Towards an Inductive Logic Programming Approach for Explaining Black-Box Preference Learning Systems

Fabio A. D'Asaro[1,3] , **Matteo Spezialetti**[1,2] , **Luca Raggioli**[1,3] , **Silvia Rossi**[1]
[1]University of Naples Federico II
[2]University of L'Aquila
[3]CRdC Tecnologie
matteo.spezialetti@univaq.it, silrossi@unina.it

Abstract

In this paper we advocate the use of Inductive Logic Programming as a device for explaining black-box models, e.g. Support Vector Machines (SVMs), when they are used to learn user preferences. We present a case study where we use the ILP system ILASP to explain the output of SVM classifiers trained on preference datasets. Explanations are produced in terms of weak constraints, which can be easily understood by humans. We use ILASP both as a global and a local approximator for SVMs, score its fidelity, and discuss how its output can prove useful e.g. for interactive learning tasks and for identifying unwanted biases when the original dataset is not available. Finally, we highlight directions for further work and discuss relevant application areas.

1 Introduction

In recent years, there has been a sharp increase of Machine Learning algorithms that are able to match or even exceed human performance in a variety of tasks. Although they are able to carry out complex tasks, these reasoning systems are often regarded as *black-boxes*, whose internal structure and functioning cannot be easily translated to human-understandable sentences. This issue has raised concern about the applicability, transparency and reproducibility of these black-box models to scenarios in which taking bad decisions can lead to severe consequences, most notably in medical applications, where no trade-offs between accuracy of the model and its interpretability are possible (He et al. 2019). For this reason, the General Data Protection Regulation (GDPR)[1] introduced the *right to an explanation*, which aims at guaranteeing the right to obtain meaningful explanations of the logic involved by an automated system to all the individuals potentially affected by its decisions. This much debated principle has provided strong rationale not only to develop *transparent* Machine Learning algorithms (i.e., algorithms that, by design, can provide explanations for their own decisions), but also to *post-hoc* methods that automatically extract the logic behind decisions taken by black-box models (see (Guidotti et al. 2018) for a survey of such methods). Transparent systems trade-off between accuracy and ability to explain. On the other hand, post-hoc methods do not alter the black-boxes' accuracy, and seek to reconstruct explanations for their output that aim to be as clear and correct as possible. In this work we follow the latter approach and propose a post-hoc method to explain black-box models for preference learning using Answer Set Programming (ASP) (Calimeri et al. 2020) and the state-of-the-art Inductive Logic Programming framework ILASP (Law, Russo, and Broda 2014; Law, Russo, and Broda 2015; Law, Russo, and Broda 2018). The standard syntax of ASP defines complex constructs such as *weak constraints* (Calimeri et al. 2020), that can be used to naturally express preference relations. In turn, these preferences can be learned by ILASP. Thanks to its ability to generalise from examples, the output of any black-box can be processed by ILASP in order to find an appropriate ASP program that mimics its behaviour. There are many advantages to this procedure: first, ASP programs can be readily translated into natural language that can be understood even by non-experts. Second, it can be used to overcome a major issue of non-transparent Machine Learning models: these are in fact often trained on enormous amounts of data which may encode unwanted (moral, racial) biases or artefacts (see e.g. (Caliskan, Bryson, and Narayanan 2017; Schramowski et al. 2020)). In situations where only the black-box model is available (and not the data it was trained on) logic-based method may expose such systematic biases by e.g. showing that decisions are being taken according to prejudices or unfair principles.

2 Background and Related Work

In this section we briefly overview related work, and demonstrate basic definitions from Explainable AI (XAI), Answer Set Programming (ASP) and Inductive Logic Programming (ILP).

XAI is a quickly growing field of study (see e.g. (Guidotti et al. 2018) for a survey) that aims to generate human-understandable explanations for decisions taken by *black-boxes*. However, little attention has been paid so far to ILP approaches within the field of XAI. Some examples are (Rabold, Siebers, and Schmid 2018; Rabold et al. 2020) and (Shakerin and Gupta 2019). These works, similarly to ours, demonstrate the use of an ILP framework for local explanation of black-boxes using a modification of LIME (Ribeiro, Singh, and Guestrin 2016). Unlike our approach, these works use the Prolog-based system Aleph (Srinivasan

[1] https://eur-lex.europa.eu/eli/reg/2016/679/oj

2004) as their ILP system. Our work, instead, is based on the ILP system ILASP which has been proven to scale favourably with respect to Aleph in terms of accuracy in a variety of tasks (Law, Russo, and Broda 2018). A further major difference is that our approach focuses specifically on *ranking* and *preference learning* tasks, in which one typically wants to predict an order relation on a collection of objects, e.g. "The user prefers fish over vegetables". It is worth noting here that ILP is only one of the possible approaches to generate explanations – other methods such as Exceptional Preferences Mining (Rebelo de Sá et al. 2016) and LIME (Ribeiro, Singh, and Guestrin 2016) are possible. We intend to investigate their relationships with ILASP in forthcoming work.

