Delivering Inflated Explanations

Yacine Izza[1], Alexey Ignatiev[2], Peter J. Stuckey[2,4], Joao Marques-Silva[3]
[1]CREATE, National University of Singapore, Singapore
[2]Monash University, Melbourne, Australia
[3]IRIT, CNRS, Toulouse, France
[4]OPTIMA ARC Industrial Training and Transformation Centre, Melbourne, Australia
izza@comp.nus.edu.sg, alexey.ignatiev@monash.edu, peter.stuckey@monash.edu, joao.marques-silva@irit.fr

Abstract

In the quest for Explainable Artificial Intelligence (XAI) one of the questions that frequently arises given a decision made by an AI system is, "why was the decision made in this way?" Formal approaches to explainability build a formal model of the AI system and use this to reason about the properties of the system. Given a set of feature values for an instance to be explained, and a resulting decision, a formal *abductive explanation* is a set of features, such that if they take the given values, will always lead to the same decision. This explanation is useful, it shows that only some features were used in making the final decision. But it is narrow, it only shows that if the selected features take their given values the decision is unchanged. It is possible that some features may change values and still lead to the same decision. In this paper we formally define *inflated explanations*, which is a set of features and for each feature a set of values (always including the value of the instance being explained), such that the decision will remain unchanged, for any of the values allowed for any of the features in the (inflated) abductive explanation. Inflated formal explanations are more informative than common abductive explanations since e.g. they allow us to see if the exact value of a feature is important, or it could be any nearby value. Overall they allow us to better understand the role of each feature in the decision. We show that we can compute inflated explanations for not that much greater cost than abductive explanations, and that we can extend duality results for abductive explanations also to inflated explanations.

Introduction

The purpose of eXplainable AI (XAI) is to help human decision makers in understanding the decisions made by AI systems. It is generally accepted that XAI is fundamental to deliver trustworthy AI (Ignatiev 2020; Marques-Silva and Ignatiev 2022). In addition, explainability is also at the core of recent proposals for the verification of Artificial Intelligence (AI) systems (Seshia, Sadigh, and Sastry 2022). Nevertheless, most of the work on XAI offers no formal guarantees of rigor (and so will be referred to as non-formal XAI in this paper). Examples of non-formal XAI include model-agnostic methods (Ribeiro, Singh, and Guestrin 2016; Lundberg and Lee 2017; Ribeiro, Singh, and Guestrin 2018), heuristic learning of saliency maps (and their variants) (Bach et al. 2015; Samek and Müller 2019; Samek et al. 2019, 2021), but also proposals of intrinsic interpretability (Rudin 2019; Molnar 2020; Rudin et al. 2022). In recent years, comprehensive evidence has been gathered that attests to the lack of rigor of these (non-formal) XAI approaches (Ignatiev 2020; Izza, Ignatiev, and Marques-Silva 2022; Yu, Ignatiev, and Stuckey 2023b).

The alternative to non-formal explainability is *formal XAI*. Formal XAI proposes definitions of explanations, and algorithms for their computation, that ensure the rigor of computed explanations (Shih, Choi, and Darwiche 2018; Ignatiev, Narodytska, and Marques-Silva 2019; Marques-Silva and Ignatiev 2022). Despite its promise, formal XAI also exhibits a number of important limitations, which include lack of scalability for some ML models, and the computation of explanations which human decision makers may fail to relate with. This paper targets mechanisms for improving the clarity of computed explanations.

An abductive explanation (Shih, Choi, and Darwiche 2018; Ignatiev, Narodytska, and Marques-Silva 2019) is a subset-minimal set of features which correspond to a rule of the form: if a conjunction of literals is true, then the prediction is the expected one. The literals associated with that rule are of the form $x_i = v_i$, i.e. a feature is tested for a concrete value in its domain. For categorical features with large domains, and more importantly for real-value features, specifying a literal that tests a single value will provide little insight. For example, in the case of an ordinal feature, stating that the height of a patient is 1.825m is less insightful that stating that the height of a patient is between 1.75m and 1.84m. Similarly, in the case of a categorical feature, an explanation that indicates that the vehicle color is one of $\{\text{Red}, \text{Blue}, \text{Green}, \text{Silver}\}$ (e.g. with colors $\{\text{Black}, \text{White}\}$ excluded), is more insightful that stating that the color must be Blue. For an explanations involving several features, the use of more expressive literals will allow a human decision maker to relate the explanation with several different instances.

This paper proposes *inflated formal explanations*. In an inflated formal explanation the literals of the form $x_i = v_i$ are replaced by literals of the form $x_i \in \mathbb{E}_i$, where $\mathbb{E}_i$ is a subset of the feature's domain. Furthermore, each $\mathbb{E}_i$ is maximally large, i.e. no proper superset of $\mathbb{E}_i$ guarantees sufficiency for the prediction. Inflated explanations can be related with previous works (Choi et al. 2020; Izza, Ignatiev, and Marques-Silva 2022; Ji and Darwiche 2023), but also with earlier work on minimum satisfying assignments (Dillig et al. 2012). For example, recent work (Izza, Ignatiev, and Marques-Silva 2022) proposed algorithms for computing explanations in the case of decision trees that report path explanations, where the literals are taken from a concrete tree path; however, no algorithm for inflating explanations is described. In contrast, our contribution is to propose algorithms for transforming abductive explanations into inflated (abductive) explanations, and such that these algorithms are shown to be efficient in practice. Furthermore, we prove that contrastive explanations can also be inflated, and that there exist different minimal hitting set duality (Ignatiev et al. 2020) relationships between inflated explanations. The duality results regarding contrastive explanations attest to the robustness of minimal hitting set duality in explainability. The experimental results illustrate that in practice features can in general be inflated, with some inflated to almost their original domain.

Preliminaries

Classification problems. We consider a classification problem, characterized by a set of features $\mathcal{F} = \{1, \ldots, m\}$, and by a set of classes $\mathcal{K} = \{c_1, \ldots, c_K\}$. Each feature $j \in \mathcal{F}$ is characterized by a domain $\mathbb{D}_j$. As a result, feature space is defined as $\mathbb{F} = \mathbb{D}_1 \times \mathbb{D}_2 \times \ldots \times \mathbb{D}_m$. A specific point in feature space represented by $\mathbf{v} = (v_1, \ldots, v_m)$ denotes an *instance* (or an *example*). Also, we use $\mathbf{x} = (x_1, \ldots, x_m)$ to denote an arbitrary point in feature space. In general, when referring to the value of a feature $j \in \mathcal{F}$, we will use a variable x_j, with x_j taking values from $\mathbb{D}_j$. We consider two types of features $j \in \mathcal{F}$: *categorical features* where $\mathbb{D}_j$ is a finite unordered set, and *ordinal features* where $\mathbb{D}_j$ is a possibly infinite ordered set. For ordinal (real-valued) features j we use range notation $[\lambda(j), \mu(j)]$ to indicate the set of values $\{d \mid d \in \mathbb{D}_j, \lambda(j) \leq d \leq \mu(j)\}$, and $[\lambda(j), \mu(j))$ to indicate the (half-open) set of values $\{d \mid d \in \mathbb{D}_j, \lambda(j) \leq d < \mu(j)\}$. A classifier implements a *total classification function* $\kappa : \mathbb{F} \to \mathcal{K}$. For technical reasons, we also require κ not to be a constant function, i.e. there exists at least two points in feature space with differing predictions.

Given the above, we represent a classifier $\mathcal{M}$ by a tuple $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$. Moreover, and given a concrete instance $(\mathbf{v}, c)$, an explanation problem is represented by a tuple $(\mathcal{M}, (\mathbf{v}, c))$.

Throughout the paper, we consider the following families of classifiers: monotonic classifiers, decision lists and tree ensembles. These families of classifiers are well-known, and have been investigated in the context of logic-based explainability (Marques-Silva et al. 2021; Ignatiev and Marques-Silva 2021; Ignatiev 2020).

