



Reason to Explain: Interactive Contrastive Explanations (REASONX)

Laura State^{1,2}, Salvatore Ruggieri¹, and Franco Turini¹

¹ University of Pisa, Pisa, Italy
laura.state@di.unipi.it

² Scuola Normale Superiore, Pisa, Italy

Abstract. Many high-performing machine learning models are not interpretable. As they are increasingly used in decision scenarios that can critically affect individuals, it is necessary to develop tools to better understand their outputs. Popular explanation methods include contrastive explanations. However, they suffer several shortcomings, among others an insufficient incorporation of background knowledge, and a lack of interactivity. While (dialogue-like) interactivity is important to better communicate an explanation, background knowledge has the potential to significantly improve their quality, e.g., by adapting the explanation to the needs of the end-user.

To close this gap, we present REASONX, an explanation tool based on Constraint Logic Programming (CLP). REASONX provides interactive contrastive explanations that can be augmented by background knowledge, and allows to operate under a setting of under-specified information, leading to increased flexibility in the provided explanations. REASONX computes factual and contrastive decision rules, as well as closest contrastive examples. It provides explanations for decision trees, which can be the ML models under analysis, or global/local surrogate models of any ML model.

While the core part of REASONX is built on CLP, we also provide a program layer that allows to compute the explanations via Python, making the tool accessible to a wider audience. We illustrate the capability of REASONX on a synthetic data set, and on a well-developed example in the credit domain. In both cases, we can show how REASONX can be flexibly used and tailored to the needs of the user.

Keywords: explainable AI · Contrastive Explanations · Background Knowledge · Interactivity · Constraint Logic Programming

1 Introduction

Contrastive (counterfactual) explanations (CEs)¹ are a popular method to provide insights into not interpretable machine learning (ML) models. Signifi-

¹ In this paper, we refer to *contrastive* over counterfactual explanations, in order to avoid confusion with the concept of counterfactuals as understood in causality, and following [28].

cant efforts are made to provide CEs for different data types such as tabular data [49], images [9] or text [53], based on numeric [31, 49], causal [24] or other approaches [38, 48]. However, some open challenges remain, such as the integration of background knowledge [5, 45], and interactivity [28, 29, 51].

To tackle these shortcomings, we present REASONX (REASON to eXplain). We display an exemplary dialogue between a fictional end-user and REASONX below. The dialogue is situated in the context of a credit application example, therefore the user of the tool is a natural person whose credit application has been rejected by an Automated Decision-Making (ADM) system. Please note that while the information content of the displayed dialogue is exactly what REASONX can provide, we enhanced the dialogue to mimic better a realistic interaction by reporting questions and answers in natural language.²

USER: I want to understand the decision better. Can I see the rule that led to the denial of my credit application?

REASONX: Your credit application was rejected, because your income is lower than 60,000 EUR/year, and you still have to pay back the lease of your car.

USER: Ok. Can you present me two different options that will lead to a change of the decision outcome? Please take into consideration that I need a credit of at least 10,000 EUR. I would like to see options that require as little change as necessary.

REASONX: You have the following two options: you pay back the lease on the car, or you increase your age by 10 years (from 35 to 45 years).

USER: The second option presented is a bit strange. I am wondering whether this is something salient in the model. Can I please see the options to obtain credit for an individual with the same properties as me, for a credit of at least 10,000 EUR, but with the feature age at 35 years or less (i.e., young applicant), instead of fixed?

REASONX: For the given profile, the credit is always rejected.

USER: Can you please show how to reverse the decision, under as few changes as possible, for the specified profile?

REASONX: Credit can be obtained, if the feature age is set to higher than 35 years.

USER: This is interesting and worth investigating further. There could be bias w.r.t. the age of the person that applies for credit.

This dialogue illustrates the capabilities of REASONX. Our tool addresses the following points:

Background Knowledge. Adding background knowledge to an explanation has the potential to significantly improve its quality, but it is seldom offered [5, 45]. Ignoring this knowledge is not necessarily wrong, but can lead to contrastive explanations that ignore the needs of the data subject under decision, or CEs that do not fit the reality of our world - this depends closely on the purpose of the explanation. An example of such knowledge in the credit application example is the minimum credit amount (“a credit of at least 10,000 EUR”, see dialogue above). For REASONX, we rely on background knowledge in the form of linear constraints over the features.

² Adding a layer that implements a natural language communication between the user and REASONX is left for future work.

Interactivity. Interactivity is a property of an explanation that is important, if the explanations should be communicated successfully to the end-user [28, 29, 51]. However, most eXplainable AI (XAI) tools do not account for this. Interactivity arises naturally in REASONX: the end-user can flexibly query the explanation tool, choosing answers that best fit her questions, adding and removing constraints, and thereby building her own, personalized explanation.

Under-Specified Information. REASONX allows computing CEs under under-specified information. For example, it is not necessary to fix all features of the data instance of interest to compute a matching CE but provide only bounds (“feature age at 35 years or less”, see dialogue above). This property is implicitly in the use of CLP, and leads to a wider flexibility in provided explanations. Further, there is a loose connection to the notion of *group contrastive explanations*, which have been shown to be beneficial to the end-user via user studies [50].

