An ASP-Based Approach to Counterfactual Explanations for Classification

Leopoldo Bertossi[⋆]
Universidad Adolfo Ibáñez

Abstract. We propose answer-set programs that specify and compute counterfactual interventions as a basis for causality-based explanations to decisions produced by classification models. They can be applied with black-box models and models that can be specified as logic programs, such as rule-based classifiers. The main focus in on the specification and computation of maximum responsibility causal explanations. The use of additional semantic knowledge is investigated.

1 Introduction

Providing explanations to results obtained from machine-learning models has been recognized as critical in many applications; and has become an active research direction in the broader area of *explainable AI*. This becomes particularly relevant when decisions are automatically made by those models, possibly with serious consequences for stake holders. Since most of those models are algorithms learned from training data, providing explanations may not be easy or possible. These models are or can be seen as *black-box models*.

In AI, explanations have been investigated, among other areas, under *actual causality* [12], where *counterfactual interventions* on a causal model are central. They are hypothetical updates on the model's variables, to explore if and how the outcome of the model changes or not. In this way, explanations for an original output are defined and computed. Counterfactual interventions have been used with ML models, in particular with classification models [16, 19, 18, 13, 7, 15, 4].

In this work we introduce the notion of causal explanation, as a feature value for the entity under classification that is *most responsible* for the outcome. The responsibility score is adopted and adapted from the general notion of responsibility used in actual causality [6]. Experimental results with the responsibility score, and comparisons with other scores are reported in [4]. We also introduce *answer-set programs* (ASPs) that specify counterfactual interventions and causal explanations, and allow to specify and compute the responsibility score. The programs can be applied with black-box models, and with rule-based classification models.

As we show in this work, our declarative approach to counterfactual interventions is particularly appropriate for bringing into the game additional, declarative semantic knowledge, which is much more complicated to do with purely procedural approaches. In this way, we can combine logic-based specifications, and use the generic and optimized solvers behind ASP implementations.

[⋆] Member of RelationalAI's Academic Network, and the Millenium Institute on Foundations of Data (IMFD, Chile). UAI, Faculty Eng. & Sciences, Santiago, Chile. leopoldo.bertossi@uai.cl

2 Counterfactual Explanations

We consider a *classification model* represented by a label function L that maps entities $\mathbf{e}$ to 0 or 1. That is, to simplify the presentation, we consider binary classifications, but this is not essential. Every input entity $\mathbf{e}$ is represented by a record (or vector), $\mathbf{e} = \langle x_1, \ldots, x_n \rangle$, where x_i is the value $F_i(\mathbf{e}) \in Dom(F_i)$ taken on $\mathbf{e}$ by a feature $F_i \in \mathcal{F} = \{F_1, \ldots, F_n\}$. We do not assume we have a causal model; actually not even an explicit classification model. It could be a black-box model. The values in $Dom(F_i)$ can be all categorical or all numerical. However, *in this work we concentrate on features that can take a finite number of categorical values*. For numerical domains, we appeal to "bucketization and one-hot-encoding" (c.f. Section 4).

The problem is the following: Given an entity $\mathbf{e}$ that has received the label $L(\mathbf{e})$, provide an "explanation" for this outcome. In order to simplify the presentation, and without loss of generality, *we assume that label* 1 *is the one that has to be explained*. It is the "negative" outcome one has to justify, such as the rejection of a loan application.

Causal explanations are defined in terms of counterfactual interventions that simultaneously change feature values in $\mathbf{e}$ in such a way that the updated record gets a new label. A *causal explanation* for the classification of $\mathbf{e}$ is then a set of its original feature values that are affected by a *minimal counterfactual interventions*. These explanations are assumed to be more informative than others. Minimality can be defined in different ways, and we adopt an abstract approach, assuming a partial order relation $\preceq$ on counterfactual interventions.

