Efficient Modal Decision Trees

F. Manzella[1], G. Pagliarini[1], G. Sciavicco[1(✉)], and I. E. Stan[2]

[1] University of Ferrara, Ferrara, Italy
federic.manzella@edu.unife.it, {pglgnn,scvgdu}@unife.it
[2] Free University of Bozen-Bolzano, Bolzano, Italy
ioneleduard.stan@unibz.it

Abstract. Modal symbolic learning is an emerging machine learning paradigm for (non)-tabular data, and modal decision trees are its most representative schema. The underlying idea behind modal symbolic learning is that non-tabular (e.g., temporal, spatial, spatial-temporal) instances can be seen as finite Kripke structures of a suitable modal logic and propositional alphabet; from a non-tabular dataset, then, modal formulas can be extracted to solve classic tasks such as classification, regression, and association rules extraction. Although this paradigm has already been proven successful in different learning tasks, a provably correct and complete formulation of modal decision trees has only recently been found. In this paper, we prove that correct and complete modal decision trees are also efficient, learning-wise.

Keywords: Modal symbolic learning · Decision trees · Efficient implementation

1 Introduction

Symbolic learning is the sub-field of machine learning focused on approaching classic tasks (such as classification or regression) via the extraction of logical formulas from data. While often seen as less versatile and statistically accurate, symbolic learning has the advantage of extracting intelligible information that can be later discussed with the domain experts, corrected if necessary, and combined with background knowledge. Out of all possible symbolic learning schemata, *decision trees* are probably the best known ones, and they are also emblematic for a whole range of other symbolic models, such as *decision lists* [13,40], bootstrap aggregation or *bagging* [7], typically based on independent decision trees as in *random forests* [8], *boosting* [25], and, in particular, *gradient boosted trees* (e.g., [12,24]); *hybrid* models combining the strengths of both symbolic and connectionist methods, when based on decision trees, become *neural-symbolic decision trees* [21,44], and *tree-based neural networks* [2,26,33,34,42,43], among others.

The origin of modern decision trees dates back to [3]. In [32] the authors proposed *Automatic Interaction Detection (AID)* as an alternative to functional

R. Basili et al. (Eds.): AIxIA 2023, LNAI 14318, pp. 381–395, 2023.
https://doi.org/10.1007/978-3-031-47546-7_26

regression. Whereas AID is used for regression tasks, *Theta AID* [31] and *Chi-Squared AID* [23] extend AID for classification tasks by introducing new impurity *information-based* functions. The *Classification and Regression Trees (CART)* [9] method follows the same greedy approach as the AID-based methods, but adds several features as, for example, pruning techniques to regularize the resulting model to cope with overfitting. Later, Quinlan [37] formalized the development of an inductive process for knowledge acquisition, which resulted in the so-called *Iterative Dichotomizer 3 (ID3)* algorithm, extended with pruning techniques some years later by the same author [39], and improved in terms of learning algorithms with the introduction of C4.5 [38] to cope with the main limitation of ID3 of handling only categorical data. A more complete survey on decision trees can be found, for example, in [18,27,28]. *Modal decision trees* have been introduced in [10], in their temporal form, as a generalization of propositional ones, and later extended and applied to a variety of *non-tabular* data, both in the temporal and in the spatial case, such as respiratory diseases diagnosis [30], land cover classification [35], and electroencephalogram recordings reading and interpreting [16], among others [41]. Modal decision trees, and modal symbolic learning in general, are based on the idea that instances of a non-tabular (that is, temporal, spatial, spatial-temporal, but also text-based and graph-based) dataset can be seen as a set of finite Kripke structures, so that modal (that is, temporal, spatial, and so on) logical formulas can be extracted from such a dataset to solve, for example, classification or regression tasks.

Propositional decision trees are *complete* for the classification task with respect to propositional logic, that is, given a dataset, there always exists an *optimal* tree for it, able to correctly classify each of its instances, whose class is identified by a propositional formula. Learning a *minimal* optimal tree from a dataset is NP-hard [40]; thus, sub-optimal algorithms such as information-based algorithms became the de-facto standard (examples of such algorithms include ID3 and C4.5, mentioned above). In this sense, propositional decision trees are also provably *efficient* in terms of learning, because an optimal (but not necessarily minimal) decision tree can be learned by a polynomial-time information-based algorithm from a dataset. While modal decision trees have already been shown to be able to extract useful, accurate, and interpretable models, their properties have only recently been studied [19]. As it turns out, modal decision trees too are complete for the classification task with respect to modal logic. This solves a problem that was open since the first proposals concerning methods for non-propositional logical formulas extraction from data [5,6,17], but leaves as open the question of whether modal decision trees are also efficient, that is, whether there exists a polynomial-time information-based algorithm that learns an optimal decision tree from a dataset. In this paper, we prove that modal decision tree are *weakly efficient*, which means that there exists a polynomial algorithm that, given a dataset, returns a *t-optimal* tree for it, able to classify in the perfect way each of its instances whose class is determined by a modal formula of length less than or equal to t; we also discuss in detail the problem of minimizing the experimental complexity of an implementation of such an algorithm, by leveraging the nature of the modal formulas that are actually examined during the learning process and by exploiting a suitable memoization approach.

