Logic Programming for XAI: A technical perspective[1]

Laura State[1,2]

[1]Department of Computer Science, University of Pisa, Largo B. Pontecorvo, 3, 56127 Pisa, Italy
[2]Scuola Normale Superiore, Piazza dei Cavalieri, 7, 56126 Pisa, Italy

Abstract

Our world is increasingly shaped by Artificial Intelligence systems, from search engines over automated hiring algorithms to self-driving cars. Being also used in high-stake decisions, their impact on the life of individuals is huge. Thus it becomes exceedingly important to sceptically review their limitations. One alarming problem is their uptake and reinforcement of existing social biases, as found in many different domains (criminal justice, facial recognition, credit scoring etc). It is complemented by the inherent opaqueness of the most accurate AI systems, making it impossible to understand details of their internal workings. The field of Explainable Artificial Intelligence is trying to address these problems. However, there are several challenges in the field, and we will start this work by pointing them out. We put forward a set of technical pathways, drawing from Logic Programming. Specifically, we propose using Constraint Logic Programming to construct explanations that incorporate prior knowledge, as well as Meta-Reasoning to track model and explanation changes over time.

1. Introduction

Artificial Intelligence (AI) systems have a huge impact on our lives. As much as they positively shape the world, for example by supporting scientific discoveries, they bring responsibility, specifically when applied to social data. As shown in many application contexts, they are susceptible to social biases. Even more, they have the potential to increase and systematise the harm done to already marginalised societal groups.
An example is commercial facial recognition systems, found to be negatively biased against darker-skinned females (error rate up to 34.7% compared to 0.8% for lighter-skinned males) [1]. Next to important questions about bias and ethical values of AI systems, we need to discuss their accountability, as the tragic case of the Uber car overrunning a pedestrian suggests [2].
One main challenge we thereby face is the opaqueness of these systems. As their internal logic is not understandable to humans, they are considered Black Boxes.
This work is putting forward a proposal on how to construct explanations for Black Boxes, which help us to address above questions. It thereby builds upon the field of *Explainable Artificial Intelligence* (XAI) or equivalently *Interpretable Machine Learning*, and *Logic Programming*.
After surveying the field of XAI, we pose the following challenges, which are forming the base of the proposal:

1. A canonical definition of an explanation and its desiderata is missing
2. Prior knowledge is not exploited in the explanation process
3. Explanations do not account for time-evolving models
4. Explanations are not sufficiently evaluated, nor specific to the end user

We will cover each of the points in sec. 3. But first, a quick overview on current approaches in XAI is given, as well as an introduction to Counterfactual Explanations and Logic Programming (sec. 2). The technical proposal, focusing on challenge 2 and 3, is put forward in sec. 4.

Why logic? Technical aspects of the posed questions can be addressed by Logic. As such, we present an application scenario for Logic Programming. It is an inherently interpretable and verifiable approach, and can easily incorporate prior knowledge. It also supports meta-reasoning, and reasoning under constraints [3] [4] [5].

Running Example A loan application scenario, including an applicant with specific features such as age, and the loan amount asked for. The algorithmic decision reduces to a binary classification problem (grant/deny loan). It is a relevant, real-world example [6] [7] [8].

2. Related work

2.1. Explainable Artificial Intelligence

The field of XAI can be described along 3 main dimensions [9] [10]. First, we distinguish between constructing *Transparent/White Box* (WB) models, and (post-hoc) explanations for *Black Boxes* (BB). WB are based on inherently interpretable approaches such as linear models, decision trees, or rule lists/sets. BB describe a wide set of models, all of them being not interpretable, or not accessible. Second, explanations for BB models are either

[1]Original Contribution.

1st Workshop on Machine Ethics and Explainability - The Role of Logic Programming, September 2021, Virtual Only
laura.state@di.unipi.it (L. State)

local or *global*. Local approaches focus on single data points, e.g. LIME [11], global on the level of the whole model by fitting an interpretable surrogate, e.g. TREPAN [12]. An approach that bridges the gap is GLocalX [8]. Third, explanations can be *model-agnostic* (applicable to any model) or *model-specific* (applicable to certain models only). E.g. LIME is model-agnostic, TREEPAN in its original form model-specific.
This concept focuses on local (first), model-agnostic explanation methods for Black Boxes.