1.1 Contributions and Structure of the Paper

With this paper, we make the following contributions

- *Explanation tool.* We propose REASONX, a novel explanation tool that provides contrastive interactive explanations, that can incorporate background knowledge, and that works under under-specified information. To the best of our knowledge, this is one of the first tools using constraint logic programming to generate contrastive explanations.
- *Synthetic data set illustration.* A first illustration of REASONX is based on a well-defined synthetic data set. We demonstrate step-by-step how REASONX operates under different (constraint) settings, including some graphical representations.
- *Credit application example.* We demonstrate the capabilities of REASONX on the Adult Income data set. This data set can be used as an approximation to determine the income of a natural person, we assume here that this approximation relates directly to a decision about credit allocation. We provide this demonstration together with an in-depth discussion of the relevant context (credit domain). We chose this context, as ADM systems for credit applications are an important and recurring topic. Further, as they are classified as “high-risk” according to the proposed AI Act of the European Union [16], they have to be considered particularly carefully.

This paper is structured as follows. In Sect. 2, we discuss background and related work. In Sect. 3, we introduce REASONX, followed by the illustration on synthetic data in Sect. 4 and the credit application example in Sect. 5. We close our paper by discussing limitations of this work in Sect. 6, and a conclusion in Sect. 7.

We provide our code (REASONX and experiments) via a public repository.³ In-depth theory and implementation details of REASONX will be provided in a companion paper.

³ <https://github.com/lstate/REASONX>.

2 Background and Related Work

2.1 Explanations

Explanations for opaque ML models (“black box models”) can be divided into two categories [19]: *global* methods, explaining the full model at once, or *local* methods, producing explanations that are only valid for a single data instance (data subject). Further, we can distinguish between *model-agnostic* and *model-specific* methods - the first can be applied to any ML model, the second only to the ML model type it was developed for.

Contrastive (Counterfactual) Explanations. Contrastive explanations (CEs) are local, model-agnostic explanations. They are computed *after* the model was trained (post-hoc), and based only on input-output pairs.

Wachter et al. [49] introduced the idea of contrastive explanations in the field of XAI. Following them, a contrastive explanation can be described as an optimization problem of the following form:

$$\arg \min_{x_{ce}} \max_{\lambda} \lambda (f_w(x_{ce}) - y_{ce})^2 + d(x_f, x_{ce}) \quad (1)$$

x_{ce} denotes the contrastive data point, x_f the original data point, $f_w(x_{ce})$ the prediction of the contrastive example and y_{ce} the desired prediction. Further, λ denotes a tuning parameter, and $d(\cdot, \cdot)$ a distance measure, usually the Manhattan distance, weighted by the median absolute deviation. Equation 1 minimizes the distance between the factual and the contrastive data point. The intuition underlying a CE is that to be realistic, it should be as close as possible to the original data point, while at the same time making sure that the prediction of the CE aligns with the desired prediction.

Contrastive explanations are well received by the community. However, they do not only have support from the technical side. While [49] is discussing the legal basis for contrastive explanations, with a focus on the European General Data Protection Regulation (GDPR), [7] argues in favor of CEs from a psychological point of view. Further, both [28] and [30] make clear that explanations in a contrastive form in general, i.e., explanations outlining why other events did not occur and differences to the actual outcome, are highly desirable for the (lay) end-user. Finally, we point to a number of surveys: [23] focusing on the computational side of CEs and actionable recourse, a closely related field, [25], critically analyzing the lack of evaluation methodology for CE methods, and two additional surveys [17, 46].

Technically Related Approaches. The approach taken in this work is inspired by two strands of research, focusing either on solving CE queries by (a) using SAT, or causality-based frameworks [22, 24] or (b) using ILP/MILP approaches [13, 21, 38, 48]. However, our approach is different in two main points: first, our focus is clearly on creating explanations for *any* ML model. Approaches (a) are generally agnostic, but the model internals need to be known, in (b) only linear

or additive models are considered. Second, to the best of our knowledge, this is one of the first approaches using CLP to generate CEs.

Group Contrastive (Counterfactual) Explanations. The notion of group contrastive explanations is relatively new and refers to a small body of work that has not yet converged on a common definition. A definition of group contrastive explanations that is based on common “feature differences”, i.e., the idea that different but similar data points share the same contrastive example is proposed in [50]. Their approach builds on a simple extension of the traditional approach for the generation of CEs such as [31]. Further, the paper benchmarks its approach via a user study, showing its usefulness. Other works include [2], focusing on explanations that refer to an ensemble of decisions, and [34], connecting group CEs to actionable recourse.

Background Knowledge. Adding background knowledge to an explanation has the potential to significantly improve its quality [5, 45]. In this work, we refer to background (or prior) knowledge as to any information that is relevant in the decision context but that does not emerge through the decision pipeline. Not only do simple facts count as such, but we would also consider a natural law as such knowledge, or as in our example in the introduction, a specific restriction a customer has related to her living reality (e.g., a minimum credit amount). However, we have to restrict ourselves to knowledge that can be formalized and thus used by the explainer.

REASONX incorporates knowledge in the form of linear constraints. Other examples of knowledge integration include the following. A local explanation tool for medical data that incorporates an ontology (the ICD-9-CM) to generate a meaningful, local neighborhood for explanation generation [33]. A discrimination discovery approach [37], where knowledge in the form of association rules is used to detect cases of indirect discrimination. Explanations are also historically connected to knowledge and logic reasoning - the first systems offered to explain AI models were expert systems [11].