2 Propositional Decision Trees

Definition 1. *A* tabular dataset *is a finite collection of* m instances $\mathcal{I} = \{\mathfrak{I}_1, \ldots, \mathfrak{I}_m\}$*, each described by the value of* n variables $\mathcal{V} = \{V_1, \ldots, V_n\}$*, and associated to a unique* label *from a set* $\mathcal{L} = \{L_1, \ldots, L_k\}$*.*

Several problems are associated to tabular datasets: classification (when labels are categorical – in this case they are also called *classes*), regression (when labels are numerical), association rules extraction (when labels are absent or ignored). In the symbolic context, datasets are naturally associated to a logical alphabet $\mathcal{P}$ of propositional letters (which represents an *inductive bias*, in a learning context), from which formulas are built. While in some cases alphabets are the result of a suitable variable selection and/or domain filtration, from a purely methodological point of view we can always assume that:

$$\mathcal{P} = \{V \bowtie v \mid V \in \mathcal{V}, a \in \mathbb{R}, \bowtie \in \{<, \leq, =, \neq, \geq, >\}\},$$

where $\bowtie$ is a *test operator*; whenever necessary, for a given variable V, we shall refer to its *domain*, defined as the set of all and only distinct values that V takes in a given dataset. Tabular datasets can be also defined as *propositional* datasets, as follows.

Definition 2. *A* propositional dataset *is a finite collection of* m instances $\mathcal{I} = \{\mathfrak{I}_1, \ldots, \mathfrak{I}_m\}$*, each described as a propositional model over a given alphabet* $\mathcal{P}$ *and associated to a unique* label *from a set* $\mathcal{L} = \{L_1, \ldots, L_k\}$*.*

The purpose of symbolic classification is to extract from a dataset $\mathcal{I}$ a (set of) logical formula(s) to be used as rule(s) for classifying instances of a dataset $\mathcal{J}$ drawn from the same distribution as $\mathcal{I}$. Decision trees allows one to do so in a very convenient way. In the classical setting, formulas are written in *propositional logic.*

Let $\tau = (\mathcal{V}, \mathcal{E})$ be a *full binary directed tree* with *nodes* in $\mathcal{V}$ and *edges* in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We denote by $\mathcal{V}^\ell$ the set of its *leaf nodes* (or, simply, *leaves*), and by $\mathcal{V}^\iota$ the set of its *internal nodes* (i.e., non-root and non-leaf nodes). Given a tree τ, we denote its *root* by $\rho(\tau)$, and its nodes (either root, internal or leaf) by $\nu, \nu_1, \nu_2, \ldots$ and leaves by $\ell, \ell_1, \ell_2, \ldots$ Each non-leaf node ν of a tree τ has precisely two *children*, the *left child* $\swarrow(\nu)$ and the *right child* $\searrow(\nu)$, and each non-root node ν has a *parent* $\wr(\nu)$. For a node ν, the set of its *ancestors* (ν included) is denoted by $\wr^*(\nu)$, where $\wr^*$ is the transitive and reflexive closure of $\wr$; we also define $\wr^+(\nu) = \wr^*(\nu) \setminus \{\nu\}$, and we say that if $\nu' \in \wr^*(\nu)$, then ν is a *descendant* of ν'. Moreover, given a tree τ, a *path* $\pi = \nu_0 \rightsquigarrow \nu_h$ *in* τ of *length* $h \geq 0$ between two nodes ν_0 and ν_h is a finite sequence of $h+1$ nodes such that $\nu_i = \wr(\nu_{i+1})$, for each $i = 0, \ldots, h-1$. We denote by $\pi_1 \cdot \pi_2$ the operation of *appending* the path π_2 to path π_1. A *branch of* τ is a path π_ℓ, for some $\ell \in V^\ell$. For a path π and a node ν, π_ν denotes the unique path $\rho(\tau) \rightsquigarrow \nu$. Finally, given two paths π_1, π_2, we denote by $\pi_1 \sqsubseteq \pi_2$ the fact that π_1 is a not necessarily proper prefix of π_2.