Definition 1. Given a binary classifier represented by its label function L, and a fixed input record $\mathbf{e} = \langle x_1, \ldots, x_n \rangle$: (a) An *intervention* ι on $\mathbf{e}$ is a set of the form $\{\langle i_1; x_{i_1}, x'_{i_1} \rangle, \ldots, \langle i_K; x_{i_K}, x'_{i_K} \rangle\}$, with $i_s \neq i_\ell$, for $s \neq \ell$, $F_{i_s} \in \mathcal{F}$, $x_{i_s} = F_{i_s}(\mathbf{e})$, $x_{i_s} \neq x'_{i_s} \in Dom(X_{i_s})$. We denote with $\iota(\mathbf{e})$ the record obtained by applying to $\mathbf{e}$ intervention ι, i.e. by replacing in $\mathbf{e}$ every x_{i_s} appearing in ι by x'_{i_s}.
(b) A *counterfactual intervention* on $\mathbf{e}$ is an intervention ι on $\mathbf{e}$ such that $L(\mathbf{e}) \neq L(\iota(\mathbf{e}))$. A $\preceq$*-minimal* counterfactual intervention is such that there is no counterfactual intervention ι' on $\mathbf{e}$ with $\iota' \prec \iota$ (i.e. $\iota' \preceq \iota$, but not $\iota \preceq \iota'$).
(c) A *causal explanation* for $L(\mathbf{e})$ is a set of the form $\epsilon = \{\langle i_1; x_{i_1} \rangle, \ldots, \langle i_K; x_{i_K} \rangle\}$ for which there is a counterfactual intervention $\iota = \{\langle i_1; x_{i_1}, x'_{i_1} \rangle, \ldots, \langle i_K; x_{i_K}, x'_{i_K} \rangle\}$ for $\mathbf{e}$. Sometimes, to emphasize the intervention, we denote the explanation with $\epsilon(\iota)$.
(d) A causal explanation ϵ for $L(\mathbf{e})$ is $\preceq$*-minimal* if it is of the form $\epsilon(\iota)$ for a $\preceq$-minimal counterfactual intervention ι on $\mathbf{e}$. □

Several minimality criteria can be expressed in terms of partial orders, such as: (a) $\iota_1 \leq^s \iota_2$ iff $\pi_{1,2}(\iota_1) \subseteq \pi_{1,2}(\iota_2)$, with $\pi_{1,2}(\iota)$ the projection of ι on the first two positions. (b) $\iota_1 \leq^c \iota_2$ iff $|\iota_1| \leq |\iota_2|$. That is, minimality under set inclusion and cardinality, resp. In the following, we will consider only these; and mostly the second.

Example 1. Consider three binary features, i.e. $\mathcal{F} = \{F_1, F_2, F_3\}$, and they take values 0 or 1; and the input/output relation of a classifier $\mathcal{C}$ shown in Table 1. Let $\mathbf{e}$ be $\mathbf{e}_1$ in the table. We want causal explanations for its label 1. Any other record in the table can be seen as the result of an intervention on $\mathbf{e}_1$. However, only $\mathbf{e}_4, \mathbf{e}_7, \mathbf{e}_8$ are (results of) counterfactual interventions in that they switch the label to 0.

$\mathcal{C}$

entity (id)	F_1	F_2	F_3	L
$\mathbf{e}_1$	0	1	1	1
$\mathbf{e}_2$	1	1	1	1
$\mathbf{e}_3$	1	1	0	1
$\mathbf{e}_4$	1	0	1	0
$\mathbf{e}_5$	1	0	0	1
$\mathbf{e}_6$	0	1	0	1
$\mathbf{e}_7$	0	0	1	0
$\mathbf{e}_8$	0	0	0	0

