Explaining Black-box Classification Models with Arguments

Leila Amgoud
IRIT–CNRS
Toulouse, France
amgoud@irit.fr

***Abstract*—Two approaches for explaining black-box classification models have been studied: a *global* approach which aims at stressing when classes are predicted independently of instances, and a *local* approach which looks for justifying individual predictions. Besides, different types of local explanations have been studied in the recent literature, however their links to global explanations remain unclear.**

The present paper proposes a unified setting for global explanations and local ones. It is based on dual concepts that provide global explanations: arguments in favour of predictions and arguments against predictions. The former justify why a class is suggested by a black-box classifier and the latter state why a class is not. We investigate the properties of both types of arguments, and provide ways for generating arguments pro a class from arguments con the class and vice versa. Finally, we define various notions of local explanations from the literature by arguments pros/con, characterizing formally their relationships and differences, and also their relations with global explanations.

***Index Terms*—Classification, Explainability, Arguments.**

I. Introduction

Recent progress in data-driven AI has been largely due to machine learning and in particular deep learning models. However, the predictions of these black-box models resist analysis due to their inherent non-linear behaviour and their vast amount of interacting parameters. This opacity impedes the relevance of those models from a theoretical point of view, since their properties are difficult to investigate, and from a practical point of view, as many applications, such as healthcare or embedded systems need guarantees to be deployed, and others, e.g in the legal or financial domain require transparency to be accepted.

Explaining the functionality of complex classification systems and their rationale thus becomes a vital need. These issues have generated a lot of effort, see [1]–[3] for recent surveys on explanations of black-box machine learning models. Most approaches consider as input a black-box model, and provide explanations of its predictions on given instances (local approach), or as a whole (global approach). They can be divided into two families: the first family opens somehow the black-box model to provide insight into the internal decision-making process, e.g. [4], [5]. The second family provides explanations without opening the black-box. They focus mainly on key factors that caused predictions, eg. [4], [6]–[10]. Several notions have been defined within that perspective. The most prominent ones are *prime implicants* [11], called also abductive explanations in [12], and *counterfactuals* [4], called *contrastive* explanations in [7], [8]. These notions have been proposed for explaining individual predictions. Their links with global explanations remain unclear.

This paper bridges this gap by providing a formal and unifying framework in which global/local explanations are defined. The framework is based on two dual concepts, which are seen as global explanations of a classifier: arguments in favour of (or pro) predictions and arguments against (or con) predictions. The former justifies why a class is suggested by a black-box classifier and the latter states why a class is not proposed. We investigate the properties of both types of arguments, and provide ways for generating arguments pro a class from arguments con the class and vice versa. Finally, we define various notions of local explanations from the literature by arguments pros/con, characterizing formally their relationships and differences, and also their relations with global explanations.

The paper is structured as follows: It starts by presenting the background on classification. Then, it introduces two types of arguments and investigates their properties and links. Then, it defines formally existing notions of local explanations from arguments. The two last sections are respectively devoted to related work and concluding remarks.

II. Classification Problem

Let $\mathcal{F} = \{f_1, \ldots, f_n\}$ be a finite and non-empty set of *features* (called also *attributes*) that take respectively their values from *finite* domains $\mathcal{D}_1, \ldots, \mathcal{D}_n$. Let $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$[1]. For every feature f and every possible value v of f, the pair (f, v) is called *literal feature*, or *literal* for short. The two functions `Feat` and `Val` return respectively the feature f and the value v of the literal (f, v). Let $\mathcal{U}$ denote the universe of all possible literal features. The set $\mathcal{X}$ contains all possible $n-$tuples of literal features or sets of the form $\{(f_1, v_{1i}), \ldots, (f_n, v_{nl})\}$, i.e., $\mathcal{X}$ contains all the possible *instantiations* of the n features of $\mathcal{F}$. $\mathcal{X}$ and its elements are called respectively *feature space* and *instances*. Note that $\mathcal{X}$ is finite since $\mathcal{F}$ and the n domains in $\mathcal{D}$ are finite. Let $\mathcal{C} = \{c_1, \ldots, c_m\}$, with $m > 1$, be a finite and non-empty set of possible distinct *classes*.

[1]We focus here on the important case of discrete features, as is the case in important applications (e.g. image or natural language processing). There are well-known ways of discretizing continuous attributes, which sometimes gives better results on some learning algorithms, see [13].