Definition 3. *Let $\mathcal{I}$ be a dataset with set of classes $\mathcal{L}$ and set of associated propositional letters $\mathcal{P}$, and define the set of* propositional decisions $\Lambda = \{p, \neg p \mid p \in \mathcal{P}\}$. *Then, a* propositional decision tree *(*over Λ*) is an object of the type:*

$$\tau = (\mathcal{V}, \mathcal{E}, l, e),$$

where $(\mathcal{V}, \mathcal{E})$ is a full binary directed tree, $l : \mathcal{V}^\ell \rightarrow \mathcal{L}$ is a leaf-labelling function *that assigns a class from $\mathcal{L}$ to each leaf node in $\mathcal{V}^\ell$, $e : \mathcal{E} \rightarrow \Lambda$ is a* edge-labelling function *that assigns a propositional decision from Λ to each edge in $\mathcal{E}$, such that $e(\nu, \swarrow(\nu)) \equiv \neg e(\nu, \searrow(\nu))$ for all non-leaf node ν. For a path $\pi = \nu_0 \rightsquigarrow \nu_h$ in τ, the* path-formula φ_π *is defined as:*

$$\varphi_\pi = \top \wedge \bigwedge_{\nu_i \in \pi, i<h} e(\nu_i, \nu_{i+1}).$$

For a leaf ℓ, the leaf-formula φ_ℓ *is defined as:*

$$\varphi_\ell = \varphi_{\pi_\ell},$$

and for a class $L \in \mathcal{L}$, the class-formula *is defined as:*

$$\varphi_L = \bigvee_{l(\ell)=L} \varphi_{\pi_\ell}.$$

Finally, the run of τ on $\mathfrak{I}$ from ν, *denoted by $\tau(\mathfrak{I}, \nu)$, is defined as follows:*

$$\tau(\mathfrak{I}, \nu) = \begin{cases} l(\nu) & \text{if} \quad \nu \in \mathcal{V}^\ell; \\ \tau(\mathfrak{I}, \swarrow(\nu)) & \text{if} \quad \mathfrak{I} \models \varphi_{\pi_{\swarrow(\nu)}}; \\ \tau(\mathfrak{I}, \searrow(\nu)) & \text{if} \quad \mathfrak{I} \models \varphi_{\pi_{\searrow(\nu)}}, \end{cases}$$

and the run $\tau(\mathfrak{I})$ *of $\mathfrak{I}$ on τ is simply $\tau(\mathfrak{I}, \rho(\tau))$. An instance $\mathfrak{I}$ is* classified into $L \in \mathcal{L}$ *by τ if and only if $\tau(\mathfrak{I}, \rho(\tau)) = L$.*

Definition 4. *A decision tree is said to be* optimal for *a dataset $\mathcal{I}$ with respect to a logic if and only if, for every instance $\mathfrak{I}$ whose class is identified by a formula of that logic, $\tau(\mathfrak{I}) = L$ if and only if $\mathfrak{I}$ is labelled by L.*

Definition 5. *A family of decision trees is* correct *if and only if every tree classifies every instance into exactly one class. Furthermore, it is* complete *with respect to a logic if and only if, for every dataset, there exists the optimal tree for it with respect to that logic. Finally, it is* efficient *with respect to a logic if and only if there exists a polynomial-time algorithm that, for every dataset, learns an optimal tree for it with respect to that logic.*

It is well-known that the family $\mathcal{DT}$ of propositional decision trees is correct, complete, and efficient with respect of propositional logic.

3 Modal Decision Trees

Symbolic learning is founded on the idea that patterns (in our context, classification patterns) are expressible in propositional logic. Non-tabular data, however, may be too complex to be adequately described using propositional logic. *Modal logic* extends propositional logic by assuming the existence of many propositional *worlds*, connected by *binary relations*. Each world plays the role of a propositional model; in the standard, philosophical formulation, the relation plays the role of *accessibility* among worlds. So, given a set of propositional letters $\mathcal{P}$, formulas of modal logic are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi,$$

where $p \in \mathcal{P}$. The remaining classic Boolean operators can be obtained as shortcuts; similarly, we use $\Box\varphi$ to denote $\neg\Diamond\neg\varphi$. The *modality* $\Diamond$ (resp., $\Box$) is usually referred to as *it is possible that* (resp., *it is necessary that*), and called *diamond* (resp., *box*). The semantics of modal logic is given in terms of Kripke structures. A *Kripke structure*, over $\mathcal{P}$, $\mathfrak{K} = (\mathcal{W}, \mathcal{R}, \mathfrak{V})$ consists of a non-empty (possible infinite, but countable) set of *(possible) worlds* $\mathcal{W}$, a *binary accessibility relation* over worlds $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$, and a *valuation function* $\mathfrak{V} : \mathcal{W} \to 2^{\mathcal{P}}$, which associates each world w with the set of proposition letters $\mathfrak{V}(w) \subseteq \mathcal{P}$ that are true on it. The *truth (relation)* $\mathfrak{K}, w \Vdash \varphi$, for a (Kripke) model $\mathfrak{K}$, a world w (in that model), and a formula φ, is defined by induction on the complexity of formulas:

$$\begin{array}{ll}
\mathfrak{K}, w \Vdash p & \text{iff } p \in \mathfrak{V}(w), \text{ for all } p \in \mathcal{P};\\
\mathfrak{K}, w \Vdash \neg\psi & \text{iff } \mathfrak{K}, w \not\Vdash \psi;\\
\mathfrak{K}, w \Vdash \psi_1 \wedge \psi_2 & \text{iff } \mathfrak{K}, w \Vdash \psi_1 \text{ and } \mathfrak{K}, w \Vdash \psi_2;\\
\mathfrak{K}, w \Vdash \Diamond\psi & \text{iff there exists } w' \text{ s.t. } w\mathcal{R}w' \text{ and } \mathfrak{K}, w' \Vdash \psi.
\end{array}$$

