Two Knowledge-driven Approaches to Explaining Black-box Models

Roberto Confalonieri[1], Pietro Galliani[1], Oliver Kutz[1], Daniele Porello[2], Guendalina Righetti[1], and Nicolas Troquard[1]

[1] Faculty of Computer Science, Free University of Bozen-Bolzano, Italy
firstname.lastname@unibz.it
[2] Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy
daniele.porello@cnr.it

Abstract. We briefly introduce two novel and promising research lines to explaining black-box models: perceptron (or threshold) connectives in the context of Description Logic, and their possible use to bridge the gap between statistical learning of models from data and logical reasoning over knowledge bases; TREPAN RELOADED, an approach that builds post-hoc explanations of black-box classifiers in the form of decision trees enhanced by domain knowledge. Our aim is to study how these two frameworks interact on a theoretical level, and secondly, to investigate use-cases in ML and AI in a comparative manner, specifically user-studies that help determine human understandability of explanations generated using these two frameworks by human users.

1 Background and Motivation

While interest in explainable Artificial Intelligence had subsided together with that in expert systems after the mid-1980s [16], more recent successes in machine learning technology have brought explainability back into the focus. This has led to a plethora of new approaches for explanations of black-box models [8], aiming to achieve explainability without sacrificing system performance. Only a few of these approaches, however, focus on global explanations, and on how to integrate and use domain knowledge to drive the explanation process (e.g., [13]) or how to measure the understandability of explanations of black-box models (e.g., [14]).

For that reason, an important foundational aspect of explainable AI has remained hitherto mostly unexplored: can the integration of domain knowledge, e.g., as modeled by means of ontologies, help human understandability of explanations?

2 Explanations via Weighted Threshold Operators

Weighted Threshold Operators are n-ary logical operators which compute a weighted sum of their arguments and verify whether it reaches a certain threshold. These operators have been extensively studied in the context of circuit

complexity theory, and they are also known in the neural network community under the alternative name of *perceptrons*. In [12], threshold operators were studied in the context of Knowledge Representation, focusing in particular on Description Logics (DLs). In brief, if $C_1 \dots C_n$ are concept expressions, $w_1 \dots w_n \in \mathbb{R}$ are weights, and $t \in \mathbb{R}$ is a threshold, we can introduce a new concept $\mathbb{W}^t(C_1 : w_1 \dots C_n : w_n)$ to designate those individuals d such that $\sum\{w_i : C_i \text{ applies to } d\} \geq t$.

In the context of DL and concept representation, such threshold expressions are natural and useful, as they provide a simple way to describe the class of the individuals that satisfy "enough" of a certain set of desiderata.

Consider the *Felony Score Sheet* used in the State of Florida[3], in which various aspects of a crime are assigned points, and a threshold must be reached to decide compulsory imprisonment. For example, possession of cocaine corresponds to 16 points if it is the primary offense and to 2.4 points otherwise, a victim injury describable as "moderate" corresponds to 18 points, and a failure to appear for a criminal proceeding results in 4 points. Imprisonment is compulsory if the total is greater than 44 points and not compulsory otherwise. A knowledge base describing the laws of Florida would need to represent this score sheet as part of its definition of its **CompulsoryImprisonment** concept, for instance as

$$\mathbb{W}^{44}(\textbf{CocainePrimary} : 16, \textbf{ModerateInjuries} : 18, \dots).$$

While it would be possible to also describe it (or any other Boolean function) in terms of more ordinary logical connectives (e.g., by a DNF expression), a definition in terms of threshold expressions is far simpler and more readable. As such, the definition is more transparent and more explainable.

We refer the interested reader to [12] and to [6] for a more in-depth analysis of the properties of this operator. Having threshold expressions in a language of knowledge representation has notable advantages. First, in psychology and cognitive science, the combination of two or more concepts has a more subtle semantics than set theoretic operations. As shown in [15], threshold operators can represent complex concepts more faithfully regarding the way in which humans think of them. For this reason, explanations provided using threshold expressions are in principle more accessible to human agents. Second, as illustrated in [6], since a threshold expression is simply a linear classification model, it is possible to use standard linear classification algorithms (such as the Perceptron Algorithm, Logistic Regression, or Linear SVM) to learn its weights and its threshold given a set of assertions about individuals (that is, given an ABox).