Table 1

For example, $\mathbf{e}_4$ corresponds to the intervention $\iota_4 = \{\langle F_1, 1, 0\rangle, \langle F_2, 0, 1\rangle\}$, for which $\pi_{1,2}(\iota_4) = \{\langle F_1, 1\rangle, \langle F_2, 0\rangle\}$, a causal explanation that tells us that the presence of values 1 and 0 for F_1, resp. F_2 is a cause for $\mathbf{e}$ to be classified as 1. By taking a projection, the partial order $\preceq^s$ does not care about the values that replace the original feature values, as long as they are changed. Then, we have three causal explanations: $\epsilon_4 := \{\langle F_1, 0\rangle, \langle F_2, 1\rangle\}$, $\epsilon_7 := \{\langle F_2, 1\rangle\}$, and $\epsilon_8 := \{\langle F_2, 1\rangle, \langle F_3, 1\rangle\}$. (Given $\mathbf{e}$, it would be good enough to indicate the features whose values are relevant, e.g. $\epsilon_7 = \{F_2\}$.) Here, $\mathbf{e}_4$ and $\mathbf{e}_8$ are incomparable under $\preceq^s$, $\mathbf{e}_7 \prec^s \mathbf{e}_4$, $\mathbf{e}_7 \prec^s \mathbf{e}_8$, and ϵ_7 turns out to be $\preceq^s$- and $\preceq^c$-minimal (actually, minimum). □

Clearly, every $\preceq^c$-minimal explanation is also $\preceq^s$-minimal. However, it is easy to produce an example showing that a $\preceq^s$-minimal explanation may not be $\preceq^c$-minimal.

Definition 2. Given a binary classifier represented by its label function L, and a fixed input record $\mathbf{e}$: (a) An *s-explanation* for $L(\mathbf{e})$ is a $\preceq^s$-minimal causal explanation for $L(\mathbf{e})$. (b) A *c-explanations* $L(\mathbf{e})$ is a $\preceq^s$-minimal causal explanation for $L(\mathbf{e})$. □

This definition characterizes explanations as sets of (interventions on) features. However, it is common that one wants to quantify the "causal strength" of a single feature value in a record representing an entity [15, 4], or a single tuple in a database (as a cause for a query answer) [17], or a single attribute value in a database tuple [1, 2], etc. Different *scores* have been proposed in this direction, e.g. SHAP in [15] and DA in [4]. One of them, in the context of actual causality [12], is that of *responsibility* of an actual cause [6], which we adapt to our setting.

Definition 3. Let $\mathbf{e}$ be an entity represented as a record of feature values $x_i = F_i(\mathbf{e})$, $F_i \in \mathcal{F}$. (a) A feature value $v = F(\mathbf{e})$, with $F \in \mathcal{F}$, is a *value-explanation* for $L(\mathbf{e})$ if there is an s-explanation ϵ for $L(\mathbf{e})$, such that $\langle F, v\rangle \in \epsilon$.
(b) The *explanatory responsibility* of a value-explanation $v = F(\mathbf{e})$ is:

$$\mathsf{x\text{-}resp}_{\mathbf{e},F}(v) := max\{\tfrac{1}{|\epsilon|} : \epsilon \text{ is s-explanation with } \langle F, v\rangle \in \epsilon\}.$$

(c) If $v = F(\mathbf{e})$ is not a value-explanation, $\mathsf{x\text{-}resp}_{\mathbf{e},F}(v) := 0$. □

Notice that (b) can be stated as $\mathsf{x\text{-}resp}_{\mathbf{e},F}(v) := \frac{1}{|\epsilon^\star|}$, with $\epsilon^\star = \mathsf{argmin}\{|\epsilon| : \epsilon$ is s-explanation with $\langle F, v\rangle \in \epsilon\}$.

