Heuristic Minimization Modulo Theory of Modal Decision Trees Class-Formulas

Giovanni Pagliarini[1], Andrea Paradiso[1], Sasha Rubin[2], Guido Sciavicco[1] and Ionel Eduard Stan[3]

[1]University of Ferrara, Italy
[2]University of Sydney, Australia
[3]Free University of Bozen-Bolzano

Abstract
One way towards the design of trustable, explainable, and interpretable artificial intelligence models is to focus on symbolic machine learning models, such as decision trees. While decision trees are already intelligible in principle, the logical rules they enclose may still be redundant, in particular with respect to some underlying theory. Moreover, propositional decision trees have been recently generalized to the case of modal logic; modal decision trees turn out to be more expressive than propositional ones, so their corresponding modal rules are proportionally harder to understand and minimize. In this paper we approach the problem of minimizing logical rules extracted from (modal) decision trees modulo some external theory.

Keywords
decision trees, formula minimization,

1. Introduction

Decision trees are part of a collection of logic-based learning methods that include learning DNF/CNF formulas, Horn formulas, and decision lists, and characterized by the implicit assumption that each instance of a dataset can be seen as a logical model. Among such methods, *decision trees* are probably the most successful one. Learning an optimal decision tree, that is, a decision tree with a minimum number of nodes, is an Σ_1^P-hard problem [1], so learning decision trees is often accomplished with heuristics such as entropy-based algorithms (e.g., CART [2], C4.5 [3], or ID3 [4]). Classic decision trees are propositional. Typically, edge conditions are simple literals (e.g., $Age > 45$, $Gender = M$), so given the set $\mathcal{V}$ of the *variables* of the problem, one can fix a *propositional vocabulary* $\mathcal{AP} = \{V \bowtie v \mid V \in \mathcal{V}, v \in \mathbb{R}\}$, where $\bowtie \in \{<, \leq, =, \geq, >\}$ and then interpret edge conditions as literals built from propositions in $\mathcal{AP}$. As a consequence, a branch in a classic propositional decision tree is seen as a logical term, that is, a conjunction

OVERLAY 2023: 5th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, November 7, 2023, Rome, Italy
giovanni.pagliarini@unife.it (G. Pagliarini); andrea.paradiso@edu.unife.it (A. Paradiso); sasha.rubin@sydney.edu.au (S. Rubin); guido.sciavicco@unife.it (G. Sciavicco); ioneleduard.stan@unibz.it (I. E. Stan)
0000-0002-8403-3250 (G. Pagliarini); 0000-0002-3614-2487 (A. Paradiso); 0000-0002-3948-129X (S. Rubin); 0000-0002-9221-879X (G. Sciavicco); 0000-0001-9260-102X (I. E. Stan)

of literals, and a class C is identified by the union of branches that are labeled with it; the corresponding formula, which is a disjunction of conjunctions of literals, that is, a DNF formula, is called *class-formula*. In this sense, each *data point*, or *instance*, of a dataset, can be seen as a propositional interpretation. If φ is the class-formula for C in the tree τ, from a logical point of view a decision tree assigns an instance I to the class C if I, viewed as propositional interpretation, satisfies φ.

