Inductive Programming as Approach to Comprehensible Machine Learning

Ute Schmid

Faculty Information Systems and Applied Computer Science
University of Bamberg
96045 Bamberg, Germany
ute.schmid@uni-bamberg.de

Abstract. In the early days of machine learning, Donald Michie introduced two orthogonal dimensions to evaluate performance of machine learning approaches – predictive accuracy and comprehensibility of the learned hypotheses. Later definitions narrowed the focus to measures of accuracy. As a consequence, statistical/neuronal approaches have been favoured over symbolic approaches to machine learning, such as inductive logic programming (ILP). Recently, the importance of comprehensibility has been rediscovered under the slogan 'explainable AI'. This is due to the growing interest in black-box deep learning approaches in many application domains where it is crucial that system decisions are transparent and comprehensible and in consequence trustworthy. I will give a short history of machine learning research followed by a presentation of two specific approaches of symbolic machine learning – inductive logic programming and end-user programming. Furthermore, I will present current work on explanation generation.

Die Arbeitsweise der Algorithmen, die über uns entscheiden, muss transparent gemacht werden, und wir müssen die Möglichkeit bekommen, die Algorithmen zu beeinflussen. Dazu ist es unbedingt notwendig, dass die Algorithmen ihre Entscheidung begründen!
Peter Arbeitsloser zu John of Us, Qualityland, Marc-Uwe Kling, 2017

1 Introduction

Machine learning research began in the 1950s with roots in two different scientific communities – artificial intelligence (AI) and signal processing. In signal processing research mainly statistical methods to generalisation learning from data were developed under the label of *pattern recognition*. In AI *machine learning* was investigated under this label as a crucial principle underlying general intelligent behaviour along with other approaches such as knowledge representation, automated reasoning, planning, game playing, and natural language processing (Minsky, 1968). Over the next decades, research between these two communities began to overlap, mainly with respect to neural information processing

research, especially artificial neural networks. An early important domain of interest outside the machine learning community itself was data bases research which recognised the usefulness of machine learning methods for the discovery of patterns ins large data bases (data mining).

In the early days of machine learning, Donald Michie introduced two orthogonal dimensions to evaluate performance of machine learning approaches – predictive accuracy and comprehensibility of the learned hypotheses (Michie, 1988). Michie proposed to characterise comprehensibility as operational effectiveness – meaning that the machine learning system can communicate the learned hypothesis (model) to a human whose performance is consequently increased to a level beyond that of the human studying the training data alone (Muggleton, Schmid, Zeller, Tamaddoni-Nezhad, & Besold, 2018). Later definitions (Mitchell, 1997), narrowed their focus to measures of accuracy. As a consequence, statistical/neuronal approaches have been favoured over symbolic approaches to machine learning, such as inductive logic programming (ILP) (Muggleton & De Raedt, 1994).

However, in recent years, there is a growing recognition that the holy grail of performance is not sufficient for successful applications of machine learning in complex real world domains. Under the label of 'explainable AI' (Ribeiro, Singh, & Guestrin, 2016) different approaches are proposed for making classification decisions of black box classifiers such as (deep) neural networks more transparent and comprehensible for the user. Such explanations are mostly given in form of visualisations. However, alternatively or additionally, verbal explanations might be helpful, similar to the approaches to explanation generation developed in early AI (Clancey, 1983). There explanations were generated based on the execution traces of rules during inference. While most current machine learning methods rely on implicit black box representations of the induced hypotheses, symbolic machine learning approaches allow to induce rules from training examples.

In the following, I will argue that symbol level approaches to machine learning, such as approaches to inductive programming, offer an interesting complement to standard machine learning. I will present ILP and end user programming as two specific approaches of interest. Furthermore, I will present current work on explanation generation.

2 Approaches to Inductive Programming

A classic white-box approach which allow the generation of hypotheses in form of symbolic representations are decisions trees and variants such as decision rules or random forests. There is empirical evidence that such machine learned hypothesis can be understood and applied by humans (Lakkaraju, Bach, & Leskovec, 2016; Fürnkranz, Kliegr, & Paulheim, 2018). Other approaches to symbol-level learning are grammar inference (Angluin, 1980; Siebers, Schmid, Seuß, Kunz, & Lautenbacher, 2016), inductive functional programming (Gulwani et al., 2015; Schmid & Kitzelmann, 2011; Kitzelmann & Schmid, 2006), and the already