Extensions of Description Logic involving threshold operators have also been discussed in [1] and [2]. The approaches presented in these two papers are, however, very different from the one summarised above: the former paper, indeed, changes the semantics of DL by associating *graded membership functions* to models and requiring them for the interpretation of expressions, while the latter one extends the semantics of the DL $\mathcal{ALC}$ by means of weighted alternating

[3] http://www.dc.state.fl.us/pub/scoresheet/cpc_manual.pdf (accessed: 7 September 2020)

parity tree automata. The approach described above is, in comparison, more direct: no changes are made to the definitions of the models of the DL(s) to which threshold operators are added, and the language is merely extended by means of the above-described operators. Provided that the language of the original DL contains the ordinary Boolean operators, adding the threshold operators to it does no increase the expressive power (as already noted in [12]), but does not increase the complexity of reasoning either [7].

3 Explanations via Decision Trees

In the ML literature, techniques for explaining black-box models are typically classified as local and global methods [8]. Whilst local methods take into account specific examples and provide local explanations, global methods aim to provide an overall approximation of the behavior of the black-box model. Global explanations are usually preferable over local explanations, because they provide a more general view about the decision making process of a black-box.

A well-known global explanation method to explaining black-box classifiers is TREPAN [4]. TREPAN is a tree induction algorithm that recursively extracts decision trees from oracles, in particular from feed-forward neural networks. The algorithm is model-agnostic, and it can in principle be applied to explain any black-box classifier (e.g., Random Forest).

TREPAN combines the learning of the decision tree with a trained machine learning classifier (the oracle). At each learning step, the oracle's predicted labels are used instead of known real labels. The use of this oracle serves two purposes: first, it helps to prevent the tree from overfitting to outliers in the training data. Second, and more importantly, it helps to build more accurate trees.

In order to produce enough examples to reliably generate test conditions on lower branches of the tree, TREPAN draws extra artificial query instances that are submitted to the neural network as if they were real data. The features of these query instances are based on the distribution of the underlying data. Both the query instances and the original data are submitted to the neural network 'oracle', and its outputs are used to build the tree.

In [3] we extended the classical TREPAN algorithm to take into account explicit knowledge, modeled by means of ontologies, to drive the explanation extraction process. In particular, we modified the information gain function for choosing, in the creation of split nodes, features associated with more general concepts in a domain ontology. This was achieved by defining a measure of information content of the concepts in an ontology using refinement operators (as a case of point, the DL $\mathcal{EL}$ was considered). Linking explanations to structured knowledge, in the form of ontologies, brings multiple advantages. It does not only enrich explanations (or the elements therein) with semantic information—thus facilitating effective knowledge transmission to users—but it also creates a potential for supporting the customisation of the levels of specificity and generality of explanations to specific user profiles [9].

To measure the effects of the ontology on the understandability of explanations with human users a preliminary on-line user study was conducted. The study showed that decision trees generated by Trepan Reloaded, thus taking domain knowledge into account, were more understandable than those generated without the use of domain knowledge [3].

4 Evaluating Human Understandability of Explanations

Decision trees and threshold expressions appear to have complementary pros and cons as explanatory tools for black-box classifiers. Decision trees have the advantage of having clear visual representations. A human user can easily follow them to understand what factors lead the classifier to reach which conclusion in which circumstances; but on the other hand, especially in the case of very large trees, it can be difficult for a user to follow the overall structure of the decision tree or use it to engage in counterfactual reasoning (e.g., "would the final decision of the classifier have been YES rather than NO if feature C1 had been different?"). Threshold expressions, on the other hand, are arguably of less immediate interpretability for a user; but have the advantage of specifying clearly which factors influence positively or negatively the decision of the classifier, and up to which (comparative) degree, thus making it easier for a user to evaluate the effect that changing certain specific input features would have on the outcome.