Our work is based on model-theoretic logic programming language ASP. As we previously mentioned, a feature of ASP that is central to our work is that it can represent *weak constraints*. We briefly show their general form, and then provide the intuition behind their semantics through an example.

Weak constraints have the form:

```
:~ b1, ..., bn [w@l, t1, ..., tm]
```

for terms `t1`, ..., `tm`, `w`, `l`, literals `b1`, ..., `bn` for `m`≥ 0, `n`≥ 0. Terms `w` and `l` are called *weight* and *level* of the constraint, respectively. Terms `t1`, ..., `tm` are used to handle *independence* among weak constraints[2]. Unlike *hard* constraints, weak constraints do not affect the answer sets of a theory. Instead, they induce a preference relation among them. Intuitively, each answer set satisfying the body of a weak constraint gets a penalty that is proportional to that weak constraint's weight. For instance, consider the ASP program P consisting of the following axioms:

```
p(a). p(b). p(c).
0{ q(X) }1 :- p(X).
```

This theory has 8 answer sets, namely: $\{p(a), p(b), p(c)\}$, $\{p(a), p(b), p(c), q(b)\}$, $\{p(a), p(b), p(c), q(c)\}$, $\{p(a), p(b), p(c), q(b), q(c)\}$, $\{p(a), p(b), p(c), q(a)\}$, $\{p(a), p(b), p(c), q(a), q(c)\}$, $\{p(a), p(b), p(c), q(a), q(b)\}$ and $\{p(a), p(b), p(c), q(a), q(b), q(c)\}$. Augmenting P with the following weak constraints:

```
:~ q(a). [1@2, a]
:~ q(b). [3@1, b]
:~ q(c). [-1@2, c]
```

results again in the 8 answer sets above, since answer sets are not modified by weak constraints. We denote the resulting augmented theory by P^+.

To describe the preference relation induced by these weak constraints, we first introduce the notion of *cost* of an answer set at some priority level `l`. This is defined as the sum of weights at priority level `l` for all the weak constraints such that their bodies are satisfied by the answer set. For instance, answer set $\{p(a), p(b), p(c), q(a), q(b), q(c)\}$ satisfies all the bodies of weak constraints in P^+. Therefore its cost at priority level 2 is 0 (which results from adding the weights of the first and third constraint above), whereas its cost at priority level 1 is 3.

We say that an answer set A *is preferred to* an answer set B (according to theory T) if the cost of A is strictly smaller than that of B at the highest level for which their costs differ. For an ASP theory T this is written $A \succ_T B$.

Recall theory P^+ above, and consider answer sets $A_1 = \{p(a), p(b), p(c)\}$ and $A_2 = \{p(a), p(b), p(c), q(a), q(b), q(c)\}$. Note that A_1 does not satisfy any of the bodies of the weak constraints in P^+, therefore its cost is 0 at all priority levels. On the other hand, A_2 satisfies all the bodies of such weak constraints, therefore its cost at priority level 2 is 0, and its cost at level 1 is 3. Therefore, these two answer sets differ at priority level 1, and we conclude that $A_1 \succ_{P^+} A_2$. It is worth noting that negative weights can be assigned to weak constraints. Therefore, for instance, $\{p(a), p(b), p(c), q(c), q(b)\} \succ_{P^+} \{p(a), p(b), p(c)\}$.

Weak constraints can be learned using ILASP (Law, Russo, and Broda 2014). To our knowledge, ILASP is the only ILP system that, to date, can learn weak constraints from *examples*, i.e. ordered pairs of partial answer sets (Law, Russo, and Broda 2015), and therefore it can naturally be applied to preference learning tasks. It should be noted that this generality comes at an increased computational cost with respect to other ILP languages that do not support them. In a nutshell, the user can provide ILASP with a number of ordered pairs, such that the user always prefers the first element of the pair to the second. ILASP, given some additional background knowledge (the *language bias*) required to define the hypothesis space, will find a suitable theory (including weak constraints) that covers the examples in the input set of preferences. ILASP also works in the case of noisy data, that is, when not all the examples can be covered. Instead, theories that does not cover some of the examples are given a (user-customisable) penalty by ILASP, which then tries to find the theory in the hypothesis space that is minimally penalised. We use ILASP version 3 which is specifically targeted to noisy learning tasks (Law, Russo, and Broda 2018).