$\mathcal{X}$	Sky	Temp.	Humidity	Wind	Jogging
x_1	Sunny	Hot	High	Low	No
x_2	Sunny	Hot	High	High	No
x_3	Cloudy	Hot	High	Low	Yes
x_4	Rainy	Mild	High	Low	Yes
x_5	Rainy	Cool	Medium	Low	Yes
x_6	Rainy	Cool	Medium	High	No
x_7	Cloudy	Cool	Medium	High	Yes
x_8	Sunny	Mild	High	Low	No
x_9	Sunny	Cool	Medium	Low	Yes
x_{10}	Rainy	Mild	Medium	Low	Yes
x_{11}	Sunny	Mild	Medium	High	Yes
x_{12}	Cloudy	Mild	High	High	Yes
x_{13}	Cloudy	Hot	Medium	Low	Yes
x_{14}	Rainy	Mild	High	High	No

Definition 1 (Theory): A *theory* is a tuple $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$.

A classification model is a function that assigns to every instance $x \in \mathcal{X}$ of a theory $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$ a single prediction, which is a class from the set $\mathcal{C}$.

Definition 2 (Classification Model): Let $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$ be a theory. A *classification model* is a function $\mathtt{f} : \mathcal{X} \to \mathcal{C}$.

Let us illustrate the above sets with a classical example.

Example 1: The attributes are sky (S), temperature (T), humidity (H) and wind (W), thus $\mathcal{F} = \{S, T, H, W\}$. They respectively take their values from the domains $\{\mathtt{Sunny}, \mathtt{Rainy}, \mathtt{Cloudy}\}$, $\{\mathtt{Hot}, \mathtt{Mild}, \mathtt{Cool}\}$, $\{\mathtt{Medium}, \mathtt{High}\}$ and $\{\mathtt{Low}, \mathtt{High}\}$. The concept to learn is whether to go jogging which is binary, $\mathcal{C} = \{\mathtt{Yes}, \mathtt{No}\}$. Example of instances are given in Table 1.

A set of literal features is consistent if it does not contain two literals having the same feature but distinct values.

Definition 3 (Consistency): A set $H \subseteq \mathcal{U}$ is *consistent* iff $\nexists (f, v), (f', v') \in H$ such that $f = f'$ and $c \neq v'$. Otherwise, H is said to be *inconsistent*.

Example 1 (Cont.) {(Wind, `Low`), (Humidity, `Medium`)} is consistent while {(Wind, `Low`), (Wind, `High`), (Humidity, `Medium`)} is inconsistent.

Note that instances of $\mathcal{X}$ are all consistent. Furthermore, any consistent set of literals is included in at least one instance.

Property 1: Let $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$ be a theory.

- For any $x \in \mathcal{X}$, x is consistent.
- For any $H \subseteq \mathcal{U}$ s.t. H is consistent, the following hold:
  - $\forall H' \subset H$, H' is consistent.
  - $\exists x \in \mathcal{X}$ s.t. $H \subseteq x$.

III. Arguments Pro and Con Classifications

This section aims at understanding how an arbitrary but fixed classifier `f` assigns classes to instances of a (arbitrary but fixed) theory $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$. The idea is to understand the general behaviour of `f` independently of instances. For that purpose, we assume the availability of an oracle of that model that can be queried on any instance. We are interested by the question: what is an argument in favour of labelling an instance or a set of instances with a class c? In what follows, we consider an argument as a set of literal features that are minimally sufficient for labelling an instance with c. In other words, it is the smallest set of literal features that always lead to the class c.

Definition 4 (Argument Pro): An *argument pro* a class $c \in \mathcal{C}$ is a pair $\langle H, c \rangle$ s.t.

- $H \subseteq \mathcal{U}$
- H is consistent
- $\forall x \in \mathcal{X}$ s.t. $H \subseteq x$, $\mathtt{f}(x) = c$
- $\nexists H' \subset H$ such that H' satisfies the third condition.

H is called *support*. Let $\mathtt{Pros}(c)$ denote the set of all arguments pro c in theory $\mathcal{T}$, and $\mathtt{arg}^+(\mathcal{T}) = \bigcup_{c \in \mathcal{C}} \mathtt{Pros}(c)$.

The consistency condition is useful for discarding irrelevant arguments of the form $\langle \{(f_1, v_1), (f_1, v_2)\}, c \rangle$.