Running examples. The monotonic classifier of Example 1 will be used throughout the paper.

Example 1 (Running example – ordinal features)**.** *We consider a monotonic classifier defined on ordinal features, adapted from (Marques-Silva et al. 2021, Example 1). The classifier serves for predicting student grades. It is assumed that the classifier has learned the following formula (after being trained with grades of students from different cohorts):*

$$\begin{aligned} S &= \max\left[0.3 \times Q + 0.6 \times X + 0.1 \times H, R\right] \\ M &= \text{ite}(S \geq 9, A, \text{ite}(S \geq 7, B, \text{ite}(S \geq 5, C, \\ & \text{ite}(S \geq 4, D, \text{ite}(S \geq 2, E, F))))) \end{aligned}$$

The features represent the different components of assessment, namely S, Q, X, H and R denote, respectively, the final score, the marks on the quiz, the exam, the homework, and the mark of an optional research project. Each mark ranges from 0 to 10. (For the optional mark R, the final mark is 0 if the student opts out.) The final score is the largest of the two marks, as shown above. The student's final grade M is defined using an ite *(if-then-else) operator, and ranges from A to F. Features Q, X, H and R are respectively numbered 1, 2, 3 and 4, and so* $\mathcal{F} = \{1, 2, 3, 4\}$. *Each feature takes values from* $[0, 10]$, *i.e.* $\lambda(i) = 0$ *and* $\mu(i) = 10$. *The set of classes is* $\mathcal{K} = \{A, B, C, D, E, F\}$, *with* $F \prec E \prec D \prec C \prec B \prec A$. *Clearly, the complete classifier (that given the different marks computes a final grade) is monotonic. Moreover, and in contrast with (Marques-Silva et al. 2021), we will consider the following point in feature space representing a student* s_1, $(Q, X, H, R) = (5, 10, 5, 8)$, *with a predicted grade of B, i.e.* $\kappa(5, 10, 5, 8) = B$, *given that* $S = 8$. *Moreover, and unless stated otherwise, the order in which features are analyzed throughout the paper will be* $\langle 1, 2, 3, 4 \rangle$.

The decision list of Example 2 will also be used throughout the paper.

Example 2 (Running example – categorical features)**.** *We also consider a decision list with categorical features. The classification problem is to assess risk accident, given two features, age segment (represented by variable A), and car color (represented by variable C). Let* $\mathcal{F} = \{1, 2\}$, $\mathcal{K} = \{0, 1\}$ $\mathbb{D}_1 = \{\text{Adult}, \text{Junior}, \text{Senior}\}$ *(where* Junior *is synonym of* Young Adult*)* $\mathbb{D}_2 = \{\text{Red}, \text{Blue}, \text{Green}, \text{Silver}, \text{Black}, \text{White}\}$. *Let the decision list be,*

IF	$A = \text{Adult}$	THEN	$\kappa(\mathbf{x}) = 0$
ELSE IF	$C = \text{Red}$	THEN	$\kappa(\mathbf{x}) = 1$
ELSE IF	$C = \text{Blue}$	THEN	$\kappa(\mathbf{x}) = 1$
ELSE IF	$C = \text{Green}$	THEN	$\kappa(\mathbf{x}) = 1$
ELSE IF	$C = \text{Black}$	THEN	$\kappa(\mathbf{x}) = 1$
ELSE			$\kappa(\mathbf{x}) = 0$

Moreover, we consider the instance $(\mathbf{v}, c) = ((\text{Junior}, \text{Red}), 1)$, *i.e. a young adult with a red colored car represents a risk of accident.*

Logic-based explainability. Two types of formal explanations have been studied: abductive (Shih, Choi, and Darwiche 2018; Ignatiev, Narodytska, and Marques-Silva 2019) and contrastive (Miller 2019; Ignatiev et al. 2020). Abductive explanations broadly answer a **Why** question, i.e. *Why the prediction?*, whereas contrastive explanations broadly answer a **Why Not** question, i.e. *Why not some other prediction?*.

Given an explanation problem, an abductive explanation (AXp) is a subset-minimal set of features $\mathcal{X} \subseteq \mathcal{F}$ which, if assigned the values dictated by the instance $(\mathbf{v}, c)$, are sufficient for the prediction. This is stated as follows, for a chosen set $\mathcal{X}$:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{i\in\mathcal{X}}(x_i = v_i)\right] \rightarrow (\kappa(\mathbf{x}) = c) \qquad (1)$$

Monotonicity of entailment implies that there exist algorithms for computing a subset-minimal set $\mathcal{X} \subseteq \mathcal{F}$ that are polynomial in the time to decide (1) (Marques-Silva and Ignatiev 2022).

An AXp $\mathcal{X}$ can be interpreted as a logic rule, of the form:

$$\text{IF } \left[\bigwedge\nolimits_{i\in\mathcal{X}}(x_i = v_i)\right] \text{ THEN } [\kappa(\mathbf{x}) = c] \qquad (2)$$

Moreover, and given an explanation problem, a contrastive explanation (CXp) is a subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ which, if the features in $\mathcal{F} \setminus \mathcal{Y}$ are assigned the values dictated by the instance $(\mathbf{v}, c)$, then there is an assignment to the features in $\mathcal{Y}$ that changes the prediction. This is stated as follows, for a chosen set $\mathcal{Y} \subseteq \mathcal{F}$:

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{i\in\mathcal{F}\setminus\mathcal{Y}}(x_i = v_i)\right] \land (\kappa(\mathbf{x}) \neq c) \qquad (3)$$

Similarly to the case of AXp's, monotonicity of entailment ensures that there exist algorithms for computing a subset-minimal set $\mathcal{Y} \subseteq \mathcal{F}$ that are polynomial in the time to decide (3) (Marques-Silva and Ignatiev 2022).

Following standard notation (Marques-Silva and Ignatiev 2022), we use the predicate WAXp (resp. WCXp) to hold true for any set $\mathcal{X} \subseteq \mathcal{F}$ for which (1) (resp. (3)) holds, and the predicate AXp (resp. CXp) to hold true for any subset-minimal (or irreducible) set $\mathcal{X} \subseteq \mathcal{F}$ for which (1) (resp. (3)) holds.

AXp's and CXp's respect a minimal-hitting set (MHS) duality relationship (Ignatiev et al. 2020). Concretely, each AXp is an MHS of the CXp's and each CXp is an MHS of the AXp's. MHS duality is a stepping stone for the enumeration of explanations.

Example 3. *For the monotonic classifier of Example 1, and instance $((5, 10, 5, 8), B)$, we can use existing algorithms (Marques-Silva et al. 2021; Cooper and Marques-Silva 2021, 2023), for computing one AXp. We use the proposed order for analyzing features: $\langle 1, 2, 3, 4\rangle$. we conclude that Q must be included in the AXp, since increasing the value of Q to 10 would change the prediction. In contrast, given the value of R, and because we have fixed Q, then both X and H are dropped from the AXp. Moreover, we conclude that R must be included in the AXp; otherwise, we would be able to change the prediction to A by increasing R. As a result, the computed AXp is $\mathcal{X} = \{1, 4\}$. (For a different order of the features, a different AXp would be obtained.) With this AXp, we associate the following rule:*

$$\text{IF } [Q = 5 \land R = 8] \text{ THEN } [\kappa(\mathbf{x}) = B]$$

Example 4. *For the decision list running example (see Example 2), and given the instance $(\mathbf{v}, c) = ((\text{Junior}, \text{Red}), 1)$, an AXp is $\{1, 2\}$, meaning that,*

$$\text{IF } [A = \text{Junior} \land C = \text{Red}] \text{ THEN } [\kappa(\mathbf{x}) = 1]$$

where $\mathbf{x} = (A, C)$.