We write $\mathfrak{K} \Vdash \varphi$ as an abbreviation of $\mathfrak{K}, w_0 \Vdash \varphi$, where w_0 is the *initial world* of $\mathfrak{K}$. Modal logic is paradigmatic for propositional temporal, spatial, spatial-temporal logics, as well as description logics, epistemic logics, and many others. Indeed, most classic temporal logics [14,22,36] and spatial logics [1,29] are in fact specializations of modal logic with more than one (possibly non-binary) accessibility relations (and associated modalities), subject to constraints that range from very simple and intuitive ones (e.g., transitivity, antisymmetry) to very complex ones (e.g., when worlds are assumed to be intervals and modalities are assumed to mimic relations between intervals).

Inspired by the generalization from propositional to modal logic, we can now define modal datasets.

Definition 6. *A* labelled modal dataset *is a finite collection of* m instances $\mathcal{I} = \{\mathfrak{I}_1, \ldots, \mathfrak{I}_m\}$*, each described as a finite Kripke structure over a given alphabet* $\mathcal{P}$ *and associated to a unique* label *from a set* $\mathcal{L} = \{L_1, \ldots, L_k\}$*.*

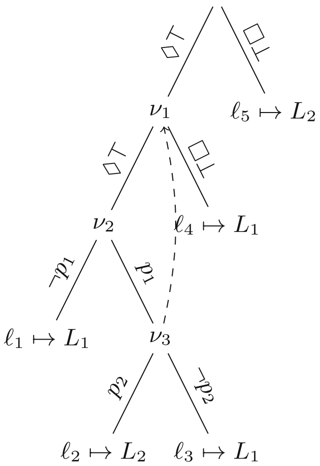
The link between non-tabular datasets and modal ones, as well as the role and the nature of the variables in modal datasets can be explained with an example.

Consider the case of temporal data. In its most general setting, a *temporal dataset* is a collection of m temporal instances, where a temporal instance is a *multivariate time series*, described by a set of n temporal variables $\mathcal{V} = \{V_1, \ldots, V_n\}$, each taking a value at each of N distinct instants. A possible way to extract logical information from a temporal dataset is to consider a set of *feature extraction functions* $\mathcal{F} = \{F_1, \ldots, F_s\}$, each defined as $F_i : \mathbb{R} \times \ldots \times \mathbb{R} \rightarrow \mathbb{R}$ (e.g., the generalized mean, or the number of local maxima), and then apply them to every *interval* (i.e., to the set of values that a temporal variable takes between an ordered pair of temporal points) of the temporal domain $[1, \ldots, N]$. In this way, each interval can be seen as a world, and the alphabet becomes:

$$\mathcal{P} = \{F(V) \bowtie v \mid F \in \mathcal{F}, V \in \mathcal{V}, v \in \mathbb{R}, \bowtie \in \{<, \leq, =, \neq, \geq, >\}\},$$

and the different relations between any two intervals can play the role of binary accessibility relations, resulting into an instance of *Halpern and Shoham's modal logic for time intervals* (HS). Reasoning with intervals is clearly not the only way to extract information from temporal data, but it is a very convenient one; it has been successfully used in [11,30], among others. This approach can be also generalized to spatial data, by considering a multi-dimensional generalization of HS, as in [35]. Other types of non-tabular data, such as graph-based data, can be even more naturally treated in the same way, as Kripke structures are, in fact, graphs. However, all kinds of non-tabular data share the notion of variable, so that the above definition of propositional letters ($\mathcal{P}$) can be considered relatively general. Therefore, in the following, we shall assume that a modal dataset is, in fact, characterized by n variables, as it is in the propositional case.

It is natural to ask if, and how, decision trees too can be generalized from propositional to modal logic. This has been first proposed in [10] in the special case of the temporal logic HS; in their prototypical version, modal decision trees were not able to express every possible modal formula. Successive extensions have improved several aspects of modal decision trees, but a provably complete version has been proposed only in [19]; notably, other, incomplete, versions of modal decision trees had been proposed earlier, again in the temporal case [6,17]. The idea behind modal decision trees is that the tree structure must be enriched in order to design an information-based learning algorithm that does not need to explore an exponential number of formulas. Completeness and efficiency can be obtained together at the propositional level because propositional formulas can always be expressed in disjunctive normal form. While there are several proposals for modal disjunctive normal forms, they all share a definition of literal whose modal prefix has an arbitrary number of modalities (see, e.g., [4]), and are therefore unsuitable for a straightforward implementation of modal decision trees. At a closer look, it appears evident that the main obstacle on the road to efficiency of modal logic decision trees learning is the fact that diamonds (resp., boxes) do not distribute over conjunctions (resp., disjunctions), that is, modal logic is not *separable*. As a consequence, learning modal formulas in an inductive, information-based fashion, requires building complex formulas along the branches of a tree.



