of literals that change value are a subset-minimal set of feature-value pairs that is sufficient for predicting δ. Finally, there are crucial differences between the duality result established in this section, which targets local explanations, and a recent result [39], which targets *global* explanations. Earlier work established a relation between prime implicants and implicates as a way to relate global abductive explanations and so-called counterexamples.

[12] Although in general not the case, in Example 5 and Example 6 an MUS of size 1 is also an MCS of size 1.

Algorithm 1. Enumeration of CXps

Function CXpEnum($\mathcal{M}_\pi$,$\mathcal{C}$, π)
Input: $\mathcal{M}_\pi$: ML model, $\mathcal{C}$: Input cube, π: Prediction
Variables: $\mathcal{N}$ and $\mathcal{P}$ defined on the variables of $\mathcal{C}$

$\mathcal{I} \leftarrow \emptyset$; // Block CXps
while true **do**
  $\mu \leftarrow$ ExtractCXp($\mathcal{M}_\pi, \mathcal{C}, \pi, \mathcal{I}$)
  if $\mu = \emptyset$ **then break**;
  ReportCXp(μ)
  $\mathcal{I} \leftarrow \mathcal{I} \cup$ NegateLiteralsOf(μ)

In contrast, we delved into the fundamentals of reasoning about inconsistency, concretely the duality between MCSes and MUSes, and established a relation between model-based *local* AXps and CXps.

4 Extracting and Enumerating Explanations

The results of Sect. 3.3 enable exploiting past work on extracting and enumerating MCSes and MUSes to the setting of contrastive and abductive explanations, respectively. Perhaps surprisingly, there is a stark difference between algorithms for extraction and enumeration of contrastive explanations and abductive explanations. Due to the association with MCSes, one contrastive explanation can be computed with a logarithmic number of calls to a decision procedure [49]. Moreover, there exist algorithms for the direct enumeration of contrastive explanations [49]. In contrast, abductive explanations are associated with MUSes. As a result, any known algorithm for extraction of one abductive explanation requires at best a linear number of calls to a decision procedure [42,44,54,55], in the worst-case. Moreover, there is no known algorithm for the direct enumeration of abductive explanations, and so enumeration can be achieved only through the enumeration of contrastive explanations [23,48,49,53,56,57].

We apply state-of-the-art algorithms for the enumeration of MUSes and MCSes [8,9,29,48,49] to find all the abductive and contrastive explanations. Note that, as in the case of enumeration of MCSes and MUSes, enumeration of CXps is comparatively easier than enumeration of AXps. Algorithm 1 shows our application of MCS enumeration algorithm to the enumeration of CXps [49]. Other alternatives [29] could be considered instead. Algorithm 1 finds a CXp, blocks it and finds the next one until no more exists. To extract a single CXp, we can use a standard algorithm, e.g. [8,53,56,57]. In principle, enumeration of AXps can be achieved by computing all CXps and then computing all the minimal hitting sets of all CXps, as proposed in the propositional setting [49]. However, there are more efficient alternatives that we can adapt here [8,9,48,63]. Algorithm 2 applies [48] to the case of computing both AXps and CXps. The algorithm simultaneously searches for AXps and CXps and is based on the hitting set duality defined above.

Algorithm 2. Enumeration of AXps (and CXps)

Function XpEnum($\mathcal{M}_\pi$,$\mathcal{C}$, π)

Input: $\mathcal{M}_\pi$: ML model, $\mathcal{C}$: Input cube, π: Prediction

Variables: $\mathcal{N}$ and $\mathcal{P}$ defined on the variables of $\mathcal{C}$

$\mathcal{K} = (\mathcal{N}, \mathcal{P}) \leftarrow (\emptyset, \emptyset)$; `// Block AXps & CXps`

while true **do**

  $(st_\lambda, \lambda) \leftarrow$ `FindMHS`$(\mathcal{P}, \mathcal{N})$; `// MHS of` $\mathcal{P}$ st $\mathcal{N}$

  if $\neg st_\lambda$ **then break**;

  $(st_\rho, \rho) \leftarrow$ `SAT`$(\lambda \land \neg\mathcal{M}_\pi)$

  if $\neg st_\rho$ **then** `// entailment holds`

    `ReportAXp`(λ)

