Reasoning about Counterfactuals and Explanations: Problems, Results and Directions

Leopoldo Bertossi

Universidad Adolfo Ibáñez
Faculty of Engineering and Sciences
and
Millennium Inst. for Foundational Research on Data (IMFD)
Santiago, Chile
leopoldo.bertossi@uai.cl

Abstract. There are some recent approaches and results about the use of *answer-set programming* for specifying counterfactual interventions on entities under classification, and reasoning about them. These approaches are flexible and modular in that they allow the seamless addition of domain knowledge. Reasoning is enabled by query answering from the answer-set program. The programs can be used to specify and compute *responsibility-based* numerical scores as attributive explanations for classification results.

1. Introduction. In this short paper we describe at a high level recent research that we have carried out in the area of score-based explanations to outcomes from classification models. We also describe how declarative specifications of- and reasoning with counterfactual explanations leading to score computation are enabled and supported by *answer-set programs* [11]. References are given for the technical details. We also discuss some relevant research directions.

2. Attribution Scores. Counterfactuals are at the very basis of the notion of *actual causality* [17]. They are hypothetical interventions (or changes) on variables of a causal structural model. Counterfactuals can be used to define and assign *responsibility scores* to the variables in the model, with the purpose of quantifying the strength of their causal contribution to a particular outcome [12, 18]. These generals notions of actual causality have been applied in databases, to investigate actual causes and responsibilities for query results [23, 24, 4].

Numerical scores have been applied in *explainable AI*, and most prominently in machine learning models for classification [25]. The general idea is that feature values in entities under classification are given numerical scores, to indicate how relevant they are for the outcome of the classification. For example, one might want to know how important is the city or the neighborhood where a client lives when a bank uses a classification algorithm to accept a loan request or not. This can be done by assigning a number to the feature value, e.g. to "Bronx in New York City". As such, it is a *local explanation*, for the entity at hand, and in relation to all its participating feature values.

A widely used score is Shap [22], which is based on the Shapley value that is used in coalition game theory [26]. It is based on *implicit* counterfactuals and a numerical aggregation of the outcomes from the classification of those different counterfactual versions of the initial entity. Accordingly, the emphasis is not on the possible counterfactuals, but on the final numerical score. However, counterfactuals are interesting *per se*. For example, we might want to know if the client, by changing his/her address, might turn a rejection into acceptance of the loan request. The so generated new entity, with a new address and a new label, is a *counterfactual version* of the original entity.

The x-Resp score was introduced in [8]. It is defined in terms of explicit counterfactuals and responsibility as found in general actual causality. A more general version of it, the Resp score, was introduced in [6], and was compared with other scores, among them, Shap.

3. Reasoning with Counterfactual and ASPs. Taking seriously the idea that counterfactuals are interesting in their own right, *counterfactual intervention programs* (CIPs) were proposed in [8]. They are *answer-set programs* (ASPs) [11, 16] that specify counterfactual versions of an initial entity, reason about them, and compute x-Resp scores for feature values.

Answer-set programming is a flexible and powerful logic programming paradigm that, as such, allows for declarative specifications and reasoning from them. The (non-monotonic) semantics of a program is given in terms of its *stable models*, i.e. special models that make the program true [15]. In our applications, the relevant counterfactual versions correspond to different models of the CIP.

CIPs can be used to specify the relevant counterfactuals (by imposing extra conditions in rule bodies or using *program constraints* that filter models where they are violated), specify "minimum-change" counterfactuals (by using *weak program constraints* that filter models where they are not minimally violated [20]), analyze different versions of them, and use them to specify and compute the x-Resp score. In particular, one can specify and compute maximum-responsibility counterfactuals (through the use of weak program constraints [8, 9].

For our examples with decision-trees [8, 9] and with naive-Bayes classifiers [10], we have used the *DLV* system (and its extensions) [20] that implements the ASP semantics. The classifiers can be specified directly inside the CIP, or can be invoked as external predicates [8]. The latter case is useful when we interact with a *black-box classifier* [27], to which scores such as Shap and x-Resp can be applied.

4. Semantics and Domain Knowledge. CIPs are very flexible in that one can easily add *domain knowledge* or *domain semantics*, in such a way that certain counterfactuals are not considered, or others are privileged. With CIPs, many kinds of changes on the specification that are of potential interest can be easily and seamlessly tried out on-the-fly, for exploration purposes [10, 9]. All these changes and alternatives are much more difficult to implement with a purely procedural approach.