Logic-based explainability is covered in a number of recent works. Explainability queries are studied in (Audemard, Koriche, and Marquis 2020; Audemard et al. 2021; Huang et al. 2021; Audemard et al. 2022; Huang, Izza, and Marques-Silva 2023; Huang et al. 2023). Probabilistic explanations are investigated in (Wäldchen et al. 2021; Arenas et al. 2022; Izza et al. 2022; Izza and Marques-Silva 2023; Izza et al. 2023). There exist proposals to account for constraints on the inputs (Gorji and Rubin 2022; Shrotri et al. 2022; Yu et al. 2023b). A distinction between contrastive and counterfactual explanations is studied in (Liu and Lorini 2023). An extension of feature selection-based abductive explanations into feature attribution-based abductive explanations is proposed in (Yu, Ignatiev, and Stuckey 2023b; Biradar et al. 2023; Yu et al. 2023a). Additional recent works include (Malfa et al. 2021; Boumazouza et al. 2021; Liu and Lorini 2022; Darwiche and Hirth 2023; Amgoud 2023; Bassan and Katz 2023; Huang and Marques-Silva 2023b,a; Carbonnel, Cooper, and Marques-Silva 2023; Hurault and Marques-Silva 2023; Yu, Ignatiev, and Stuckey 2023a). In addition, there exist recent surveys summarizing the progress observed in formal XAI (Marques-Silva and Ignatiev 2022; Marques-Silva 2022).

Motivating examples & related work. For the two running examples, let us anticipate what we expect to obtain with inflated explanations.

Example 5 (Explanations with more expressive literals – ordinal features.)**.** *For the monotonic classifier of Example 1, and instance $((5, 10, 5, 7), B)$, one AXp is $\mathcal{X} = \{1, 4\}$. Our goal is to identify more general literals than the equality relational operator. Similarly to recent work (Izza, Ignatiev, and Marques-Silva 2022), we use the set-membership ($\in$) operator. The purpose of the paper is to propose approaches for obtaining more expressive rules using the $\in$ operator. As a result, instead of the rule from Example 3, a more expressive rule would be,*

$$\text{IF } [Q \in [0, 6.6] \land R \in [7, 8.8]] \text{ THEN } [\kappa(\mathbf{x}) = B]$$

(Depending on how the domains are expanded, larger intervals could be obtained. Later in the paper, we explain how the above values were obtained.) Clearly, the modified rule is more informative, about the marks that yield a grade of B, than the rule shown in Example 3.

Example 6 (Explanations with more expressive literals – categorical features.)**.** *For the decision list of Example 2, and instance $((\text{Junior}, \text{Red}), 1)$, the only AXp is $\mathcal{X} = \{1, 2\}$. The purpose of this paper is to identify more general literals. For this example, instead of the rule from Example 4, a more expressive rule would be,*

$$\text{IF } \Big[A \in \{\text{Junior}, \text{Senior}\} \land \\ C \in \{\text{Red}, \text{Blue}, \text{Green}\}\Big] \text{ THEN } [\kappa(\mathbf{x}) = 1]$$

As in the previous example, the modified rule is more informative than the rule shown in Example 4.

The use of generalized explanation literals is formalized in earlier work (Amgoud 2021; Amgoud and Ben-Naim

2022; Amgoud 2023). Initial approaches for computing explanations with more expressive literals include (Choi et al. 2020; Izza, Ignatiev, and Marques-Silva 2022; Ji and Darwiche 2023).

Inflated Abductive Explanations

In order to account for more expressive literals in the definition of abductive explanations, we consider an extended definition of AXp.

Definition of Inflated AXp's

Given an AXp $\mathcal{X} \subseteq \mathcal{F}$, an Inflated abductive explanation (iAXp) is a tuple $(\mathcal{X}, \mathbb{X})$, with $\mathcal{X} \subseteq \mathcal{F}$ is an AXp of the explanation problem $\mathcal{E}$, and $\mathbb{X}$ is a set of pairs $(j, \mathbb{E}_j)$, one for each $j \in \mathcal{X}$, such that the following logic statement holds true,

$$\forall(\mathbf{x} \in \mathbb{F}).\left[\bigwedge\nolimits_{j\in\mathcal{X}}(x_j \in \mathbb{E}_j)\right] \rightarrow (\kappa(\mathbf{x}) = c) \tag{4}$$

where $v_j \in \mathbb{E}_j, \forall j \in \mathcal{X}$, and where $\mathbb{E}_j$ is a maximal set such that (4) holds. (Concretely, for any $j \in \mathcal{X}$, and for any $\mathbb{I}_j \subseteq \mathbb{D}_j \setminus \mathbb{E}_j$, it is the case that (4) does not hold when $\mathbb{E}_j$ is replaced with $\mathbb{E}_j \cup \mathbb{I}_j$.) Clearly, (4) is a stronger statement than (3), provided that $\mathbb{E}_j \supset \{v_j\}$ for each $j \in \mathcal{X}$.

After computing one AXp $\mathcal{X}$, it is the case that either $\mathbb{E}_j = \{v_j\}$ (for a categorical feature) or $\mathbb{E}_j = [v_j, v_j]$ (for an ordinal feature). Our purpose is to find ways of augmenting $\mathbb{E}_j$ maximally[1]. This leads to formulate the following problem.

Problem 1 (Inflate Explanation)**.** *Given a classifier* $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ *and an input instance* $(\mathbf{v}, c)$ *to explain. The output computes a set of pairs* $(j, \mathbb{E}_j)$ *that is an explanation of* $(\mathbf{v}, c)$*, where* j *is a feature and* $\mathbb{E}_j \subsetneq \mathbb{D}_j$ *is admissible subdomain of* j*.*

Example 7. *For the monotonic classifier of Example 1, and the proposed inflated AXp of Example 5, for feature 1, we can conclude that we get* $\mathbb{E}_1 = [3.4, 6.6]$.

Observe that the decision to start from an AXp aims at practical efficiency. Algorithm 1 could be adapted to start from the set of features $\mathcal{F}$ instead of $\mathcal{X}$. Any feature for which the expansion includes its domain is removed from the AXp. The set of inflated features for which the expansion did not yield their domain represents one AXp. We underline that similarly to the computation of explanations in general, the order of features can be very important in enabling the expansion of $\mathbb{E}_j$.

Computation of Inflated Explanations

Algorithm 1 summarizes the basic algorithm for inflating a given AXp. As shown, the algorithm picks some order for the features in the AXp $\mathcal{X}$. The features not included in the AXp $\mathcal{X}$ will not be analyzed, i.e. we are only interested in inflating features that are not already inflated. The precision δ is user-specified and it is only relevant for ordinal features, where it is used to expand the value of the feature such that for any value in the resulting interval, sufficiency of prediction is guaranteed.

Algorithm 1: Computing inflated explanations

```
Input: $\mathcal{E} = (\mathcal{M}, (\mathbf{v}, c))$, AXp $\mathcal{X} \subseteq \mathcal{F}$, Precision $\delta$
function InflateAXp($\mathcal{E}, \mathcal{X}$)
    $\mathbb{X} \leftarrow \emptyset$        ▷ $\mathbb{X}$: Sets composing inflated explanation
    $\iota \leftarrow$ PickSomeOrder($\mathcal{X}$)
    for all $j \in \iota$ do
        if Categorical($j$) then
            $\mathbb{E}_j \leftarrow \{v_j\}$
            $\mathbb{E}_j \leftarrow$ InflateCategorical($j, \mathbb{E}_j, \mathcal{E}, \mathcal{X}$)
        else
            inf($j$) = sup($j$) = $v_j$
            $\mathbb{E}_j \leftarrow$ [inf($j$), sup($j$)]        ▷ Initial $\mathbb{E}_j$
            $\mathbb{E}_j \leftarrow$ InflateOrdinal($j, \mathbb{E}_j, \mathcal{E}, \mathcal{X}, \delta$)
        $\mathbb{X} \leftarrow \mathbb{X} \cup \{(j, \mathbb{E}_j)\}$
    return $(\mathcal{X}, \mathbb{X})$
```

Algorithm 2: Inflate categorical feature

```
Input: Feature $j$, $\mathbb{E}_j$, $\mathcal{E}$, and AXp $\mathcal{X} \subseteq \mathcal{F}$
function InflateCategorical($j, \mathbb{E}_j, \mathcal{E}, \mathcal{X}$)
    $\mathbb{R}_j \leftarrow \mathbb{D}_j \setminus \mathbb{E}_j$        ▷ $\mathbb{R}_j$: what remains of $\mathbb{D}_j$
    $\eta \leftarrow$ PickSomeOrder($\mathbb{R}_j$)
    for all $u_{ji} \in \eta$ do
        $\mathbb{E}_j \leftarrow \mathbb{E}_j \cup \{u_{ji}\}$    ▷ Expand $\mathbb{E}_j$, conditionally
        if (4) does not hold then
            $\mathbb{E}_j \leftarrow \mathbb{E}_j \setminus \{u_{ji}\}$  ▷ Revert expansion of $\mathbb{E}_j$
    return $\mathbb{E}_j$
```

The procedure to inflate the values associated with a feature $i \in \mathcal{X}$ depends on the type of the feature. In the next sections, we detail how categorical and ordinal features can be inflated.