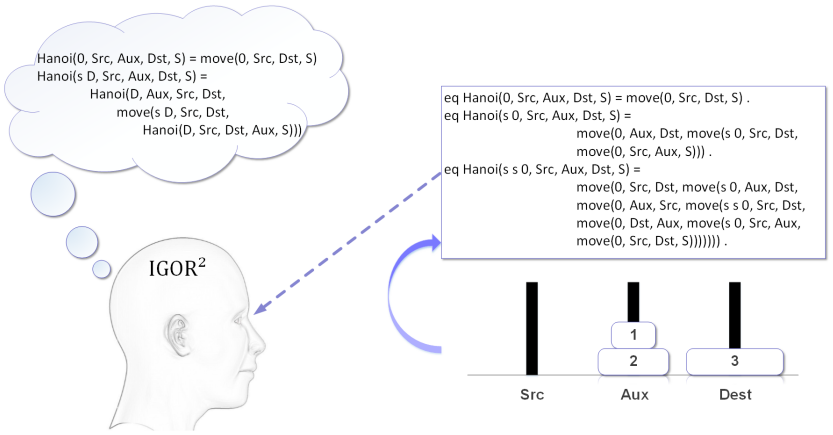


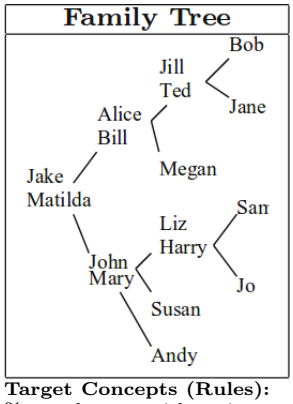
Fig. 1. Recursive functional program to solve Tower of Hanoi problems learned from a small set of examples, taken from Gulwani et al., 2015.

mentioned inductive logic programming. An example for a learned functional program is given in Figure 1, an example for ILP is given in Figure 2

In contrast to standard machine learning, these approaches are strongly related to formalisms of computer science. The expressiveness of learned hypotheses goes beyond conjunctions over feature values. In principle, arbitrary computer programs – for instance in the declarative language Haskell (for inductive functional) or Prolog (for inductive logic programming) can be induced from training examples (Gulwani et al., 2015; Muggleton et al., 2018). The learned programs or rules are naturally incorporable in rule-based systems. Because the learned hypotheses are represented in symbolic form they are inspectable by humans and therefore provide transparency and comprehensibility of the machine learned classifiers. In contrast to many standard machine learning approaches, learning is possible from few examples. This corresponds to the way humans learn in many high-level domains (Marcus, 2018). However, there are also some draw-backs for these symbolic approaches to machine learning: They are inherently brittle and are not designed to deal well with noise and it is in general not easy to express complex non-linear decision surfaces in logic. While (deep) neural networks and other black box approaches often reach high predictive accuracy, symbolic approaches such as inductive functional or logic programming are superior with respect to comprehensibility. Consequently, research on hybrid approaches combining both methods seems a promising domain of research (Rabold, Siebers, & Schmid, 2018).

2.1 Inductive Logic Programming

Inductive logic programming has been established in the 1990ies as a symbol-level approach to machine learning where logic (Prolog) programs are used as



Background Knowledge (Observations):

father(jake,alice).	mother(matilda,alice).
father(jake,john).	mother(matilda,john).
father(bill,ted).	mother(alice,ted).
father(bill,megan).	mother(alice,megan).
father(john,harry).	mother(mary,harry).
father(john,susan).	mother(mary,susan).
	mother(mary,andy).
father(ted,bob).	mother(jill,bob).
father(ted,jane).	mother(jill,jane).
father(harry,sam).	mother(liz,sam).
father(harry,jo).	mother(liz,jo).

Target Concepts (Rules):

% grandparent without invented pred.
p(X,Y) :- father(X,Z), father(Z,Y).
p(X,Y) :- father(X,Z), mother(Z,Y).
p(X,Y) :- mother(X,Z), mother(Z,Y).
p(X,Y) :- mother(X,Z), father(Z,Y).

% ancestor without invented predicate
p(X,Y) :- father(X,Y).
p(X,Y) :- mother(X,Y).
p(X,Y) :- father(X,Z), p(Z,Y).
p(X,Y) :- mother(X,Z), p(Z,Y).

