Explaining Black-box Classifiers with ILP – Empowering LIME with Aleph to Approximate Non-linear Decisions with Relational Rules

Johannes Rabold, Michael Siebers, Ute Schmid

Cognitive Systems, University of Bamberg, Germany

Abstract. We propose an adaption of the explanation-generating system LIME. While LIME relies on sparse linear models, we explore how richer explanations can be generated. As application domain we use images which consist of a coarse representation of ancient graves. The graves are divided into two classes and can be characterised by meaningful features and relations. This domain was generated in analogy to a classic concept acquisition domain researched in psychology. Like LIME, our approach draws samples around a simplified representation of the instance to be explained. The samples are labelled by a generator – simulating a black-box classifier trained on the images. In contrast to LIME, we feed this information to the ILP system Aleph. We customised Aleph's evaluation function to take into account the similarity of instances. We applied our approach to generate explanations for different variants of the ancient graves domain. We show that our explanations can involve richer knowledge thereby going beyond the expressiveness of sparse linear models.

1 Introduction

While during the last 20 years machine learning approaches have been mainly evaluated with respect to their accuracy, recently, the importance of explainability of machine learned classifiers has been recognised. In the context of many application domains, it is crucial that system decisions are transparent and comprehensible and in consequence trustworthy (Guidotti *et al.*, 2018; Muggleton *et al.*, 2018; Pu and Chen, 2007). An illustration of how classifier decisions can be made transparent by explanations is given in Figure 1. Here an aerial view of an ancient site is classified as belonging to the iron age – for instance, in contrast to the Viking age. The user can ask for an explanation for the systems decision. A verbal explanation is given, involving different objects which are part of the site and relations between them. For example, there is a circle of stones *included* into another circle. Current systems typically focus on explanations based on simple features. However, there are many domains where object classification depends on relations between primitive constituents – such as molecular chemistry (King *et al.*, 1996). Furthermore, current approaches to explanation generation either focus on textual or on visual explanations (Guidotti *et al.*, 2018). Typically,

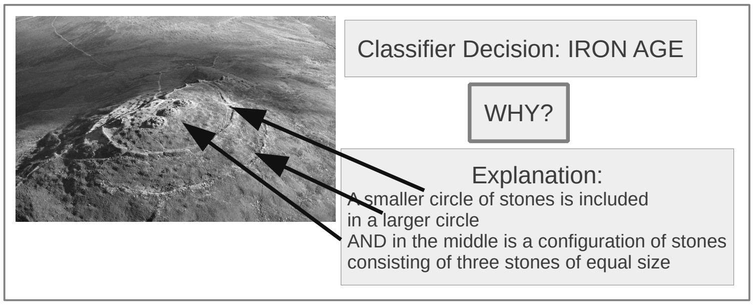
Classifier Decision: IRON AGE
WHY?
Explanation:
A smaller circle of stones is included
in a larger circle
AND in the middle is a configuration of stones
consisting of three stones of equal size

Fig. 1. Image of an ancient site classified to belong to iron age with a textual explanation related to the image. (Aerial View of the Iron Age Site Foel Drygarn Hillfort adapted from http://orapweb.rcahms.gov.uk/coflein/D/DI2006_1293.jpg)

images are explained by highlighting pixels which strongly contributed to the classifier decision (Samek *et al.*, 2017). Verbal explanations are used in the context of text classification (Ribeiro *et al.*, 2016) and in the context of learning from symbolic features (Lakkaraju *et al.*, 2016). However, there is evidence that humans can profit from a dual representation relating visual and verbal information (Mayer and Sims, 1994).

In principle, there are two possibilities to address explainability: On the one hand, machine learning approaches – such as inductive logic programming (ILP) – can be designed which learn interpretable rules, on the other hand, black-box approaches – such as (deep) neural networks – can be extended by an interface for explanation generation.