Categorical features. Algorithm 2 summarizes the algorithm for a given categorical feature $j \in \mathcal{X}$. Given an initial $\mathbb{E}_j$, we traverse all the values in the domain of the feature, with the exception of the values in $\mathbb{E}_j$. For each value, we check whether inflating $\mathbb{E}_j$ with that value maintains the sufficiency of prediction, i.e. Equation (4) still holds. If sufficiency of prediction still holds, then the value is kept, and the algorithm moves to another value. Otherwise, the value is removed from $\mathbb{E}_j$, and another value will be considered.

Example 8. *For the decision list of Example 2, and the instance* $(\mathbf{v}, c) = ((\text{Junior}, \text{Red}), 1)$*, from Example 4, we know that (the only) AXp is* $\{1, 2\}$*. Let us pick the order of features* $\iota = (1, 2)$*. For feature 1, the order of values is (for example)* $(\text{Adult}, \text{Senior})$*, and initially* $\mathbb{E}_1 = \{\text{Junior}\}$*. Clearly, we cannot consider the value* Adult *for feature 1, as this would change the prediction. In contrast, the value* Senior *can be added to* $\mathbb{E}_1$*. With respect to feature 2, the order of values is (for example)* $(\text{Blue}, \text{Green}, \text{Silver}, \text{Black}, \text{White})$*, and ini-*

[1] It should be noted that, for the algorithms described later in the paper, the replacement of literals of the form $(x_j = v_j)$ by literals of the form $x_j \in \mathbb{E}_j$ is straightforward in terms of logic encodings, e.g. by using well-known solutions for clausification of logic formulas (Biere et al. 2021).

Algorithm 3: Inflate ordinal feature

```
Input: Feat. $j$, $\mathbb{E}_j$, $\mathcal{E}$, AXp $\mathcal{X} \subseteq \mathcal{F}$, $\delta$
function InflateOrdinal($j, \mathbb{E}_j, \mathcal{E}, \mathcal{X}, \delta$)
    if UnconstrainedSup($j, \mathcal{E}, \mathcal{X}$) then
        $\mathsf{sup}(j) \leftarrow \mu(j)$
        if $\mu(j) = +\infty$ then
            $\mathbb{E}_j \equiv [\mathsf{inf}(j), +\infty)$
    else
        $\mathsf{sup}(j) \leftarrow$ ExpandSup($j, \mathcal{E}, \mathcal{X}, \delta$)
    if UnconstrainedInf($j, \mathcal{E}, \mathcal{X}$) then
        $\mathsf{inf}(j) \leftarrow \lambda(j)$
        if $\lambda(j) = -\infty$ then
            $\mathbb{E}_j \equiv (-\infty, \mathsf{sup}(j)]$
    else
        $\mathsf{inf}(j) \leftarrow$ ExpandInf($j, \mathcal{E}, \mathcal{X}, \delta$)
    return $\mathbb{E}_j$
```

Algorithm 4: Inflating the supremum with linear search

```
Input: Feature $j$, $\mathcal{E}$, AXp $\mathcal{X} \subseteq \mathcal{F}$, $\delta$
function ExpandSup($j, \mathcal{E}, \mathcal{X}, \delta$)
    while true do
        $\mathsf{sup}(j) \leftarrow \mathsf{sup}(j) + \delta$
        if (4) does not hold then
            $\mathsf{sup}(j) \leftarrow \mathsf{sup}(j) - \delta$
            return $\mathsf{sup}(j)$
```

Algorithm 5: Inflating the infimum with linear search

```
Input: Feature $j$, $\mathcal{E}$, AXp $\mathcal{X} \subseteq \mathcal{F}$, $\delta$
function ExpandInf($j, \mathcal{E}, \mathcal{X}, \delta$)
    while true do
        $\mathsf{inf}(j) \leftarrow \mathsf{inf}(j) - \delta$
        if (4) does not hold then
            $\mathsf{inf}(j) \leftarrow \mathsf{inf}(j) + \delta$
            return $\mathsf{inf}(j)$
```

tially $\mathbb{E}_2 = \{\text{Red}\}$. *Clearly, by inspection of the decision list, we can conclude that, for a non-Adult, any of the colors Red, Blue, Green and Black will yield prediction 1. Hence,* $\mathbb{E}_2$ *can be inflated from* $\{\text{Red}\}$ *to* $\{\text{Red, Blue, Green, Black}\}$*, as these values do not change the sufficiency for the prediction. As a result, the inflated explanation denotes the rule,*

$$\text{IF } \Big[A \in \{\text{Junior, Senior}\} \wedge \\ C \in \{\text{Red, Blue, Green, Black}\} \Big] \text{ THEN } [\kappa(\mathbf{x}) = 1]$$

Hence, from the decision list, and given the list, it becomes clear that a driver does not pose a risk of accident if he/she is an Adult, or otherwise drives a car colored Silver or White.

Ordinal features. Algorithm 3 summarizes the algorithm for inflating a given ordinal feature $j \in \mathcal{X}$. With the purpose of assessing the largest and smallest possible values for feature j, we initially check whether the feature can take its upper bound $\mu(j)$ (or can be unbounded when $\mu(j) = +\infty$), and later check whether the feature can take its lower bound. (Evidently, no feature j in $\mathcal{X}$ can be allowed to take any value between its lower and upper bounds. Given that the feature j is included in the AXp, then it cannot be allowed to take *any* value in its domain, since otherwise, it could be dropped from the AXp.) If the feature is unconstrained for either its largest or smallest value, then $\mathbb{E}_j$ is updated accordingly (with the case of $+\infty/-\infty$ being handled differently). Otherwise, we seek to inflate $\mathbb{E}_j$, either by increasing values or by decreasing values. The search for increasing values is illustrated by Algorithm 4, and the search for decreasing values is illustrated by Algorithm 5. For simplicity, the algorithms implement a (simple) linear search, using only the value of precision δ for approximating the solution. If δ is small this is inefficient. A simple improvement consists in using two positive values, β and δ, where (the larger) β is used for a rough approximation of the solution, and (the smaller) δ is then used to refine the coarser approximation obtained with β. Moreover, assuming the upper (or lower) bound is finite, then standard binary search can be used. If the upper (or lower) bound is infinite, and not constrained, then one can use exponential (binary) search[2], with a subsequent binary search step to zoom in on the largest (or smallest) value of the feature's values that guarantee sufficiency for the prediction.