Target Concepts (Rules):

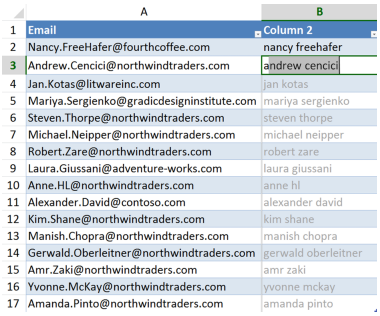
p(X,Y) :- p1(X,Z), p1(Z,Y).
p1(X,Y) :- father(X,Y).
p1(X,Y) :- mother(X,Y).

% ancestor with invented predicate
p(X,Y) :- p1(X,Y).
p(X,Y) :- p1(X,Z), p(Z,Y).
p1(X,Y) :- father(X,Y).
p1(X,Y) :- mother(X,Y).

Fig. 2. An example family tree and two possible target predicates which can be learned from a small set of positive and negative examples, taken from Muggleton et al., 2018.

a uniform representation for examples, background knowledge and hypotheses. In Figure 2 the well-known family tree example is shown. Here the father and mother relations are given as background knowledge as observed/known facts concerning a specific family. Target predicates can be as simple as *grandfather* which can be described as the father of a parent of a person or a bit more general such as *grandparent* or more complex, such as the *ancestor* relation which involves recursion. For learning, some positive and negative examples for the target predicate are presented. For example *ancestor(Jake, Bob)* is a positive example and *ancestor(Harry, Mary)* is a negative example for *ancestor*. Given this information, an ILP system can derive a hypothetical logic program which entails all the positive and none of the negative examples. Some relaxation of this brittle criterion is possible (Siebers & Schmid, 2018).

One of the main advantages of ILP over other symbolic approaches to machine learning is that it allows to consider n-ary relations. Although it is possible to transform relations into features, such a transformation can be tedious, result in sparse feature vectors, and cannot be performed without loss of information. The advantage of ILP in structural domains such as chemistry has for example been demonstrated for the identification of mutagenic chemical structures (King, Muggleton, Srinivasan, & Sternberg, 1996). That humans can comprehend and



	A	B
1	Email	Column 2
2	Nancy.FreeHafer@fourthcoffee.com	nancy freehafer
3	Andrew.Cencici@northwindtraders.com	andrew cencici
4	Jan.Kotas@litwareinc.com	jan kotas
5	Mariya.Sergienko@gradicdesigninstitute.com	mariya sergienko
6	Steven.Thorpe@northwindtraders.com	steven thorpe
7	Michael.Neipper@northwindtraders.com	michael neipper
8	Robert.Zare@northwindtraders.com	robert zare
9	Laura.Giussani@adventure-works.com	laura giussani
10	Anne.HL@northwindtraders.com	anne hl
11	Alexander.David@contoso.com	alexander david
12	Kim.Shane@northwindtraders.com	kim shane
13	Manish.Chopra@northwindtraders.com	manish chopra
14	Gerwald.Oberleitner@northwindtraders.com	gerwald oberleitner
15	Amr.Zaki@northwindtraders.com	amr zaki
16	Yvonne.McKay@northwindtraders.com	yvonne mckay
17	Amanda.Pinto@northwindtraders.com	amanda pinto

Fig. 3. Flashfill as example for a successful application of inductive programming to end-user programming. Here a user edits the first line of column 2 and Flashfill can infer the intended transformation, taken from Gulwani et al., 2015.

successfully apply ILP learned rules has been demonstrated in two experiments (Muggleton et al., 2018)

2.2 End-user Programming

Inductive programming approaches typically are defined in a generic way. In the context of end-user programming, Gulwani and colleagues demonstrated that such techniques can be highly efficient in complex practical applications if they are restricted for a specific domain (Gulwani et al., 2015). For example, since 2013 the system Flashfill is included in Excel. It can learn table manipulations from observing user actions.

Flashfill is a convincing example for end-user programming which gives users without background in computer programming the possibility to work more efficiently with their software applications (Cypher, 1995; Schmid & Waltermann, 2004). Users with a background in computer programming have the possibility to inspect the programs induced by Flashfill. For end-users, transparency is gained by the possibility to directly observe the effect of giving an example on the actions of the program.