Adopting the usual terminology in actual causality [12], a *counterfactual value-explanation* for $\mathbf{e}$'s classification is a value-explanation v with $\mathsf{x\text{-}resp}_{\mathbf{e}}(v) = 1$, that is, it suffices, without company of other feature values in $\mathbf{e}$, to justify the classification. Similarly, an *actual value-explanation* for $\mathbf{e}$'s classification is a value-explanation v with $\mathsf{x\text{-}resp}_{\mathbf{e}}(v) > 0$. That is, v appears in an s-explanation ϵ, say as $\langle F, v\rangle$, but possibly in company of other feature values. In this case, $\epsilon \setminus \{\langle F, v\rangle\}$ is called a *contingency set* for v [17]. It turns out that maximum-responsibility value-explanations appear in c-explanations.

Example 2. (ex. 1 cont.) ϵ_7 is the only c-explanation for entity $\mathbf{e}_1$'s classification. Its value 1 for feature F_2 is a value-explanation, and its explanatory responsibility is $\mathsf{x\text{-}resp}_{\mathbf{e}_1,F_2}(1) := 1$. □

3 Specifying Causal Explanations in ASP

Entities will be represented by a predicate with $n+2$ arguments $E(\cdot;\cdots;\cdot)$. The first one holds a record (or entity) id (which may not be needed when dealing with single entities). The next n arguments hold the feature values. The last argument holds an annotation constant from the set $\{\mathsf{o}, \mathsf{do}, \star, \mathsf{s}\}$. Their semantics will be specified below, by the generic program that uses them.

Initially, a record $\mathbf{e} = \langle x_1, \ldots, x_n\rangle$ has not been subject to interventions, and the corresponding entry in predicate E is of the form $E(\mathbf{e}; \bar{x}; \mathsf{o})$, with $\bar{x}$ an abbreviation for $x_1, \ldots, x_n$, and constant o standing for "original entity".

When the classifier gives label 1 to $\mathbf{e}$, the idea is to start changing feature values, one at a time. The intervened entity becomes then annotated with constant do in the last argument. When the resulting intervened entities are classified, we may not have the classifier specified within the program. For this reason, the program uses a special predicate $\mathcal{C}[\cdot;\cdot]$, whose first argument takes (a representation of) an entity under classification, and whose second argument returns the binary label. We will assume this predicate can be invoked by an answer-set program as an external procedure, much in the spirit of HEX-programs [8, 9]. Since the original instance may have to go through several interventions until reaching one that switches the label to 0, the intermediate entities get the "transition" annotation $\star$. This is achieved by a generic program.

The Counterfactual Intervention Program:

1. The facts of the program are all the atoms of the form $Dom_i(c)$, with $c \in Dom_i$, plus the initial entity $E(\mathbf{e}; \bar{f}; \mathsf{o})$, where $\bar{f}$ is the initial vector feature values.
2. The transition entities are obtained as initial, original entities, or as the result of an intervention: (here, $\mathbf{e}$ is a variable standing for a record id)

$$E(\mathbf{e}; \bar{x}; \star) \longleftarrow E(\mathbf{e}; \bar{x}; \mathsf{o}).$$
$$E(\mathbf{e}; \bar{x}; \star) \longleftarrow E(\mathbf{e}; \bar{x}; \mathsf{do}).$$

3. The program rule specifying that, every time the entity at hand (original or obtained after a "previous" intervention) is classified with label 1, then a new value has to be picked from a domain, and replaced for the current value. The new value is chosen via the non-deterministic "choice operator", a well-established mechanism in ASP [11]. In this case, the values are chosen from the domains, and are subject to the condition of not being the same as the current value:

$$E(\mathbf{e}; x_1', x_2, \ldots, x_n, \mathsf{do}) \vee \cdots \vee E(\mathbf{e}; x_1, x_2, \ldots, x_n', \mathsf{do}) \longleftarrow E(\mathbf{e}; \bar{x}; \star), \mathcal{C}[\bar{x}; 1],$$
$$Dom_1(x_1'), \ldots, Dom_n(x_n'), x_1' \neq x_1, \ldots, x_n' \neq x_n,$$
$$choice(x_1'), \ldots, choice(x_n').$$