To overcome some of the limitations of propositional decision trees, several generalizations have been proposed; a very recent one consists of replacing propositional logic with *modal logic* [5]. *Modal decision trees* are based on the idea that instances of non-scalar datasets (e.g., time series, texts, images, videos, and graphs) can be seen as finite Kripke structure (i.e., a certain type of transition system). Thus, in the simplest case, a (non-scalar) instance I is described as a tuple $I = (W, R, V)$, where W is a finite set of *worlds* (among which an *initial world* w_0 is identified), $R \subseteq W \times W$ is an *accessibility relation*, and $V : W \rightarrow 2^{\mathcal{A}P}$ is a *valuation function* that maps every world to the subset of propositional letters that are true on it. It can be shown that most types of data points can be, in fact, seen as Krikpe structure, allowing one to effectively generalize propositional learning to modal (propositional) learning (instances with a single world are, in fact, propositional). Modal decision trees have been introduced in [6] in their temporal form, and later extended and applied to a variety of domains and tasks [7, 8, 9, 10, 11]), and their properties have been studied in [5]. Modal decision trees have a general, syntactical definition, which can be instantiated with a specific modal language $\mathcal{L}$; real-world non-scalar datasets can be seen as sets of multi-relation Kripke models, and real-world modal logics used in learning are in fact multi-modal logics with high expressive power. However, most basic results can be stated in the simple case of single-relation structures and uni-modal logic, and then easily generalized. Learning modal decision trees is accomplished with heuristic learning algorithms inspired by their propositional counterparts. Class-formulas from modal decision trees can be extracted as in the propositional case, and they have the form of conjunctions of disjunctions of formulas (each called a *path-formula*); as before, an instance I is classified into a class C by a decision tree τ if and only if I, seen as a Kripke interpretation, satisfies φ on the initial world w_0, where φ is the class-formula extracted from τ for C.

Since decision tree learning algorithms are sub-optimal, class-formulas may display some kind of redundancy. At the propositional level, for example, a very simple case occurs when a product includes the conjunction of two literals of the type $V > v$ and $V > v'$, being $v \geq v'$; clearly, $V > v'$ can be omitted. More generally, one can consider the situation in which a given learning problem is linked to a finite *theory* $\mathcal{T}$, possibly provided by an expert and/or induced by the nature of propositional letters, that may help simplifying the learned model; for example, we may know that $Fever > 38$ implies $Headache = True$, so that a term that contains both literals may, again, be simplified. In this paper, we want to define the problem of simplifying class-formulas modulo a theory, and discuss an initial approach to its solution.

2. Minimization of Modal Class-Formulas Modulo Theory

Given a (modal) decision tree for a fixed language $\mathcal{L}$ and vocabulary $\mathcal{AP}$, a theory $\mathcal{T}$ written in the language $\mathcal{L}$, a class C, and the class-formula φ for C, we can formulate the problem of

Algorithm 1: TheoryMinimize.

function *TheoryMinimize*($\varphi, \mathcal{T}$):
  return *TMin*($\varphi, Null, \mathcal{T}$)
end
function *TMin*($\varphi, \varphi_{parent}, \mathcal{T}$):
  if $\varphi \neq \lambda_{lit}$ **then**
    if $\varphi = \xi_1 \wedge \xi_2$ **then** $\varphi \leftarrow$ (*TMin*($\xi_1, \varphi, \mathcal{T}$) $\wedge$ *TMin*($\xi_2, \varphi, \mathcal{T}$))
    else if $\varphi = \xi_1 \rightarrow \xi_2$ **then** $\varphi \leftarrow$ (*TMin*($\xi_1, \varphi, \mathcal{T}$) $\rightarrow$ *TMin*($\xi_2, \varphi, \mathcal{T}$))
    else if $\varphi = \Diamond\xi$ **then** $\varphi \leftarrow \Diamond$*TMin*($\xi, \varphi, \mathcal{T}$)
    else if $\varphi = \Box\xi$ **then** $\varphi \leftarrow \Box$*TMin*($\xi, \varphi, \mathcal{T}$)
    if $\varphi \neq \xi_1 \wedge \xi_2$ **then** $\varphi \leftarrow Mark(\varphi)$
    else if $\varphi = \xi_1 \wedge \xi_2$ **and** $\varphi_{parent} \neq \xi_1 \wedge \xi_2$ **then**
      $(\overline{\varphi}, Map) \leftarrow$ *ReplaceMarkedSubFormulasWithNewLetters*(φ)
      $\overline{\varphi}' \leftarrow$ *CNFMin*($\overline{\varphi} \wedge \mathcal{T}$)
      $\varphi \leftarrow$ *ReplaceLettersWithSubFormulas*($\overline{\varphi}', Map$)
  return φ
end