Example 1 (Cont.) Assume that Table 1 contains the only "reasonable" instances. The class `No` is supported by two arguments a_1 and a_2 while `Yes` is supported by b_1, b_2, b_3.

- $a_1 = \langle \{(\mathtt{Sky}, \mathtt{Sunny}), (\mathtt{Humidity}, \mathtt{High})\}, \mathtt{No} \rangle$,
- $a_2 = \langle \{(\mathtt{Sky}, \mathtt{Rainy}), (\mathtt{Wind}, \mathtt{High})\}, \mathtt{No} \rangle$.
- $b_1 = \langle \{(\mathtt{Sky}, \mathtt{Cloudy})\}, \mathtt{Yes} \rangle$,
- $b_2 = \langle \{(\mathtt{Sky}, \mathtt{Sunny}), (\mathtt{Humidity}, \mathtt{Medium})\}, \mathtt{Yes} \rangle$.
- $b_3 = \langle \{(\mathtt{Sky}, \mathtt{Rainy}), (\mathtt{Wind}, \mathtt{Low})\}, \mathtt{Yes} \rangle$.

A class may have zero, one, or several arguments pro. The first case holds when the class is not assigned by the ML model `f` to any instance. When the same class is assigned to all instances of $\mathcal{X}$, then the set of arguments would contain a single argument, which is in favour of the class and its support is the empty set. Furthermore, from a theory $\mathcal{T}$, it is possible to generate arguments in favour of any class provided that the latter is ascribed to at least one instance. Finally, every argument refers to at least one instance of $\mathcal{X}$. Note that from the same instance, it is possible to generate more than one argument in favour of a class.

Proposition 1: Let $c \in \mathcal{C}$.

- $(\mathtt{arg}^+(\mathcal{T}) = \{\langle \emptyset, c \rangle\}) \iff (\forall x \in \mathcal{X}, \mathtt{f}(x) = c)$
- If $\exists x \in \mathcal{X}$ s.t. $\mathtt{f}(x) = c$, then $\exists \langle H, c \rangle \in \mathtt{Pros}(c)$. Furthermore, $H \subseteq x$.
- If $\exists \langle H, c \rangle \in \mathtt{Pros}(c)$, then $\exists x \in \mathcal{X}$ s.t. $\mathtt{f}(x) = c$.
- $\mathtt{Pros}(c) = \emptyset$ iff $\forall x \in \mathcal{X}$, $\mathtt{f}(x) \neq c$.

The following result shows that the supports of any pair of arguments pro distinct classes are inconsistent.

Proposition 2: Let $c_i, c_j \in \mathcal{C}$ with $c_i \neq c_j$. For all $\langle H, c_i \rangle, \langle H', c_j \rangle \in \mathtt{arg}^+(\mathcal{T})$, the set $H \cup H'$ is inconsistent.

We show next that the arguments that can be generated from a theory define a partition of the set $\mathcal{X}$ of instances.

Proposition 3: Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ and $i \in \{1, \ldots, m\}$,

$$\mathcal{X}_i = \{x \in \mathcal{X} \mid \exists \langle H, c_i \rangle \in \mathtt{arg}^+(\mathcal{T}) \text{ and } H \subseteq x\}.$$

The following properties hold:

- For all $i, j \in \{1, \ldots, m\}$ such that $i \neq j$, $\mathcal{X}_i \cap \mathcal{X}_j = \emptyset$.
- $\mathcal{X} = \mathcal{X}_1 \cup \ldots \cup \mathcal{X}_m$.

We now introduce the notion of argument *against* or *con* a class, say c. It is a minimal set of literals that is sufficient for not assigning the class c to any instance. In other terms,

it is the minimal amount of information that is sufficient for labelling instances with any other class than c.

Notation: For $c \in \mathcal{C}$, $\overline{c}$ denotes that c is not recommended.

Definition 5 (Argument Con): Let $c \in \mathcal{C}$. An *argument con* c is a pair $\langle H, \overline{c} \rangle$ s.t.

- $H \subseteq \mathcal{U}$
- H is consistent
- $\forall x \in \mathcal{X}$ s.t. $H \subseteq x$, $\mathtt{f}(x) \neq c$
- $\nexists H' \subset H$ such that H' satisfies the third condition.

Let $\mathtt{Cons}(c)$ be the set of all arguments con c and $\mathtt{arg}^{-}(\mathcal{T}) = \bigcup_{c \in \mathcal{C}} \mathtt{Cons}(c)$.