4. The following rule specifies that we can "stop", hence annotation s, when we reach an entity that gets label 0.

$$E(\mathbf{e}; \bar{x}; \mathsf{s}) \longleftarrow E(\mathbf{e}; \bar{x}; \mathsf{do}), \mathcal{C}[\bar{x}; 0].$$

5. We add a *program constraint* specifying that we prohibit going back to the original entity via local interventions:

$$\longleftarrow E(\mathbf{e}; \bar{x}; \mathsf{do}), E(\mathbf{e}; \bar{x}; \mathsf{o}).$$

6. The causal explanations can be collected by means of predicates $Expl_i(\cdot; \cdot)$ specified by means of:

$$Expl_i(\mathbf{e}; x_i) \longleftarrow E(\mathbf{e}; x_1, \ldots, x_n; \mathsf{o}), E(\mathbf{e}; x'_1, \ldots, x'_n; \mathsf{s}), x_i \neq x'_i.$$

Actually, each of these is a value-explanation. □

The program will have several stable models due to the disjunctive rule and the choice operator. Each model will hold intervened versions of the original entity, and hopefully versions for which the label is switched, i.e. those with annotation s. If the classifier never switches the label, despite the fact that local interventions are not restricted (and this would be quite an unusual classifier), we will not find a model with a version of the initial entity annotated with s. Due to the program constraint in 5., none of the models will have the original entity annotated with do, because those models would be discarded [14].

The choice operator can be replaced by additional rules that contain non-stratified negation [11]. So, its use hides occurrences of non-stratified negation. Another implicit source of non-stratified negation is the use of disjunction in the head of the rule. By the way, the semantics of ASP, which involves model minimality, makes only one of the atoms in the disjunct true (unless forced otherwise by the program itself).

Example 3. (ex. 1 cont.) Most of the *Counterfactual Intervention Program* above is generic. In this particular example, the have the following facts: $Dom_1(0), Dom_1(1)$, $Dom_2(0), Dom_2(1), Dom_3(0), Dom_3(1)$ and $E(\mathbf{e}_1; 0, 1, 1; \mathsf{o})$, with $\mathbf{e}_1$ a constant, the record id of the first row in Table 1.

In this very particular situation, the classifier is explicitly given by Table 1. Then, predicate $\mathcal{C}[\cdot; \cdot]$ can be specified with a set of additional facts: $\mathcal{C}[0, 1, 1; 1]$, $\mathcal{C}[1, 1, 1; 1]$, $\mathcal{C}[1, 1, 0; 1]$ $\mathcal{C}[1, 0, 1; 0]$ $\mathcal{C}[1, 0, 0; 1]$ $\mathcal{C}[0, 1, 0; 1]$ $\mathcal{C}[0, 0, 1; 0]$ $\mathcal{C}[0, 0, 0; 0]$.

The stable models of the program will contain all the facts above. One of them, say $\mathcal{M}_1$, will contain (among others) the facts: $E(\mathbf{e}_1; 0, 1, 1; \star), E(\mathbf{e}_1; 0, 1, 1; \star)$. The presence of the last atom activates rule 3., because $\mathcal{C}[0, 1, 1; 1]$ is true (for $\mathbf{e}_2$ in Table 1). New facts are produced for $\mathcal{M}_1$ (the new value due to an intervention is underlined): $E(\mathbf{e}_1; \underline{1}, 1, 1; \mathsf{do})$, $E(\mathbf{e}_1; \underline{1}, 1, 1; \star)$. Due to the last fact and $\mathcal{C}[1, 0, 1; 0]$, rule 3. is activated again. Choosing the value 0 for the second disjunct, atoms $E(\mathbf{e}_1; \underline{1}, \underline{0}, 1; \mathsf{do})$, $E(\mathbf{e}_1; \underline{1}, \underline{0}, 1; \star)$ are generated. For the latter, $\mathcal{C}[1, 0, 1; 0]$ is true (coming from $\mathbf{e}_4$ in Table 1), switching the label to 0. Rule in no longer activated, and we can apply rule 4., obtaining $E(\mathbf{e}_1; \underline{1}, \underline{0}, 1; \mathsf{s})$.