$$
\begin{aligned}
\varphi_{\wr(\nu)\rightsquigarrow\nu} &= e(\wr(\nu),\nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau)\},\\
\varphi_{\nu_1\rightsquigarrow\ell_1} &= \Diamond(\top \wedge \neg p_1),\\
\varphi_{\rho(\tau)\rightsquigarrow\nu_2} &= \Diamond(\top \wedge \Diamond\top),\\
\varphi_{\rho(\tau)\rightsquigarrow\ell_1} &= \Diamond(\top \wedge \Diamond(\top \wedge \neg p_1)),\\
\varphi_{\nu_1\rightsquigarrow\nu_3} &= \Box(\top \rightarrow p_1),\\
\varphi_{\rho(\tau)\rightsquigarrow\nu_3} &= \Box(\top \rightarrow \Box(\top \rightarrow p_1)),\\
\varphi_{\rho(\tau)\rightsquigarrow\ell_2} &= \Diamond(\Box(\top \rightarrow p_1) \wedge p_2),\\
\varphi_{\rho(\tau)\rightsquigarrow\ell_3} &= \Box(\Box(\top \rightarrow p_1) \rightarrow \neg p_2),\\
\varphi_{\rho(\tau)\rightsquigarrow\ell_4} &= \Box(\top \rightarrow \Box\bot),\\
\varphi_{\ell} &= \textstyle\bigwedge_{\pi \sqsubseteq \pi_\ell} \varphi_\pi,\\
\varphi_{L_1} &= \varphi_{\ell_1} \vee \varphi_{\ell_3} \vee \varphi_{\ell_4},\\
\varphi_{L_2} &= \varphi_{\ell_2} \vee \varphi_{\ell_5}.
\end{aligned}
$$

Fig. 1. A modal decision tree τ (right) and all its relevant path-, leaf-, and class-formulas (left). For each node, non-displayed backward-edges and forward-edges are assumed to be self-loops.

Definition 7 ([19], modified). *Let $\mathcal{I}$ be a modal dataset with set of classes $\mathcal{L}$ and set of associated propositional letters $\mathcal{P}$, and define the set of* modal decisions $\Lambda = \{p, \neg p, \mid p \in \mathcal{P}\} \cup \{\top, \bot, \Diamond\top, \Box\bot\}$. *Then, a* modal decision tree *(*over Λ*) is an object of the type:*

$$\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f),$$

where $(\mathcal{V}, \mathcal{E})$ is a full binary directed tree, $l : \mathcal{V}^\ell \rightarrow \mathcal{L}$ is a leaf-labelling function *that assigns a class from $\mathcal{L}$ to each leaf node in $\mathcal{V}^\ell$, $e : \mathcal{E} \rightarrow \Lambda$ is a* edge-labelling function *that assigns a modal decision to each edge in $\mathcal{E}$, $b : \mathcal{V}^\iota \rightarrow \mathcal{V}^\iota$ is a* backward-edge function *that links an internal node to one of its ancestors, and $f : \mathcal{V} \setminus \mathcal{V}^\ell \rightarrow \mathcal{V}^\iota$ is a* forward-edge function *that links a non-leaf node to one of its descendants, such that, for all $\nu, \nu'\nu'' \in \mathcal{V}$:*

1. *if $\nu, \nu' \in \mathcal{V}^\ell$ and $\wr(\nu) = \wr(\nu')$, then $l(\nu) \neq l(\nu')$,*
2. *if $\nu \notin \mathcal{V}^\ell$, then $e(\nu, \swarrow(\nu)) \equiv \neg e(\nu, \searrow(\nu))$,*
3. *if $b(\nu) = \nu'$, then $\nu' \in \wr^*(\nu)$,*
4. *if $b(\nu) \neq \nu$ and $b(\nu') \neq \nu'$, then $b(\nu) \neq b(\nu')$,*
5. *if $b(\nu) = \nu', \nu' \in \wr^+(\nu'')$, and $\nu'' \in \wr^+(\nu)$, then $\nu' \in \wr^+(b(\nu''))$,*
6. *if $(\nu, \nu') \in \mathcal{E}$, $\nu' \notin \mathcal{V}^\ell$, and $e(\nu, \nu') \in \{\bot, \Box\bot\}$, then $b(\nu') \neq \nu'$, and*
7. *if $f(\nu) = \nu'$, then $\nu \in \wr^*(\nu)$.*