Interactivity. Interactivity as property of an explanation aligns closely with our working definition: “an explanation, or explainability is about an interaction, or an exchange of information”, where it crucially matters to *whom* the explanation is given, and for *what* purpose [45].

While being acknowledged as an important property of an explanation that is successfully communicated to the end-user [28, 29, 51] - interactivity is only in very few cases incorporated into XAI methods. Sokol and Flach [44] outline the usefulness of interactivity prominently in their paper: “Truly interactive explanations allow the user to tweak, tune and personalise them (i.e., their content) via an interaction, hence the explainee is given an opportunity to guide them in a direction that helps to answer selected questions” Also, they present a first solution (the glass-box tool [42]): an interactive explanation tool that provides

explanations in natural language and that can be queried either by voice or via a chat. Similar to our work, it relies on a decision tree to generate the explanations. The tool was tested in different environments, this helped the authors to also develop a mapping of desiderata of interpretable explanations.

We further point to [27], presenting an interview study with practitioners that revealed that interactivity for explanations is strongly preferred, and discussing some ideas about how to achieve this. As a last example, we point to the what-if tool, a commercial application provided by Google.⁴

2.2 (Constraint) Logic Programming

Logic programming (LP) is a declarative approach to problem-solving, based on logic rules in the form of (Horn) clauses [1]. It supports reasoning under various settings, i.e., deductive, inductive, abductive and meta-reasoning [12, 39]. Starting with the Prolog programming language [10], programming in logic has been extended in several directions, as per expressivity and efficiency [26]. Constraint logic programming (CLP) augments logic programming with the ability to solve constrained problems in some domain [20]. We rely on $CLP(R)$, which consists of linear constraints over the reals, as implemented in the SWI Prolog system [52]. $CLP(R)$ adds to logic programming rules the ability to test for linear constraint satisfiability, entailment, equivalence, and projection, as well as for solving MILP optimization problems.

(C)LP for XAI Logic programming is considered symbolic reasoning - contrary to what is commonly referred to as sub-symbolic reasoning, or ML. While the first set of approaches is inherently transparent, most ML models are not, but come with other advantages, such as the ability to work on large amounts of data. Combining both is a promising synergy - exactly what we do in this paper.

A related approach to our work is a body of papers by Sokol et al. [41–43]. While the first introduces explanations for decision trees (among others, through contrastive explanations), the second generalizes it to be used as a local surrogate model. In the third paper, the focus is explicitly set on interactivity. Our work is closely linked to that. The main difference is in methodology, i.e., we rely on CLP in our computations, thus allowing for the integration of background knowledge, interactivity, and operation under under-specified information, making REASONX unique against previous approaches. Another related work is [6]. It provides CEs - but treats them from the angle of (actual) causality. Methodologically, it relies on answer set programming (ASP), and uses also its straightforward ability to integrate knowledge into the explanations.

A survey that discusses how to combine both symbolic (logic) and sub-symbolic systems, with a specific focus on explainable AI is [8]. Further, [47] adopts ASP to compute both local and global explanations for tree ensemble models. While it is methodologically related to our approach (relying on logic programming and on tree structures), it does not discuss the notion of

⁴ <https://pair-code.github.io/what-if-tool/>.

CEs. There are also a few approaches using logic rules as explanations, but that methodologically do not rely on LP, e.g., [18] and [40].

3 REASONX: Reason to Explain

We propose REASONX, a novel tool to generate contrastive explanations that can account for background knowledge, that works under under-specified information and is highly interactive. REASONX can be used to obtain factual and contrastive decision rules about the classification of the data point (or profile) of interest, and provides the closest CE (via optimization). In all settings, background knowledge can be added in the form of linear constraints.

REASONX is strongly guided by a decision tree model, called the *base model*. Such a decision tree can be: (a) the model to be explained and reasoned about; (b) a global surrogate model of a black box; (c) a local surrogate model of a black box decision in the neighborhood of an instance to explain. In cases (b) and (c), the surrogate model is assumed to have good fidelity in reproducing the black box decisions. This is reasonable for local models, i.e., in case (c), by learning the tree over a localized neighborhood as common in perturbation-based explanation methods such as LIME [35]. Regarding case (b), we point out that our approach works with any type of decision tree, including axis-parallel, oblique, and complete decision trees – the last one offering very good performances. For presentation purposes, we assume the case (a), in order to disentangle the properties of REASONX in detail under no gap in fidelity between the surrogate model and the black box that should be explained.

The base model is translated into a set of Prolog facts, one for each path in the decision tree. In fact, a split in a decision tree is a linear constraint over features \mathbf{x} of the form $c^T \mathbf{x} \geq b$ or $c^T \mathbf{x} < b$. A path from the root to a leaf is then a conjunction of linear constraints, represented by a Prolog fact such as:

$$path(m, [\mathbf{x}], [c_1^T \mathbf{x} \geq b_1, \dots, c_k^T \mathbf{x} \geq b_k], c, p).$$

where m is an id of the path, $[\mathbf{x}]$ the list of features, c the class predicted at the leaf, p the confidence of the prediction, and $[c_1^T \mathbf{x} \geq b_1, \dots, c_k^T \mathbf{x} \geq b_k]$ the list of k splits from the root to the leaf. Such linear constraints can be combined with constraints on data types (modeling for instance one-hot encoded features), on distance functions (modeling norms⁵ to be used in computing contrastive examples), and user-provided ones (the background knowledge). Further, a constraint φ can be reasoned about in the following forms:

- checking satisfiability, i.e., whether $\exists_{\mathbf{x}} \varphi$ holds, also considering some features from \mathbf{x} over the domain of integers, e.g., the one-hot encoded features or ordinal features;
- projecting over some features $\mathbf{w} \subseteq \mathbf{x}$, i.e., computing $\exists_{\mathbf{x} \setminus \mathbf{w}} \varphi$, as a way to express φ w.r.t. features in \mathbf{w} only;

⁵ REASONX currently implements the L1 norm.