3 Explanation Generation

Over the last years there is a growing interest in artificial intelligence from industry and it became a topic of discussion in politics and society in general. Mostly AI is used a label for machine learning – ignoring all other subject domains which constitute AI, among them knowledge representation and learning. Furthermore, machine learning is mostly exclusively referring to artificial neural networks, especially variants of deep learning. Following the big bang of deep learning (LeCun, Bengio, & Hinton, 2015), there is a growing recognition from practitioners, for example in medicine, in connected industry, and in automotive,

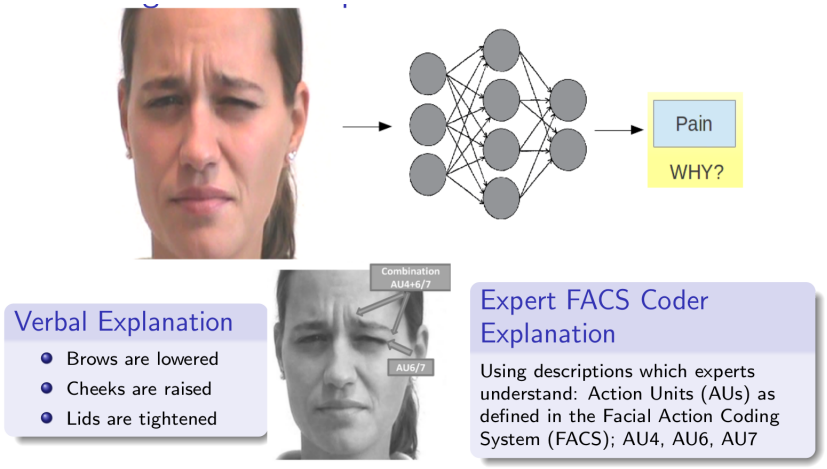


Fig. 4. Illustration of an explanation of the classifier decision that the person is in pain.

that back box classifiers have the disadvantage of being intransparent. Evaluation of the safety of software involving such classifiers is problematic and naive as well as expert users will ultimately hesitated to trust such systems.

It is widely recognized that explanations are crucial for comprehensibility and trust (Tintarev & Masthoff, 2015; Ribeiro et al., 2016). For image classification, explanations are typically given in a visual form. For example, the system LIME provides an explanation interface where such parts of an image are highlighted which are relevant for the classification decision (Ribeiro et al., 2016). However, visual explanations are restricted to conjunctions of isolated information. Verbal explanations in addition allow for the use of relational information, for recursion, and even negation:

- This is a grave of the iron age because there is one stone circle *within* another one.
- This is a correct Tower of Hanoi because starting with the largest disk at the bottom, *each* disk is smaller then the previous one.
- This is a peaceful person because he/she is *not* holding a weapon (but a flower).

Often, it might be especially helpful to combine visual and verbal explanations. An example is given in Figure 4. Here the explanation involves distortions of specific regions of the face which are typically characterized by so called action units used in the facial action coding systems (FA CS) (Ekman & Friesen, 1971; Siebers et al., 2016). The explanation can be given either in generally understandable terms or for an expert knowledgeable in FA CS coding.

Explanations are important for transparency of a machine learned classifier. Experts might demand an explanation if their own decision deviates from that

Fig. 5. Illustration of an explanation of the classifier decision that the person is in pain.

of the classifier. The explanation might be convincing for the expert or not and consequently, the expert might revise his decision or override the classification.

Explanations can also be helpful in education and training. Here, contrasting examples might be helpful to make explanations more comprehensible. In many realistic contexts, it is not so easy to relate ones perception to a specific feature value. For example, to classify a mushroom it might be necessary to decide whether the cap has the shape of a bell or whether it is conical (Schlimmer, 1987). To understand the difference between these feature values, contrasting examples might be helpful. Similarly, facial expressions for pain and disgust share a number of action units. Consequently, sometimes an observer might confuse these states and might profit from an explicit contrast (see Fig, 5.

It is an open research question which type of contrast is helpful to support understanding. In cognitive science research on similarity and analogy it is recognized that alignment is an important aspect for concept acquisition (Markman & Gentner, 1996; Goldwater & Gentner, 2015). Exemplars must be sufficiently similar that differences between them are helpful: To understand the concept of a bottle, it might be helpful to contrast it with some other receptacle such as a mug but it will not be helpful to contrast it with an arbitrary object, say a chair. To understand the concept of a grandparent (see Fig. 2) it might be helpful to replace one attribute, such as male – contrasting with grandmother – or to invert relations, contrasting with grandchild.

4 Conclusions and Further Work

It is my strong conviction that future applications of machine learning should not aim at replacing human decision makers but at supporting them in complex domains. It is still desirable, in my opinion, to develop machine learning approaches which fulfill the ultra-strong machine learning criterion proposed by Michie. Thereby, rather than creating autonomous systems which replace humans, we can create intelligent companion systems (Forbus & Hinrichs, 2006). Such companions might make it possible that humans can perform better when

supported by the system than when they are on their own. 'Better' might address efficiency, correctness, or simply feeling more secure or more relaxed. I hope that approaches to inductive programming will be a useful building block in such an endeavour.