Finally, we make a terminological remark. In Explainable AI terminology (Guidotti et al. 2018), the extent up to which a transparent model (ILASP, in our case) is able to imitate a black-box model is known as *fidelity*. In the remainder of this paper, we use and measure fidelity in terms of accuracy scores.

3 Explaining Black-boxes With ILASP

Similarly to other work on Machine Learning techniques for small preference datasets (see e.g. (Qomariyah and Kazakov 2017; Law, Russo, and Broda 2018)), we demonstrate our proposed approach by applying it to Support Vector Machines (SVMs) trained on the SUSHI preference dataset[3] (Kamishima 2003). The SUSHI preference dataset comprises sushi preferences of 5000 users. Each user was asked

[2] We are not going to discuss this syntax here, since all weak constraints in the remainder of this paper are independent from each other. The interested reader can find the full syntax and semantics in (Calimeri et al. 2020).

[3] http://www.kamishima.net/sushi/

Feature	Type	Range	ASP Encoding
style	cat	{0,1}	maki/nothing
major group	cat	{0,1}	seafood/nothing
minor group	cat	$\{0,\ldots,11\}$	minor_group(·)
oiliness	cont	$[0,4]$	value(oil,·)
frequency	cont	$[0,3]$	value(freq,·)
price	cont	$[0,5]$	value(price,·)
frequency2	cont	$[0,1]$	value(freq2,·)

Table 1: Features of the SUSHI dataset. Types "cat" and "cont" correspond to categorical and continuous variables, respectively. Feature "style" indicates whether the sushi is maki (0) or not (1), "major group" indicates whether it is made with seafood (0) or not (1), "minor group" describes the subcategory (aomono, akami, shiromi, tare, clam or shell, squid or octopus, shrimp or crab, roe, other seafood, egg, meat, vegetables), "frequency" indicates how often the sushi is eaten and "frequency2" how frequently it is sold.

to rank 10 types of sushi (*ebi*, *anago*, *maguro*, *ika*, *uni*, *tako*, *ikura*, *tamago*, *tekka maki* and *kappa maki*) in the form of a total ordering. Each sushi is associated to the 7 features described in Table 1. We considered a subset of 10 users. For each of them, we trained an SVM (with polynomial kernel) to learn his/her preferences. Given two sushi items, the trained SVM guesses which sushi is preferred over the other according to the user. For instance, if B is an SVM (or, more generally, any kind of black-box) trained on a user's preferences and $Q = (ika, tamago)$ is a *query*, the output of the SVM $B(Q) = 1$ (resp. $B(Q) = -1$) means that the SVM thinks that the user prefers sushi type *ika* over sushi type *tamago* (resp. the user prefers *tamago* over *ika*).

Note that as far as we are concerned, accuracy scores of the SVMs on the underlying dataset are not relevant, as we are working under the hypothesis that the dataset may not be available[4].

We have tested ILASP both as a *global* and *local approximator* for the SVM.

ILASP as global approximator: In the case of *global* approximation, one is concerned with finding a transparent representation of the whole black-box, i.e. the logic that may lead to every possible outcome or decision. The procedure for using ILASP as a global approximator of a preference learning system is given in Algorithm 1. It samples N pairs of items at random from the feature space. Each pair (i_1, i_2) is collected in a list $\mathcal{O}$ together with the black-box's prediction $B(i_1, i_2)$. Note that, thanks to the information recorded in $B(i_1, i_2)$, the set $\mathcal{O}$ can be considered as a set ordered pairs (or *examples*, in the terminology introduced in Section 2) such that the first element of the pair is always preferred to the second element by the underlying black-box. ILASP is then trained on these ordered pairs in $\mathcal{O}$ using an appropriate language bias L.