In particular, one can specify domain-dependent *actionable counterfactuals* [8], that, in certain applications, make more sense or may lead to feasible changes of feature values for an entity to reverse a classification result [30, 19].

The definitions of attribution scores explicitly or implicity consider all counterfactual version of the entity under explanation. However, both their definition and computation should be influenced by the domain semantics, which could lead to ignore some counterfactuals or to give more importance to others. This could be done by declaratively specifying which is the case for different counterfactuals. Probabilistic constraints could be declared and imposed, affecting the underlying population, to which counterfactual versions belong (c.f. Section 6. below).

5. Queries and Reasoning. Reasoning is enabled by query answering, for which two semantics are offered. Under the *brave semantics* one obtains as query answers those that hold in *some* model of the CIP. This can be useful to detect if there is minimum-change counterfactual version of the initial entity where the city is changed together with the salary.

Under the *cautious semantics* one obtains answers that hold in all the models of the CIP, which could be used to identify feature values that have to be changed no matter what if we want to reverse the outcome.

As components of a same program, we could, for example, interact at the same time with two different classifiers. It would be easy to compare their classifications and counterfactuals by means of query answering.

Query answering on ASPs offers many opportunities. Actually, there have been some efforts to design and investigate query languages for explanations [29]. ASP offers a query language for this task, and as a part of the same system that does the reasoning and computation [10]. The investigation of its full potential (or shortcomings) for this tasks remains to be carried out. This analysis has to done more on the basis of practical needs than at that of the expressive power of the query language (which has been investigated in the case of ASP [14]).

6. Room for Probabilistic Reasoning. Attribution scores are usually of a probabilistic nature in that they consider a -possibly implicit- distribution on the entity population. This is the case of Shap, Resp; and also the Causal-Effect, used in [28] for tuple-attribution w.r.t. query answering in databases. The distribution is an important element to consider when analyzing the complexity of score computation [1, 31, 6].

The first generation of answer-sets programming, the one that is mostly used, is not probabilistic, and does not provide much support for probabilistic reasoning. With some difficulty, one can do probabilistic reasoning through numerical aggregations (as with naive-Bayes classifiers in [10]).

A probabilistic extension of a logic-based declarative semantics, as is the case of *ProbLog* [13], would be welcome for ASP. Actually, there are probabilistic extensions of the ASP-semantics [2] that could be tried in this direction, and not only for probabilistic classifiers, probabilistic counterfactual reasoning, or probabilistic score computation, but also for exploring semantic changes and conditions that are reflected on modified distributions [8]. The need for systems for probabilistic-ASP reasoning becomes crucial.

7. Contexts and Interpretations. When producing explanations, one should have in mind who is going to receive and analyze them, in particular, those based on attribution scores. The final user, possibly a non-expert in explanation methodologies, has to make sense of them. For this reason, explanations should be conveyed *in terms of the context* of this user. Through this context the user will be in position to *interpret the explanations*. We have argued that formal ontologies are appropriate for describing and specifying contexts [5, 3]. For this purpose, the ASP-based specification and computation of explanations could interact "at a similar logical level" with formal ontologies, e.g. conveying results from the former to the latter. This is a promising research direction. The integration of ASP and ontologies has been considered in general terms (cf. [21] for a discussion and references).

Acknowledgments: Part of this work was funded by ANID - Millennium Science Initiative Program - Code ICN17002.

References

[1] Arenas, M., Barcelo, P., Bertossi, L. and Monet, M. The Tractability of SHAP-scores over Deterministic and Decomposable Boolean Circuits. Proc. AAAI 2021.

[2] Baral, C., Gelfond, M. and Rushton, N. Probabilistic Reasoning with Answer Sets. *Theory and Practice of Logic Programming*, 2009, 9(1):57-144.

[3] Bertossi, L., Rizzolo, F. and Lei, J. Data Quality is Context Dependent. Proc. WS on Enabling Real-Time Business Intelligence (BIRTE 2010). Springer LNBIP 84, 2011, pp. 52-67.

[4] 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.