    $\mathcal{N} \leftarrow \mathcal{N} \cup$ `NegateLiteralsOf`(λ)

  else

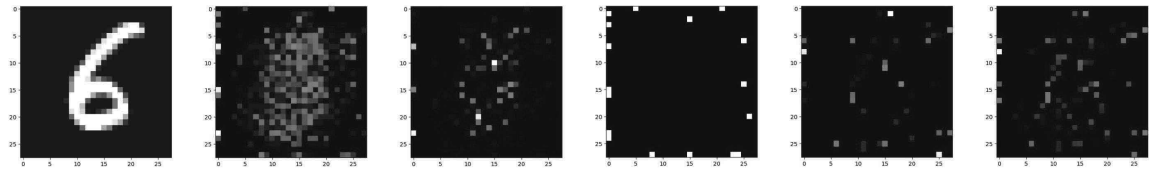
    $\mu \leftarrow$ `ExtractCXp`$(\mathcal{M}_\pi, \rho, \pi)$

    `ReportCXp`(μ)

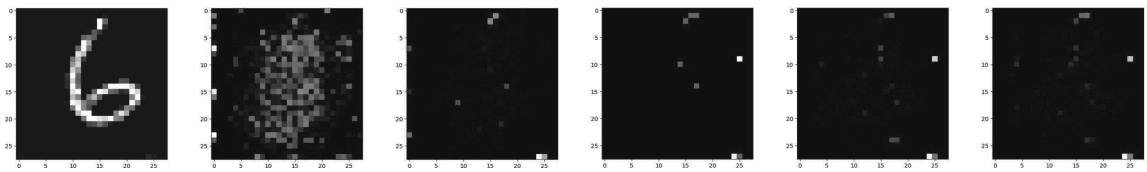
    $\mathcal{P} \leftarrow \mathcal{P} \cup$ `UseLiteralsOf`(μ)

5 Experimental Evaluation

This section details the experimental evaluation to assess the practical feasibility and efficiency of the enumeration of abductive and contrastive explanations for a few real-world datasets, studied in the context of explainability and algorithmic fairness. To evaluate, we use Algorithm 1 and Algorithm 2 in Sect. 4[13].



(a) Real 6 (b) XGBoost (c) SHAP (d) CXp^1 (e) CXp^2 (f) CXp$^{1–3}$



(g) Fake 6 (h) XGBoost (i) SHAP (j) CXp^3 (k) CXp^4 (l) CXp$^{3–5}$

Fig. 2. The 'real vs fake' images. The first row shows results for the real image 6; the second – results for the fake image 6. The first column shows examples of inputs; the second – heatmaps of XGBoost's important features; the third – heatmaps of SHAP's explanation. Last three columns show heatmaps of CXp of different cardinality. The brighter pixels are more influential features.

[13] The prototype and the experimental setup are available at https://github.com/alexeyignatiev/xdual.

5.1 Enumeration of CXps

Our experiments demonstrate a novel, unexpected practical use case of CXps enumeration algorithms. In particular, we show that our method gives a *new fine-grained view* on both global and local standard explanations extracted from ML models. The goal of these experiments is to *gain better understanding of existing explainers* rather than generate all CXps for a given input.

Setup. To perform enumeration of CXps in our first experiment, we use a constraint programming solver, ORtools [65]. To encode the enumeration problem with ORtools we converted scores of XGBoost models into integers keeping 5 digits precision. We enumerate contrastive explanations in the increasing order by their cardinality. This can be done by a simple modification of Algorithm 1 forcing it to return CXps in this order. So, we first obtain all minimum size contrastive explanations, and so on.

We conduct two sets of experiments. The first experiment, called "real vs fake", distinguishes real from fake images. A dataset contains two classes of images: (a) original MNIST digits and (b) fake MNIST digits produced by a standard DCGAN model [67] (see Fig. 2a and Fig. 2g for typical examples). The second experiment, called "3 vs 5 digits", uses a dataset that contains digits "3" and "5" from the standard MNIST dataset (these digits are similar in writing) and we distinguish "3" from "5" images.

Brief Overview of the SHAP Explainer. Given a classifier f and an explainer model g, SHAP aims to train g be similar to f in the neighborhood of some given point x. The objective function for SHAP is designed so that: (1) g approximates the behavior of the black box f accurately within the vicinity of x, and (2) g achieves lower complexity and is interpretable: $\xi(x) = \arg\min_{g \in G} L(\pi_x, g, f) + \Omega(g)$, where the loss function L is defined to minimize the distance between f and g in the neighborhood of x using a weight function π_x and $\Omega(g)$ quantifies the complexity of g; $\Omega(g)$ and π_x are defined based on game-theoretic notions [51]. We chose SHAP for our experiments due to its efficiency to generate an explanation (within seconds per input).

"Real vs Fake" Experiment. First, we discuss the results of the "real vs fake" experiment in details. We train an XGBoost model [14] with 100 trees of depth 6 (accuracy 0.85/0.80 on train/test sets). We quantized images so that each pixel takes a value between 0 and 15, image pixels are categorical features in the model.