The first perspective on explainability can be characterised as knowledge-level learning (Dietterich, 1986) which has been also proposed to be a constituent of (human and artificial) cognitive systems (Langley, 2016). Typical approaches of this category are decision trees and variants such as decision sets (Lakkaraju *et al.*, 2016). Decision trees generalise over feature vectors and express concepts as disjunction of conjunctions of constraints over features. More expressive approaches are offered by inductive logic programming (Muggleton and De Raedt, 1994). In ILP training instances are characterised by relations and induced concepts are represented by Horn clauses, that is, relational and even infinite (recursive) concepts can learned. Common to such symbol-level machine learning approaches is that prediction and description of a concept are addressed within the same representation. Learned hypotheses are represented as rules and therefore can be naturally included in rule-based systems such as decision support or expert systems. Techniques for explanation generation which have been already proposed in the early days of AI can be directly applied (Clancey, 1983), for example, the trace of rules which have been applied to reach a conclusion can be shown to the user.

Symbol-level machine learning inherently is explainable AI (Gulwani *et al.*, 2015; Muggleton *et al.*, 2018). However, it often is outperformed by back-box approaches with respect to accuracy since symbolic approaches typically are less suited to highly non-linear decision functions. Therefore, currently there is a lot of research interest addressing the second perspective on explainability, that is to preserve the strong representational power of non-linear classifiers and provide additional methods which can be applied ex-post to make the decisions of a black-box classifier transparent to the user (Guidotti *et al.*, 2018). One of the most considered approaches of this type is LIME (Local Interpretable Model-Agnostic Explanations) which generates sparse linear models as local approximations for a (non-linear) classifier decision (Ribeiro *et al.*, 2016). For image classification domains, LIME identifies that group of pixels (called super-pixel) with the strongest contribution to the classifier decision.

Currently LIME is limited to simple linear explanations, such as which set of pixels are responsible for classification decision 'electric guitar' (Ribeiro *et al.*, 2016). Such simple explanations might be not enough for more domains involving relational concepts. Therefore, we propose an extension of LIME with the ILP approach Aleph (Srinivasan, 2004) for generation of explanations for more complex domains. We demonstrate the approach for coarse images of ancient graves. In the next section (Sect. 2), we first introduce LIME's explanation generation approach in more detail. Afterwards (Sect. 3) we present our modified approach and specifically how similarity between images in the non-linear decision space can be included in Aleph's evaluation function. We present experiments with two variants of the ancient grave domain in Section 4 and conclude in Section 5 with a short discussion and pointers to future work.

2 Generating Local Explanations in LIME

LIME addresses the problem of finding an easy to understand explanation for the classification result of a more complex black-box classifier. The original paper describes the work-flow of LIME for two types of input, images and text. For every instance $x \in X$, there exists an interpretable representation $x' \in X'$. For text, x' consists of a binary vector stating if a word occurs in the text or not. For images, x is a tensor containing the pixel values and x' is a binary vector describing whether small parts (super-pixels) of the image are present or not. The words or the super-pixels can be seen as features which can be used to explain the original classifier function $f(x)$. This function typically has a multi-dimensional and non-linear decision surface. LIME aims at finding a linear model which helps to explain classifier decisions for the original instances in X. This model is a local approximation of the original classifier function where the simple features in the vector representations x' are weighted by their relevances w. The model construction is described in Algorithm 1.

LIME constructs a sparse linear model by sampling N instances around x' where each sample represents a perturbed version of this instance. Each perturbation z' is sampled uniformly at random drawing non-zero elements of x'. For

Algorithm 1 Linear Model Generation with LIME (adapted from Ribeiro *et al.*, 2016)

```
**Require:** Classifier $f$, Number of samples $N$
**Require:** Instance $x$ and its interpretable version $x'$
**Require:** Similarity kernel $\pi_x$, Length of explanation $K$
    $\mathcal{Z} \leftarrow \{\}$
    **for** $i \in \{1, 2, 3, \ldots, N\}$ **do**
        $z'_i \leftarrow$ sample_around($x'$)
        $\mathcal{Z} \leftarrow \mathcal{Z} \cup (z'_i, f(z_i), \pi_x(z_i))$
    **end for**
$w \leftarrow$ K-Lasso($\mathcal{Z}, \mathcal{K}$)$\triangleright$ with $z'_i$ as features, $f(z)$ as target
**return** $w$
```