For a path $\pi = \nu_0 \rightsquigarrow \nu_h$ in τ, with $h > 1$, the contributor *of π, denoted by $\zeta(\pi)$, is defined as the only node $\nu_i \in \pi$ such that $\nu_i \neq \nu_1$, with $0 < i < h$, and $b(\nu_i) = \nu_1$, if it exists, and ν_1, otherwise. Moreover, given two nodes $\nu_i, \nu_j \in \pi$, with $i, j < h$, we say that they* agree, *denoted by $\mathfrak{A}(\nu_i, \nu_j)$, if $\nu_{i+1} = \swarrow(\nu_i)$ (resp., $\nu_{i+1} = \searrow(\nu_i)$) and $\nu_{j+1} = \swarrow(\nu_j)$ (resp., $\nu_{j+1} = \searrow(\nu_j)$); otherwise, we say that they* disagree, *denoted by $\mathfrak{D}(\nu_i, \nu_j)$. Furthermore, we say that a modal formula φ is* implicative *if it has the form $\varphi_1 \rightarrow \varphi_2$ or $\Box(\varphi_1 \rightarrow \varphi_2)$, and we denote by*

Im the set of implicative formulas. *The* path-formula φ_π *is defined inductively as* $\top$ *if* $h = 0$, $e(\nu_0, \nu_1)$ *if* $h = 1$, *and, if* $h > 1$, $\lambda = e(\nu_0, \nu_1), \pi_1 = \nu_1 \rightsquigarrow \zeta(\pi)$, *and* $\pi_2 = \zeta(\pi) \rightsquigarrow \nu_h$, *then* φ_π *is:*

– $\lambda \wedge (\varphi_{\pi_1} \wedge \varphi_{\pi_2})$, *if* $\lambda \neq \Diamond\top, \mathfrak{A}(\nu_0, \zeta(\pi))$, *and* $\varphi_{\pi_2} \notin Im$, *or* $\lambda \neq \Diamond\top, \mathfrak{D}(\nu_0, \zeta(\pi))$ *and* $\varphi_{\pi_2} \in Im$;
– $\lambda \rightarrow (\varphi_{\pi_1} \rightarrow \varphi_{\pi_2})$, *if* $\lambda \neq \Diamond\top, \mathfrak{D}(\nu_0, \zeta(\pi))$, *and* $\varphi_{\pi_2} \notin Im$, *or* $\lambda \neq \Diamond\top, \mathfrak{A}(\nu_0, \zeta(\pi))$, *and* $\varphi_{\pi_2} \in Im$;
– $\Diamond(\varphi_{\pi_1} \wedge \varphi_{\pi_2})$, *if* $\lambda = \Diamond\top, \mathfrak{A}(\nu_0, \zeta(\pi))$ *and* $\varphi_{\pi_2} \notin Im$, *or* $\lambda = \Diamond\top, \mathfrak{D}(\nu_0, \zeta(\pi))$, *and* $\varphi_{\pi_2} \in Im$;
– $\Box(\varphi_{\pi_1} \rightarrow \varphi_{\pi_2})$, *if* $\lambda = \Diamond\top, \mathfrak{D}(\nu_0, \zeta(\pi))$ *and* $\varphi_{\pi_2} \notin Im$, *or* $\lambda = \Diamond\top, \mathfrak{A}(\nu_0, \zeta(\pi^\tau))$ *and* $\varphi_{\pi_2} \in Im$.

For each leaf $\ell \in \mathcal{V}^\ell$, *the* leaf-formula φ_ℓ *is defined as:*

$$\varphi_\ell = \bigwedge_{\pi \sqsubseteq \pi_\ell} \varphi_\pi,$$

and for each class L, *the* class-formula φ_L *is defined as:*

$$\varphi_L = \bigvee_{l(\ell)=L} \varphi_{\pi_\ell}.$$

Finally, the run of τ on $\mathfrak{I}$ from ν, *denoted by* $\tau(\mathfrak{I}, \nu)$, *is defined as follows:*

$$\tau(\mathfrak{I}, \nu) = \begin{cases} l(\nu) & \text{if} \quad \nu \in \mathcal{V}^\ell; \\ \tau(\mathfrak{I}, \swarrow(f(\nu))) & \text{if} \quad \mathfrak{I} \Vdash \varphi_{\pi_{\swarrow(f(\nu))}}; \\ \tau(\mathfrak{I}, \searrow(f(\nu))) & \text{if} \quad \mathfrak{I} \Vdash \varphi_{\pi_{\searrow(f(\nu))}}, \end{cases}$$

and the run of τ on $\mathfrak{I}$, *denoted by* $\tau(\mathfrak{I})$, *is defined as* $\tau(\mathfrak{I}, \rho(\tau))$. *An instance* $\mathfrak{I}$ *is* classified into $L \in \mathcal{L}$ *by* τ *if and only if* $\tau(\mathfrak{I}, \rho(\tau)) = L$.

