Learning Decision Rules or Learning Decision Models?

Christian de Sainte Marie(✉)

IBM France Lab, Bâtiment Lisp 2 rue d'Arsonval, 91400 Orsay, France
csma@fr.ibm.com

Abstract. In this position paper, we discuss the reasons for the lack of success of rule learning, as witnessed by the quasi-absence of commercial applications, and what can be done to revive the domain and possibly to kick-start the kind of explosive development that statistical Machine Learning and Neural Networks have experienced over the past 15 years. The root cause of the problem is well-known, and it is not the rule learning algorithms themselves: if the representation language to which a rule learning algorithm has access – often only the representation model of the input data – is not appropriate to represent the decision rules, the algorithm has no way to generate a decision ruleset that generalizes well. Feature generation and other methods that have been proposed to augment the data representation language are useful, but we argue that the focus should be on discovering the conceptual model that underly the decision. We claim that this amounts to discovering the structure of decisions, that is, to learn decision models. We outline some potentially fruitful research directions, and how this topic is central to neuro-symbolic learning.

Keywords: Rule learning · Concept discovery · Decision modeling · Neuro-symbolic learning

1 Introduction

Learning rules from precedents has been a subject since Artificial Intelligence emerged as a research domain in the 50's. Rule-based systems have been mainstream in commercial decision assistance and automation at least since the 90's, e.g. as business rules management systems. Still, rule learning from data is nowhere close to be as successful as other Machine Learning techniques such as Neural Networks (NN), be it in terms of scientific, technical, social or commercial impact: none of the main commercial business rules management systems offers even the simplest rule learning capability, for instance.

In this paper, we examine what, we believe, are the reasons why learning rules from precedents has not been more successful, and what these reasons tell us about potentially fruitful research directions.

Let us start with a look at another machine learning technology, namely Neural Networks. The bases for most of modern machine learning technology existed essentially already in the early 90' (see e.g. [1]). Of course, progresses were made after that, but why

S. Moschoyiannis et al. (Eds.): RuleML+RR 2021, LNCS 12851, pp. 276–283, 2021.
https://doi.org/10.1007/978-3-030-91167-6_19

is it that we had to wait until the second half of the 2000's to witness the rebirth of Artificial Intelligence in the guise of (NN-based) Machine Learning and that explosive growth of scientific results, technology and applications? One usual answer to that question is that what unlocked the domain was the new availability (and affordability) of both data and computing power. But there was something else, otherwise (i) the onset of modern AI would have started earlier, and (ii) the affordable access to data and computing power should have benefited to symbolic ML as much as it did to statistical ML. That something else, which is to be found, at least for an important part, in Hinton's et al. paper [2], was a way to train arbitrarily deep neural networks by training each layer independently.

The situation of rule learning– and more generally of symbolic learning – is comparable to that of neural networks in the early 2000's: the technology and the algorithmic basis for learning rules from precedents, essentially existed already in the early 90's, e.g. CN2, FOIL, RIPPER [3], or decision trees, e.g. C4.5 [4], and improvements since then focused on performance and handling noisy data, but no breakthrough like Deep Learning has happened yet in symbolic learning to unlock the field.

In Sect. 2, we explain with an experiment what we believe is the main lock that holds rule learning from progressing at the same speed as neural networks and other kinds of statistical machine learning, and why decision rule learning should now focus on generating the best hypotheses space rather than on generating the best hypotheses in the given representation space. In Sect. 3, we outline some consequences of this change of focus on research directions and priorities. In Sect. 4, we point to relevant related work. Finally, in the conclusion, we propose a new interpretation of decision model, in view of our previous analysis, that justifies the title of this paper.

2 What is the Problem?

Rule learning algorithms can only learn rules that are accessible in their hypotheses space [5]. As obvious as this statement may seem, we believe that this is the root cause of the lack of success of rule learning technology.

Indeed, in the absence of additional knowledge, the representation language that is accessible to the rule learning algorithm to generate the conditions of candidate rules is made of straightforward tests on the input data attributes: rule learning algorithms typically generate hyper-rectangles in the input data space as Boolean combinations of tests that compare the value of an attribute of the tested instance in the input data space with a constant value in the domain of that attribute.