References

Angluin, D. (1980). Inductive inference of formal languages from positive data. *Information and control*, *45*(2), 117–135.

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

Cypher, A. (1995). Eager: Programming repetitive tasks by example. In *Readings in human–computer interaction* (pp. 804–810). Elsevier.

Ekman, P., & Friesen, W. V. (1971). Constants across cultures in the face and emotion. *Journal of Personality and Social Psychology*, *17*(2), 124.

Forbus, K. D., & Hinrichs, T. R. (2006). Companion cognitive systems – a step toward human-level AI. *AI Magazine, special issue on Achieving Human-Level Intelligence through Integrated Systems and Research*, *27*(2), 83-95.

Fürnkranz, J., Kliegr, T., & Paulheim, H. (2018). On cognitive preferences and the interpretability of rule-based models. *CoRR*, *abs/1803.01316*. Retrieved from http://arxiv.org/abs/1803.01316

Goldwater, M. B., & Gentner, D. (2015). On the acquisition of abstract knowledge: Structural alignment and explication in learning causal system categories. *Cognition*, *137*, 137–153.

Gulwani, S., Hernandez-Orallo, J., Kitzelmann, E., Muggleton, S. H., Schmid, U., & 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., & 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. *Proceedings of the National Academy of Sciences*, *93*(1), 438–442.

Kitzelmann, E., & Schmid, U. (2006). Inductive synthesis of functional programs: An explanation based generalization approach. *Journal of Machine Learning Research*, *7*, 429–454. Retrieved from http://www.jmlr.org/papers/v7/kitzelmann06a.html

Lakkaraju, H., Bach, S. H., & 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* (pp. 1675–1684).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436.

Marcus, G. (2018). Deep learning: A critical appraisal. *CoRR*, *abs/1801.00631*. Retrieved from http://arxiv.org/abs/1801.00631

Markman, A. B., & Gentner, D. (1996). Commonalities and differences in similarity comparisons. *Memory & Cognition*, *24*(2), 235–249.

Michie, D. (1988). Machine learning in the next five years. In *Proceedings of the Third European Working Session on Learning* (pp. 107–122). Pitman.

Minsky, M. (Ed.). (1968). *Semantic information processing.* MIT Press.

Mitchell, T. (1997). *Machine learning.* McGraw Hill.

Muggleton, S., & 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., Schmid, U., Zeller, C., Tamaddoni-Nezhad, A., & Besold, T. (2018). Ultra-strong machine learning: comprehensibility of programs learned with ilp. *Machine Learning*, *107*(7), 1119–1140.

Rabold, J., Siebers, M., & Schmid, U. (2018). Explaining black-box classifiers with ILP – empowering LIME with Aleph to approximate non-linear decisions with relational rules. In F. Riguzzi, E. Bellodi, & R. Zese (Eds.), *Proc of the 28th International Conference (ILP 2018, Ferrara, Italy, September 2-4)* (Vol. 11105, pp. 105–117).

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. In *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144).

Schlimmer, J. C. (1987). Concept acquisition through representational adjustment.

Schmid, U., & Kitzelmann, E. (2011). Inductive rule learning on the knowledge level. *Cognitive Systems Research*, *12*(3), 237-248.

Schmid, U., & Waltermann, J. (2004). Automatic synthesis of xsl-transformations from example documents. In M. Hamza (Ed.), *Artificial Intelligence and Applications Proceedings (AIA 2004)* (pp. 252–257).

Siebers, M., & Schmid, U. (2018). Was the year 2000 a leap year? stepwise narrowing theories with metagol. In F. Riguzzi, E. Bellodi, & R. Zese (Eds.), *Proc of the 28th International Conference (ILP 2018, Ferrara, Italy, September 2-4)* (Vol. 11105, pp. 141–156). Retrieved from https://doi.org/10.1007/978-3-319-99960-9_9 doi: 10.1007/978-3-319-99960-9_9

Siebers, M., Schmid, U., Seuß, D., Kunz, M., & Lautenbacher, S. (2016). Characterizing facial expressions by grammars of action unit sequences–A first investigation using ABL. *Information Sciences*, *329*, 866–875.

Tintarev, N., & Masthoff, J. (2015). Explaining recommendations: Design and evaluation. In *Recommender systems handbook* (pp. 353–382). Springer.