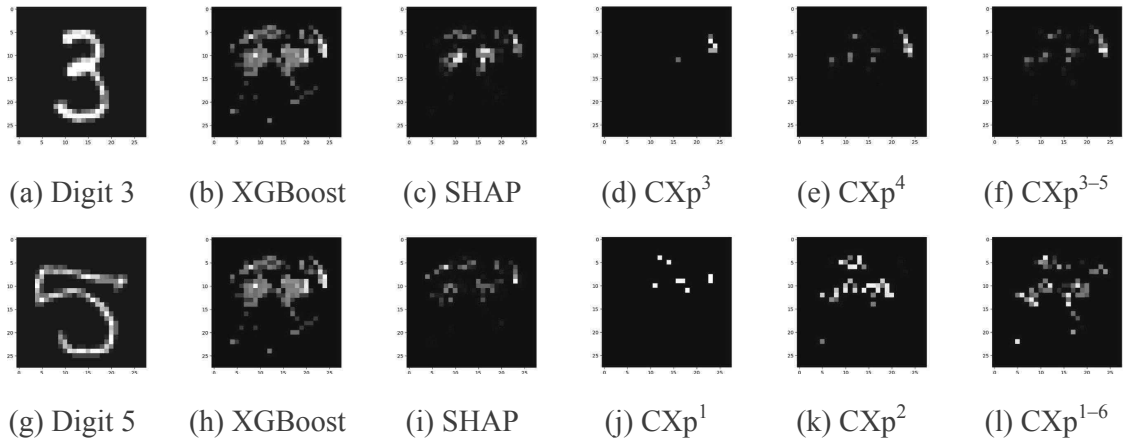
Global and Local Explainers. We start by discussing our results on a few samples (Fig. 2a and Fig. 2g). First, we extract important features provided by XGBoost. As these features are *global* for the model, they are the same for all inputs (Fig. 2b and Fig. 2h are identical for real and fake images). Figure 2b shows that these important features are *no very informative* for this dataset as these pixels

form a blob of pixels that cover an image. Then we compute an image-specific explanation using the standard explainer SHAP (see Fig. 2c for the real image and Fig. 2i for the fake image). SHAP explanations are more focused on specific parts of images compared to XGBoost. However, it is still not easy to gain insights about which areas of an image are more important as pixels all over the image participate in the explanations of SHAP and XGBoost. For example, both XGBoost and SHAP distinguish some edge and middle pixels as key pixels (the bright pixels are more important) but it is not clear why these are important pixels.

Enumeration. Our goal here is to investigate whether there is a connection between the important pixels that SHAP/XGBoost finds and CXps for a given image. The most surprising result is that, indeed, a connection exists and, for example, it reveals that the edge pixels of an image, highlighted by both SHAP and XGBoost as important pixels, are, reveal in fact, CXps of small cardinalities. For an image, we enumerate first 2000 CXps. Given all CXps of size k, we plot a heatmap of occurrences of each pixel in these CXps of size k. Let us focus on the first row with the real 6. Consider the heatmap CXp^1 at Fig. 2d that shows all CXps of size one for the real 6. It shows that most of important pixels of XGBoost and SHAP are actually CXps of size one. This means that *it is sufficient to change a single pixel value to some other value to obtain a different prediction.* Note that these results lead us to an interesting observation. DCGAN generates images with a few gray edges pixels (see Fig. 4. Indeed, some of them have several edge pixels in gray.) This 'defect' does not happen often for real MNIST images. Therefore, the classifier 'hooks' on this issue to classify an image as fake. Now, consider the heatmap CXp^2 at Fig. 2e of CXps of size two. It overlaps a lot with SHAP important pixels in the middle of the image explaining *why* these are important. Only a *pair* of these pixels can be changed to get a different prediction.

A Correlation Between CXps and SHAP's Important Features. To qualitatively measure our observations on correlation between key features of CXps and SHAP, we conducted the same experiment as above on 100 random images and measured the correlation between first CXps and SHAP features. First, we compute a set T of pixels that is the union of the first (top) 100 smallest size CXps. On average, we have 60 pixels in T. Note that the average 60 pixels represent a small fraction (7%) of the total number of pixels. Then we find a set S of $|T|$ SHAP pixels with highest absolute weights. Finally, we compute $corr = |S \cap T|/|S|$ as the correlation measure. Note that $corr = 0.4$ on average, i.e. our method hits 40% of best SHAP features. As the chances of two tools independently hitting the same pixel (out of 784) are quite low, the fact that 40% of $|S|$ are picked indicates a significant correlation.

"3 vs 5 Digits" Experiment. Consider our second the "3 vs 5 digits" experiment. We use a dataset that contains digits "3" (class 0) and "5" (class 1) from the standard MNIST (see Fig. 3a and Fig. 3g for representative samples).