From rules 6., we obtain as explanations: $Expl_1(\mathbf{e}_1; 0), Expl_1(\mathbf{e}_1; 1)$, showing the values in $\mathbf{e}_i$ that were changed. All this in model $\mathcal{M}_1$. There are other models, and one of them contains $E(\mathbf{e}_1; 0, \underline{0}, 1; \mathsf{s})$, the minimally intervened version of $\mathbf{e}_1$, i.e. $\mathbf{e}_7$. □

3.1 C-explanations and maximum responsibility

There is no guarantee that the intervened entities $E(\mathbf{e}; c_1, \ldots, c_n; \mathsf{s})$ will correspond to a c-explanations, which are the main focus of this work. In order to obtain them (and only them), we add *weak program constraints* (WCs) to the program. They can be violated by a stable model of the program (as opposed to (strong) program constraints that have to be satisfied). However, they have to be violated in a minimal way. We use WCs, whose *number* of violations have to be minimized, in this case, for $1 \leq i \leq n$:

$$\Longleftarrow \; E(\mathbf{e}; x_1, \ldots, x_n, \mathsf{o}), E(\mathbf{e}; x'_1, \ldots, x'_n, \mathsf{s}), x_i \neq x'_i.$$

Only the stable models representing an intervened version of $\mathbf{e}$ with a minimum number of value discrepancies with $\mathbf{e}$ will be kept.

In each of these "minimum-cardinality" stable models $\mathcal{M}$, we can collect the corresponding c-explanation as the set $\epsilon^{\mathcal{M}} = \{Expl_i(\mathbf{e}; c_i) \mid Expl_i(\mathbf{e}; c_i) \in \mathcal{M}\}$. This can be done within a ASP system such as *DLV*, which allows set construction and aggregation, in particular, counting [14], . Actually, counting comes handy to obtain the cardinality of $\epsilon^{\mathcal{M}}$. The responsibility of a value-explanation $Expl_i(\mathbf{e}; c_i)$ will then be: $\mathsf{x\text{-}resp}_{\mathbf{e},F_i}(c_i) = \frac{1}{|\epsilon^{\mathcal{M}}|}$.

4 Semantic Knowledge

Counterfactual interventions in the presence of semantic conditions requires consideration. As the following example shows, not every intervention, or combination of them, may be admissible [3]. It is in this kind of situations that declarative approaches to counterfactual interventions, like the one presented here, become particularly useful.

Example 4. A moving company makes automated hiring decisions based on feature values in applicants' records of the form $R = \langle appCode, \textit{ability to lift}, \textit{gender}, \textit{weight}, \textit{height}, age\rangle$. Mary, represented by $R^\star = \langle 101, 1, F, \textit{160 pounds}, \textit{6 feet}, 28\rangle$ applies, but is denied the job, i.e. the classifier returns: $L(R^\star) = 1$. To explain the decision, we can hypothetically change Mary's gender, from F into M, obtaining record $R^{\star\prime}$, for which we now observe $L(R^{\star\prime}) = 0$. Thus, her value F for *gender* can be seen as a counterfactual explanation for the initial decision.