Example 9. *For the example monotonic classifier (see Example 1), with instance* $((5, 10, 5, 8), B)$*, the computed AXp is* $\{1, 2, 4\}$*. Let* $\delta = 0.2$*. We illustrate the execution of Algorithm 3, in the concrete case where we increase the maximum value that Q (i.e. feature 1) can take such that the prediction does not change. Since the classifier is monotonically increasing, we can fix H to 10. The value of X is set to 10 and R is set to 8. Clearly, Q cannot take value 10; otherwise the prediction would become A. By iteratively increasing the largest possible value for Q, with increments of δ (see Algorithm 4), we conclude that Q can increase up to 6.6 while ensuring that the prediction does not change to A. If Q were assigned value 6.8 (the next δ increment), the prediction would change to A. In terms of the smallest value that can be assigned to Q, we assign H to 0. In this case, we conclude that Q can take the (lower bound) value of 0, because* $R = 8$*. Hence, feature Q can take values in the range* $[0, 6.6]$*. Finally, in the case of R, its value cannot get to 9, since otherwise the prediction would become A. Hence, the value of R before considering 9 is 8.8 (see Example 5). In terms of the smallest possible value for R, it is clear that its value cannot be less than 7, as this would serve to change the prediction. Hence, feature R can take values in the range* $[7, 8.8]$*. Given the above, the rule associated with the inflated AXp is:*

$$\text{IF } [Q \in [0, 6.6] \wedge R \in [7, 8.8]] \text{ THEN } [\kappa(\mathbf{x}) = B]$$

Ordinal features and tree-based models. Tree-based machine learning models such as decision trees, random for-

[2] Exponential binary search is a common algorithm for finding a value when the domain is unbounded (Bentley and Yao 1976), which is guaranteed to terminate if one knows that the target value exists, and given some value of precision.

est, and boosted trees allow a simpler treatment of ordinal features j since the trees will only compare to a finite set of feature values $V_j \subset \mathbb{D}_j$, we assume with comparison $x_j \geq d, d \in V_j$. Let $[d_1, \ldots, d_m]$ be the V_j is sorted order. We can construct disjoint intervals, given by $I_1 = [\min(\mathbb{D}_j), d_1)$, $I_2 = [d_1, d_2), \ldots, I_{m+1} = [d_m, \max(\mathbb{D}_j)]$. By construction no two values in any interval can be treated differently by the tree-base model, hence we can use these intervals as a finite categorical representation of the feature j.

Complexity of Inflated Explanations

As can be concluded from the algorithms described in the previous section, sufficiency for prediction is checked using Equation (4), which mimics the oracle call for finding the AXp. Furthermore, the features in inflated AXp's match those in the AXp that serves as the seed for computing the inflated AXp; this decision serves to curb the run time complexity of the algorithms proposed in the paper. Nevertheless, inflated explanations require a number of calls to Equation (4) that grows with the number of values in the features domains (for categorical features), or the computed supremum (or the infimum) divided by δ, for categorical features when linear search is used. (The analysis for (unbounded) binary search is beyond the goals of the paper.)

Although other (more sophiscated) variants can be envisioned (e.g. different variants of globally optimally inflated AXp's), the algorithm proposed in this paper ensures that the run time complexity is only affected by the features domains. It is conjectured that finding inflated AXp's with stronger guarantees of optimality would increase the problem's complexity. This is the subject of future research.

Inflated Contrastive Explanations & Duality

This section investigates the differences that must be accounted for in the case of contrastive explanations.

Plain Inflated CXp's & MHS Duality

In contrast with AXp's, in the case of CXp's expansion takes place in the features *not* in the explanation, i.e. in the features whose value remains fixed.

As with AXp's, we propose a modification to the definition of CXp, as follows:

$$\exists(\mathbf{x} \in \mathbb{F}).\left[\bigwedge\nolimits_{j \in \mathcal{F}\setminus\mathcal{Y}}(x_j \in \mathbb{E}_j)\right] \land (\kappa(\mathbf{x}) \neq c) \tag{5}$$

Minimal hitting set duality between inflated AXp's and inflated CXp's is immediate and follows from the duality result relating AXp's and CXp's (Ignatiev et al. 2020). To motivate the argument, let us define the following sets:

$$\mathbb{A}(\mathcal{E}) = \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\} \tag{6}$$

$$\mathbb{C}(\mathcal{E}) = \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{Y})\} \tag{7}$$

Given the discussion above, the following result follows:

Proposition 1. *$\mathbb{A}(\mathcal{E})$ and $\mathbb{C}(\mathcal{E})$ remain unchanged for inflated explanations.*

Given Proposition 1 and from earlier work (Ignatiev et al. 2020), it follows that inflated AXp's and minimal hitting sets of the set of CXp's and vice-versa.

Nevertheless, and in contrast with (4), (5) is a weaker definition that the original definition of CXp. The next section proposes a stronger definition of CXp.

Generalized CXp's & Extended Duality

Due to the use of sets $\mathbb{E}_j \subset \mathbb{D}_j$ instead of concrete feature-values $v_j \in \mathbb{D}_j$, the straightforward definition of inflated contrastive explanations above provides a weaker explanation, than the uninflated definition (3). Nevertheless, it allows us to establish a simple minimal hitting set duality between inflated AXp's and inflated CXp's by building directly on (Ignatiev et al. 2020). In particular, each inflated CXp is a minimal hitting set of all the inflated AXp's and vice versa, in the sense that given the set of all inflated AXp's (resp. inflated CXp's), an uninflated CXp (resp. uninflated AXp) can be constructed by this duality and then inflated afterwards. Note that this *set-wise* duality requires us to first reconstruct uninflated "versions" of the dual explanations and only then inflate them. We can define a stronger form of inflated contrastive explanation as follows, which will enable us to construct inflated CXp's (resp. inflated AXp's) directly from inflated AXp's (resp. inflated CXp's). Namely, given an instance $(\mathbf{v}, c)$ an inflated CXp is a pair $(\mathcal{Y}, \mathbb{Y})$ s.t $\mathbb{Y}$ is a set of pairs $(j, \mathbb{G}_j)$ for each feature $j \in \mathcal{Y}$, such that the following holds:

$$\exists(\mathbf{x} \in \mathbb{F}). \quad \left[\bigwedge\nolimits_{j \in \mathcal{F}\setminus\mathcal{Y}}(x_j = v_j) \land \bigwedge\nolimits_{j \in \mathcal{Y}}(x_j \in \mathbb{G}_j)\right] \land (\kappa(\mathbf{x}) \neq c) \tag{8}$$

We can assume $v_j \notin \mathbb{G}_j, j \in \mathcal{Y}$ since otherwise we could eliminate j from $\mathcal{Y}$ and have a tighter contrastive explanation. Also, observe that in contrast to (3), Equation (8) considers each of the features $j \in \mathcal{Y}$ to belong to some set $\mathbb{G}_j \subset \mathbb{D}_j$ rather than assuming them to be free.

The algorithms proposed in the previous section can be adapted for inflating CXp's by checking whether (8) holds instead of (4). An immediate observation here is that these algorithms would need to be updated such that given a feature $j \in \mathcal{Y}$, we need to *shrink* the set of allowed values starting from $\mathbb{D}_j$ rather than inflating it starting from $\{v_j\}$, which was the case for inflated AXp's.

In order to facilitate the duality relationship between iAXp's and iCXp's, observe that given an instance $(\mathbf{v}, c)$, an iAXp $(\mathcal{X}, \mathbb{X})$, as defined in (4), can be equivalently reformulated as follows:

$$\forall(\mathbf{x} \in \mathbb{F}). \quad \left[\bigwedge\nolimits_{j \in \mathcal{X}}(x_j \in \mathbb{E}_j) \land \bigwedge\nolimits_{j \in \mathcal{F}\setminus\mathcal{X}} x_j \in \mathbb{D}_j\right] \rightarrow (\kappa(\mathbf{x}) = c) \tag{9}$$

In other words, simply let $\mathbb{E}_j = \mathbb{D}_j$ for all $j \notin \mathcal{X}$. Similarly, given an instance $(\mathbf{v}, c)$, an iCXp $(\mathcal{Y}, \mathbb{Y})$, s.t. $\mathbb{Y}$ is a set of pairs $(j, \mathbb{G}_j)$, can be reformulated such as the following holds:

$$\exists(\mathbf{x} \in \mathbb{F}). \tag{10}$$
$$\left[\bigwedge\nolimits_{j\in\mathcal{F}\setminus\mathcal{Y}}(x_j \in \{v_j\}) \wedge \bigwedge\nolimits_{j\in\mathcal{Y}}(x_j \in \mathbb{G}_j)\right] \wedge (\kappa(\mathbf{x}) \neq c)$$

In other words, $\mathbb{G}_j = \{v_j\}$ for all $j \notin \mathcal{Y}$.