- checking entailment, i.e., whether they entail some other linear constraint, e.g., in order to test if solutions of φ satisfy some property;
- solving a MILP minimization problem $\min f(\mathbf{x})$ s.t. φ where $f(\cdot)$ is a linear function, e.g., in the distance minimization when computing contrastive instances.

The reasonings above are implemented through the $\text{CLP}(R)$ functionalities. However, to allow for wider accessibility of REASONX, we build a Python layer on top of the $\text{CLP}(R)$ core, which translates the base model and the user queries into Prolog queries, and translates back the results into Python data structures. Python also adds some syntactic sugar to express more concisely a few categories of constraints, such as the immutability of features, or to generate constraints from the features of an instance.

4 Synthetic Data Set Illustration

To illustrate the capabilities of REASONX, we introduce a simple synthetic data set, comprised of two classes with each 1,000 data instances. These were sampled from different random normal distributions over two independent features. The distributions were generated to be almost separable by an axis-parallel decision tree. We choose a data instance from class 0, called the *factual instance* (“factual” in the figures), to illustrate a few of the functionalities of REASONX.

As a first operation, we use REASONX to provide us with the factual rule (plotted as factual region in Fig. 1, left) that is satisfied by the data instance. Next, we ask for the contrastive rules, i.e., admissible contrastive regions where potential contrastive examples are located (plotted as contrastive regions in Fig. 1, right). In this example, three of those regions exist.

In Fig. 2 we demonstrate the capability of REASONX to account for background knowledge. We use a constraint that ensures that feature 2 stays constant between the factual instance and the contrastive example (Fig. 2, left), or a constraint that ensures that instead feature 1 stays constant (Fig. 2, right). While the first leads to an admissible contrastive region in the form of a region (a line), the second constraint suppresses any solution (no solution exists).

Now, we extend the example along two dimensions: asking for the *closest* contrastive example under a specified constraint setting, and admitting under-specified information in the factual instance. In Fig. 3, left, we run REASONX under a linear constraint between the two features (feature 1 and feature 2 have to be equal to each other). Instead of solving for admissible contrastive regions, we ask for the *closest* CE. The solution is provided as red dots. We observe the following: first, all solutions lie on the line that marks the linear constraint as introduced. Second, REASONX provides not only one but three solutions with a different distance to the factual instance. This stems from the fact that in this example, three admissible contrastive regions (see also Fig. 1, right) are given.

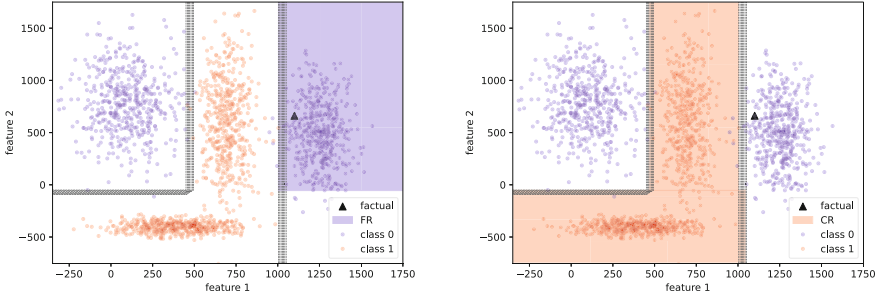


Fig. 1. Left: factual region (FR) as provided by REASONX. Right: contrastive regions (CR) as provided by REASONX. Grey lines refer to the decision boundary of the tree.

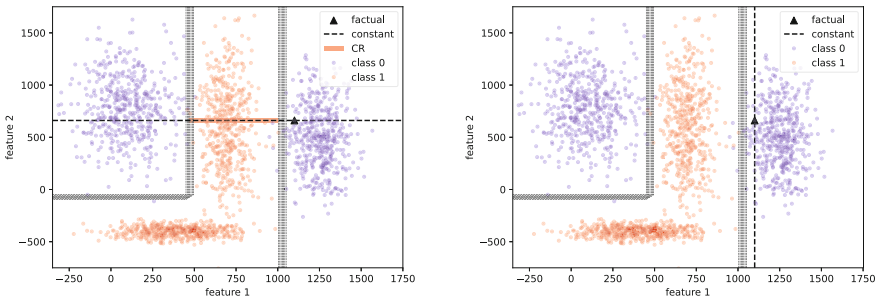


Fig. 2. Left: contrastive region (CR) as provided by REASONX, and given a constraint on feature 2. Right: given the constraint on feature 1, no solution provided by REASONX. The dashed line refers to the enforced constraint. Grey lines refer to the decision boundary of the tree.

REASONX solves the optimization for each of those independently, and provides us therefore not with the global optimum (one solution), but three local optima. This can be an advantage over conventional tools that provide contrastive explanations, as it leaves more flexibility to the end-user. A further refinement of the results, e.g. filtering for the global optimum, can be implemented.