Algorithm 1 ILASP as a global black-box approximator

Input: Black-box B, language bias L, natural number N
Output: An ASP theory T_B

```
1: $\mathcal{O} \leftarrow \{\}$
2: for $j \in \{1, 2, \ldots, N\}$ do
3:     $i_1 \leftarrow randomly\_sample\_from\_feature\_space()$
4:     $i_2 \leftarrow randomly\_sample\_from\_feature\_space()$
5:     $\mathcal{O} \leftarrow \mathcal{O} \cup ((i_1, i_2), B(i_1, i_2))$
6: end for
7: $T_B \leftarrow ILASP(L, \mathcal{O})$
8: return $T_B$
```

An output theory for our example is:

```
:~ minor_group(3). [1@5, 5]
:~ minor_group(1). [1@4, 3]
:~ seafood. [1@3, 2]
:~ value(price,V0), maki. [V0@2, 4, V0]
:~ minor_group(9). [-1@1, 1]
```

meaning that, according to ILASP, the SVM seems to think that (in order of priority) the User 1 does not like sushis of minor groups 3 and 1, that s/he does not like seafood sushi, that s/he does not like expensive sushis of style maki, and that s/he likes sushis of minor group 9.

The graph on Figure 1 plots the fidelity of ILASP as a function of the number of items it is trained on. Note that, based on a theory, it is not always possible to decide whether a particular type of sushi is preferred over another as different items may get the same penalty score. For example, the output theory above does not allow one to decide which sushi is preferred between two non-maki, non-seafood sushi of minor group 0. We refer to these pairs as *unclassified* pairs. In Figure 1 we plot both the fidelity for the case where unclassified pairs are regarded as errors, and the fidelity for the case where unclassified pairs are simply discarded and are not considered when computing the fidelity.

ILASP as a local approximator: A *local* approximator for a black-box B, a query Q and an associated distance metric π_Q produces an ASP theory $T_{B,Q}$ that seeks to explain why the black-box B produces output $B(Q)$ on input Q. The procedure for using ILASP as a local approximator is given in Algorithm 2 and it is a modification of the LIME method (Ribeiro, Singh, and Guestrin 2016). It departs from Algorithm 1 in that the two items i_1 and i_2 are obtained by *sampling around* Q (compare lines 3 and 4 of Algorithm 1 with line 3 of Algorithm 2), meaning that pairs that are *closer* to Q according to a user-defined distance metric π_Q have a greater probability of being sampled. In addition, these distances are recorded in list $\mathcal{O}$ and we make ILASP to take them into account during the training phase. Theories that cover examples that are far from Q are penalised more than theories that cover examples that are close to Q. Pairs of items (i_1, i_2) are weighted by considering a distance metric $\pi_Q(i_1, i_2)$ to calculate their distance from Q. In our experiments we considered the following metric:

$$\pi_Q(i_1, i_2) = \sum_{j \in \{1,2\}} \sqrt{\sum_f \left(d(i_j(f), q_j(f))\right)^2}$$

where $i_j(f)$ is the value of feature f for item i_j, and $d(i_j(f), q_j(f))$ evaluates to $i_j(f) - q_j(f)$ when f is a con-

[4] For reference, SVM with a polynomial kernel reached an accuracy of $\approx 80.4\%$ using 10-folds cross validation on the first 10 users of the dataset

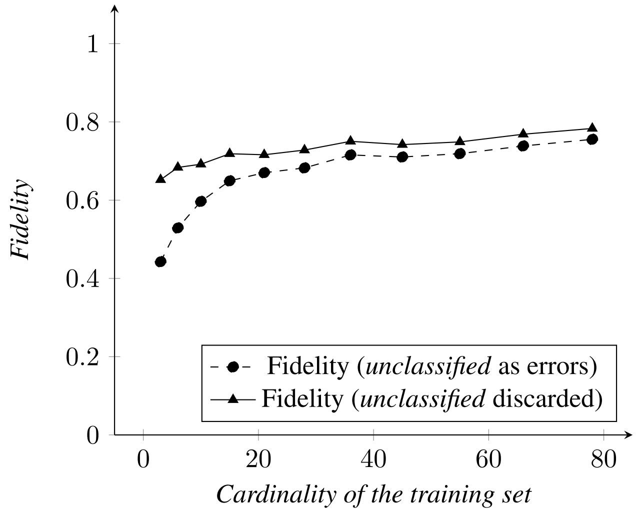

Figure 1: Fidelity of ILASP as a global approximator vs. cardinality of training set

tinuous feature, whereas it evaluates to

$$d(i_j(f), q_j(f)) = \begin{cases} 0 & \text{if } i_j(f) = q_j(f) \\ 3 & \text{otherwise} \end{cases}$$

when f is a categorical feature[5]. During the training phase, we make ILASP take into account distances by making theories that do not cover a pair (i_1, i_2) pay a penalty that is inversely proportional to the distance of (i_1, i_2) from Q.