each sample z', the classifier decision $f(z)$ and the distance $\pi_x(z)$ between the perturbed and the original instance. The distance measure is highly dependent on the form of input. For text the cosine distance can be used. An assumption-free distance measure for images is the Mean Squared Error (MSE). The distance measure represents the relative importance of an instance with respect to fitting a locally faithful model. The final step of the LIME algorithm is picking the K most important features from the input by fitting a regression model on the data. The weights w are learned via least squares (K-Lasso, see Ribeiro *et al.* (2016) for details).

The simpler linear model characterises the original input x in terms of weighted attributes of x'. These weights can be exploited to highlight super-pixels that the classifier seems to find important. In the text domain, the words with the highest (or lowest) weights can be highlighted to show positive (or negative) importance for the classification result. Such linear models are sufficient for some tasks like showing what the important parts for the classification result of an image are. However, there are more intricate tasks such as explaining why a sepia toned image is considered to be retro (Ribeiro *et al.*, 2016). Also problematic is explaining classifications which rely on combinations of features or relations between objects. LIME in its current version could at best highlight the pixels where the relation is located at, but there is no possibility to describe the constellation further.

3 Model Agnostic Explanation Generation with LIME-Aleph

In the following, we describe our adaptation of the LIME system in order to get richer explanations for the classification of images. Specifically, we will apply the ILP approach Aleph (Srinivasan, 2004) to generate explanations in terms of logic rules. In the context of images, logical rules can capture combinations of features and also relations between objects. For example, to explain why an image is classified as a face does not only depend on the occurrence of eyes, nose and mouth, but also on the relations between these entities.

Currently, we presuppose that for a sample of instances there exists not only the image but additionally a logic description. In a next step, our approach can be extended to an interactive system which presents the super-pixels identified by LIME and asks the user for meaningful labels for single super-pixels as well as pairs or even larger tuples. However, even without such interactive labelling, logical rules defined with anonymous predicate names might be helpful: In a recent empirical study in the context of the family domain, it has been shown that for some classification tasks, logical rules can be comprehensible even if predicate names are not meaningful (Muggleton *et al.*, 2018).

The LIME-Aleph algorithm differs from the original LIME as given in Algorithm 1 in lines 6 and 9: Sampling is done by picking N instances with an equal proportion of positive and negative instances. Furthermore, instances are picked such that they are distributed over the complete instance space – and not sampled around the current to be explained x'. The distance to x' will however become relevant during learning with Aleph (see Sect. 3.1). The instance to explain is also part of this sample. As in the original LIME, the sample is stored together with the classification calculated by the black-box classifier and with the distance to the currently to be explained original instance x. Like in the original, this measure indicates how close two images are. We need this information later in order to decide which examples are more important for the explanation of our image. For the original LIME, an L2 distance which is basically the Mean Squared Error between the two images has been used.

The images considered in Ribeiro *et al.* (2016) are photographic images. For more sparse image representations such as black and white line drawings, the L2 measure would over-estimate the distance of two equal sketches that are just shifted by one pixel. Consequently, for the coarse images of ancient graves which we will investigate (see Sect. 4) we first down-sample the images, before we calculate the L2 measure, thereby gaining translation invariance.

Line 9 of Algorithm 1 is replaced by Aleph. That is, now a model in form of logical rules is learned instead of a set of weights for a linear classifier. In the following we give a short description of Aleph followed by a proposition for a new evaluation function to guide Aleph's refinement search.

3.1 The Aleph System

The inductive logic programming system Aleph is based on a specific-to-general refinement search. The induction algorithm can be described as follows (Srinivasan, 2004):

1. Take one example from the example set. If none exists stop.
2. Build the most-specific-clause that entails the example selected.
3. Search for a clause that is more general than the current clause.
   - A more general clause is defined as a subset of the current set of literals.
   - Search is guided by an evaluation function.
4. Remove the redundant examples (all examples covered by the current clause).
5. Repeat from the step 1.