In Fig. 3, right, we re-initialize the factual instance and allow feature 2 to be under-specified, i.e., setting it to a region instead of a fixed value. We omit all constraints, and solve again for the *closest* CE. We display one solution of this query in Fig. 3, right. For the same reasons as above, there exist three solutions (other two solutions in Fig. 4). We observe that the provided solution is also a region in the data space.

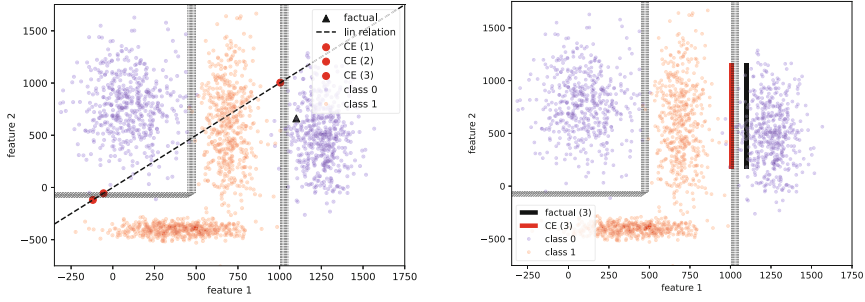


Fig. 3. Left: closest CEs as provided by REASONX, given a constraint on feature 1 and feature 2. The dashed line refers to the enforced constraint. Right: one closest CE as provided by REASONX, feature 2 of factual is provided as region (solution 1). Grey lines refer to the decision boundary of the tree.

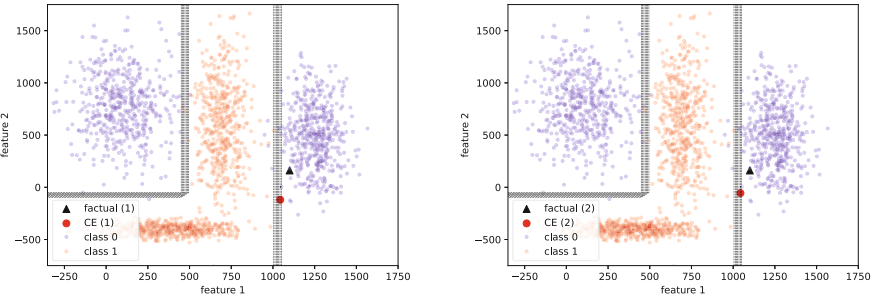


Fig. 4. Closest CE as provided by REASONX, feature 2 of factual is provided as region (solution 2 and 3). Grey lines refer to the decision boundary of the tree.

5 Credit Application Example

This example focuses on contrastive explanations for an ADM system that assesses credit applications, for example, provided by a bank. The main *purpose* of the explanation is for a customer (our imagined end-user) to make sense of the decision, and to receive insights about possible opportunities of action. Secondary purposes are legal compliance of the ADM, as well as increasing the trust of its customers in internal decisions, both in the interest of the bank. Next to demonstrating the capabilities of REASONX, this example serves as a (qualitative) *validation* of the tool. This is justified by the strong qualitative nature of REASONX, and ties closely to the fact that an explanation tool always has to be discussed in context [32].

We run the example on a processed version of the Adult Income data set⁶, which has a binary response variable. It contains twelve features, and we restrict ourselves to a subset: three categorical features (*workclass*, *race*, *sex*), one

⁶ <https://archive.ics.uci.edu/ml/datasets/Adult>.

ordinal (`education`) and four continuous (`age`, `capitalgain`, `capitalloss`, `hoursperweek`). Categorical features are one-hot encoded. The queries as displayed below work iteratively, i.e., background knowledge that is added to the program is always kept (but can be retracted). The `verbose` parameter determines how the output of the query is displayed. The `project` parameter is used to map results of the set of variables onto a subset.

After initializing REASONX with the base decision tree trained on a subset of available data, we turn towards a first question of the user (corresponding to the factual instance): *Why was my application rejected?* We answer it by showing the rule behind the decision. This requires naming an instance ('F' in the code below, for 'factual') and passing the feature values (`InstFeatures`) and the decision of the base model (`InstDecision`).

```

USER:    r.instance('F', features=InstFeatures, label=InstDecision)
         r.solveopt(verbose=2)
REASONX: Rule satisfied by F:
         IF F.capitalgain<=5119.0,F.education<=12.5,F.age<=30.5
         THEN <=50K [0.9652]

```

The rule as returned by REASONX explains the classification of the factual instance. It refers to a specific region in the data input space as characterized by the base decision tree. A second answer can be given by comparing the factual instance against a contrastive rule, using the differences as an explanation. This requires naming a second instance ('CE' in the code below, for 'contrastive example'), and possibly a minimum confidence of the rule leading to the contrastive decision. We obtain two rules by running the following query:

```

USER:    r.instance('CE', label=1-InstDecision, minconf=0.8)
         r.solveopt(verbose=2)
REASONX: Rule satisfied by CE:
         IF CE.capitalgain>5119.0,CE.capitalgain<=5316.5
         THEN >50K [1.0],
         Rule satisfied by CE:
         IF CE.capitalgain>7055.5,CE.age>20.0
         THEN >50K [0.9882]

```

By comparing this output with the answer to the previous question, the user can understand the factual decision of the ADM better. This is especially relevant to answer a second question: *What are my options to change the outcome of the ADM, and receive the credit?* For example, comparing the first contrastive with the factual rule, an increase of the feature `capitalgain` will lead to a change in the predicted outcome of the ADM. Similarly, an increase in the `capitalgain`, and a change in the `age` (from 19 to 20 or higher) will alter it.