[5] Bertossi, L. and Milani, M. Ontological Multidimensional Data Models and Contextual Data Quality. *Journal of Data and Information Quality*, 2018, 9(3):14.1-14.36.

[6] Bertossi, L., Li, J., Schleich, M., Suciu, D. and Vagena, Z. Causality-Based Explanation of Classification Outcomes. Proc. Fourth Workshop on Data Management for End-To-End Machine Learning (DEEM@SIGMOD), 2020, pp. 6:1-6:10.

[7] Bertossi, L. An ASP-Based Approach to Counterfactual Explanations for Classification. Proc. RuleML-RR 2020, Springer LNCS 12173, pp. 70-81.

[8] Bertossi, L. Declarative Approaches to Counterfactual Explanations for Classification. arXiv Paper 2011.07423, 2020. Journal submission after revisions. Extended version of [7].

[9] Bertossi, L. Score-Based Explanations in Data Management and Machine Learning: An Answer-Set Programming Approach to Counterfactual Analysis. arXiv Paper 2106.10562. To appear in *Reasoning Web, 2021.*

[10] Bertossi, L. and Reyes, G. Answer-Set Programs for Reasoning about Counterfactual Interventions and Responsibility Scores for Classification. To appear in Proc. 1st International Joint Conference on Learning and Reasoning (IJCLR), 2021. arXiv Paper 2107.10159.

[11] Brewka, G., Eiter, T. and Truszczynski, M. Answer Set Programming at a Glance. *Commun. ACM*, 2011, 54(12):92-103.

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

[13] De Raedt, L. and Kimmig, A. Probabilistic (Logic) Programming Concepts. *Machine Learning*, 2015, 100(1):5-47.

[14] Dantsin, E., Eiter, T., Gottlob, G. and Voronkov, A. Complexity and Expressive Power of Logic Programming. *ACM Computing Surveys*, 2001, 33(3):374-425.

[15] Gelfond, M. and Lifschitz, V. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing*, 1991, 9:365-385.

[16] Gelfond, M. and Kahl, Y. *Knowledge Representation and Reasoning, and the Design of Intelligent Agents*. Cambridge Univ. Press, 2014.

[17] Halpern, J. and Pearl, J. Causes and Explanations: A Structural-Model Approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 2005, 56(4):843-887.

[18] Halpern, J. Y. A Modification of the Halpern-Pearl Definition of Causality. Proc. IJCAI 2015, pp. 3022-3033.

[19] Karimi, A-H., von Kügelgen, B. J., Schölkopf, B. and Valera, I. Algorithmic Recourse under Imperfect Causal Knowledge: A Probabilistic Approach. Proc. NeurIPS, 2020.

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

[21] Lukumbuzya, S., Ortiz, M. and Šimkus, M. Resilient Logic Programs: Answer Set Programs Challenged by Ontologies. Proc. AAAI 2020.

[22] Lundberg, S., Erion, G., Chen, H., DeGrave, A., Prutkin, J., Nair, B., Katz, R., Himmelfarb, J., Bansal, N. and Lee, S.-I. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nature Machine Intelligence*, 2020, 2(1):2522-5839.

[23] 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.

[24] Meliou, A., Gatterbauer, W., Halpern, J.Y., Koch, C., Moore, K. F. and Suciu, D. Causality in Databases. *IEEE Data Engineering Bulletin*, 2010, 33(3):59-67.

[25] Molnar, C. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book, 2020.

[26] Roth, A. E. (ed.) *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[27] Rudin, C. Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead. *Nature Machine Intelligence*, 2019, 1:206-215. Also arXiv:1811.10154,2018.

[28] Salimi, B., Bertossi, L., Suciu, D. and Van den Broeck, G. Quantifying Causal Effects on Query Answering in Databases. Proc. 8th USENIX Workshop on the Theory and Practice of Provenance (TaPP), 2016.

[29] Subercaseaux, B., Perez, J. and Barcelo, P. Foundations of Languages for Interpretability and Bias Detection. AFCI WS at NeurIPS, 2020.

[30] Ustun, B., Spangher, A. and Liu, Y. Actionable Recourse in Linear Classification. Proc. FAT 2019, pp. 10-19.

[31] Van den Broeck, G., Lykov, A., Schleich, M. and Suciu, D. On the Tractability of SHAP Explanations. Proc. AAAI 2021.