For instance, consider the simple example of learning rules to automate the decision to accept or reject loan applications from customers, based on customer details collected on their application forms. In the "miniloan" use case from the IBM ODM tutorial [6]. the data that is required for the sample ruleset to make the decision is the applicant and loan identifiers, the requested amount, interest rate and duration of the loan, and the revenue and credit score of the applicant[1].

[1] Typically, additional data such as the applicant home address, household etc. will be available, some of it completely unrelated to the decision at hand and making the learning problem more difficult. However, this simple example will be enough for our purpose.

A simple experiment will help make our point clear. We generated instances in the "miniloan" input space described above, with a uniform distribution, and we labelled them using the "miniloan" ruleset. Then we trained a decision tree with the instances[2]. Here is a typical example of a rule extracted from the decision tree[3]:

```
   if CreditScore > 199 and ApplicantIncome <= 17020 and
LoanAmount <= 118303
   then "Reject" loan request
```

The performance on the test set is acceptable (score >97% with 4,000 training instances), but the rules learnt do not generalize well, in the sense that adding new data points to the training set will generate additional rules: indeed, the size of the decision tree grows linearly with the number of training instances (see Fig. 1a, below).

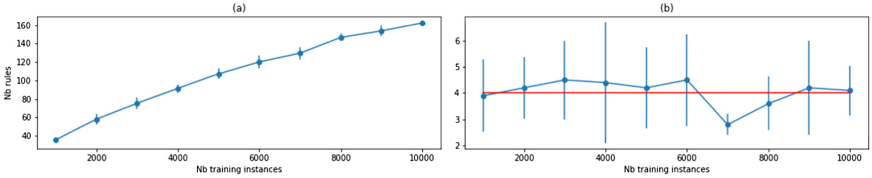

Fig. 1. Number of rules learnt as a function of the number of training instances, when the decision tree is trained on: (a) the initial representation of the instances; (b) *blue*: the representation learnt from a trained 5-5-5-3 multi-layer perceptron[4]; *red*: the initial representation augmented with repayment amount and debt to revenue ratio. (Color figure online)

One way to explain this is because, in the general case, the conceptual model that is required to make the decision is not the same as the conceptual model that is relevant to represent the data: in our credit approval example, it may make sense for a bank to describe loan requests with the attributes listed above, but when it comes to approve or reject them, one will probably rather think in terms of the reimbursement capacity of the applicant, of the sustainability of that capacity over the duration of the loan etc. A simplified meaningful rule of thumb for approving a loan could be:

If the Applicant is Reliable and Capable of Paying the Terms, Then Approve the Loan
We can see the traces of such a rule in our example, above: the test on '`CreditScore`' checks the applicant reliability, the pair of tests on '`ApplicantIncome`' and '`LoanAmount`' checks the repayment capacity. But except for the credit score attribute, which is readily available in the input data, the hypotheses space does not contain a compact and robust representation for the concepts related to reimbursement capacity, for instance, such as that of periodic repayment amount and debt to revenue

[2] We used Scikit Learn DecisionTreeClassifier, which is an optimised version of the CART algorithm [7] and Scikit Learn MLPClassifier for the further experiment, reported below. In both cases, the performance is computed using the classifier "score" method on a test set.

[3] Each leave in a decision tree defines a rule, where the condition is the conjunction of the tests along the path and the leave specifies the corresponding decision.

[4] Other configurations produce similar results. See: https://github.com/cfmrsma/RuleML21.

ratio. When the decision tree is trained, instead, on instances that include these additional features, the number of rules remains constant as soon as the number of training instances has been sufficient for the algorithm to learn the initial “miniloan” rules (Fig. 1b, red line).

The well-known tendency of rule learning algorithms to overfit the training data with large number of rules, especially in the presence of noise, and the brittleness of the learnt rulesets are consequences of trying to approximate the decision rules in an inappropriate representation space. In addition, in the absence of an explicit representation of the relevant concepts, the rules cannot be easily interpreted from a decision maker or business point of view.