It is easy to show that when the concept to learn is binary, then the arguments pro one class are con the other.

Proposition 4: If $\mathcal{C} = \{c_1, c_2\}$, then $\mathtt{Pros}(c_1) = \{\langle H, c_1 \rangle \mid \langle H, \overline{c_2} \rangle \in \mathtt{Cons}(c_2)\}$ and $\mathtt{Cons}(c_1) = \{\langle H, \overline{c_1} \rangle \mid \langle H, c_2 \rangle \in \mathtt{Pros}(c_2)\}$.

Example 1 (Cont.) Since the concept to learn is binary, then $\mathtt{Cons}(\mathtt{Yes}) = \{\langle \{(\mathtt{Sky}, \mathtt{Sunny}), (\mathtt{Humidity}, \mathtt{High})\}, \overline{\mathtt{Yes}} \rangle, \langle \{(\mathtt{Sky}, \mathtt{Rainy}), (\mathtt{Wind}, \mathtt{High})\}, \overline{\mathtt{Yes}} \rangle\}$ and $\langle \{(\mathtt{Sky}, \mathtt{Cloudy})\}, \overline{\mathtt{No}} \rangle \in \mathtt{Cons}(\mathtt{No})$.

In case of non-binary concepts, an argument that is against a given class does not necessarily support another class. Let us consider the following abstract example.

Example 2: Let $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C} \rangle$ be such that $\mathcal{F} = \{f_1, f_2\}$, $\mathcal{D}_1 = \mathcal{D}_2 = \{0, 1\}$, and $\mathcal{C} = \{c_1, c_2, c_3\}$. Assume the following assignments of classes to instances by a classifier $\mathtt{f}$.

$\mathcal{X}$	f_1	f_2	c
x_1	0	0	c_1
x_2	0	1	c_2
x_3	1	0	c_3
x_4	1	1	c_3

The argument $\langle \{(f_1, 0)\}, \overline{c_3} \rangle$ is against c_3, however the set $\{(f_1, 0)\}$ is not sufficient for supporting any other class.

Naturally, the support of every argument against a class is inconsistent with the support of any argument pro that class.

Proposition 5: Let $c \in \mathcal{C}$. For all $\langle H, c \rangle \in \mathtt{Pros}(c)$, $\langle H', \overline{c} \rangle \in \mathtt{Cons}(c)$, the set $H \cup H'$ is inconsistent.

The following results show the relationship between an argument against a class and those supporting other classes.

Proposition 6: Let $c \in \mathcal{C}$.

- $\langle \emptyset, \overline{c} \rangle \in \mathtt{Cons}(c)$ iff $\forall x \in \mathcal{X}, \mathtt{f}(x) \neq c$.
- If $\exists \langle H, \overline{c} \rangle \in \mathtt{Cons}(c)$, then $\exists x \in \mathcal{X}$ s.t. $\mathtt{f}(x) \neq c$. Furthermore, $H \subseteq x$.
- If $\exists x \in \mathcal{X}$ s.t. $\mathtt{f}(x) \neq c$, then $\exists \langle H, \overline{c} \rangle \in \mathtt{Cons}(c)$ s.t. $H \subseteq x$.
- If $\langle H, c \rangle \in \mathtt{Pros}(c)$, then $\forall c' \in \mathcal{C} \setminus \{c\}$, $\exists \langle H', \overline{c'} \rangle \in \mathtt{Cons}(c')$ s.t. $H' \subseteq H$.

While a class that is not assigned to any instance has no pros, we show that it has a single argument con whose support is the empty set.

Proposition 7: Let $c \in \mathcal{C}$.

- $(\mathtt{Pros}(c) = \emptyset) \iff (\mathtt{Cons}(c) = \{\langle \emptyset, \overline{c} \rangle\})$
- $(\mathtt{Cons}(c) = \emptyset) \iff (\mathtt{Pros}(c) = \{\langle \emptyset, c \rangle\})$

We show next that the two notions of pros and cons are *dual*. We start by the following straightforward property.