As an illustration, consider the following pair of items $Q = (q_1, q_2)$: $q1 = [style = 1,\ major\ group = 0,\ minor\ group = 11,\ oiliness = 4,\ frequency = 0,\ price = 2,\ frequency2 = 0]$ and $q2 = [style = 1,\ major\ group = 0,\ minor\ group = 6,\ oiliness = 3,\ frequency = 1,\ price = 3,\ frequency2 = 0]$. Our local approximator procedure, when run on an SVM trained on User 3's preferences, produces the following output:

```
:~ value(freq,V0),
        minor_group(11).[-V0@4, 3, V0]
:~ value(oil,V0).[-V0@2, 2, V0]
:~ minor_group(6).[1@1, 1]
```

This ASP theory provides information about why q_1 is preferred over q_2 by the SVM. In fact, it seems that in a neighbourhood of Q the SVM gives maximum priority to the items of minor group 11 with high non-zero frequency. However, although q_1 is of minor group 11, it has frequency 0. The second priority is to maximise oiliness. Sushi q_1 has higher oiliness than q_2. According to ILASP, this is why the SVM prefers q_1 to q_2. In addition, the third weak constraint gives more information on what is preferred when two sushis also have the same oiliness.

The example above shows that it is not always the case that all weak constraints contribute towards explaining why a sushi is preferred over another: for instance, the reader may verify that axiom

```
:~ value(freq,V0),
    minor_group(11).[-V0@4, 3, V0]
```

above does not help differentiating between items q_1 and q_2, as it assigns a penalty of 0 to both these sushis.

Note that this does not constitute a limitation: it is rather the case that sometimes too many axioms are being produced by ILASP in the output theory. Therefore, to improve readability, one might want to filter out these constraints and only output those that are relevant to the local explainability task, e.g. one might want to output only the weak constraint

```
:~ value(oil,V0). [-V0@2, 2, V0]
```

in this case.

Algorithm 2 ILASP as a local black-box approximator

```
Input: Black-box $B$, language bias $L$, natural number $N$,
    Query $Q$ to be explained, distance $\pi_Q$
Output: An ASP theory $T_{B,Q}$
$\mathcal{O} \leftarrow \{\}$
for $j \in \{1, 2, \ldots, N\}$ do
    $i_1, i_2 \leftarrow sample\_around(Q)$
    $\mathcal{O} \leftarrow \mathcal{O} \cup ((i_1, i_2), B(i_1, i_2), \pi_Q(i_1, i_2))$
end for
$T_{B,Q} \leftarrow ILASP(L, \mathcal{O})$
return $T_{B,Q}$
```

To score the fidelity of our ILASP-based local approximator we performed an experiment where we considered pairs sampled from a neighborhood of Q, both during the training and the testing of ILASP (with disjoint training and test sets). Running Algorithm 2 for the first 10 users on a training set of cardinality 45 for 100 times on random queries resulted in $\approx 91\%$ fidelity and ≈ 0.04 standard deviation.

4 Conclusion

In this work, we performed preliminary experiments on the use of ILP for explaining black-box learning systems. Specifically, we tested the ILASP framework both as global and local approximator of a classic machine learning algorithm (SVM), trained to classify user's preferences regarding sushi. When employed as global estimator, our approach achieved 75.5% and 78.3% of fidelity considering unclassified data as errors or not taking it in account, respectively. Best results were obtained by training the system with a set of about 80 samples (dichotomic preferences), but, as shown in Figure 1, after 45 samples, fidelity values both converge to a plateau (71.0% - 74.2%). For this reason, we chose 45 as training set size for testing our local approximator. In this case ILASP scored an average fidelity of 91.0%, producing a quite good local explanation for SVM queries. At the moment, fidelity is the only metric we adopt to evaluate our approach. If, in addition, absolute preferences of the users were available (e.g., "User1 rates oily sushis 7 out of 10") we would be able to validate output theories on them. For this reason, we plan to build a dataset that also includes such information.

[5] We chose 3 as an arbitrary value for distance between categorical features, as ILASP works with integer weights and 3 was a reasonable choice given the ranges of continuous features defined in Table 1

Ongoing work includes efforts to incorporate fragments of this framework, alongside machine learning black-boxes and other logic-based systems (D'Asaro et al. 2017; D'Asaro et al. 2020), into a decision-making support system in the context of healthcare project AVATEA (D'Asaro, Origlia, and Rossi 2019). The project requires to take decisions and provide natural language explanations and feedback about users' preferred rehabilitation strategies. Often, these decision are taken by black-box classifiers, and being able to provide therapists with useful information about them is a non-trivial task that can be addressed using some of the techniques described here.