These problems are not specific to our decision tree algorithm: any rule learning algorithm will exhibit them in one form or another, including ones using completely different approaches to candidate rule generation, such as column generation [8].

Another simple experiment will help clarify our point that an appropriate change of representation can be learnt that decorrelates the number of rules from the number of training instances (thus relieving our primary symptom that the problem is the representation space rather than the learning algorithm). We trained a multi-layer perceptron with the same training instances as our initial decision tree, then we retrieved and discretized, for each training instance, the values of the nodes in the last hidden layer of the neural network, and we trained a decision tree in that transformed representation space. Figure 1b clearly shows that, with the learnt representation, the number of rules does not depend on the number of training instances (with similar performance).

We have stated earlier that the root cause of the lack of success of rule learning was that rule learning algorithms can only learn rules that are accessible in their hypotheses space. More precisely, the problem is that rule learning algorithms focus on generating the best candidate rules in their hypotheses space, which is of course required. But to be useful – and thus to be more widely used – they will have to focus also on generating the best hypotheses space for the decision problem at hand.

3 Research Directions and Approaches

We claim that the focus of rule learning, and more generally symbolic learning, should move to discovering the best hypotheses space, which requires discovering concepts and learning representations. The capability to learn representations is what made the success of Deep Learning and other statistical learning approaches, and it is what is missing for rule learning to succeed; with the difference that, here, we need to learn symbolic representations.

By learning symbolic representations, we imply (i) discovering the concepts that are useful or required to learn good decision rules and (ii) grounding these concepts in the input data space. This, in turn, requires a definition of what are good decision rules as well as research on how it can be measured; the same holds for what makes a concept useful or required to learn good rules. The next question is: how to learn those symbolic concepts? And, finally, how to ground them in the input space, that is, how to define them in terms of the observed features.

Understanding What Makes a Good Rule Good and a Useful Concept Useful
Ideally, we want to learn rules that are both correct and necessary. A rule is (i) correct if it proposes the correct decision in all instances that satisfy its condition, regardless of their other features: such rules will explain past decisions, propose the correct decision in future cases and permit counterfactual reasoning and what-if analyses; (ii) necessary if it cannot be replaced by a more general rule, covering, in the input space, more instances of interest with respect to the decision: such rules are more likely to rely on features that are essential to the decision. We want to learn a ruleset that covers the whole input space[5], but we also want to learn rules that generalize as much as possible, in the sense that they cover as many instances of interest as possible.

In addition, we want to learn rules that make sense to the user: indeed, ease of modification by the decision authority is, beside auditability and traceability, one of the most important motivations for rule-based decision automation or assistance.

In short, we want rules and rulesets that are compact, robust, necessary, general and understandable, all of which characterize explanations with a high explanatory power [9]: indeed, what makes a rule good is its capacity to explain a decision.

Therefore, we propose that the measure of the quality of a rule or ruleset (its goodness), and the objective that we want a learning algorithm to maximize for the learnt ruleset, should be its explanatory power. Let us stress, here, that we are not seeking high explanatory power because we want to use rules as a mean to provide explainable decisions, but that explanatory power seems to be an adequate measure of the good properties that we want from our decision rules, whatever the reasons why we want to use a rule-based approach: from the point of view of rule learning, the capacity of good rules to provide good explanations is a (much appreciated!) side benefit.

Our proposal does not offer ready-made solutions, but it may indicate fruitful new research directions. A good test for explanatory power could be, for instance, the capacity of a rule to explain noisy data away: a rule with a high explanatory power is robust to noise because it is "truer than the data", in the same sense that, in physics, theory is stronger than measurements[6]: if your instrument measures a force between two macroscopic masses that disagrees with Newton's attraction law, you will suspect your instrument, not Newton. In the same way, if the rule says that loans to reliable applicant with the capacity to reimburse are to be approved, if there is data about an applicant who satisfied the condition and was rejected, one should be able to conclude that either the data about the applicant is wrong, or the rejection was a mistake.

Learning a Useful Conceptual Model
The possibility to learn good rules depends critically on the rule representation space, as shown also by our simple experiment above: per the proposed criterion for rule goodness, rule learning requires the identification or discovery of the features or concepts that contribute most to the explanatory power of a rule, e.g. the features that are inherent to (or constitutive of) the different decisions in the problem at hand.