Proposition 8: Let $c \in \mathcal{C}$. It holds that $\mathcal{X} = \mathcal{Y} \cup \mathcal{Z}$ where

$$\mathcal{Y} = \{x \in \mathcal{X} \mid \exists \langle H, \overline{c} \rangle \in \mathtt{Cons}(c) \text{ and } H \subseteq x\},$$

$$\mathcal{Z} = \{x \in \mathcal{X} \mid \exists \langle H, c \rangle \in \mathtt{Pros}(c) \text{ and } H \subseteq x\}.$$

Arguments against a class can be generated from arguments pro the class and vice versa. Let us define the set of all minimal and consistent subsets of literals that are inconsistent with any argument against a given class.

*Definition 6 (*Supp*):* Let $c \in \mathcal{C}$. We define $\mathtt{Supp}(c) = \{H_1, \ldots, H_k\}$ such that for every $i = 1, \ldots, k$,

- $H_i \subseteq \mathcal{U}$
- H_i is consistent
- $\forall \langle H, \overline{c} \rangle \in \mathtt{Cons}(c)$, $H \cup H_i$ is inconsistent
- $\nexists H' \subset H_i$ s.t. H' satisfies the third condition.

The following result shows that every element of the set $\mathtt{Supp}(c)$ yields an argument pro the class c. In other words, we show how to generate arguments pro a class from its cons.

Theorem 1: Let $c \in \mathcal{C}$. $\mathtt{Pros}(c) = \{\langle H, c \rangle \mid H \in \mathtt{Supp}(c)\}$.

In Example 1, the arguments b_1, b_2, b_3 can be generated automatically from the two arguments a_1, a_2 and vice versa.

Let us now introduce the set of all minimal consistent subsets of literals that are inconsistent with any argument pro a class in a given theory.

*Definition 7 (*Att*):* Let $c \in \mathcal{C}$. We define $\mathtt{Att}(c) = \{H_1, \ldots, H_k\}$ such that for every $i = 1, \ldots, k$,

- $H_i \subseteq \mathcal{U}$
- H_i is consistent
- $\forall \langle H, c \rangle \in \mathtt{Pros}(c)$, $H \cup H_i$ is inconsistent
- $\nexists H' \subset H_i$ s.t. H' satisfies the third condition.

The following result shows that every element of the set $\mathtt{Att}(c)$ yields an argument against the class c. Hence, we show how to generate arguments against a class from its pros.

Theorem 2: Let $c \in \mathcal{C}$. $\mathtt{Cons}(c) = \{\langle H, \overline{c} \rangle \mid H \in \mathtt{Att}(c)\}$.

IV. Global vs. local explanations

This section investigates the links between global explanations provided by pros/cons and different types of local explanations, which are input-dependent. We show that explanations, whatever their type, are generated from arguments pro/con classes.

Abductive explanations answer the question: "why $\mathtt{f}(x) = c$?", i.e., why does the outcome c hold for x? The answer consists in highlighting *factors that caused the given class*. In [11], [14]–[16], an abductive explanation, called also prime implicant, is defined as a *minimal* (for set inclusion) set of literals that is sufficient for predicting a class. Such explanations are closely tied to arguments pro classes. They are definitely the supports of arguments pro classes.

Definition 8 (Abductive Explanation): Let $x \in \mathcal{X}$ and $c \in \mathcal{C}$ s.t. $\mathtt{f}(x) = c$. An *abductive explanation* of (x, c) is any member of the set:

$$\mathtt{AE}(x, c) = \{H \subseteq \mathcal{U} \mid H \in \mathtt{Supp}(c) \text{ and } H \subseteq x\}.$$

Example 1 (Cont.) The pair $(x_1, \texttt{No})$ has a single abductive explanation, which is the support of the argument a_1, i.e., $\{(\texttt{Sky}, \texttt{Sunny}), (\texttt{Humidity}, \texttt{High})\}$. Note that the second argument pro No is not an explanation of the pair. The pair $(x_7, \texttt{Yes})$ also has a single abductive explanation: $\{(\texttt{Sky}, \texttt{Cloudy})\}$.

From the results presented in the previous section, it follows that abductive explanations exist for every instance, and a class that is assigned to all instances has a unique explanation, which is the emptyset.

Proposition 9: Let $x \in \mathcal{X}$ and $c \in \mathcal{C}$ s.t. $\texttt{f}(x) = c$.

- $\texttt{AE}(x, c) \neq \emptyset$.
- $\texttt{AE}(x, c) = \{\emptyset\}$ iff $\forall y \in \mathcal{X}$, $\texttt{f}(y) = c$.
- $\texttt{AE}(x, c) \subseteq \{H \subseteq \mathcal{U} \mid \langle H, c\rangle \in \texttt{Pros}(c)\}$

It is worth noticing that local explanations (for individual instances) coincide with the global (instance-independent) explanations of the classifier.