(a) Digit 3 (b) XGBoost (c) SHAP (d) CXp^3 (e) CXp^4 (f) $CXp^{3–5}$

(g) Digit 5 (h) XGBoost (i) SHAP (j) CXp^1 (k) CXp^2 (l) $CXp^{1–6}$

Fig. 3. Results of the 3 vs 5 digits experiments. The first row shows results for the image 3. The second row shows results for the image 5. The first column shows examples of inputs; the second column shows heatmaps of XGBoost's global important features; the third column shows heatmaps of SHAP's important features. Last three columns show heatmaps of CXp of different cardinality.

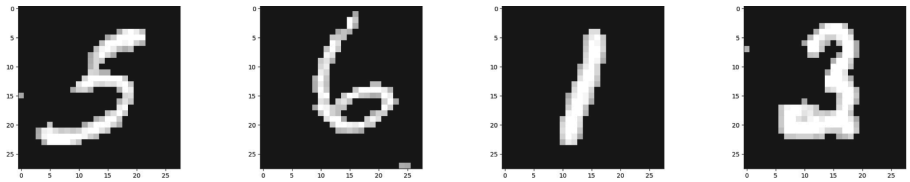


Fig. 4. Additional fake images. We reduced values of zero-valued pixels to highlight gray pixels on the edges for some fake images.

XGboost model has 50 trees of depth 3 with accuracy 0.98 (0.97) on train/test sets. We quantized images so that each pixel takes a value between 0 and 15. As before, each pixel corresponds to a feature. So, we have 784 features in our XGBoost model.

Global and Local Explainers. We start by discussing our results on few random samples (Fig. 3a and Fig. 3g). First, we obtain the important features from XGBoost. As these features are *global* for the model so they are the same for all inputs (Fig. 3b and Fig. 3h are identical for 3 and 5 images). Figure 2b shows that these important features. The important pixels highlight that the top parts of images are important, which is a plausible high-level explanation of the classifier behavior. Digits 3 and 5 are mostly differ in the top part of the image. However, some pixels are way more important than other and it is hard to understand why.

Next, we compute an image-specific explanation using the standard explainer SHAP (see Fig. 3c for the digit 3 and Fig. 3c for the digit 5). While SHAP explanations mimic XGBoost important features, they do provide additional insights for the user. Note that both XGBoost and SHAP mark a "belt" of pixels in the upper middle part that as important (bright pixels is the most important pixels).

Enumeration. We run our enumeration algorithm to produce CXps of increasing cardinality. For each image, we enumerate first 2000 CXps. Given all CXps of size k, we plot a heatmap of occurrences of each pixel in these CXps of size k. Let us focus on the second row with the digit 5. For example, CXp^2 (Fig. 3k) shows the heatmap of CXps of size two for the digit 5. As we mentioned above, both XGBoost and SHAP hint that the 'belt' of important pixels in the middle. Again, our method can explain *why* this is the case. Consider the heatmap CXp^1 at Fig. 3j. This picture shows all CXps of size one for the digit 5. It reveals that most of important pixels of XGBoost and SHAP are actually CXps of size one. We reiterate that it is sufficient to change a *single* pixel value to some other value to obtain a different prediction. Now, consider the heatmap $\text{CXp}^{1\text{–}6}$ at Fig. 3l. This figure shows 2000 CXps (from size 1 to size 6). It overlaps a lot with SHAP important pixels in the middle of the image. So, these pixels occur in many small size CXps and changing their values leads to misclassification.

Correlation Between CXps and SHAP Features. To qualitatively measure our observations on correlation between key features of CXps and SHAP, we conducted the same experiment as above on 100 random images and measured the correlation between CXps and SHAP features. First, we compute a set T of pixels that is the union of the first (top) 100 smallest size CXps. On average, we have 38 pixels in T. Note that the average 38 pixels represent a small fraction (5%) of the total number of pixels. Then we find a set S of $|S|$ SHAP pixels with highest absolute weights. Finally, we compute $\mathit{corr} = |S \cap T|/|S|$ as the correlation measure. Note that $\mathit{corr} = 0.6$ on average, i.e. our method hits 60% of best SHAP features. As the chances of two tools independently hitting the same pixel (out of 784) are quite low, the fact that 60% of $|T|$ are picked indicates a significant correlation.