3.2 Adapted Evaluation Function

By default, the search in Aleph is guided by a coverage measure that takes into account how many positive (p) and negative (n) examples a clause covers. The evaluation score is simply calculated by $score = p - n$. The clause with the highest *score* is selected. However, in our implementation of LIME, the distance measure plays an important role in defining the locality of an image to the image we want to explain. Consequently, we adapted Aleph's evaluation function to take into account distances between examples. Having a clause cover many positive examples with a small distance needs to be preferred over having a clause covering many negative examples with a small distance. We can state this in a cost measure $cost(Clause)$ of a clause $Clause$ with respect to positive and negative examples:

$$cost(Clause) =$$

$$\sum_{e \in E^+} \begin{cases} -c(e), & \text{if e gets covered by the clause} \\ 0, & \text{otherwise} \end{cases}$$

$$+$$

$$\sum_{e \in E^-} \begin{cases} 0, & \text{if e does not get covered by the clause} \\ c(e), & \text{otherwise} \end{cases}$$

where $c(e)$ is defined as $\frac{1}{(1+d(e))^2}$ with $d(e)$ being the distance assigned to e. E^+ is the set of all positive examples and E^- the set of all negative examples.

4 Experiments

To investigate how LIME can profit from explanation generation with ILP, we used a concept learning domain introduced by Medin and Schaffer (1978) which has been investigated extensively in cognitive psychology (Goodman *et al.*, 2008). The concepts are typically depicted graphically, mostly by simple line drawings. The concept to be learned for the original domain can be represented by a decision tree (Lafond *et al.*, 2009) or three rules with a conjunction of two feature attributes each. The representation of this type of concept goes already beyond the simple single feature approximation of LIME. In a next step, we extended the concept learning problem to include a conjunction of binary relations, structurally similar to the grandparent rule which has been extensively studied in ILP (Muggleton *et al.*, 2018). Bitmaps of training examples for both variants were produced by a generator program. The images together with the classification decision were input for the LIME-Aleph algorithm.

Table 1. Abstract structure of the concept learning domain introduced by Medin and Schaffer (1978)

Category A	Category B	Transfer
A1 0001	B1 0011	T1 0110
A2 0101	B2 1001	T2 0111
A3 0100	B3 1110	T3 0000
A4 0010	B4 1111	T4 1101
A5 1000		T5 1010
		T6 1100
		T7 1011

4.1 The Medin and Schaffer Concept Acquisition Task

The classical domain introduced by Medin and Schaffer (1978) is characterised by four binary features. A concept *A* has to be learned from five positive examples and four negative examples (concept *B*). The remaining seven instances are used as transfer items to explore how humans generalise under different experimental conditions. The abstract learning problem is given in Table 1. It has been instantiated with different graphical domains. The most basic domain investigated are geometrical forms with figure as triangle or circle, colour as red or green, size as small or large, position as left or right. A more natural domain is Brunswick faces where metric information is given in two discrete instantiations for eye height, eye distance, nose length, mouth height (Medin and Schaffer, 1978; Nosofsky *et al.*, 1994).

We instantiate the domain with fictitious patterns of ancient graves which either from iron age or from Viking age. The four features are not given by two specific instantiations but by decision boundaries (Goodman *et al.*, 2008):

- form: narrow/round (axis ratio < 0.5),
- number of stones: many/few (number of stones > 10),
- corner stones: think/normal (circumference $> 2\times$ of average size),
- orientation: north/west (angle between -45 and $+45$ from vertical).

Examples for the training instances are given in Figure 2. Using decision boundaries rather than Boolean features, the 16 distinct patterns can be instantiated with arbitrarily many images. Nevertheless, linguistic labels can be used to characterise membership to a category.