In the next step, we add some background knowledge to the explanations. We apply an immutability constraint on the feature `age`:

```

USER:    r.constraint('CE.age = F.age')
         r.solveopt(verbose=2)
REASONX: Rule satisfied by CE:
         IF CE.capitalgain>5119.0,CE.capitalgain<=5316.5
         THEN >50K [1.0]

```

As expected, the solution has changed: by adding the above-stated constraint, the admissible region for CEs becomes smaller, and only one contrastive rule remains. Last, we ask for the CE that is *closest* to the factual instance:

```

USER:      r.solveopt(minimize='l1norm(F, CE)', project=['CE'],
              verbose = 2)
REASONX: Answer constraint:
          CE.race=Black, CE.sex=Male,
          CE.workclass=Private,
          CE.education=10.0,
          CE.age=19.0,
          CE.capitalgain=5119.0,
          CE.capitalloss=0.0,
          CE.hoursperweek=40.0

```

REASONX returned the *closest* CE. This corresponds to the most common notion of a CE in the literature. However, REASONX also took care of the specified background knowledge - the CE is returned under the above specified constraints on the feature `age`. This is a similar case as described in Sect. 4, Fig. 3, left. We extend the example to account for under-specified information. E.g., it can be interesting to ask for the *closest* CE in the case the feature `age` is not fixed, but restricted to 19 years or lower.

```

USER:      r.retract('F.age=19.0')
          r.constraint('F.age<=19.0')
          r.solveopt(minimize='l1norm(F, CE)', project=['CE', 'F.age'],
              verbose = 2)
REASONX: Answer constraint:
          CE.race=Black, CE.sex=Male,
          CE.workclass=Private,
          CE.education=10.0,
          CE.age=F.age,
          CE.capitalgain=5119.0,
          CE.capitalloss=0.0,
          CE.hoursperweek=40.0

```

The returned solution of REASONX is similar to the previous one, but we observe that also in the CE, the feature `age` is not fixed to a single value but is provided as an equality region. This is a nice demonstration of how both in the input and the output, REASONX does work under under-specified information. For a similar example based on the synthetic data set, see Sect. 4, Fig. 3, right.

Since queries work iteratively, the flow of the above corresponds exactly to how an *interaction* with REASONX could look like, forming what we call an explanation (dialogue). Repeated specification of background knowledge and querying are a central part of this dialogue between the end-user and REASONX, allowing to build individual, personalized explanations.

6 Limitations

Evaluation So far, we demonstrated the capabilities of REASONX via a synthetic illustration and an example in the credit domain, and - by surveying relevant

literature - we showed how REASONX is novel against other XAI tools. The most prominent difference between REASONX and other XAI methods is that it relies on symbolic reasoning, based on CLP capabilities, to generate (contrastive) explanations. This is exactly what makes it possible for REASONX to combine a set of useful properties: background knowledge integration, interactivity, and operation under under-specified knowledge. However, a thorough evaluation of REASONX is currently *missing* from this paper. While there is yet no fixed consensus on how to evaluate an explanation, in the case of REASONX, a qualitative and quantitative evaluation is especially tricky: no method exists so far for explanation tools utilizing CLP. Developing a proposal of a set of suitable methods is therefore a priority for future work. We also point out the possibility of user studies for evaluation [36], or validation via real-world data [32].

Technical. We demonstrated the capabilities of REASONX on (nominal, ordinal, and continuous) tabular data, and for a binary decision problem. While an extension to multi-class problems is straightforward, the extension to unstructured data, such as text and images, needs to be strictly formalized. In the domain of images, a solution can be the integration of concepts, as demonstrated by [14].

Social. We discussed the delivery of explanations to enable the end-user to understand the decision better. Another reason to compute explanations is to understand algorithmic harms such as bias. An important question is: *Is the protected attribute (e.g., gender, race, age [15]) determining the decision outcome?* The answer can be given by e.g., comparing a factual instance and the CE, with a focus on whether only protected attributes alter the decision outcome.

Another important point to consider is the fairness of explanations. For two popular model-agnostic, local methods it has been shown that explanations can be unfair: different fidelity values can apply for different subgroups [3].

Last, we would like to discuss two basic assumptions of REASONX that *must* be taken care of in social application contexts. We do not explicitly talk about recommendations based on contrastive explanations but must consider this as a secondary application, as discussed under the keyword of *algorithmic recourse* [23]. First, we assume that our model is *stable over time*, making it possible not only to obtain several contrastive explanations but proposing that the CE suggests a change that indeed alters the outcome of the ML model. While this might hold in a toy setting, it likely does not in practice and can create some wrong promises [4, 45]. Second, by querying the *closest* CE, there is often the implicit assumption that the closer such a CE, the easier the proposed change. This does not necessarily hold in practice, as pointed out already [23]. While an extension of REASONX to account for a variety of norms in the optimization is possible and part of future work, we acknowledge the importance of the choice of a suitable distance function, depending on the context of the application.