finding the smallest (in terms of number of symbols) φ' equivalent to φ in every model in which $\mathcal{T}$ holds universally; in other words, we ask that for every instance I such that for every world w it is the case that $I, w \Vdash \psi$ for each $\psi \in \mathcal{T}$, it so happens that $I, w_0 \Vdash \varphi$ if and only if $I, w_0 \Vdash \varphi'$. In the general case of modal decision trees, class-formulas have the form

$$(\varphi_1^1 \wedge \varphi_2^1 \wedge \ldots \wedge \varphi_{n_1}^1) \vee \ldots \vee (\varphi_1^m \wedge \varphi_2^m \wedge \ldots \wedge \varphi_{n_m}^m),$$

where each φ_i^j (called *path-formula*) is a formula that belongs to a specific grammar. So, this problem is at least Σ_2^P-hard, as it can be reduced to the propositional DNF minimization with an empty theory [12], and it is its natural generalization to the case of decision trees.

A two-step sub-optimal approach towards the solution to the above problem, in the particular case in which $\mathcal{T}$ only contains propositional implications of the type $(\lambda_1 \wedge \lambda_2 \wedge \ldots \wedge \lambda_z \rightarrow \lambda)$ (where λ and each λ_i is a propositional literal); the efficiency of such an approach depends, among other aspects, on the efficiency of the algorithm *Norm(φ)* that, given φ, returns its logical negation φ' (in the same grammar).

In step one (Alg. 1) we take advantage from the fact that class-formulas present typical patterns for which the theory can be exploited towards a simplification, considering each path-formula of each term $(\varphi_1^j \wedge \varphi_2^j \wedge \ldots \wedge \varphi_{n_j}^j)$ individually. First, the procedure inductively searches for a maximal subtree of the syntax tree of the considered formula that is a conjunction. Then, it substitutes every conjunct that is not a literal with a fresh propositional letter (*ReplaceMarkedSubformulasWithNewLetters*), using a map to keep trace of each substitution. Finally, it delegates the process of minimizing the size of the obtained conjunctive formula within the theory to an heuristic minimization algorithm *CNFMin*, before replacing back the fresh letters with the original subtrees (*ReplaceLettersWithSubFormulas*). The call *CNFMin($\lambda_1 \wedge \lambda_2 \wedge \ldots \wedge \lambda_n \wedge \mathcal{T}$)* returns a conjunction $\lambda_{s_1} \wedge \ldots \wedge \lambda_{s_l}$ ($l \leq n$, $s_1, \ldots, s_l \in [1, n]$) such that for every λ_i with

Algorithm 2: PropositionalMinimize.

```
function PropositionalMinimize(φ):
    (φ̄, Map) ← ReplaceConjunctsWithNewLiterals(φ)
    φ̄′ ← DNFMin(φ̄)
    return ReplaceLiteralsWithConjuncts(φ̄′, Map)
end
function ReplaceSubFormulasWithNewLiterals(φ):
    Map ← ∅
    foreach ξ ∈ Conjuncts(φ) do
        if Map[ξ] does not exist then
            t̃ ← NewLetter()
            Map[ξ] ← t̃
            Map[Norm(¬ξ)] ← ¬t̃
        ξ ← Map[ξ]
    return (φ, Map)
end
```

$i \notin \{s_1, \ldots, s_l\}$ it is the case that $\lambda_{s_1} \wedge \ldots \wedge \lambda_{s_l} \wedge \mathcal{T} \rightarrow \lambda_i$ is valid, and that the subset $\{\lambda_{s_1}, \ldots, \lambda_{s_l}\}$ is minimal. A procedure *CNFMin* as we have described it can be obtained by simply adapting a deletion-based procedure, namely, *plain deletion-based MES extraction*, from [13]. As an example, the term

$$\Box(p \wedge \neg q \wedge \Diamond r \wedge s) \wedge t$$

would be reduced to the term

$$\Box(\neg q \wedge \Diamond r) \wedge t$$

if $\mathcal{T}$ contained the implications $\neg q \rightarrow p$ and $\neg q \rightarrow s$.