Proposition 2. *Given an explanation problem $\mathcal{E}$, let $\mathbb{A}'(\mathcal{E})$ denote the set of all iAXp $(\mathcal{X}, \mathbb{X})$ while $\mathbb{C}'(\mathcal{E})$ denote the set of all iCXps $(\mathcal{Y}, \mathbb{Y})$. Then each iAXp $(\mathcal{X}, \mathbb{X}) \in \mathbb{A}'(\mathcal{E})$ minimally "hits" each iCXp $(\mathcal{Y}, \mathbb{Y}) \in \mathbb{C}'(\mathcal{E})$ s.t. if feature $j \in \mathcal{F}$ is selected to "hit" iAXp $(\mathcal{X}, \mathbb{X})$ then $\mathbb{G}_j \cap \mathbb{E}_j = \emptyset$, and vice versa.*

Proof. Suppose given an iAXp $(\mathcal{X}, \mathbb{X})$ and an iCXp $(\mathcal{Y}, \mathbb{Y})$, feature $j \in \mathcal{F}$ is selected to hit the CXp such that $\mathbb{G}_j \cap \mathbb{E}_j \neq \emptyset$. Then we have a contradiction. Indeed, we can extract an instance $\mathbf{v}' = (v'_1, \ldots, v'_m)$ s.t. $v'_j \in \mathbb{G}_j \cap \mathbb{E}_j$ satisfying Equation (10), with class $\kappa(\mathbf{v}') \neq c$, which by definition (9) violates the iAXp condition since $v'_j \in \mathbb{G}_j \cap \mathbb{E}_j$. □

Given the above proposition, we can construct iCXp's from iAXp's as follows. Given a complete set $\mathbb{A}'(\mathcal{E})$ of iAXp's, we select one feature $\theta(\mathcal{X}) \in \mathcal{X}$ in the features for explanation $(\mathcal{X}, \mathbb{X})$ and construct an iCXp defined by subset of features $\mathcal{Y} = \{\theta(\mathcal{X}) \mid (\mathcal{X}, \mathbb{X}) \in \mathbb{A}'(\mathcal{E})\}$ and $\mathbb{G}_j = \bigcap_{(\mathcal{X},\mathbb{X})\in\mathbb{A}'(\mathcal{E}),\theta(\mathcal{X})=j}(\mathbb{D}_j \setminus \mathbb{E}_j^{\mathcal{X}})$. Similarly, given a complete set $\mathbb{C}'(\mathcal{E})$ of iCXp's, we can construct an iAXp, by selecting one feature $\phi(\mathcal{Y}) \in \mathcal{Y}$ from each CXp $(\mathcal{Y}, \mathbb{Y})$, and defining $\mathcal{X} = \{\phi(\mathcal{Y}) \mid \mathcal{Y} \in \mathbb{C}'(\mathcal{E})\}$ and $\mathbb{E}_j = \bigcap_{(\mathcal{Y},\mathbb{Y})\in\mathbb{C}'(\mathcal{E}),\phi(\mathcal{Y})=j}(\mathbb{D}_j \setminus \mathbb{G}_j^{\mathcal{Y}})$. Note that the key difficulty here is to organize efficient *minimal* selection of features used to "hit" the given set of explanations (either $\mathbb{A}'(\mathcal{E})$ or $\mathbb{C}'(\mathcal{E})$). In various settings of minimal hitting set duality (Ignatiev et al. 2020), this is typically done by invoking a modern mixed-integer linear programming (MILP) or maximum satisfiability (MaxSAT) solver. Similar ideas can be applied in this case as well.

Experiments

This section presents a summary of empirical assessment of computing inflated abductive and contrastive explanations for the case study of random forests (RFs) trained on some widely studied datasets.

Experimental setup. The experiments are conducted on a MacBook Pro with a Dual-Core Intel Core i5 2.3GHz CPU with 8GByte RAM running macOS Ventura. The reported results do not impose any time or memory limit.

Benchmarks. The assessment is performed on benchmarks of (Izza and Marques-Silva 2021), where we selected 35 RF models trained on well-known tabular datasets (all publicly available and originate from UCI repository (Markelle Kelly 2020) and PMLB (Olson et al. 2017)). The number of trees in each RF model is set to 100, while tree depth varies between 3 and 10. The accuracy of the models collection varies between 61% to 100% (avg. 88.33%). Besides, our formal explainers are set to compute a single AXp and then apply the expansion method, per data instance from the selected set of instances and 200 samples are randomly to be tested for each dataset.

Prototype implementation. A prototype implementation of the outlined algorithms (Algorithms 1 to 5) were developed as a Python script. It builds on RFxpl[3] (Izza and Marques-Silva 2021) and makes heavy use of the latest version of the PySAT toolkits (Ignatiev, Morgado, and Marques-Silva 2018) to generate the CNF formulas of the RF encodings and afterwards instrument incremental SAT oracle calls to compute explanations.

Results. Summary results of computing iAXp's for RFs on the selected datasets is reported in Table 1. As can be observed from the results, and with three exceptions, our method succeeds in expanding all individual set $\mathbb{E}_i$ of features involved in the AXp. Moreover, we observe that for 19 out of 35 datasets, the average increase in sub-domain $\mathbb{E}_i$ varies between 100 and 720, and for 13 over 35 datasets this number varies between 13 to 99. The domain coverage of iAXp's can increase until 10^{40} times the coverage of AXp's and on average 2×10^{39} for all tested datasets. In terms of performance, the results clearly demonstrate that our approach scales to (realistically) large data and large tree ensembles considered in the assessment. It is plain to see that for most datasets the proposed method takes a few seconds on average to deliver an iAXp, thus the average (resp. minimum and maximum) runtime for all datasets is 7.72 seconds (resp. 0.12 and 144.48 seconds). Even though a few outliers were observed in continuous data where the number of splits (intervals) generated by the trees is fairly large, this does not contrast the effectiveness of our technique since that the largest running time that could be registered is less than 3 minutes.

In the final analysis, we further assessed iCXp's (see the results shown in Table 1). In summary, our results for iCXps indicate no observable difference in performance with respect to iAXps. Moreover, we observe that for 7 out of 35 datasets, the average increase in sub-domain $\mathbb{G}_i$ varies between 100 and 289, and for 21 over 35 datasets this number varies between 7 to 98; runtimes are less a second for 33 datasets and the maximum runtime is 2.25 seconds.

To review, our extensive evaluation performed on a large range of real world data and RFs of large sizes, allows us to conclude that our solution is effective in practice to produce more expressive explanations than standard AXp's/CXp's and more importantly in a short time.

Conclusions

One limitation of logic-based abductive or contrastive explanations is that these are based on fairly restricted literals, of the form $x_i = v_i$. This paper formalizes the concept of *inflated explanation*, which applies either in the case of abductive or contrastive explanations. Furthermore, the paper proposes algorithms for the rigorous computation of inflated explanations, and demonstrates the existence of minimal hitting set duality between inflated abductive and inflated con-

[3] Available at https://github.com/izzayacine/RFxpl.