Previous work attempting to measure the understandability of symbolic decision models, and decision trees in particular [10,11], proposed syntactic complexity measures based on the model's structure. The syntactic complexity of an explanation can be measured, for instance, in the case of decision trees, by counting the number of internal nodes or leaves, or in the case of logical formulas, by counting the number of symbols adopted. Having a measure like syntactic complexity, that can be easily computed, is useful from an application perspective. E.g., it may be used to prevent excessive complexity in building decision trees and threshold expressions when explaining a black-box. On the other hand, the syntactic complexity does not necessarily capture precisely the understandability of explanations by users. A direct measure of user understandability is how accurately a user can employ a given explanation to perform a decision. Another measure of cognitive difficulty is the reaction time (RT) or response latency [5]. RT is a standard measure used by cognitive psychologists and has become a staple measure of complexity in the domain of design and user interfaces [17]. Understandability depends on the cognitive load experienced by users, e.g., in using the decision model to classify instances and in understanding the features in the model itself. However, for practical processing human understandability needs to be approximated by an objective measure.

We will compare two characterisations of the understandability of explanations: (i) Understandability based on the syntactic complexity of an explanation (number of internal nodes, leaves, symbols used in a weighted formulas, etc.), and (ii) Understandability based on users' performances and subjective ratings,

reflecting, for instance, the cognitive load by users in carrying out tasks using a given explanation format.

We aim at designing and conducting a user study to measure and compare the understandability of explanations given in the form of decision trees and threshold expressions with human users. This could be done in domains where explanations are critical, such as justice, finance or medicine. Conducting and analysing such experiments can provide useful recommendations and insights under which conditions and tasks one representation is deemed more understandable than the other one by users.

References

1. Baader, F., Brewka, G., Gil, O.F.: Adding threshold concepts to the description logic $\mathcal{EL}$. In: Lutz, C., Ranise, S. (eds.) Frontiers of Combining Systems. pp. 33–48. Springer International Publishing, Cham (2015)
2. Baader, F., Ecke, A.: Reasoning with prototypes in the description logic $\mathcal{ALC}$ using weighted tree automata. In: Language and Automata Theory and Applications. pp. 63–75. Springer International Publishing, Cham (2016)
3. Confalonieri, R., Weyde, T., Besold, T.R., del Prado Martín, F.M.: Trepan Reloaded: A Knowledge-driven Approach to Explaining Black-box Models. In: Proc. of the 24th European Conference on Artificial Intelligence. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 2457–2464. IOS press (2020)
4. Craven, M.W., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: NIPS 1995. pp. 24–30. MIT Press (1995)
5. Donders, F.C.: On the speed of mental processes. Acta Psychologica 30, 412–31 (1969)
6. Galliani, P., Kutz, O., Porello, D., Righetti, G., Troquard, N.: On knowledge dependence in weighted description logic. In: Proc. of the 5th Global Conference on Artificial Intelligence (GCAI 2019). pp. 17–19 (2019)
7. Galliani, P., Righetti, G., Kutz, O., Porello, D., Troquard, N.: Perceptron connectives in knowledge representation. In: Proceedings of 22nd International Conference on Knowledge Engineering and Knowledge Management (EKAW 2020) (2020)
8. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comp. Surv. 51(5), 1–42 (2018)
9. Hind, M.: Explaining Explainable AI. XRDS 25(3), 16–19 (2019)
10. Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., Baesens, B.: An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. Decision Support Systems 51(1), 141–154 (2011)
11. Piltaver, R., Luštrek, M., Gams, M., Martinčić-Ipšić, S.: What makes classification trees comprehensible? Expert Syst. Appl. 62(C), 333–346 (Nov 2016)
12. Porello, D., Kutz, O., Righetti, G., Troquard, N., Galliani, P., Masolo, C.: A toothful of concepts: Towards a theory of weighted concept combination. In: Proc. of the 32nd International Workshop on Description Logics. vol. 2373. CEUR-WS (2019)
13. Renard, X., Woloszko, N., Aigrain, J., Detyniecki, M.: Concept tree: High-level representation of variables for more interpretable surrogate decision trees. CoRR abs/1906.01297 (2019), http://arxiv.org/abs/1906.01297
14. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: AAAI. pp. 1527–1535. AAAI Press (2018)

15. Righetti, G., Porello, D., Kutz, O., Troquard, N., Masolo, C.: Pink panthers and toothless tigers: Three problems in classification. In: Proc. of the 5th Int. Workshop on Artificial Intelligence and Cognition. Manchester, September 10–11 (2019)
16. Wick, M.R., Thompson, W.B.: Reconstructive expert system explanation. Art. Intelligence 54(1-2), 33–70 (Mar 1992)
17. William Lidwell, Kritina Holden, J.B.: Universal. Principles of Design. Rockport (2003)