In [17], an abductive explanation of an instance x is defined as the minimal set of *features* (instead of literals) that caused the prediction $\texttt{f}(x)$. In other words, the set of explanations of x is: $\{\{f \in \mathcal{F} \mid (f, t) \in H\}$ where $H \in \texttt{AE}(x, c)\}$. This feature-based definition is reasonable when providing local explanations (i.e., for instances), however it may be *incomplete* or explaining the classifier independently of instances.

Example 2 (Cont.) The feature f_1 causes the prediction c_3, Hence $\{f_1\}$ is an explanation of x_3 and x_4. However, this is true only when f_1 gets the value 1. Indeed, if f_1 receives 0, then c_3 is not recommended by the classifier.

Why-Not questions While abductive explanations answer the question "why $\texttt{f}(x) = c$?", *why-not question* looks for "why $\texttt{f}(x) \notin \mathcal{C} \setminus \{c\}$?", i.e., why x cannot be labelled by any other class. Arguments pro a class can be seen as arguments in favour of discarding all the other classes. Hence, abductive explanations already answer the above why-not question.

Another type of why-not questions is "why $\texttt{f}(x) \neq c'$?", with $c \neq c'$. One looks for reasons behind avoiding c' in case of input x. Note that providing arguments pro $\texttt{f}(x) = c$ is not suitable since irrelevant information would be included in explanations, namely those that concern discarding the other classes (in case of multiple classes). The idea is them to look for a set of literals $H \subseteq x$ that caused avoiding c', hence an argument con c'.

Definition 9: Let $x \in \mathcal{X}$ and $c \in \mathcal{C}$ s.t $\texttt{f}(x) \neq c$. An explanation for (x, c) is any member of the set:

$$g(x, c) = \{H \subseteq \mathcal{U} \mid H \subseteq x \text{ and } H \in \texttt{Cons}(c')\}.$$

Example 2 (Cont.) The answer to the question why $\texttt{f}(x_4) \neq c_2$ is unique: $\{(f_1, 1)\}$. Recall that there are two arguments against c_2: $\{(f_1, 1)\}$ and $\{(f_2, 0)\}$.

Notation: For $x \in \mathcal{X}$, $h \subseteq \mathcal{U}$ s.t. h is consistent, $x_{\downarrow h}$ denotes the set of literals obtained by replacing the values of features in x by those in h and keeping the remaining ones unchanged.

Proposition 10: Let $x \in \mathcal{X}$ and $c \in \mathcal{C}$ s.t $\texttt{f}(x) \neq c$.

- $g(x, c) \neq \emptyset$.
- $\forall H \in g(x, c)$, $\exists H' \in \texttt{Pros}(\texttt{f}(x))$ s.t. $H \subseteq H'$.
- $\forall H \in g(x, c)$, $\exists y \in \mathcal{X} \setminus \{x\}$ s.t. $y = x_{\downarrow H}$ and $\texttt{f}(y) = c$.

Note that such explanations do not shed light on how to get the desired outcome, especially in case of non-binary features.

Counterfactuals, called also *contrastive* explanations, are widely used for interpreting predictions of black-box ML models, see eg. [4], [9], [18]. They state how the instance would have to be different for getting another outcome. In other words, they look for information capable of altering the prediction of an input. They are of two kinds: those that look for *information capable of altering the prediction of an input* to *whatever class*, and those that focus on altering the prediction to a *target* one. We call the first kind *general counterfactuals* and the second *specific counterfactuals*. General counterfactuals of (x, c) corresponds to arguments against c.

Definition 10 (General Counterfactuals): Let $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C}\rangle$ be a theory, $x \in \mathcal{X}$ and $c \in \mathcal{C}$ s.t. $\texttt{f}(x) = c$. A *general counterfactual* of (x, c) is any member of the set:

$$\texttt{CF}(x, c) = \{H \setminus x \mid \langle H, \overline{c}\rangle \in \texttt{Cons}(c)\}.$$

Unlike abductive explanations, counterfactuals may not exist. This is particularly the case when the class of the input at hand is assigned to all instances of the feature space. However, a counterfactual can never be the empty set.