In the absence of further knowledge, the conceptual model that is most useful to learn or discover would be the, possibly latent, variable model that best explains the

[5] More precisely: the part of the input space that is relevant with respect to the decision at hand.

[6] That is, until measurements break the theory, and the theory must be changed, of course....

relations between input data and observed decisions. We will not review here the abundant literature on identifying latent variables and on learning representations (but see [10]). Let us only stress that learning causal representations [11] might prove especially important, since our proposed definition of a useful concept may well boil down to that of a causal variable in a decision… It might also be useful to revisit earlier work on induced generalization structures such as formal concept analysis [15].

Grounding Symbolic Concepts
The reader will have noticed that there is no restriction to learning *symbolic* representations in the text above. Indeed, once a useful latent feature has been identified, it can be assigned a symbol, and that symbol used to learn rules – symbolic rules – as any other symbol in the representation language used by the learning algorithm, as we did in the experiment described above. If identified as a node in the hidden layer of a neural network, that feature is specified implicitly by its grounding in the input data.

If an explicit definition of the symbols is required, approaches such as symbolic regression [ref] can be applied. Symbolic representations and their explicit definitions can also be learnt bottom up from the data: constructive induction, predicate invention, pattern mining (see review in [12]), and automated feature construction (see e.g. [13]) are important research topics in the proposed change of focus in rule learning.

Let us, however, notice that an explicit definition is not always required: the explicit semantics of symbols that have only a sub-symbolic grounding (e.g. in a neural network) would be defined by their use in a knowledge base. Garnelo et al. describe interesting preliminary work in that direction [14]. That approach to the combination of symbolic and sub-symbolic AI seems like a worthy research direction to us.

Once symbols have been identified, the concept discovery process can be iterated to identify another layer of concepts (lower or higher level, depending whether a top-down or bottom-up approach is used), each layer being defined in terms of lower layers, until the most basic concepts can be meaningfully defined in terms of the input features. Indeed, in most cases, we expect that the conceptual model that is useful to explain a decision will be a graph structure where higher level concepts are defined in terms of lower-level ones.

4 Related Works

The subject matter of this paper is, obviously, closely related to the research on feature generation, representation learning, representation change, neuro-symbolic learning, as well as to approaches to combine logic and neural networks, and probably machine learning in general, and we have tried to make that relation clear by referencing relevant work and surveys in the previous sections.

But we are aware of only few articles that analyze the missing link between symbolic learning and success, as we do in this paper. Two recent papers put the same stress as we do on learning higher-level representations and how it could unlock the progress of symbolic learning. Kramer [12] reviews techniques to learn symbolic higher-level representations and concludes, as we do, that they are useful both to improve symbolic learning and as a first step towards converging symbolic and sub-symbolic learning, because they are able to learn structures of symbolic representations with different levels of abstraction.

Fürnkranz et al. [16] focus on learning structured rule sets as a way to avoid artificial rule ordering mechanisms such as weights and claim that learning auxiliary concepts is useful for that purpose.

Kramer's and Fürnkranz et al. conclusions are close to ours, and so is probably their initial thinking as well. However, neither goes as far as claiming, as we do, that learning the appropriate structured conceptual models is a general requisite for successful symbolic learning and should therefore become the focus of this community.

Let us also stress that Bengio et al. review on representation learning [10] is extremely relevant to (and congruent with) the analysis presented here, although it is concerned with sub-symbolic representations: discovering the useful representations is a necessary step and, as we have claimed above, the grounding step may be separated from the discovery step, when symbolic grounding is required – and it is not necessarily required.

Finally, recent works on differentiable logics (e.g. [17]) may open different doors to make symbolic learning benefit from the advances in neural networks.

5 Revisiting Decision Modeling as a Conclusion

We are referring here to decision modeling as a method used by business analysts for identifying, specifying, analyzing and communicating decision, separately from (and possibly in conjunction with) the specification of business processes [18]. The method and a notation for decision models are the subject of the Decision Model and Notation standard [19]. An important characteristic of Decision Models is that they enforce a clean separation between the structure of a decision and the decision logic; that is, between the data requirements for a decision, and the decision rules.