As an alternative, we might keep the value of *gender*, and counterfactually change other feature values. However, we might be constrained or guided by an ontology containing, e.g. the denial semantic constraint $\neg(R[2] = 1 \wedge R[6] > 80)$ (2 and 6 indicating positions in the record) that prohibits someone over 80 to be qualified as fit to lift. We could also have a rule, such as $(R[3] = M \wedge R[4] > 100 \wedge R[6] < 70) \rightarrow R[2] = 1$, specifying that men who weight over 100 pounds and are younger than 70 are automatically qualified to lift weight. □

In situations like that described in the example, we can add to the ASP we had before: (a) program constraints that prohibit certain models, e.g. $\longleftarrow \; R(\mathbf{e}; x, \mathsf{1}, y, z, u, w; \star)$, $w > \mathsf{80}$; (b) additional rules, e.g. $R(\mathbf{e}; x, \mathsf{1}, y, z, u, w; \star) \longleftarrow R(\mathbf{e}; x, y, \mathsf{M}, z, u, w; \star)$, $z > \mathsf{100}, w < \mathsf{70}$, that may automatically generate additional interventions. In a similar way, one could accommodate certain preferences using weak program constraints.

Buckets and one-hot-encodings. A common situation where not all interventions are admissible occurs when features take continuous values. In these cases, it is common to appeal to *bucketization*, i.e. the feature range is first discretized by splitting it into finitely many, usually non-overlapping intervals. This makes the feature basically categorical (each interval becoming a categorical value). Next, one applies *one-hot-encoding*, that represents the original feature as a vector of indicator functions, one for each categorical value (intervals here) [4]. For example, if we have a continuous feature ExternalRiskEstimate, its buckets could be: $[0, 64)$, $[64, 71)$, $[71, 76)$, $[76, 81)$, $[81, \infty)$. Accordingly, if for an entity $\mathsf{ExternalRiskEstimate}(\mathbf{e}) = 65$, then, after one-hot-encoding, this value is represented as the vector $[0, 1, 0, 0, 0, 0]$.

In a case like this, it is clear that counterfactual interventions are constrained by the assumptions behind bucketization and one-hot-encoding. For example, the vector cannot be updated into, say $[0, 1, 0, 1, 0, 0]$, meaning that the feature value for the entity falls both in intervals $[64, 71)$ and $[76, 81)$. Bucketization and one-hot-encoding can make good use of program constraints, such as $\longleftarrow ERE(\mathbf{e}; x, 1, y, 1, z, w; \star)$, etc. Of course, admissible interventions on predicate ERE could be easily handled with a disjunctive rule like that in 3., but without the "transition" annotation $\star$. However, the ERE record is commonly a component of a larger record containing all the feature values for an entity [4]. Hence the need for a more general and uniform form of specification.

5 Discussion

In this work we have treated classifiers as black-boxes that are represented by external predicates in the ASP. However, in some cases it could be the case that the classifier is given by a set of rules, which, if compatible with ASPs, could be appended to the program, to define the classification predicate $\mathcal{C}$. The domains used by the programs can be given explicitly. However, they can be specified and extracted from other sources. For example, for the experiments in [4], the domains were built from the training data, a process that can be specified and implemented in ASP.

The ASPs we have used are inspired by *repair programs* that specify and compute the repairs of a database that fails to satisfy the intended integrity constraints [5]. Actually, the connection between database repairs and actual query answer causality was established and exploited in [1]. ASPs that compute attribute-level causes for query answering were introduced in [2]. They are much simpler that those presented here, because in that scenario, changing attribute values by nulls is good enough to invalidate the query answer (the "equivalent" in that scenario to switching the classification label here). Once a null is introduced, there is no need to take it into account anymore, and a single "step" of interventions is good enough.

Here we have considered only s- and c-explanations, specially the latter. Both embody specific and different, but related, minimization conditions. However, counterfactual explanations can be cast in terms of different optimization criteria [13, 18]. One could investigate in this setting other forms on preferences, the generic $\preceq$ in Definition 1, by using ASPs as those introduced in [10].

This article reports on preliminary work that is part of longer term and ongoing research. In particular, we are addressing the following: (a) multi-task classification. (b)

inclusion of rule-based classifiers. (c) scores associated to more than one intervention at a time [4], in particular, to full causal explanations.