5.2 Enumeration of CXps and AXps

Datasets. Here, we aim at testing the *scalability* of explanation enumeration. The results are obtained on the six well-known and publicly available datasets. Three of them were previously studied in [70] in the context of heuristic explanation approaches, namely, Anchor [70] and LIME [69], including *Adult*, *Lending*, and *Recidivism*. These datasets were processed the same way as in [70]. The *Adult* dataset [46] is originally taken from the Census bureau and targets predicting whether or not a given adult person earns more than \$50K a year depending on various attributes, e.g. education, hours of work, etc. The *Lending* dataset aims at predicting whether or not a loan on the Lending Club website will turn out bad. The *Recidivism* dataset was used to predict recidivism for individuals released from North Carolina prisons in 1978 and 1980 [73]. Two more datasets were additionally considered including *Compas* and *German* that were previously studied in the context of the FairML and Algorithmic Fairness projects [21, 22, 24, 25], an area in which the need for explanations is doubtless. *Compas* is a popular dataset, known [4] for exhibiting racial bias of the COMPAS algorithm used for scoring criminal defendant's likelihood of reoffending.

Table 2. Results of the computational experiment on enumeration of AXps and CXps.

	Dataset					
	Adult	Lending	Recidivism	Compas	German	Spambase
# of instances	5579	4414	3696	778	1000	2344
Total time (sec.)	7666.9	443.8	3688.0	78.4	16943.2	6859.2
Minimal time (sec.)	0.1	0.0	0.1	0.0	0.2	0.1
Average time (sec.)	1.4	0.1	1.0	0.1	16.9	2.9
Maximal time (sec.)	13.1	0.8	8.9	0.5	193.0	23.1
Total oracle calls	492990	69653	581716	21227	748164	176354
Minimal oracle calls	14	11	17	13	23	12
Average oracle calls	88.4	15.8	157.4	27.3	748.2	75.2
Maximal oracle calls	581	73	1426	134	7829	353.
Total # of AXps	52137	8105	60688	1931.0	59222	18876
Average # of AXps	9.4	1.8	16.4	2.5	59.2	8.1
Average AXp size	5.3	1.9	6.4	3.8	7.5	4.6
Total # of CXps	66219	8663	77784	3558.0	66781	24774
Average # of CXps	11.9	2.0	21.1	4.6	66.8	10.6
Average CXp size	2.4	1.4	2.6	1.5	3.6	2.3

The latter dataset is a German credit data (e.g. see [22,25]), which given a list of people's attributes classifies them as good or bad credit risks. Finally, we consider the *Spambase* dataset from the UCI repository [20]. The main goal is to classify an email as spam or non-spam based on the words that occur in this email. Due to scalability constraints, we preprocessed the dataset to keep ten words per email that were identified as the most influential words by a random forest classifier.

Implementation and Setup. A prototype implementing Algorithm 2 targeting the enumeration of either (1) all abductive or (2) all contrastive explanations was created. In the experiment, the prototype implementation is instructed to enumerate all abductive explanations. (Note that, as was also mentioned before, no matter what kind of explanations Algorithm 2 aims for, all the dual explanations are to be computed as a side effect of the hitting set duality.) The prototype is able to deal with tree ensemble models trained with XGBoost [14]. For that purpose, a simple encoding of tree ensembles into satisfiability modulo theories (SMT) was developed. Concretely, the target formulas are in the theory of linear arithmetic over reals (RIA formulas). (Note that encodings of a decision tree into logic are known [12,50,78]. The final score summations used in tree ensembles can be encoded into RIA formulas.)

Due to the twofold nature of Algorithm 2, it has to deal with (1) implicit hitting set enumeration and (2) entailment queries with SMT. The former part is implemented using the award-winning maximum satisfiability solver RC2 [37] written on top of the PySAT toolkit [36]. SMT solvers are accessed through the

PySMT framework [27], which provides a unified interface to a variety of state-of-the-art SMT solvers. In the experiments, we use Z3 [17] as one of the best performing SMT solvers. The conducted experiment was performed in Debian Linux on an Intel Xeon E5-2630 2.60 GHz processor with 64GByte of memory. Given a dataset, we trained an XGBoost model containing 50 trees per class, each tree having depth 3. (Further increasing the number of trees per class and also increasing the maximum depth of a tree did not result in a significant increase of the models' accuracy on the training and test sets for the considered datasets.) All abductive explanations for every instance of each of the six datasets were exhaustively enumerated using the duality-based approach (Algorithm 2 in Algorithm 4). This resulted in the computation of all contrastive explanations as well).

Evaluation Results. Table 2 shows the results. There are several points to make. First, although it seems computationally expensive to enumerate all explanations for a data instance, it can still be achieved effectively for the medium-sized models trained for all the considered datasets. This may on average require from a few dozen to several hundred of oracle calls per data instance (in some cases, the number of calls gets up to a few thousand). Also observe that enumerating all explanations for an instance takes from a fraction of a second to a couple of seconds on average. These results demonstrate that our approach is practical.