Proposition 11: Let $\mathcal{T} = \langle \mathcal{F}, \mathcal{D}, \mathcal{C}\rangle$ be a theory and $x \in \mathcal{X}$ s.t. $\texttt{f}(x) = c$. The following hold:

- $\texttt{CF}(x, c) \subseteq \{H \setminus x \mid \langle H, \overline{c}\rangle \in \texttt{Cons}(c)\}$,
- $\texttt{CF}(x, c) = \emptyset$ iff $\forall y \in \mathcal{X}$, $\texttt{f}(y) = c$,
- $\emptyset \notin \texttt{CF}(x, c)$

General counterfactuals provide all the possible changes for an instance that lead to another outcome. In what follows, we provide a function which returns only the minimal adjusting of features in an instance.

Definition 11 (Minimal Counterfactuals): Let $x \in \mathcal{X}$. A *minimal counterfactual* of $(x, \texttt{f}(x))$ is a set $H \subseteq \mathcal{U}$ s.t.

- H is consistent
- $\texttt{f}(x_{\downarrow H}) \neq \texttt{f}(x)$
- $\nexists H' \subset H$ s.t. H' satisfies the above conditions.

The following result relates minimal counterfactuals with arguments con a class.

Proposition 12: Let $x \in \mathcal{X}$ s.t $\texttt{f}(x) = c$.

- If H is a minimal counterfactual of (x, c), then $\exists \langle H', \overline{c}\rangle \in \texttt{Cons}(c)$ s.t. $H = H' \setminus x$.
- If $\langle H, \overline{c}\rangle \in \texttt{Cons}(c)$, then $\exists H' \subseteq H \setminus x$ s.t. H' is a minimal counterfactual of x.

Remark: In [17], a counterfactual is defined as a set of features. Such a definition is suitable only when features are all binary since the modified value of each attribute is implicit. This is however not true in the general case.

Specific counterfactual provide the changes that should minimally occur for being classified in an expected class c.

Definition 12 (Specific Counterfactuals): Let $x \in \mathcal{X}$, $c \in \mathcal{C}$ s.t. $\texttt{f}(x) \neq c$. A *specific counterfactual* of (x, c) is a set $H \subseteq \mathcal{U}$ s.t.

- $\exists y \in \mathcal{X}$ s.t. $\mathtt{f}(y) = c$ and $y = x_{\downarrow H}$
- $\nexists H' \subset H$ s.t. H' satisfies the above conditions.

Obviously, every specific counterfactual is also a minimal counterfactual. The converse does not hold.

V. Related Work

Most work on finding explanations in the ML literature is experimental, focusing on specific models, exposing their internal representations to find correlations *post hoc* between these representations and the predictions, and is thus more about the arguably vaguer notion of interpretability.

There haven't been a lot of formal characterizations of explanations in AI, with the exception of [12], which defines abductive explanations, counterexamples and adversarial examples in a fragment of first order logic or [11]. Both works focused on local explanations, and proposed specific notions that are suitable in the case of binary classifiers with binary features. Furthermore, they did not relate the proposed notions to global behaviour, thus global explanations, of classifiers.

Unlike our work, which explains existing Black-box models, [19], [20] proposed novel classification models that are based on arguments. Their explanations are defined in dialectical way as fictitious dialogues between a proponent (supporting an output) and an opponent (attacking the output) following [21]. The authors in [22]–[25] followed the same approach for defining explainable multiple decision systems, recommendation systems, or scheduling systems. In the above papers an argument is simply an instance and its label while our arguments pro/con are much richer. This shows that they are proposed for different purposes.

VI. Conclusion

This paper investigates the different notions of (local, global) explanation that have been discussed in the literature for interpreting black-box classifiers without "opening" them. It proposes the first formal framework for *defining, generating*, and *comparing* the most prominent notions. The framework is based on two dual types of arguments for justifying predictions, and used them as building blocks of existing explanations. This work lays the foundations for formal comparisons with other types of explanation.

Acknowledgements

Support from the ANR-3IA Artificial and Natural Intelligence Toulouse Institute is gratefully acknowledged.

References

[1] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Computing Surveys*, vol. 51, no. 5, pp. 93:1–93:42, 2019.

[2] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[3] O. Biran and C. Cotton, "Explanation and justification in machine learning: A survey," in *IJCAI Workshop on Explainable Artificial Intelligence (XAI)*, 2017, pp. 1–6.