Given the pattern in Table 1, the classification rules which correctly discriminate iron age graves from Viking age graves are

```
iron(X) :- narrow(X), north(X).
iron(X) :- narrow(X), thick(X).
iron(X) :- thick(X), north(X).
```

That is, one feature (the number of stones) is irrelevant and a grave can be classified as belonging to iron age by a combination of a specific value of either two of the remaining features. For Aleph, training examples are represented in the usual way as

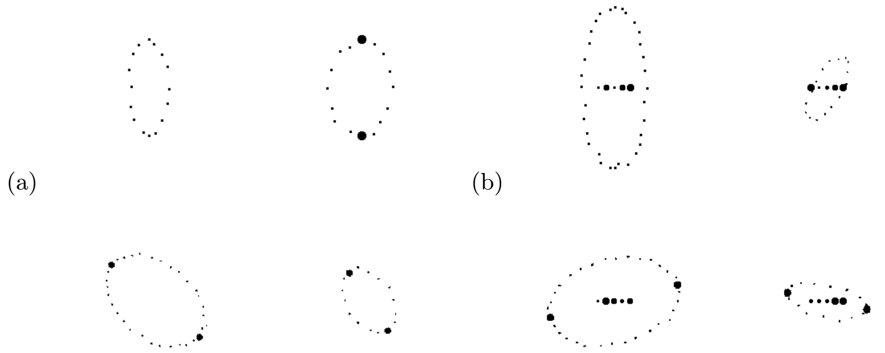

Fig. 2. Examples for the ancient grave domain based on the Medin and Schaffer (1978) structure (a) and extended for learning concepts involving relations (b), top row shows positive and bottom row negative examples.

```
iron(iron93).          not iron(viking87).
narrow(iron93).        many(viking87).
thick(iron93).         north(viking87).
many(iron93).
```

4.2 Complex Relational Concepts

To extend the classification problem such that relational information has to be taken into account, we modified the ancient graves domain such that there is an additional pattern of stones within the stone circle. A grave now shall be of iron age, if the included stones contain a sequence of exactly three stones growing in size. Examples are given in Figure 2. The target predicate *iron(X)* still is defined for individual graves, that is, it is not itself relational, such as for instance *grandparent(A,B)*. However, relations have to be used in the rule body to model a sequence of stones growing in size. For a grave belonging to the iron age, the following rules have to hold:

```
iron(X) :- outer(X,A), next(A,B), next(B,C), larger(A,B), larger(B,C).
iron(X) :- outer(X,A), next(A,B), next(B,C), next(C,D), larger(B,C),
           larger(C,D).
iron(X) :- outer(X,A), next(A,B), next(B,C), next(C,D), next(D,E),
           larger(C,D), larger(D,E).
```

Examples are described in the following way:

```
iron(iron79).                        not iron(viking9).
thick(iron79).                       thick(viking9).
north(iron79).                       outer(viking9, stone_a_viking9).
outer(iron79, stone_a_iron79).       smallest(stone_a_viking9).
medium(stone_a_iron79).              largest(stone_b_viking9).
```

```
smallest(stone_b_iron79).             next(stone_a_viking9, stone_b_viking9).
next(stone_a_iron79, stone_b_iron79). medium(stone_c_viking9).
small(stone_c_iron79).                next(stone_b_viking9, stone_c_viking9).
next(stone_b_iron79, stone_c_iron79). largest(stone_d_viking9).
largest(stone_d_iron79).              next(stone_c_viking9, stone_d_viking9).
next(stone_c_iron79, stone_d_iron79). largest(stone_e_viking9).
small(stone_e_iron79).                next(stone_d_viking9, stone_e_viking9).
next(stone_d_iron79, stone_e_iron79).
```

The binary relation *larger/2* is given as background knowledge for the five sizes *smallest/1*, *small/1*, *medium/1*, *large/1*, *largest/1* – for example, *larger(X, Y) :- smallest(X), small(Y).*

4.3 A Generator for Coarse Images

For the construction of training examples we let the graves be created by a generator. By using a generator instead of real images we are able to derive the classification result right away since the attributes for the graves are known by construction. Instead of using a black-box classifier, we can simulate such a classifier by the generator. Using a generator has also the advantage that one can have an almost infinitely large variety in the examples. Furthermore, different classification rules can be created.