Second, the total number of AXps is typically lower than the total number of their contrastive counterparts. The same holds for the average numbers of abductive and contrastive explanations per data instance. Third and finally, AXps for the studied datasets tend to be larger than contrastive explanations. The latter observations imply that contrastive explanations may be preferred from a user's perspective, as the smaller the explanation is the easier it is to interpret for a human decision maker. (Furthermore, although it is not shown in Table 2, we noticed that in many cases contrastive explanations tend to be of size 1, which makes them ideal to reason about the behaviour of an ML model.) On the other hand, exhaustive enumeration of contrastive explanations can be more time consuming because of their large number.

Summary of Results. We show that CXps enumeration gives us an insightful understanding of a classifier's behaviour. First, even in cases when we cannot enumerate all of CXps to compute AXps by duality, we can still draw some conclusions, e.g. CXps of size one are exactly features that occur in all AXps. Next, we clearly demonstrate the feasibility of the duality-based exhaustive enumeration of both AXps and CXps for a given data instance using a more powerful algorithm that performs enumeration of AXps and CXps.

6 Conclusions

This paper studies local model-based abductive and contrastive explanations. Abductive explanations answer 'Why?' questions, whereas contrastive

explanations answer 'Why Not?' questions. Moreover, the paper relates explanations with the analysis of inconsistent theories, and shows that abductive explanations correspond to minimal unsatisfiable subsets, whereas contrastive explanations can be related with minimal correction subsets. As a consequence of this result, the paper exploits a well-known minimal hitting set relationship between MUSes and MCSes [11,68] to reveal the same relationship between abductive and contrastive explanations. In addition, the paper exploits known results on the analysis of inconsistent theories, to devise algorithms for extracting and enumerating abductive and contrastive explanations.

Acknowledgments. This work is supported by the AI Interdisciplinary Institute ANITI (Artificial and Natural Intelligence Toulouse Institute), funded by the French program "Investing for the Future – PIA3" under Grant agreement no ANR-19-PI3A-0004.

References

1. Achinstein, P.: The Nature of Explanation. Oxford University Press, Oxford (1980)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
3. Alonso, J.M., Castiello, C., Mencar, C.: A bibliometric analysis of the explainable artificial intelligence research field. In: Medina, J., et al. (eds.) IPMU 2018. CCIS, vol. 853, pp. 3–15. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91473-2_1
4. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias (2016). http://tiny.cc/dd7mjz
5. Anjomshoae, S., Najjar, A., Calvaresi, D., Främling, K.: Explainable agents and robots: results from a systematic literature review. In: AAMAS, pp. 1078–1088 (2019)
6. Asher, N., Paul, S., Russell, C.: Adequate and fair explanations. CoRR, abs/2001.07578 (2020)
7. Bacchus, F., Katsirelos, G.: Using minimal correction sets to more efficiently compute minimal unsatisfiable sets. In: Kroening, D., Păsăreanu, C.S. (eds.) CAV 2015. LNCS, vol. 9207, pp. 70–86. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21668-3_5
8. Bailey, J., Stuckey, P.J.: Discovery of minimal unsatisfiable subsets of constraints using hitting set dualization. In: Hermenegildo, M.V., Cabeza, D. (eds.) PADL 2005. LNCS, vol. 3350, pp. 174–186. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-30557-6_14
9. Bendík, J., Černá, I., Beneš, N.: Recursive online enumeration of all minimal unsatisfiable subsets. In: Lahiri, S.K., Wang, C. (eds.) ATVA 2018. LNCS, vol. 11138, pp. 143–159. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01090-4_9
10. Biran, O., Cotton, C.: Explanation and justification in machine learning: a survey. In: IJCAI-17 Workshop on Explainable AI (XAI), vol. 8, p. 1 (2017)
11. Birnbaum, E., Lozinskii, E.L.: Consistent subsets of inconsistent systems: structure and behaviour. J. Exp. Theoret. Artif. Intell. **15**(1), 25–46 (2003)
12. Bonfietti, A., Lombardi, M., Milano, M.: Embedding decision trees and random forests in constraint programming. In: Michel, L. (ed.) CPAIOR 2015. LNCS,