2.2. Counterfactuals

Counterfactuals (CF) give an answer to "what-if" questions. Conceptually, they highlight the smallest feature changes that are necessary to alter the (undesired) prediction. A common approach to generate CF is the following optimisation problem. Denoting the CF by x', the original (factual) data point by x, the prediction of the CF as $f_w(x')$ and the new (desired) prediction as y', it reads

$$arg\min_{x}\max_{\lambda}\lambda(f_w(x')-y')^2+d(x,x') \tag{1}$$

λ denotes a tuning parameter. The distance $d(.,.)$ needs to be chosen carefully, a standard choice is the Manhattan Distance weighted by the Median Absolute Deviation. The approach was first put forward in [6].

2.3. Logic Programming

Relevant approaches are: *Constraint Logic Programming* and *Meta-Reasoning*. Both build on Logic Programs (LP), bringing forward the following advantages as declarative paradigms: a strict separation of knowledge and inference, no encoded directionalities, as well as interpretability and verifiability [3] [4] [5].

Constraint Logic Programming (CLP) Augmenting LP by constraints to solve optimisation problems. They are classified based on constraint type: (non-) linear, by the number of involved variables (arity), by preference, or by domain: e.g. integer on finite domains [4] [5].

Meta-Reasoning Enables reasoning over LP, and integration of knowledge. Standard Meta-Reasoning approaches such as meta-interpreters [5] can be augmented by theories and operators over the programs [13].

3. Main Challenges

3.1. Defining Explanations

A canonical definition of explanations and its desiderata in the domain of XAI is missing, see also [14] [15]. We don't attempt to do so, but rather want to point to the following facets:

Explanation or Interpretation? Explanations are closely related to interpretability but a distinct concept: interpretability is a (passive) property, whereas an explanation, or explainability is about an interaction, or an exchange of information. For interpretability, the following definition is adopted "the ability to explain or to present in understandable terms *to a human*" [2] [16].

Explanation to whom? The receiver of an explanation is a specific person, in most cases a *lay user* [14]. What constitutes an explanation that manages to transport its content well, can be learnt from literature in the Social Sciences. Main points are that explanations should be social/interactive, contrastive, selective, and that probabilities do not matter much [17] [18].

Explanation of what? Not only does the audience of the explanation matter, but also its specific purpose, defining further form and content. Purposes can be loosely grouped into the following: moral/ethical [19] [20], [21], including safety concerns [16] and efforts to increase trust in the user [11], legal motivations [6], or understanding/debugging [20] [22] [16].

3.2. Prior Knowledge

Prior knowledge is rarely incorporated into the explanation process. An example where this matters is the generation of CF, including some advice to change the (unfavourable) outcome. Prior knowledge that needs to be considered can be a real-world constraint, or a user-based preference. Exemplary approaches put forward are [23] [24]. The term *real-world constraints* is used in our work to refer to the subset of constraints that encode knowledge about the world.
These considerations are also relevant in our loan application scenario. Consider the following [3]

`Applicant`	$age = 45 \wedge job = none \wedge amount = 10k$
`Factual`	$age > 40 \wedge job = none \wedge amount > 5k$
`CF`	$age \leq 40 \wedge job = none \wedge amount \leq 5k$

The CF suggests to the applicant to decrease the age and the loan amount, while the feature "job" stays constant. While decreasing the age is invalidating a real-world constraint, decreasing the loan amount could also be impossible for the applicant, depending on the intended use of the loan (user-based constraint).

3.3. Changing Models

Explanations do not take into account that a model (and thus the explanation) can change over time. However,

[2] Emphasis added.
[3] Adapted from LORE [25]. The form is the same, content is changed. LORE provides each a local factual and a counterfactual rule as explanation.

this is not realistic in deployment. The model can change if the training data distribution shifts, e.g. by new incoming data points, and retraining, or by adjustments at the model itself, e.g. following a new regulation. Explanations need, therefore, to be extended by a time dimension. In the loan application scenario, it would be important to see which parts of an explanation change over time, specifically if we want to provide counterfactuals as actionable advice. As pointed out by [26], the danger is that although recommendations are followed, the model does not alter its prediction because it changed as well.