[4] S. Wachter, B. D. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *CoRR*, vol. abs/1711.00399, 2017. [Online]. Available: http://arxiv.org/abs/1711.00399

[5] A. Ignatiev, N. Narodytska, and J. Marques-Silva, "Abduction-based explanations for machine learning models," in *The Thirty-Third Conference on Artificial Intelligence, AAAI*, 2019, pp. 1511–1519.

[6] O. Biran and K. R. McKeown, "Human-centric justification of machine learning predictions," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*, 2017, pp. 1461–1467.

[7] R. Luss, P. Chen, A. Dhurandhar, P. Sattigeri, K. Shanmugam, and C. Tu, "Generating contrastive explanations with monotonic attribute functions," *CoRR*, 2019. [Online]. Available: http://arxiv.org/abs/1905.12698

[8] A. Dhurandhar, P. Chen, R. Luss, C. Tu, P. Ting, K. Shanmugam, and P. Das, "Explanations based on the missing: Towards contrastive explanations with pertinent negatives," in *Annual Conference on Neural Information Processing Systems, NeurIPS*, 2018, pp. 590–601.

[9] B. Mittelstadt, C. Russell, and S. Wachter, "Explaining explanations in AI," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 279–288.

[10] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018, pp. 1527–1535.

[11] A. Shih, A. Choi, and A. Darwiche, "A symbolic approach to explaining bayesian network classifiers," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*, 2018, pp. 5103–5111.

[12] A. Ignatiev, N. Narodytska, and J. Marques-Silva, "On relating explanations and adversarial examples," in *Thirty-third Conference on Neural Information Processing Systems, NeurIPS*, 2019, pp. 15 857–15 867.

[13] S. Kotsiantis and D. Kanellopoulos, "Discretization techniques: A recent survey," *GESTS International Transactions on Computer Science and Engineering*, vol. 32(1), pp. 47–58, 2006.

[14] Y. Dimopoulos, S. Dzeroski, and A. Kakas, "Integrating explanatory and descriptive learning in ILP," in *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI*, 1997, pp. 900–907.

[15] A. Kakas and F. Riguzzi, "Abductive concept learning," *New Generation Computing*, vol. 18, no. 3, pp. 243–294, 2000.

[16] A. Darwiche and A. Hirth, "On the reasons behind decisions," *CoRR*, vol. abs/2002.09284, 2020. [Online]. Available: https://arxiv.org/abs/2002.09284

[17] A. Ignatiev, N. Narodytska, N. Asher, and J. Marques-Silva, "From contrastive to abductive explanations and back again," in *AIxIA 2020 - Advances in Artificial Intelligence - XIXth International Conference of the Italian Association for Artificial Intelligence*, ser. Lecture Notes in Computer Science, vol. 12414. Springer, 2020, pp. 335–355.

[18] R. Byrne, "Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI*, 2019, pp. 6276–6282.

[19] L. Amgoud and M. Serrurier, "Agents that argue and explain classifications," *Autonomous Agents and Multi-Agent Systems*, vol. 16, no. 2, pp. 187–209, 2008.

[20] O. Cocarascu, A. Stylianou, K. Cyras, and F. Toni, "Data-empowered argumentation for dialectically explainable predictions," in *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence, ECAI*, 2020, p. In press.

[21] P. M. Dung, "On the Acceptability of Arguments and its Fundamental Role in Non-Monotonic Reasoning, Logic Programming and n-Person Games," *Artificial Intelligence*, vol. 77, pp. 321–357, 1995.

[22] Q. Zhong, X. Fan, X. Luo, and F. Toni, "An explainable multi-attribute decision model based on argumentation," *Expert Systems with Applications*, vol. 117, pp. 42–61, 2019.

[23] K. Cyras, D. Birch, Y. Guo, F. Toni, R. Dulay, S. Turvey, D. Greenberg, and T. Hapuarachchi, "Explanations by arbitrated argumentative dispute," *Expert Systems with Applications*, vol. 127, pp. 141–156, 2019.

[24] K. Cyras, D. Letsios, R. Misener, and F. Toni, "Argumentation for explainable scheduling," in *The Thirty-Third Conference on Artificial Intelligence, AAAI*, 2019, pp. 2752–2759.

[25] A. Rago, O. Cocarascu, and F. Toni, "Argumentation-based recommendations: Fantastic explanations and how to find them," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI*, 2018, pp. 1949–1955.