References

[1] Bertossi, L. and Salimi, B. From Causes for Database Queries to Repairs and Model-Based Diagnosis and Back. *Theory of Computing Systems*, 2017, 61(1):191-232.

[2] Bertossi, L. Characterizing and Computing Causes for Query Answers in Databases from Database Repairs and Repair Programs. Proc. FoIKs, 2018, Springer LNCS 10833, pp. 55-76. Revised and extended version as Corr Arxiv Paper cs.DB/1712.01001.

[3] Bertossi, L. and Geerts, F. Data Quality and Explainable AI. To appear in *ACM Journal of Data and Information Quality*, 2020.

[4] Bertossi, L., Li, J., Schleich, M., Suciu, D. and and Vagena, Z. "Causality-based Explanation of Classification Outcomes". To appear in Proc. 4th International Workshop on "Data Management for End-to-End Machine Learning" (DEEM) at ACM SIGMOD, 2020. Posted as Corr Arxiv Paper arXiv:2003.0686.

[5] Caniupan, M. and Bertossi, L. The Consistency Extractor System: Answer Set Programs for Consistent Query Answering in Databases. *Data & Knowledge Engineering*, 2010, 69(6):545-572.

[6] Chockler, H. and Halpern, J. Y. Responsibility and Blame: A Structural-Model Approach. *J. Artif. Intell. Res.*, 2004, 22:93-115.

[7] Datta, A., Sen, S. and Zick, Y. Algorithmic Transparency via Quantitative Input Influence: Theory and Experiments with Learning Systems. IEEE Symposium on Security and Privacy, 2016.

[8] Eiter, T., Germano, S., Ianni, G., Kaminski, T., Redl, C., Schüller, P. and Weinzierl, A. The DLVHEX System. *Künstliche Intelligenz*, 2019, 32(2-3):187-189.

[9] Eiter, T., Kaminski, T., Redl, C., Schüller, P. and Weinzierl, A. Answer Set Programming with External Source Access. *Reasoning Web*, Springer LNCS 10370, 2017, pp. 204-275.

[10] Gebser, M., Kaminski, R. and Schaub, T. Complex Optimization in Answer Set Programming. *Theory Pract. Log. Program.*, 2011, 11(4-5):821-839.

[11] Giannotti, F., Greco, S., Sacca, D. and Zaniolo, C. Programming with Non-Determinism in Deductive Databases. *Ann. Math. Artificial Intelligence*, 1997, 19(12):97125.

[12] Halpern, J. and Pearl, J. Causes and Explanations: A Structural-Model Approach: Part 1. *British J. Philosophy of Science*, 2005, 56:843-887.

[13] Karimi,A. H., Barthe, G., Balle, B. and Valera, I. Model-Agnostic Counterfactual Explanations for Consequential Decisions. Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), 2020. arXiv: 1905.11190.

[14] Leone, N., Pfeifer, G., Faber, W., Eiter, T., Gottlob, G., Koch, C., Mateis, C., Perri, S. and Scarcello, F. The DLV System for Knowledge Representation and Reasoning. *ACM Transactions on Computational Logic*, 2006, 7(3):499-562.

[15] Lundberg, S. and Lee, S.-I. A Unified Approach to Interpreting Model Predictions. Proc. NIPS 2017, pp. 4765-4774.

[16] Martens, D. and Provost, F. J. Explaining Data-Driven Document Classifications. *MIS Quarterly*, 2014, 38(1):73-99.

[17] Meliou, A., Gatterbauer, W., Moore, K. F. and Suciu, D. The Complexity of Causality and Responsibility for Query Answers and Non-Answers. Proc. VLDB, 2010, pp. 34-41.

[18] Russell, Ch. Efficient Search for Diverse Coherent Explanations. Proc. FAT 2019, pp. 20-28. arXiv:1901.04909.

[19] Wachter, S., Brent D. Mittelstadt, B. D. and Chris Russell, C. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR. CoRR abs/1711.00399, 2017.