3.4. Customisation + Evaluation

Customisation Explanations are, in their final version, user-specific and thus need to be adapted to the audience [14]. This goes together with the specific purpose of the explanation (see 3.1).
Consider the loan application scenario. The bank clerk who is in charge of communicating the loan decision needs similar information to the applicant. For example, both might be interested in advice on how to change an undesired outcome (purpose: moral/ethical, legal). However, the manager of the bank is not interested in receiving explanations in such details, but rather in summary statistics, or general information about how decisions were derived (purpose: understanding). Still, explanations could be generated under the same framework, e.g. local explanations by LORE [25], an aggregation by GLocalX [8], building upon the former.

Evaluation Alarmingly little has been done in this field yet: considering the case of CF explanations, a recent survey found that only 21% of the approaches are validated with human subject experiments [27]. However, the call for evaluations based on this type of experiments is not new, and holds for the whole field [16]. If computational evaluation is preferred, attention needs to be drawn to carefully validate the proxy variables that are used to simulate human behaviour [27].
In our loan application scenario, an ideal evaluation would involve human subject experiments with the applicant, the bank clerk, and the manager separately.

4. The (technical) path forward

In this section, we put forward concrete ideas to address two of the above challenges (3.2 and 3.3). It is out of scope of this proposal to answer to the full set of challenges.

4.1. Prior Knowledge

To compute CF that incorporate prior knowledge, we revisit CLP. The CF generation is encoded as an optimisation problem over the *distance* between the factual and the CF [4], subject to the following constraints [5]:

- the prediction is *opposite* to the factual prediction (binary decision problem)
- restricting the domain/range
- restricting feasibility (immutable/actionable), including encoding relations/monotonicity
- enforcing diversity/sparsity

Whereas the first constraint is absolutely necessary, others depend on the use-case of the CF. When focusing on understanding/debugging the model, no other constraints, or a restricted set (e.g. domain/range) suffices. This also holds, if we want to learn about bias in the decision pipeline. If we are rather interested in actionable advice, the full set of the posed constraints can be used. Real-world constraints are mandatory in these cases, others depend on the person that is subject to the decision. The first constraint is mandatory, but hardest to encode, as we cannot call the BB from within the logic program. A possible solution connects our idea to LORE [25]. This method provides local explanations as of logic rules. The rules are read from a decision tree, which is grown on the local neighbourhood around the instance that is in focus. Using the split criteria put forward by the tree, we can construct regions which hold possible CF, and use these as inputs for the optimisation problem.
For other types of constraints, we provide exemplary implementations, based on Prolog/Eclipse [28] [29]. A CF feature is denoted by subscript CF, the original by F.

Range Constraints Restricting the numerical range in one or both directions, by absolute numbers or relative to a constant (line 1/2). Another option (line 3) allows the variable to take only very specific values. In the loan application scenario, this could be the total loan duration, encoded in months, allowed to change in multiples of three (quarterly).

```
1| age_CF #>= 0.
2| initial_payment_CF #<= 0.2 * loan_amount.
3| (loan_duration_CF mod 3) #= 0.
```

Feasibility Constraints Immutable features (line 1), actionable depending on its previous value such as age in our scenario (line 2) or depending on the change of another feature. In our scenario, this could be the first instalment to be paid, that needs to increase if the total loan amount (now a variable) does (line 3-7).

```
1| birthplace_CF #= birthplace_F.
2| age_CF #>= age_F.
3| dependency(loan_amount_CF, loan_amount_F,
```

[4] Optimal encoding of distance is an open problem. As a start, (a combination of) the L_1, L_2 or L_{inf} norm can be used.
[5] Loosely based on [7].

```
4|   instalment_CF, instalment_F) :-
5|   ((loan_amount_CF - loan_amount_F) #> 0
6|          ->  instalment_CF #> instalment_F
7|          ;   instalment_CF #>= instalment_F).
```

Our approach is inspired by (1) [24] [30] (using SAT/ causal framework) and (2) [23] [31] [32] [33] (using ILP/ MILP). However, there are two main points that distinguishes the approach put forward: first, the focus is clearly on creating explanations for *any* BB. Approaches (1) are generally agnostic, but the model internals need to be known, in (2) only linear or additive models are considered. Second, to the best of our knowledge, this is the first approach using LP to generate CF.