In step two (Alg. 2), we operate on the whole class-formula by uniformly substituting every (top-level) conjunct in every term of the class-formula being considered with a fresh literal (*ReplaceConjunctsWithNewLiterals*), delegating the size minimization of the resulting DNF formula to a procedure *DNFMin* [14, 15], and performing a backward, consistent substitution (*ReplaceLiteralsWithConjuncts*). The correctness of this approach is based on the auxiliary function *Norm*, whose existence we assumed. As an example, the class-formula

$$(\varphi_1 \wedge \neg\varphi_2 \wedge \varphi_3) \vee (\varphi_1 \wedge \varphi_3)$$

would be reduced to the class-formula

$$\varphi_1 \wedge \varphi_3.$$

3. Conclusions

We defined the problem of minimization of class-formulas extracted from (modal) decision trees, and we proposed an initial, heuristic approach to its solution.

References

[1] L. Hyafil, R. L. Rivest, Constructing optimal binary decision trees is NP-complete, Information Processing Letters 5 (1976) 15–17.

[2] L. Breiman, J. Friedman, R. Olshen, C. Stone, Classification and regression trees, Wadsworth Publishing Company, 1984.

[3] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[4] J. R. Quinlan, Simplifying decision trees, International Journal of Human-Computer Studies 51 (1999) 497–510.

[5] D. Della Monica, G. Pagliarini, G. Sciavicco, I. Stan, Decision trees with a modal flavor, in: Proc. of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA), number 13796 in LNCS, Springer, 2023, pp. 47 – 56.

[6] A. Brunello, G. Sciavicco, I. Stan, Interval temporal logic decision tree learning, in: Proc. of the 16th European Conference on Logics in Artificial Intelligence (JELIA), volume 11468 of *LNCS*, Springer, 2019, pp. 778–793.

[7] F. Manzella, G. Pagliarini, G. Sciavicco, I. Stan, Interval Temporal Random Forests with an Application to COVID-19 Diagnosis, in: Proc. of the 28th International Symposium on Temporal Representation and Reasoning (TIME), volume 206 of *LIPIcs*, Schloss Dagstuhl, 2021, pp. 7:1–7:18.

[8] F. Manzella, G. Pagliarini, G. Sciavicco, I. Stan, The voice of COVID-19: Breath and cough recording classification with temporal decision trees and random forests, Artificial Intelligence in Medicine (2023) 1 – 14.

[9] G. Pagliarini, G. Sciavicco, Decision Tree Learning with Spatial Modal Logics, in: Proc. of the 12th International Symposium on Games, Automata, Logics, and Formal Verification (GANDALF), volume 346 of *EPTCS*, 2021, pp. 273–290.

[10] M. Coccagna, F. Manzella, S. Mazzacane, G. Pagliarini, G. Sciavicco, Statistical and Symbolic Neuroaesthetics Rules Extraction from EEG Signals, in: Proc. of the 9th International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC), volume 13258 of *LNCS*, Springer, 2022, pp. 536–546.

[11] G. Sciavicco, I. Stan, Knowledge extraction with interval temporal logic decision trees, in: Proc. of the 27th International Symposium on Temporal Representation and Reasoning (TIME), volume 178 of *LIPIcs*, Schloss Dagstuhl, 2020, pp. 9:1–9:16.

[12] C. Umans, T. Villa, A. L. Sangiovanni-Vincentelli, Complexity of two-level logic minimization, IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. 25 (2006) 1230–1246.

[13] A. Belov, M. Janota, I. Lynce, J. Marques-Silva, Algorithms for computing minimal equivalent subformulas, Artificial Intelligence 216 (2014) 309–326.

[14] J. Hlavicka, P. Fiser, Boom-a heuristic boolean minimizer, in: Proc. of the IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2001, pp. 439–442.

[15] R. Rudell, A. Sangiovanni-Vincentelli, Multiple-valued minimization for pla optimization, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 6 (1987) 727–750.