An example of modal decision tree can be seen in Fig. 1. The idea behind modal decision trees is that backward-edges allow one to add conjuncts and disjuncts to a leaf-formula at any modal depth. Compared with the original definition in [19], the addition of forward-edges straightforwardly allows one to improve the completeness result by formalizing the idea of *lookahead*, that is, the possibility for a modal decision tree to classify using complex formulas on top of simple decisions.

Theorem 1 ([19], modified). *The family* $\mathcal{MDT}$ *of modal decision trees is correct and complete with respect to modal logic.*

As it turns out, a learning algorithm for modal decision trees can still be implemented in an efficient way, but to a lesser extent compared with propositional ones; intuitively, in order to have polynomial time learning, we can only guarantee optimality up to a certain formula length. *ModalCART*, shown in Algorithm 1, is the adaptation of the well-known (family of) algorithm(s) known as *CART*, on which the more famous C4.5, ID3, among many others, are based.

It is an information-based approach to decision tree learning, whose main step is founded on computing the amount of information contained in a dataset (via *entropy*, *Gini index*, or similar measures), which drives a locally optimal choice. The function *SubTrees*, given the set of decisions Λ, a height i and a set of ancestors $\mathcal{N}$, returns all trees of height i with exactly two nodes at any given level greater than 0, with edge labels chosen from Λ, where (consistently with Definition 7) all outgoing backward-edges lead to a node in $\mathcal{N}$, and whose root is linked via forward-edge to the (only) non-leaf node at level $i-1$. Thus, *FindBestSubTree* generalizes the operation of finding the best split to the case of lookahead t; in this way, if a class is determined by a modal formula of length less than or equal to t, such a formula will be certainly found and expressed a branch of the tree, which implies that Algorithm 1 correctly finds a t-optimal tree.

Definition 8. *A decision tree is said to be* t-optimal *for a dataset* $\mathcal{I}$ *with respect to a logic if and only if, for every instance* $\mathfrak{I}$ *whose class is identified by a formula of that logic with length less than or equal to* t*,* $\tau(\mathfrak{I}) = L$ *if and only if* $\mathfrak{I}$ *is labelled by* L*. A family of decision trees is* weakly efficient *with respect to a logic if and only if there exists a polynomial-time algorithm that, for every dataset, learns a* t*-optimal tree for it with respect to that logic when* t *is constant.*

Algorithm 1: High-level description of *ModalCART*.

function $ModalCART(\mathcal{I}, \Lambda, t)$**:**
  $\tau \leftarrow Initialize()$
  $\rho(\tau) \leftarrow Learn(\mathcal{I}, \Lambda, \emptyset, t)$
  return τ
end
function $Learn(\mathcal{I}, \Lambda, \mathcal{N}, t)$**:**
  if *no stopping condition applies* **then**
    $\nu \leftarrow FindBestSubTree(\mathcal{I}, \Lambda, \mathcal{N}, t)$
    $f(\nu).left \leftarrow Learn(\mathcal{I}_{\swarrow(f(\nu))}, \Lambda, \mathcal{N} \cup \{\nu\}, t)$
    $f(\nu).right \leftarrow Learn(\mathcal{I}_{\searrow(f(\nu))}, \Lambda, \mathcal{N} \cup \{\nu\}, t)$
  else
    $\nu \leftarrow CreateLeafNode(\mathcal{I})$
  return ν
end
function $FindBestSubTree(\mathcal{I}, \Lambda, \mathcal{N}, t)$**:**
  $(\epsilon, \epsilon_\nu) \leftarrow (-\infty, nil)$
  foreach $i \in 1, \ldots, t$ **do**
    foreach $\nu \in SubTrees(\Lambda, \mathcal{N}, i)$ **do**
      $\mathcal{I}_{\swarrow(f(\nu))} \leftarrow$ subset of $\mathcal{I}$ satisfying $\varphi_{\swarrow(f(\nu))}$
      $\mathcal{I}_{\searrow(f(\nu))} \leftarrow$ subset of $\mathcal{I}$ satisfying $\varphi_{\searrow(f(\nu))}$
      if $\epsilon < Info(\mathcal{I}_{\swarrow(f(\nu))}, \mathcal{I}_{\searrow(f(\nu))})$ **then**
        $(\epsilon, \epsilon_\nu) \leftarrow (Info(\mathcal{I}_{\swarrow(f(\nu))}, \mathcal{I}_{\searrow(f(\nu))}), \nu)$
  return ϵ_ν
end