4.2. Changing Models

This problem can be addressed by Meta-Reasoning. The standard meta-interpreter is extended by theories [13]. We introduce the meta-interpreter (line 1-8). Also, we will need the union operator (line 9-14) [6].

```
1|  solve(true, _).
2|  solve((G1,G2), T) :- solve(G1, T), solve(G2, T).
3|  solve(A, T) :- clause_in_th(A, B, T),
4|     solve(B, T).
5|  clause_in_th(A, B, T) :-
6|     clause(A, (theory(T), B)).
7|  clause_in_th(A, true, T) :-
8|     clause(A, theory(T)).
9|  solve(A, union(T1, T2)) :-
10|    clause_in_th(A, B, T1),
11|    solve(B, union(T1, T2)).
12|  solve(A, union(T1, T2)) :-
13|    clause_in_th(A, B, T2),
14|    solve(B, union(T1, T2)).
```

Now, let us look at a simple toy example. We specify two different time points ($t1/t2$), each defined by a set of rules (line 1-15) and facts (line 16-21), mirroring the change of the model/applicant over time.

```
1|  theory(rule_t1).
2|  grant(Person) :- theory(rule_t1),
3|      highincome(Person), savings(Person).
4|  grant(Person) :- theory(rule_t1),
5|      car(Person), savings(Person).
6|  deny(Person) :- theory(rule_t1),
7|      lowincome(Person), savings(Person).
8|  theory(rule_t2).
9|  grant(Person) :- theory(rule_t2),
10|     highincome(Person), savings(Person).
11|  grant(Person) :- theory(rule_t2),
12|     car(Person), highincome(Person).
13|  deny(Person) :- theory(rule_t2),
15|     lowincome(Person), savings(Person).
16|  theory(fact_t1).
17|  savings(applicant) :- theory(fact_t1).
18|  lowincome(applicant) :- theory(fact_t1).
19|  theory(fact_t2).
20|  savings(applicant) :- theory(fact_t2).
21|  highincome(applicant) :- theory(fact_t2).
```

According to the rules and facts at $t1$, the applicant will not receive a loan (deny). The following changes can be advised, based on information available at $t1$, e.g. by generating a CF: to increase the income (line 2/3), *or* to buy a car (line 4/5). Whereas increasing the income will change the prediction at $t2$ (line 9/10), buying a car will not, as we observe a change in this rule (line 11/12). We update the facts at $t2$ according to the first advice and check this outcome by posing the following query, which returns `Person = applicant`.

```
1| cf_condition(Person) :-
2|    solve(deny(Person), union(rule_t1, fact_t1)),
3|    solve(grant(Person), union(rule_t2, fact_t2)).
```

Although we presented only a toy example, that is restricted in applicability, we could demonstrate the importance of reasoning over time. As a possible next step, we propose integrating [25] or [8], and reasoning directly over the extracted BB explanations.

5. Conclusion

This paper presented two concrete ideas that apply LP to address challenges in XAI. Specifically, we proposed a CLP-based approach to generate CF that can incorporate prior knowledge, and a Meta-Reasoning approach to account for changes in models and explanations.
As such, they can be seen as one answer to the GDPR, to provide "meaningful information about the logic involved" [34] to any person under an automated decision.
To summarise, we want to point out two aspects of the field of XAI: first, it is a highly *interdisciplinary* endeavour that will only manage to address the challenges of AI, and its own, by calling to participation scholars from Computer Science, Social Sciences, Law and others. Second, explanations are always *context-dependent*, addressing a specific problem, user group, and purpose. This needs to be considered when they are constructed, used, and evaluated.

Acknowledgments

I want to thank my supervisors Salvatore Ruggieri and Franco Turini for many fruitful discussions and advice. This work is supported by the project "NoBias - Artificial Intelligence without Bias," which has received funding from the European Union's Horizon 2020 research and innovation programme, under the Marie Skłodowska-Curie (Innovative Training Network) grant agreement no. 860630.

[6] Code contributed by F. Turini.

References

[1] J. Buolamwini, T. Gebru, Gender shades: Intersectional accuracy disparities in commercial gender classification, in: FAT, volume 81 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 77–91.

[2] BBC News, Tech, Uber's self-driving operator charged over fatal crash, 2020. URL: https://www.bbc.com/news/technology-54175359.

[3] A. Cropper, S. Dumancic, Inductive logic programming at 30: a new introduction, CoRR abs/2008.07912 (2020).

[4] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 2 ed., Pearson Education, 2003.

[5] K. Apt, From logic programming to Prolog, Prentice Hall, London New York, 1997.