For the original ancient graves domain we only need the predicates *narrow(X)*, *thick(X)*, *many(X)* and *north(X)*. The generator first produces random truth values for the four predicates and then draws images corresponding to the predicates. By allowing small random variations we get a large variety of images of grave sites. Next, the generated examples are filtered according to the pre-defined classifier rules for *iron(X)*. That way we obtain the classification result and can simulate a black-box classifier. For the relational graves domain we use a similar approach. In addition to generating the boundary of a grave, the generator now sets stones of different sizes in the middle. First, sizes are assigned randomly to the stones and then images are filtered according to the classification rules.

4.4 Results for the Original Ancient Graves Domain

To test LIME-Aleph for the original ancient graves domain we generated 100 images together with their corresponding logic representations. To simulate the intended application of explanation generation for a classifier decision, we can pick one of the images as new image x_1 as input to a black-box classifier which returns a classification decision, for instance, that x_1 belongs to iron age. Image x_1 is associated with a logic representation x'_1. An illustration is given in Figure 3: A narrow grave with thick corner stones and oriented towards north is presented to a classifier which labels it as belonging to the iron age. That it does not hold that the predicate has *many* stones is inferred due to closed world assumption.

LIME-Aleph now samples 20 images – 10 positives, including the to be explained image, and 10 negatives. The distances from all images to x_1 are calculated and Aleph generates the rule which classifies x'_1 from this sample. For the given example, only one, very simple rule has been learned:

```
iron(X) :- north(X).
```

Although not the original rule *iron(X) :- thick(X), north(X).* has been induced but a more general one, the result is a reasonable approximation. Which rule is generated for an explanation depends on the sample from which is learned.

In a second trial, an image labelled as belonging to iron age had the following attributes:

```
narrow(x2).
many(x2).
thick(x2).
north(x2).
```

and Aleph induced the hypothesis

```
iron(X) :- north(X).
iron(X) :- narrow(X), thick(X).
```

which covers all relevant features of the image for the class *iron*.

LIME-Aleph is also capable of showing why an instance is not part of the target class *iron*. We picked an instance with the features

```
  narrow(x3).
  many(x3).
```

without thick corner stones and not oriented towards north (which Aleph infers due to closed world assumption). Aleph learned the rules

```
iron(X) :- north(X).
iron(X) :- narrow(X), thick(X).
```

Here LIME-Aleph explains why the example does not belong to iron age because it is not north or not simultaneously narrow and with thick corner stones.

The three examples give a proof-of-concept that LIME-Aleph can induce explanatory rules from small samples and with the modified evaluation function which takes into account distances of examples to the to be explained instance.

`Left grave: x1': narrow(x1'), thick(x1'), north(x1')`

Fig. 3. The left grave is the query image, the right image the closest neighbour according to our distance measure.

4.5 Results for the Relational Ancient Graves Domain

We also explored LIME-Aleph's ability to generate helpful explanations for the relational variant of the graves domain. For an instance, which can be characterised in the following way:

```
many(x4).     medium(a).
thick(x4).    smallest(b).
north(x4).    smallest(c).
outer(x4,a).  small(d).
next(a,b).    largest(e).
next(b,c).
next(c,d).
next(d,e).
```

LIME-Aleph induced the following rules to explain the current instance in contrast to a sample of other instances:

```
iron(X) :- outer(X,A), next(A,B), next(B,C), next(C,D), larger(C,D),
           next(D,E), larger(C,E).
iron(X) :- outer(X,A), next(A,B), next(B,C), larger(B,C), next(C,D),
           larger(C,D).
```

While the first rule did not quite get the correct relation (larger relation between D and E must not hold), the second rule matches exactly one of the correct rules. The example can be explained to belong to the iron age because the line of stones in its centre contains a sequence of three stones with ascending size from west to east.

5 Conclusions and Further Work

We presented an extension to the model agnostic explanation generation approach LIME. This extension allows to generate explanations which take into account combinations of features and relations between parts of an to be classified object. As illustrated in Figure 1, there exist domains, where relations between constituents are crucial for a concept. This is obviously true for molecules, but has also been demonstrated in early days of AI by Winston (1970) who showed how the relational concept of an arc can be learned from near misses. We presented ancient grave sites as an example domain and explored it in a classic feature based variant and in a relational variant.