vol. 9075, pp. 74–90. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18008-3_6
13. Bromberger, S.: An approach to explanation. In: Butler, R. (ed.) Analytical Philsophy, pp. 72–105. Oxford University Press, Oxford (1962)
14. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: KDD, pp. 785–794. ACM (2016)
15. Darwiche, A.: Three modern roles for logic in AI. In: PODS, pp. 229–243 (2020)
16. Darwiche, A., Hirth, A.: On the reasons behind decisions. In: ECAI, pp. 712–720 (2020)
17. de Moura, L., Bjørner, N.: Z3: an efficient SMT solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 337–340. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78800-3_24
18. Dhurandhar, A., et al.: Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: NIPS, pp. 590–601 (2018)
19. Dosilovic, F.K., Brcic, M., Hlupic, N.: Explainable artificial intelligence: a survey. In: MIPRO, pp. 210–215 (2018)
20. Dua, D., Graff, C.: UCI machine learning repository (2017)
21. Auditing black-box predictive models (2016). http://tiny.cc/6e7mjz
22. Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and removing disparate impact. In: KDD, pp. 259–268. ACM (2015)
23. Felfernig, A., Schubert, M., Zehentner, C.: An efficient diagnosis algorithm for inconsistent constraint sets. Artif. Intell. Eng. Des. Anal. Manuf. **26**, 53–62 (2012)
24. Friedler, S., Scheidegger, C., Venkatasubramanian, S.: On algorithmic fairness, discrimination and disparate impact (2015)
25. Friedler, S.A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E.P., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: FAT, pp. 329–338. ACM (2019)
26. Frosst, N., Hinton, G.E.: Distilling a neural network into a soft decision tree. In: CEx@AI*IA (2017)
27. Gario, M., Micheli, A.: PySMT: a solver-agnostic library for fast prototyping of SMT-based algorithms. In: SMT Workshop (2015)
28. Google. AI Explainability Whitepaper (2019). http://tiny.cc/tjz2hz
29. Grégoire, É., Izza, Y., Lagniez, J.: Boosting MCSes enumeration. In: IJCAI, pp. 1309–1315 (2018)
30. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 93:1–93:42 (2019)
31. Hoffman, R.R., Klein, G.: Explaining explanation, part 1: theoretical foundations. IEEE Intell. Syst. **32**(3), 68–73 (2017)
32. Hoffman, R.R., Miller, T., Mueller, S.T., Klein, G., Clancey, W.J.: Explaining explanation, part 4: a deep dive on deep nets. IEEE Intell. Syst. **33**(3), 87–95 (2018)
33. Hoffman, R.R., Mueller, S.T., Klein, G.: Explaining explanation, part 2: empirical foundations. IEEE Intell. Syst. **32**(4), 78–86 (2017)
34. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Metrics for explainable AI: challenges and prospects. CoRR, abs/1812.04608 (2018)
35. Ignatiev, A.: Towards trustable explainable AI. In: IJCAI, pp. 5154–5158 (2020)
36. Ignatiev, A., Morgado, A., Marques-Silva, J.: PySAT: a Python toolkit for prototyping with SAT Oracles. In: Beyersdorff, O., Wintersteiger, C.M. (eds.) SAT 2018. LNCS, vol. 10929, pp. 428–437. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94144-8_26

37. Ignatiev, A., Morgado, A., Marques-Silva, J.: RC2: an efficient MaxSAT solver. J. Satisf. Boolean Model. Comput. **11**, 53–64 (2019)
38. Ignatiev, A., Narodytska, N., Marques-Silva, J.: Abduction-based explanations for machine learning models. In: AAAI, pp. 1511–1519 (2019)
39. Ignatiev, A., Narodytska, N., Marques-Silva, J.: On relating explanations and adversarial examples. In: NeurIPS, pp. 15857–15867 (2019)
40. Ignatiev, A., Narodytska, N., Marques-Silva, J.: On validating, repairing and refining heuristic ML explanations. CoRR, abs/1907.02509 (2019)
41. Izza, Y., Ignatiev, A., Marques-Silva, J.: On explaining decision trees. CoRR, abs/2010.11034 (2020)
42. Janota, M., Marques-Silva, J.: On the query complexity of selecting minimal sets for monotone predicates. Artif. Intell. **233**, 73–83 (2016)
43. Jha, S., Sahai, T., Raman, V., Pinto, A., Francis, M.: Explaining AI decisions using efficient methods for learning sparse Boolean formulae. J. Autom. Reasoning **63**(4), 1055–1075 (2019)
44. Junker, U.: QUICKXPLAIN: preferred explanations and relaxations for over-constrained problems. In: AAAI, pp. 167–172 (2004)
45. Klein, G.: Explaining explanation, part 3: the causal landscape. IEEE Intell. Syst. **33**(2), 83–88 (2018)
46. Kohavi, R.: Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In: KDD, pp. 202–207 (1996)
47. Kroening, D., Strichman, O.: Decision Procedures - An Algorithmic Point of View. Texts in Theoretical Computer Science. An EATCS Series, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/s10601-015-9183-0
48. Liffiton, M.H., Previti, A., Malik, A., Silva, J.M.: Fast, flexible MUS enumeration. Constraints **21**(2), 223–250 (2016). https://doi.org/10.1007/s10601-015-9183-0
49. Liffiton, M.H., Sakallah, K.A.: Algorithms for computing minimal unsatisfiable subsets of constraints. J. Autom. Reasoning **40**(1), 1–33 (2008). https://doi.org/10.1007/s10817-007-9084-z
50. Lombardi, M., Milano, M., Bartolini, A.: Empirical decision model learning. Artif. Intell. **244**, 343–367 (2017)
51. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: NIPS, pp. 4765–4774 (2017)
52. Marques-Silva, J., Gerspacher, T., Cooper, M.C., Ignatiev, A., Narodytska, N.: Explaining Naive Bayes and other linear classifiers with polynomial time and delay. In: NeurIPS (2020)
53. Marques-Silva, J., Heras, F., Janota, M., Previti, A., Belov, A.: On computing minimal correction subsets. In: IJCAI, pp. 615–622 (2013)
54. Marques-Silva, J., Janota, M., Belov, A.: Minimal sets over monotone predicates in Boolean formulae. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 592–607. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39799-8_39
55. Marques-Silva, J., Janota, M., Mencía, C.: Minimal sets on propositional formulae. Problems and reductions. Artif. Intell. **252**, 22–50 (2017)
56. Mencía, C., Ignatiev, A., Previti, A., Marques-Silva, J.: MCS extraction with sublinear Oracle queries. In: Creignou, N., Le Berre, D. (eds.) SAT 2016. LNCS, vol. 9710, pp. 342–360. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40970-2_21
57. Mencía, C., Previti, A., Marques-Silva, J.: Literal-based MCS extraction. In: IJCAI, pp. 1973–1979 (2015)