Beside its explanation power, the proposed method could be useful to discover unwanted biases and artefacts in the training set the black-box was originally trained upon. This could make black-boxes users aware of such biases and enable them to provide feedback on such explanations (e.g., it might be the case that a user disagrees with the decisions taken). This could in principle be used to *interact* with the user and ask him/her to provide more data if such biases seem to be present. Following this idea, future developments will include the gathering of a large dataset, also featuring meta-information on the user preferences about items' features.

Acknowledgements

This work was partially supported by the Italian Ministry of University and Research (MIUR) within the POR Campania FESR 2014-2020 AVATEA "Advanced Virtual Adaptive Technologies e-hEAlth" research project, and within the PRIN2015 research project "UPA4SAR—User-centered Profiling and Adaptation for Socially Assistive Robotics" (grant no. 2015KB-L78T).

References

Calimeri, F.; Faber, W.; Gebser, M.; Ianni, G.; Kaminski, R.; Krennwallner, T.; Leone, N.; Maratea, M.; Ricca, F.; and Schaub, T. 2020. ASP-Core-2 input language format. *Theory and Practice of Logic Programming* 20(2):294–309.

Caliskan, A.; Bryson, J. J.; and Narayanan, A. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356(6334):183–186.

D'Asaro, F. A.; Bikakis, A.; Dickens, L.; and Miller, R. 2017. Foundations for a probabilistic event calculus. In Balduccini, M., and Janhunen, T., eds., *Proceedings of the 14th International Conference Logic Programming and Nonmonotonic Reasoning, LPNMR 2017*, 57–63. Springer.

D'Asaro, F. A.; Bikakis, A.; Dickens, L.; and Miller, R. 2020. Probabilistic reasoning about epistemic action narratives. *Artificial Intelligence,* https://doi.org/10.1016/j.artint.2020.103352.

D'Asaro, F. A.; Origlia, A.; and Rossi, S. 2019. Towards a logic-based approach for multi-modal fusion and decision making during motor rehabilitation sessions. In *Proceedings of the 20th Workshop "From Objects to Agents" (WOA).*

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)* 51(5):1–42.

He, J.; Baxter, S. L.; Xu, J.; Xu, J.; Zhou, X.; and Zhang, K. 2019. The practical implementation of artificial intelligence technologies in medicine. *Nature medicine* 25(1):30–36.

Kamishima, T. 2003. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 583–588.

Law, M.; Russo, A.; and Broda, K. 2014. Inductive learning of answer set programs. In *European Workshop on Logics in Artificial Intelligence*, 311–325. Springer.

Law, M.; Russo, A.; and Broda, K. 2015. Learning weak constraints in answer set programming. *Theory and Practice of Logic Programming* 15(4-5):511–525.

Law, M.; Russo, A.; and Broda, K. 2018. Inductive learning of answer set programs from noisy examples. *arXiv preprint arXiv:1808.08441.*

Qomariyah, N. N., and Kazakov, D. 2017. Learning binary preference relations. In *4th Joint Workshop on Interfaces and Human Decision Making for Recommender Systems (IntRS)*, 30.

Rabold, J.; Deininger, H.; Siebers, M.; and Schmid, U. 2020. Enriching visual with verbal explanations for relational concepts – combining lime with aleph. In Cellier, P., and Driessens, K., eds., *Machine Learning and Knowledge Discovery in Databases*, 180–192. Springer.

Rabold, J.; Siebers, M.; and Schmid, U. 2018. Explaining black-box classifiers with ILP – empowering LIME with Aleph to approximate non-linear decisions with relational rules. In *International Conference on Inductive Logic Programming*, 105–117. Springer.

Rebelo de Sá, C.; Duivesteijn, W.; Soares, C.; and Knobbe, A. 2016. Exceptional preferences mining. In Calders, T.; Ceci, M.; and Malerba, D., eds., *Discovery Science*, 3–18. Cham: Springer International Publishing.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why should I trust you?": explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 1135–1144. New York, NY, USA: Association for Computing Machinery.

Schramowski, P.; Turan, C.; Jentzsch, S.; Rothkopf, C.; and Kersting, K. 2020. The moral choice machine. *Frontiers in Artificial Intelligence* 3:36.

Shakerin, F., and Gupta, G. 2019. Induction of non-monotonic logic programs to explain boosted tree models using lime. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3052–3059.

Srinivasan, A. 2004. The Aleph manual.