[6] S. Wachter, B. D. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, CoRR abs/1711.00399 (2017).

[7] A. Karimi, G. Barthe, B. Schölkopf, I. Valera, A survey of algorithmic recourse: definitions, formulations, solutions, and prospects, CoRR abs/2010.04050 (2020).

[8] M. Setzu, R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, F. Giannotti, Glocalx - from local to global explanations of black box AI models, Artif. Intell. 294 (2021) 103457.

[9] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2019) 93:1–93:42.

[10] C. Molnar, Interpretable Machine Learning, 2019. https://christophm.github.io/interpretable-ml-book/.

[11] M. T. Ribeiro, S. Singh, C. Guestrin, "why should I trust you?": Explaining the predictions of any classifier, in: KDD, ACM, 2016, pp. 1135–1144.

[12] M. W. Craven, J. W. Shavlik, Extracting tree-structured representations of trained networks, in: NIPS, MIT Press, 1995, pp. 24–30.

[13] A. Brogi, P. Mancarella, D. Pedreschi, F. Turini, Theory construction in computational logic, in: ICLP Workshop on Construction of Logic Programs, Wiley, 1991, pp. 241–250.

[14] B. D. Mittelstadt, C. Russell, S. Wachter, Explaining explanations in AI, in: FAT, ACM, 2019, pp. 279–288.

[15] Z. C. Lipton, The mythos of model interpretability, Commun. ACM 61 (2018) 36–43.

[16] F. Doshi-Velez, B. Kim, Towards a rigorous science of interpretable machine learning, 2017. arXiv:1702.08608.

[17] T. Miller, P. Howe, L. Sonenberg, 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).

[18] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artif. Intell. 267 (2019) 1–38.

[19] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis, I. Kompatsiaris, K. Kinder-Kurlanda, C. Wagner, F. Karimi, M. Fernández, H. Alani, B. Berendt, T. Kruegel, C. Heinze, K. Broelemann, G. Kasneci, T. Tiropanis, S. Staab, Bias in data-driven artificial intelligence systems - an introductory survey, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 10 (2020).

[20] D. Pedreschi, F. Giannotti, R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, Meaningful explanations of black box AI decision systems, in: AAAI, AAAI Press, 2019, pp. 9780–9784.

[21] M. J. Kusner, J. R. Loftus, C. Russell, R. Silva, Counterfactual fairness, in: NIPS, 2017, pp. 4066–4076.

[22] P. Lertvittayakumjorn, L. Specia, F. Toni, FIND: human-in-the-loop debugging deep text classifiers, in: EMNLP (1), Association for Computational Linguistics, 2020, pp. 332–348.

[23] B. Ustun, A. Spangher, Y. Liu, Actionable recourse in linear classification, in: FAT, ACM, 2019, pp. 10–19.

[24] A. Karimi, G. Barthe, B. Balle, I. Valera, Model-agnostic counterfactual explanations for consequential decisions, in: AISTATS, volume 108 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 895–905.

[25] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, CoRR abs/1805.10820 (2018).

[26] S. Barocas, A. D. Selbst, M. Raghavan, The hidden assumptions behind counterfactual explanations and principal reasons, in: FAT*, ACM, 2020, pp. 80–89.

[27] M. T. Keane, E. M. Kenny, E. Delaney, B. Smyth, If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual XAI techniques, CoRR abs/2103.01035 (2021).

[28] SWI Prolog, Website swi-prolog, 2021. URL: https://www.swi-prolog.org/.

[29] Eclipse, Website eclipse, 2021. URL: https://eclipseclp.org/.

[30] A. Karimi, B. Schölkopf, I. Valera, Algorithmic recourse: from counterfactual explanations to interventions, in: FAccT, ACM, 2021, pp. 353–362.

[31] C. Russell, Efficient search for diverse coherent explanations, in: FAT, ACM, 2019, pp. 20–28.

[32] K. Kanamori, T. Takagi, K. Kobayashi, H. Arimura,

DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization, in: IJCAI, ijcai.org, 2020, pp. 2855–2862.
[33] Z. Cui, W. Chen, Y. He, Y. Chen, Optimal action extraction for random forests and boosted trees, in: KDD, ACM, 2015, pp. 179–188.
[34] European Union, General data protection regulation, 2016. URL: https://gdpr.eu/article-15-right-of-access/.