7 Conclusion

We presented REASONX, an explanation tool that interactively provides contrastive explanations that can account for background knowledge in the form of constraints, and that works under-specified information. Background knowledge has the potential to significantly improve the quality of an explanation, while interactivity is an important aspect of explanations to be communicated to the end-user. Further, our tool works on under-specified information (profiles) which leads to increased flexibility. REASONX provides explanations for decision trees, which can be the ML models under analysis, or global/local surrogate models of any ML model. We demonstrated REASONX both on a synthetic data set, and an example in the credit domain.

We aim at extending REASONX along several directions: *i*) the implementation of more constraints, *ii*) an extension to account for more expressive trees such as oblique/optimal decision trees, *iii*) the application to local explanations. Implementing other constraints (e.g., diversity of a set of CEs, or sparsity of changes from the factual instance to the CE) will lead to a more flexible tool, and possibly better explanations, depending on the context that will necessitate these constraints. Relying on oblique/optimal decision trees instead of standard decision trees that use axis-parallel splits will allow to better fit the model onto the data, for higher accuracies, and therefore improved quality of the explanations, while not departing from reasoning over linear constraints. Last, REASONX can be easily adapted as a local explanation tool. The necessary changes include learning the tree not over the input data but a local neighborhood, which can, for example, be generated using random perturbations as in LIME [35].

We would also like to refer to the option to extend REASONX in a direction that allows a tighter integration between symbolic and sub-symbolic approaches. One possibility is an extension that allows REASONX to extract the rules which are fundamental to REASONX not via a tree, but directly from an underlying sub-symbolic structure.

Acknowledgments. Work supported by the European Union’s Horizon 2020 research and innovation programme under Marie Skłodowska-Curie Actions for the project NoBIAS (g.a. No. 860630), and under the Excellent Science European Research Council (ERC) programme for the XAI project (g.a. No. 834756). This work reflects only the authors’ views and the European Research Executive Agency (REA) is not responsible for any use that may be made of the information it contains.

References

1. Apt, K.: From Logic Programming to Prolog. Prentice Hall, London New York (1997)
2. Artelt, A., Vrachimis, S.G., Eliades, D.G., Polycarpou, M.M., Hammer, B.: One explanation to rule them all - ensemble consistent explanations. CoRR abs/2205.08974 (2022)

3. Balagopalan, A., Zhang, H., Hamidieh, K., Hartvigsen, T., Rudzicz, F., Ghassemi, M.: The road to explainability is paved with bias: measuring the fairness of explanations. In: FAccT, pp. 1194–1206. ACM (2022)
4. Barocas, S., Selbst, A.D., Raghavan, M.: The hidden assumptions behind counterfactual explanations and principal reasons. In: FAT*, pp. 80–89. ACM (2020)
5. Beckh, K., et al.: Explainable machine learning with prior knowledge: an overview. CoRR abs/2105.10172 (2021)
6. Bertossi, L.E.: Declarative approaches to counterfactual explanations for classification. CoRR abs/2011.07423 (2020)
7. Byrne, R.M.J.: Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: IJCAI, pp. 6276–6282. ijcai.org (2019)
8. Calegari, R., Ciatto, G., Omicini, A.: On the integration of symbolic and sub-symbolic techniques for XAI: a survey. *Intelligenza Artificiale* **14**(1), 7–32 (2020)
9. Chang, C., Creager, E., Goldenberg, A., Duvenaud, D.: Explaining image classifiers by counterfactual generation. In: ICLR (Poster). OpenReview.net (2019)
10. Clocksin, W.F., Mellish, C.S.: *Programming in Prolog. Using the ISO Standard*. Springer, Heidelberg (2003). <https://doi.org/10.1007/978-3-642-55481-0>
11. Confalonieri, R., Coba, L., Wagner, B., Besold, T.R.: A historical perspective of explainable artificial intelligence. *WIREs Data Mining Knowl. Discov.* **11**(1), e1391 (2021)
12. Cropper, A., Dumancic, S.: Inductive logic programming at 30: a new introduction. *J. Artif. Intell. Res.* **74**, 765–850 (2022)
13. Cui, Z., Chen, W., He, Y., Chen, Y.: Optimal action extraction for random forests and boosted trees. In: KDD, pp. 179–188. ACM (2015)
14. Donadello, I., Dragoni, M.: Sexai: introducing concepts into black boxes for explainable artificial intelligence. In: XAI.it@AI*IA. CEUR Workshop Proceedings, vol. 2742, pp. 41–54. CEUR-WS.org (2020)
15. European Union: Charter of Fundamental Rights of the European Union (2007)
16. European Commission: Proposal for a regulation of the European Parliament and the Council laying down harmonized rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts (2021)
17. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery* (2022). <https://doi.org/10.1007/s10618-022-00831-6>
18. Guidotti, R., Monreale, A., Giannotti, F., Pedreschi, D., Ruggieri, S., Turini, F.: Factual and counterfactual explanations for black box decision making. *IEEE Intell. Syst.* **34**(6), 14–23 (2019)
19. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. *ACM Comput. Surv.* **51**(5), 93:1–93:42 (2019)
20. Jaffar, J., Michaylov, S., Stuckey, P.J., Yap, R.H.C.: The CLP(R) language and system. *ACM Trans. Program. Lang. Syst.* **14**(3), 339–395 (1992)
21. Kanamori, K., Takagi, T., Kobayashi, K., Arimura, H.: DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In: IJCAI, pp. 2855–2862. ijcai.org (2020)
22. Karimi, A., Barthe, G., Balle, B., Valera, I.: Model-agnostic counterfactual explanations for consequential decisions. In: AISTATS. Proceedings of Machine Learning Research, vol. 108, pp. 895–905. PMLR (2020)
23. Karimi, A., Barthe, G., Schölkopf, B., Valera, I.: A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. CoRR abs/2010.04050 (2020)