Goal of this paper was to present a first proof-of-concept how explanation generating approaches which has been developed in the context of end-to-end learning of images could profit from more sophisticated, logic based learning mechanisms such as ILP. We could demonstrate, that combining LIME with Aleph can indeed help to generate more complex explanations. However, currently, LIME-Aleph needs not only the images but also their logic descriptions as input. In the context of ILP, it is quite usual, to generate logical descriptions from images (Farid and Sammut, 2014). Nevertheless, automatic or at least semi-automatic extraction of features and relations from images would be a large step towards more general applicability. We plan to investigate whether information represented in convolutionary layers of a deep network can be used for generating meaningful features and (spatial) relations. Thereby, we can build on ongoing research in computer visions (Rohrbach *et al.*, 2013).

Acknowledgements. The research reported is funded by the German Research Society (DFG) within the Priority Research Program on 'Intentional Forgetting in Organisations' (DFG-SPP 1921), project Dare2Del (SCHM1239/10-1). The authors cordially thank Laura Henning who proposed the ancient grave domain as an instantiation of the Medin and Schaffer concept learning task.

References

Clancey, W. J. (1983). The epistemology of a rule-based expert systema framework for explanation. *Artificial Intelligence*, **20**(3), 215–251.

Dietterich, T. G. (1986). Learning at the knowledge level. *Machine Learning*, **1**(3), 287–315.

Farid, R. and Sammut, C. (2014). Plane-based object categorisation using relational learning. *Machine Learning*, **94**(1), 3–23.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., and Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive Science*, **32**(1), 108–154.

Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., and Giannotti, F. (2018). A survey of methods for explaining black box models. *arXiv preprint arXiv:1802.01933*.

Gulwani, S., Hernandez-Orallo, J., Kitzelmann, E., Muggleton, S. H., Schmid, U., and Zorn, B. (2015). Inductive programming meets the real world. *Communications of the ACM*, **58**(11), 90–99.

King, R. D., Muggleton, S. H., Srinivasan, A., and Sternberg, M. (1996). Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. In *Proceedings of the National Academy of Sciences*, volume 93, pages 438–442. National Acad Sciences.

Lafond, D., Lacouture, Y., and Cohen, A. L. (2009). Decision-tree models of categorization response times, choice proportions, and typicality judgments. *Psychological Review*, **116**(4), 833.

Lakkaraju, H., Bach, S. H., and Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684. ACM.

Langley, P. (2016). The central role of cognition in learning. *Advances in Cognitive Systems*, **4**, 312.

Mayer, R. E. and Sims, V. K. (1994). For whom is a picture worth a thousand words? extensions of a dual-coding theory of multimedia learning. *Journal of Educational Psychology*, **86**(3), 389.

Medin, D. L. and Schaffer, M. M. (1978). Context theory of classification learning. *Psychological Review*, **85**(3), 207.

Muggleton, S. and De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming, Special Issue on 10 Years of Logic Programming*, **19-20**, 629–679.

Muggleton, S. H., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., and Besold, T. (2018). Ultra-strong machine learning: comprehensibility of programs learned with ILP. *Machine Learning*, pages 1–22.

Nosofsky, R. M., Palmeri, T. J., and McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review*, **101**(1), 53.

Pu, P. and Chen, L. (2007). Trust-inspiring explanation interfaces for recommender systems. *Knowledge-Based Systems*, **20**(6), 542–556.

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*, pages 1135–1144. ACM.

Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., and Schiele, B. (2013). Translating video content to natural language descriptions. In *International Conference on Computer Vision (ICCV)*, pages 433–440. IEEE.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., and Müller, K.-R. (2017). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems*, **28**(11), 2660–2673.

Srinivasan, A. (2004). *The Aleph Manual*. http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/.

Winston, P. H. (1970). Learning structural descriptions from examples. Technical Report MIT/LCS/TR-76, MIT.