Dataset	(m, K)	AXp		Inflated AXp			CXp		Inflated CXp		
		Len	Time	avg	ratio	Time	Len	Time	avg	ratio	Time
adult	(12 2)	5.6	0.19	7.0	$2e+02$	0.40	1.6	0.19	0.8	$2e+00$	0.18
ann-thyroid	(21 3)	1.7	0.20	15.8	$2e+05$	0.41	1.3	0.20	51.6	$3e+03$	0.26
appendicitis	(7 2)	3.7	0.05	54.2	$4e+05$	0.33	2.5	0.05	45.8	$3e+04$	0.09
banknote	(4 2)	2.2	0.05	156.6	$6e+04$	0.22	1.0	0.05	96.1	$1e+02$	0.11
biodegradation	(41 2)	16.7	0.33	273.1	$6e+23$	6.46	4.2	0.33	73.9	$7e+10$	0.23
ecoli	(7 5)	3.6	0.20	128.4	$2e+06$	1.39	1.1	0.20	63.3	$6e+02$	0.34
german	(21 2)	12.3	0.36	149.0	$2e+06$	4.20	4.4	0.36	14.8	$4e+02$	0.34
glass2	(9 2)	4.6	0.07	92.9	$2e+09$	0.41	1.9	0.07	50.7	$1e+04$	0.11
heart-c	(13 2)	5.5	0.07	89.1	$4e+05$	0.23	1.9	0.07	22.0	$8e+02$	0.08
ionosphere	(34 2)	21.5	0.06	193.5	$1e+29$	0.27	6.0	0.06	25.0	$2e+09$	0.10
iris	(4 3)	2.2	0.11	20.7	$9e+02$	0.15	1.3	0.11	17.1	$2e+02$	0.17
lending	(9 2)	2.2	0.10	1.7	$7e+00$	0.14	1.4	0.10	0.9	$2e+00$	0.07
magic	(10 2)	6.3	0.25	831.7	$8e+17$	6.84	1.8	0.25	289.6	$1e+08$	0.63
mushroom	(22 2)	9.2	0.07	14.5	$2e+04$	0.15	4.1	0.07	7.0	$9e+01$	0.07
new-thyroid	(5 3)	2.9	0.12	60.2	$1e+04$	0.40	1.5	0.12	39.8	$5e+02$	0.23
pendigits	(16 10)	9.8	1.07	281.0	$1e+18$	12.50	1.7	1.07	115.5	$9e+03$	0.93
phoneme	(5 2)	3.0	0.18	582.5	$2e+10$	2.62	1.5	0.18	370.1	$2e+06$	0.62
promoters	(57 2)	25.0	0.05	23.0	$5e+07$	0.12	8.8	0.05	3.3	$2e+01$	0.08
recidivism	(15 2)	7.9	0.48	2.2	$5e+00$	0.53	1.6	0.48	0.8	$2e+00$	0.60
ring	(20 2)	9.6	0.35	720.5	$1e+35$	1.35	2.1	0.35	128.6	$3e+11$	0.31
segmentation	(19 7)	8.1	0.39	59.7	$1e+10$	1.99	3.2	0.39	56.2	$3e+05$	0.60
shuttle	(9 7)	2.3	0.22	19.8	$1e+03$	0.42	2.1	0.22	18.1	$4e+02$	0.36
sonar	(60 2)	35.9	0.11	242.8	$3e+35$	4.10	6.7	0.11	27.3	$2e+06$	0.14
soybean	(35 18)	15.6	2.05	13.1	$2e+05$	3.39	3.2	2.05	4.5	$4e+01$	0.71
spambase	(57 2)	18.9	0.15	211.6	$4e+21$	0.94	4.1	0.15	73.8	$5e+08$	0.22
spectf	(44 2)	19.7	0.10	108.9	$2e+26$	2.11	6.1	0.10	22.9	$3e+09$	0.10
texture	(40 11)	22.8	1.06	338.4	$5e+38$	13.19	3.1	1.06	100.7	$8e+11$	1.04
twonorm	(20 2)	10.4	0.05	99.2	$1e+13$	0.15	3.3	0.05	35.3	$3e+06$	0.09
vowel	(13 11)	7.7	1.28	401.5	$4e+15$	28.05	2.1	1.28	245.0	$2e+07$	2.25
waveform-21	(21 3)	9.6	0.50	373.6	$4e+22$	6.28	2.0	0.50	100.0	$1e+06$	0.57
waveform-40	(40 3)	14.5	0.90	365.0	$5e+30$	8.21	3.5	0.90	98.8	$9e+09$	0.66
wdbc	(30 2)	11.9	0.08	119.8	$3e+20$	0.45	4.1	0.08	49.1	$4e+08$	0.11
wine-recog	(13 3)	4.5	0.15	39.0	$2e+07$	0.35	2.1	0.15	27.8	$7e+02$	0.23
wpbc	(33 2)	19.9	0.80	211.2	$1e+27$	144.48	8.2	0.80	36.7	$2e+08$	0.19

Table 1: Detailed performance evaluation of inflating AXp's and CXp's for RFs. The table shows results for 35 datasets, which contain categorical and ordinal data. Columns 'm' and 'K' report, respectively, the number of features and classes in the dataset. Column 'Len' reports the average explanation length (i.e. average number of features contained in the explanations). Column 'Time' shows the average runtime for extracting an explanation. Column 'avg' reports the average number of values/intervals (for categorical /continuous domain) computed in the expansion of $\mathbb{E}_j$ (resp. $\mathbb{G}_j$) for the iAXp (resp. iCXp) and 'ratio' shows the average ratio between domain coverage of iAXp and AXp (resp. iCXp and CXp).

trastive explanations. The experimental results validate the practical interest of computing inflated explanations.

The paper's results can be extended in a number of ways. First, expanded *probabilistic abductive explanations* (Arenas et al. 2022) can be investigated for the specific case of tree ensemble and neural network models. Second, SAT encoding of random forests in (Izza and Marques-Silva 2021) can be adapted for computing *optimal* (w.r.t cardinality) inflated abductive and contrastive explanations.

Acknowledgments

This work was supported by the AI Interdisciplinary Institute ANITI, funded by the French program "Investing for the Future – PIA3" under Grant agreement no. ANR-19-PI3A-0004, by the H2020-ICT38 project COALA "Cognitive Assisted agile manufacturing for a Labor force supported by trustworthy Artificial intelligence", by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme, and by the Australian Government through the Australian Research Council Industrial Transformation Training Centre in Optimisation Technologies, Integrated Methodologies, and Applications (OPTIMA), Project ID IC200100009.

References

Amgoud, L. 2021. Non-monotonic Explanation Functions. In *ECSQARU*, 19–31.

Amgoud, L. 2023. Explaining black-box classifiers: Properties and functions. *Int. J. Approx. Reason.*, 155: 40–65.

Amgoud, L.; and Ben-Naim, J. 2022. Axiomatic Foundations of Explainability. In *IJCAI*, 636–642.

Arenas, M.; Barceló, P.; Romero, M.; and Subercaseaux, B. 2022. On Computing Probabilistic Explanations for Decision Trees. In *NeurIPS*.

Audemard, G.; Bellart, S.; Bounia, L.; Koriche, F.; Lagniez, J.; and Marquis, P. 2021. On the Computational Intelligibility of Boolean Classifiers. In *KR*, 74–86.

Audemard, G.; Bellart, S.; Bounia, L.; Koriche, F.; Lagniez, J.; and Marquis, P. 2022. On the explanatory power of Boolean decision trees. *Data Knowl. Eng.*, 142: 102088.

Audemard, G.; Koriche, F.; and Marquis, P. 2020. On Tractable XAI Queries based on Compiled Representations. In *KR*, 838–849.

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7): e0130140.

Bassan, S.; and Katz, G. 2023. Towards Formal XAI: Formally Approximate Minimal Explanations of Neural Networks. In *TACAS*, 187–207.

Bentley, J. L.; and Yao, A. C. 1976. An Almost Optimal Algorithm for Unbounded Searching. *Inf. Process. Lett.*, 5(3): 82–87.

Biere, A.; Heule, M.; van Maaren, H.; and Walsh, T., eds. 2021. *Handbook of Satisfiability*. IOS Press. ISBN 978-1-64368-160-3.

Biradar, G.; Izza, Y.; Lobo, E.; Viswanathan, V.; and Zick, Y. 2023. Axiomatic Aggregations of Abductive Explanations. *CoRR*, abs/2310.03131.