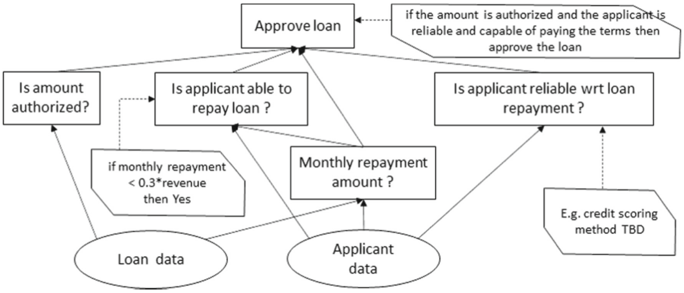
Approve loan
if the amount is authorized and the applicant is reliable and capable of paying the terms then approve the loan
Is amount authorized?
Is applicant able to repay loan ?
Is applicant reliable wrt loan repayment ?
if monthly repayment < 0.3*revenue then Yes
Monthly repayment amount ?
E.g. credit scoring method TBD
Loan data
Applicant data

Fig. 2. A DMN decision model for the "miniloan" example [6]. Input data is represented as rounded forms, decisions as rectangles, decision logic as rectangles with two cut angles; plain arrows represent data flows, dashed arrows represent knowledge flow.

Figure 2 shows a DMN decision model for our loan approval example. Not surprisingly, the structure of the sub-decisions matches exactly the conceptual model that is required to explain the decision and to make it in a reasoned way (as opposed to making it on a purely statistical basis). We claim that this is an essential property of decision models: each sub-decision represents a required concept, and the associated decision logic specifies how that concept is grounded in other concepts, down to the input data.

A decision model does not only specify the conceptual model that is required to make a decision: it specifies also the chain of representation changes that ground that decision into the input data.

The introduction of decision modeling is a major paradigm shift in the decision automation industry, as it shifts the focus from the decision rules to the complete decision structure. We claim that the same shift is necessary to the success of symbolic (decision logic) learning, and that the symbolic learning research community should move its focus from rule learning to learning decision models.

References

1. Haohan, W., Bhiksha R.: On the origin of deep learning. arXiv:1702.07800 (2017)
2. Hinton, G.E., et al.: A fast learning algorithm for deep belief nets. Neural Comput. **18**, 1527–1554 (2006)
3. Furnkranz, J.: Separate-and-conquer rule learning. Artif. Intell. Rev. **13**(1), 3–54 (1999)
4. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers (1993)
5. Mitchell, T.: Generalization as Search. Artif. Intell. **18**, 203–226 (1982)
6. https://github.com/ODMDev/odm-for-dev-getting-started
7. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. 1st edn. Routledge (1984). https://doi.org/10.1201/9781315139470
8. Dash, S., Günlük, O., Wei, D.: Boolean decision rules via column generation. In: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018) (2018)
9. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intell. **267**, 1–38 (2019)
10. Bengio, Y., et al.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 1798–1828 (2013)
11. Scholkopf, B., et al.: Toward causal representation learning. Proc. IEEE **109**, 612–634 (2021)
12. Kramer, S.: A brief history of learning symbolic higher-level representations from data (and a curious look forward). In: IJCAI (2020)
13. Sondhi, P.: Feature construction methods: a survey (2009)
14. Garnelo, M., et al.: Towards deep symbolic reinforcement learning. ArXiv abs/1609.05518 (2016)
15. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Berlin (1999). https://doi.org/10.1007/978-3-642-59830-2
16. Fürnkranz, J., et al.: Learning structured declarative rule sets - a challenge for deep discrete learning. ArXiv abs/2012.04377 (2020)
17. Shindo, H., et al.: Differentiable inductive logic programming for structured examples. In: AAAI (2021)
18. Fish, A.: Melding process models and decision models. Modeling decision-making processes. https://dmcommunity.files.wordpress.com/2016/06/decisioncamp2016-alanfish.pdf
19. OMG, Decision Model and Notation. https://www.omg.org/spec/DMN