58. Miller, T.: Contrastive explanation: a structural-model approach. CoRR, abs/1811.03163 (2018)
59. Miller, T.: "but why?" Understanding Explainable artificial intelligence. ACM Crossroads **25**(3), 20–25 (2019)
60. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intell. **267**, 1–38 (2019)
61. Mittelstadt, B.D., Russell, C., Wachter, S.: Explaining explanations in AI. In: FAT, pp. 279–288 (2019)
62. Montavon, G., Samek, W., Müller, K.: Methods for interpreting and understanding deep neural networks. Digital Signal Process. **73**, 1–15 (2018)
63. Narodytska, N., Bjørner, N., Marinescu, M.V., Sagiv, M.: Core-guided minimal correction set and core enumeration. In: IJCAI, pp. 1353–1361 (2018)
64. Narodytska, N., Shrotri, A., Meel, K.S., Ignatiev, A., Marques-Silva, J.: Assessing heuristic machine learning explanations with model counting. In: Janota, M., Lynce, I. (eds.) SAT 2019. LNCS, vol. 11628, pp. 267–278. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-24258-9_19
65. Perron, L., Furnon, V.: Or-tools
66. Poole, D., Mackworth, A.K.: Artificial Intelligence - Foundations of Computational Agents. Cambridge University Press, Cambridge (2010)
67. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. In: ICLR (2016)
68. Reiter, R.: A theory of diagnosis from first principles. Artif. Intell. **32**(1), 57–95 (1987)
69. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why should I trust you?": explaining the predictions of any classifier. In: KDD, pp. 1135–1144 (2016)
70. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: high-precision model-agnostic explanations. In: AAAI, pp. 1527–1535 (2018)
71. Samek, W., Montavon, G., Vedaldi, A., Hansen, L.K., Müller, K.-R. (eds.): Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. LNCS (LNAI), vol. 11700. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-28954-6
72. Samek, W., Müller, K.: Towards explainable artificial intelligence. In: Samek, et al. [71], pp. 5–22
73. Schmidt, P., Witte, A.D.: Predicting recidivism in North Carolina, 1978 and 1980. Inter-University Consortium for Political and Social Research (1988)
74. Shih, A., Choi, A., Darwiche, A.: Formal verification of Bayesian network classifiers. In: PGM, pp. 427–438 (2018)
75. Shih, A., Choi, A., Darwiche, A.: A symbolic approach to explaining Bayesian network classifiers. In: IJCAI, pp. 5103–5111 (2018)
76. Shih, A., Choi, A., Darwiche, A.: Compiling Bayesian network classifiers into decision graphs. In: AAAI, pp. 7966–7974 (2019)
77. Tran, S.N., d'Avila Garcez, A.S.: Deep logic networks: inserting and extracting knowledge from deep belief networks. IEEE Trans. Neural Netw. Learn. Syst. **29**(2), 246–258 (2018)
78. Verwer, S., Zhang, Y., Ye, Q.C.: Auction optimization using regression trees and linear models as integer programs. Artif. Intell. **244**, 368–395 (2017)
79. Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., Zhu, J.: Explainable AI: a brief survey on history, research areas, approaches and challenges. In: Tang, J., Kan, M.-Y., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2019. LNCS (LNAI), vol. 11839, pp. 563–574. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32236-6_51