Boumazouza, R.; Alili, F. C.; Mazure, B.; and Tabia, K. 2021. ASTERYX: A model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations. In *CIKM*, 120–129.

Carbonnel, C.; Cooper, M. C.; and Marques-Silva, J. 2023. Tractable Explaining of Multivariate Decision Trees. In *KR*, 127–135.

Choi, A.; Shih, A.; Goyanka, A.; and Darwiche, A. 2020. On Symbolically Encoding the Behavior of Random Forests. *CoRR*, abs/2007.01493.

Cooper, M. C.; and Marques-Silva, J. 2021. On the Tractability of Explaining Decisions of Classifiers. In *CP*, 21:1–21:18.

Cooper, M. C.; and Marques-Silva, J. 2023. Tractability of explaining classifier decisions. *Artif. Intell.*, 316: 103841.

Darwiche, A.; and Hirth, A. 2023. On the (Complete) Reasons Behind Decisions. *J. Log. Lang. Inf.*, 32(1): 63–88.

Dillig, I.; Dillig, T.; McMillan, K. L.; and Aiken, A. 2012. Minimum Satisfying Assignments for SMT. In *CAV*, 394–409.

Gorji, N.; and Rubin, S. 2022. Sufficient Reasons for Classifier Decisions in the Presence of Domain Constraints. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*, 5660–5667. AAAI Press.

Huang, X.; Cooper, M. C.; Morgado, A.; Planes, J.; and Marques-Silva, J. 2023. Feature Necessity & Relevancy in ML Classifier Explanations. In *TACAS*, 167–186.

Huang, X.; Izza, Y.; Ignatiev, A.; and Marques-Silva, J. 2021. On Efficiently Explaining Graph-Based Classifiers. In *KR*, 356–367.

Huang, X.; Izza, Y.; and Marques-Silva, J. 2023. Solving Explainability Queries with Quantification: The Case of Feature Relevancy. In *AAAI*, 3996–4006.

Huang, X.; and Marques-Silva, J. 2023a. From Decision Trees to Explained Decision Sets. In *ECAI*, 1100–1108.

Huang, X.; and Marques-Silva, J. 2023b. From Robustness to Explainability and Back Again. *CoRR*, abs/2306.03048.

Hurault, A.; and Marques-Silva, J. 2023. Certified Logic-Based Explainable AI - The Case of Monotonic Classifiers. In *TAP*, 51–67.

Ignatiev, A. 2020. Towards Trustable Explainable AI. In *IJCAI*, 5154–5158.

Ignatiev, A.; and Marques-Silva, J. 2021. SAT-Based Rigorous Explanations for Decision Lists. In *SAT*, 251–269.

Ignatiev, A.; Morgado, A.; and Marques-Silva, J. 2018. PySAT: A Python Toolkit for Prototyping with SAT Oracles. In *SAT*, 428–437.

Ignatiev, A.; Narodytska, N.; Asher, N.; and Marques-Silva, J. 2020. From Contrastive to Abductive Explanations and Back Again. In *AIxIA*, 335–355.

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019. Abduction-Based Explanations for Machine Learning Models. In *AAAI*, 1511–1519.

Izza, Y.; Huang, X.; Ignatiev, A.; Narodytska, N.; Cooper, M. C.; and Marques-Silva, J. 2023. On computing probabilistic abductive explanations. *Int. J. Approx. Reason.*, 159: 108939.

Izza, Y.; Ignatiev, A.; and Marques-Silva, J. 2022. On Tackling Explanation Redundancy in Decision Trees. *J. Artif. Intell. Res.*, 75: 261–321.

Izza, Y.; Ignatiev, A.; Narodytska, N.; Cooper, M. C.; and Marques-Silva, J. 2022. Provably Precise, Succinct and Efficient Explanations for Decision Trees. *CoRR*, abs/2205.09569.

Izza, Y.; and Marques-Silva, J. 2021. On Explaining Random Forests with SAT. In *IJCAI*, 2584–2591.

Izza, Y.; and Marques-Silva, J. 2023. On Computing Relevant Features for Explaining NBCs. In *ENIGMA@KR*, 75–86.

Ji, C.; and Darwiche, A. 2023. A New Class of Explanations for Classifiers with Non-binary Features. In *JELIA*, 106–122.

Liu, X.; and Lorini, E. 2022. A Logic of "Black Box" Classifier Systems. In *WoLLIC*, 158–174.

Liu, X.; and Lorini, E. 2023. A unified logical framework for explanations in classifier systems. *J. Log. Comput.*, 33(2): 485–515.

Lundberg, S. M.; and Lee, S. 2017. A Unified Approach to Interpreting Model Predictions. In *NeurIPS*, 4765–4774.

Malfa, E. L.; Michelmore, R.; Zbrzezny, A. M.; Paoletti, N.; and Kwiatkowska, M. 2021. On Guaranteed Optimal Robust Explanations for NLP Models. In *IJCAI*, 2658–2665.

Markelle Kelly, K. N., Rachel Longjohn. 2020. The UCI Machine Learning Repository. https://archive.ics.uci.edu.

Marques-Silva, J. 2022. Logic-Based Explainability in Machine Learning. In *Reasoning Web*, 24–104.

Marques-Silva, J.; Gerspacher, T.; Cooper, M. C.; Ignatiev, A.; and Narodytska, N. 2021. Explanations for Monotonic Classifiers. In *ICML*, 7469–7479.

Marques-Silva, J.; and Ignatiev, A. 2022. Delivering Trustworthy AI through Formal XAI. In *AAAI*, 12342–12350.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267: 1–38.

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

Olson, R. S.; La Cava, W.; Orzechowski, P.; Urbanowicz, R. J.; and Moore, J. H. 2017. PMLB: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(1): 36.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *KDD*, 1135–1144.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *AAAI*, 1527–1535.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5): 206–215.

Rudin, C.; Chen, C.; Chen, Z.; Huang, H.; Semenova, L.; and Zhong, C. 2022. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16: 1–85.

Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C. J.; and Müller, K. 2021. Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. *Proc. IEEE*, 109(3): 247–278.

Samek, W.; Montavon, G.; Vedaldi, A.; Hansen, L. K.; and Müller, K., eds. 2019. *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer. ISBN 978-3-030-28953-9.

Samek, W.; and Müller, K. 2019. Towards Explainable Artificial Intelligence. In (Samek et al. 2019), 5–22.

Seshia, S. A.; Sadigh, D.; and Sastry, S. S. 2022. Toward verified artificial intelligence. *Commun. ACM*, 65(7): 46–55.

Shih, A.; Choi, A.; and Darwiche, A. 2018. A Symbolic Approach to Explaining Bayesian Network Classifiers. In *IJCAI*, 5103–5111.

Shrotri, A. A.; Narodytska, N.; Ignatiev, A.; Meel, K. S.; Marques-Silva, J.; and Vardi, M. Y. 2022. Constraint-Driven Explanations for Black-Box ML Models. In *AAAI*, 8304–8314.

Wäldchen, S.; MacDonald, J.; Hauch, S.; and Kutyniok, G. 2021. The Computational Complexity of Understanding Binary Classifier Decisions. *J. Artif. Intell. Res.*, 70: 351–387.

Yu, J.; Farr, G.; Ignatiev, A.; and Stuckey, P. J. 2023a. Anytime Approximate Formal Feature Attribution. *CoRR*, abs/2312.06973.

Yu, J.; Ignatiev, A.; and Stuckey, P. J. 2023a. From Formal Boosted Tree Explanations to Interpretable Rule Sets. In *CP*, 38:1–38:21.

Yu, J.; Ignatiev, A.; and Stuckey, P. J. 2023b. On Formal Feature Attribution and Its Approximation. *CoRR*, abs/2307.03380.

Yu, J.; Ignatiev, A.; Stuckey, P. J.; Narodytska, N.; and Marques-Silva, J. 2023b. Eliminating The Impossible, Whatever Remains Must Be True: On Extracting and Applying Background Knowledge In The Context Of Formal Explanations. In *AAAI*.