24. Karimi, A., Schölkopf, B., Valera, I.: Algorithmic recourse: from counterfactual explanations to interventions. In: FAccT, pp. 353–362. ACM (2021)
25. Keane, M.T., Kenny, E.M., Delaney, E., Smyth, B.: If only we had better counterfactual explanations: five key deficits to rectify in the evaluation of counterfactual XAI techniques. In: IJCAI, pp. 4466–4474. ijcai.org (2021)
26. Körner, P., et al.: Fifty years of Prolog and beyond. *Theory Pract. Log. Program.* **22**(6), 776–858 (2022)
27. Lakkaraju, H., Slack, D., Chen, Y., Tan, C., Singh, S.: Rethinking explainability as a dialogue: a practitioner’s perspective. *CoRR abs/2202.01875* (2022)
28. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. *Artif. Intell.* **267**, 1–38 (2019)
29. Miller, T., Howe, P., Sonenberg, L.: Explainable AI: beware of inmates running the asylum or: how I learnt to stop worrying and love the social and behavioural sciences. *CoRR abs/1712.00547* (2017)
30. Mittelstadt, B.D., Russell, C., Wachter, S.: Explaining explanations in AI. In: FAT, pp. 279–288. ACM (2019)
31. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: FAT*, pp. 607–617. ACM (2020)
32. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Interpretable machine learning: definitions, methods, and applications. *CoRR abs/1901.04592* (2019)
33. Panigutti, C., Perotti, A., Pedreschi, D.: Doctor XAI: an ontology-based approach to black-box sequential data classification explanations. In: FAT*, pp. 629–639. ACM (2020)
34. Rawal, K., Lakkaraju, H.: Beyond individualized recourse: interpretable and interactive summaries of actionable recourses. In: *NeurIPS* (2020)
35. Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should I trust you?”: explaining the predictions of any classifier. In: *KDD*, pp. 1135–1144. ACM (2016)
36. Rong, Y., et al.: Towards human-centered explainable AI: user studies for model explanations. *CoRR abs/2210.11584* (2022)
37. Ruggieri, S., Pedreschi, D., Turini, F.: Data mining for discrimination discovery. *ACM Trans. Knowl. Discov. Data* **4**(2), 9:1–9:40 (2010)
38. Russell, C.: Efficient search for diverse coherent explanations. In: FAT, pp. 20–28. ACM (2019)
39. Russell, S.J., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 2nd edn. Pearson Education, London (2003)
40. Setzu, M., Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., Giannotti, F.: GLocalX - from local to global explanations of black box AI models. *Artif. Intell.* **294**, 103457 (2021)
41. Sokol, K.: Towards intelligible and robust surrogate explainers: a decision tree perspective. Ph.D. thesis, School of Computer Science, Electrical and Electronic Engineering, and Engineering Maths, University of Bristol (2021)
42. Sokol, K., Flach, P.A.: Glass-box: explaining AI decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In: IJCAI, pp. 5868–5870. ijcai.org (2018)
43. Sokol, K., Flach, P.A.: LIMETree: interactively customisable explanations based on local surrogate multi-output regression trees. *CoRR abs/2005.01427* (2020)
44. Sokol, K., Flach, P.A.: One explanation does not fit all. *Künstliche Intell.* **34**(2), 235–250 (2020)
45. State, L.: Logic programming for XAI: a technical perspective. In: *ICLP Workshops*. CEUR Workshop Proceedings, vol. 2970. CEUR-WS.org (2021)

46. Stepin, I., Alonso, J.M., Catalá, A., Pereira-Fariña, M.: A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. *IEEE Access* **9**, 11974–12001 (2021)
47. Takemura, A., Inoue, K.: Generating explainable rule sets from tree-ensemble learning methods by answer set programming. In: *ICLP Technical Communications. EPTCS*, vol. 345, pp. 127–140 (2021)
48. Ustun, B., Spangher, A., Liu, Y.: Actionable recourse in linear classification. In: *FAT*, pp. 10–19. ACM (2019)
49. Wachter, S., et al.: Counterfactual explanations without opening the black box. *Harv. JL Tech.* **31**, 841 (2017)
50. Warren, G., Keane, M.T., Guéret, C., Delaney, E.: Explaining groups of instances counterfactually for XAI: a use case, algorithm and user study for group-counterfactuals. *CoRR* abs/2303.09297 (2023)
51. Weld, D.S., Bansal, G.: The challenge of crafting intelligible intelligence. *Commun. ACM* **62**(6), 70–79 (2019)
52. Wielemaker, J., Schrijvers, T., Triska, M., Lager, T.: SWI-Prolog. *Theory Pract. Log. Program.* **12**(1–2), 67–96 (2012)
53. Wu, T., Ribeiro, M.T., Heer, J., Weld, D.S.: Polyjuice: generating counterfactuals for explaining, evaluating, and improving models. In: *ACL/IJCNLP* (1), pp. 6707–6723. Association for Computational Linguistics (2021)