Lemma 1. *Given a dataset $\mathcal{I}$ with m instances, each with n variables and N distinct worlds, a set of nodes $\mathcal{N}$, and given a lookahead amount t, the running time of FindBestSubTree($\mathcal{I}, \Lambda, \mathcal{N}, t$) is:*

$$\mathcal{O}(t(nm^2N)^t mN).$$

Proof. As per Definition 7, a single split in a tree is the result of model-checking one path-formula (and its negation) on every instance $\mathfrak{I}$ of the current dataset, at the world w_0; given a generic modal formula φ, one can simply apply a finite model-checking algorithm that labels every world with the truth value of all sub-formulas (in non-decreasing order of length). Using a technique similar to [15], the cost of such a check, for all instances, is $\mathcal{O}(|\varphi|N^2m)$. Approaching *FindBestSubTree* naïvely would imply to repeat such a process for all possible formulas that could be generated at a given step; moreover, with lookahead t, this single step requires trying all sub-trees of height at most t. However, the weak completeness guarantees that one may limit the exploration of incomplete sub-trees with exactly two nodes at each given height. Given a set of ancestors $\mathcal{N}$ to which backward-edges may lead, the number of structurally different such sub-trees, of height $1 \leq i \leq t$, is bounded by $2^{i-1}(|\mathcal{N}| + i - 1)^i$; for each of such sub-trees, we must check $(2nmN + 4)^i$ different formulas which are obtained by choosing a different decision on each edge (observe that, given that one formula differs from another one by at least one of the i decisions, the number of different formulas is determined by the number of decisions $|\Lambda|$; the latter, in turn, is bounded by the number of different variables, n, times the cardinality of the domain of an variable, mN, times 2 different test operators – observe that with numerical variables and finite domains, limiting $\bowtie \in \{\geq, <\}$ suffices to explore all propositional letters, plus 4, which is the cardinality of $\{\Diamond\top, \Box\bot, \top, \bot\}$). Summarizing, a naïve implementation of *FindBestSubTree* runs in time:

$$\mathcal{O}(\Sigma_{i=1}^{t} 2^{i-1}(m + i - 1)^i(2nmN + 4)^i m^2N^2) = \mathcal{O}(t(nm^2N)^t m^2N^2),$$

considering that, during the execution, both the number of any set of ancestors and the length of any formula are bounded by m and that every element of the summation can be bounded by $2^{t-1}(m + t - 1)^t(2nmN + 4)^t$. For a small enough t, it makes sense to reduce the above complexity by exploiting the nature of propositional letters via memoization. Observe that at any given step some formulas to be checked have the same structure, only differing by the numerical constants within the propositional letters. By way of example, consider checking a set of formulas $\Diamond(V < a)$ on a single world w of an instance $\mathfrak{I}$. The number of worlds that are accessible from w is bounded by N, and the domain of the variable by mN. Thus, a single check of all such formulas takes $\mathcal{O}(mN^2)$. If, instead, we compute and store the minimum of the values for V on every world accessible from w (which takes time $\mathcal{O}(N)$), we can check each of the mN similar formulas in time $\mathcal{O}(1)$, by comparing a with the computed value. Generalizing, consider now the case where $t = 1$. In such a case, the execution of *FindBestSubTree* consists of exploring at most $|\Lambda|m$ different formulas; the factor m depends on the possible backward-edges of the newly formed tree, and the factor $|\Lambda|$ depends on the element that changes from one formula to another. The set of formulas can, then, be partitioned into $2n+4$ groups. All formulas within a single group are *siblings*, that is, they share the structure of the syntax tree and differ by exactly a numerical constant at a leaf; the groups that emerge from decisions that are not propositional letters are singletons. The key idea is that we can compute, save, and use a single scalar value for each of such groups in order to check all siblings within the group. We now describe a memoization structure, keyed in a triple formed by a formula φ, an instance $\mathfrak{I}$, and a world w, and returning a single number that suffices to check the truth of every sibling formula of φ on $\mathfrak{I}, w$. For a given φ generated by the grammar in Definition 7, we denote the biggest sub-formula that contains the variable part (that is, the single leaf that identifies an element which its group of its siblings) by $\bar{\varphi}$. For convenience, let us define the special values $\top(\geq) = \infty, \top(<) = -\infty, \bot(\geq) = -\infty, \top(<) = \infty$. We inductively define the memoization structure, denoted by T, fixed $\mathfrak{I}$ and w, and fixed V and $\bowtie$ that identify the variable part of φ, as follows:

$$T[\top] = \top(\bowtie) \qquad T[\Diamond\top] = \begin{cases} \top(\bowtie) \text{ if } |\{(w,w') \in R\}| > 0, \\ \bot(\bowtie) \text{ otherwise} \end{cases}$$

$$T[\bar{p}] = H[\mathfrak{I}, w, V] \qquad T[V' \bowtie' a] = \begin{cases} \top(\bowtie) \text{ if } H[\mathfrak{I}, w, V'] \bowtie' a, \\ \bot(\bowtie) \text{ otherwise} \end{cases}$$

$$T[\varphi \wedge \bar{\varphi}'] = \begin{cases} T[\bar{\varphi}'] & \text{if } \mathfrak{I}, w \Vdash \varphi \\ \bot(\bowtie) & \text{otherwise} \end{cases} \qquad T[\Diamond(\varphi \wedge \bar{\varphi}')] = \begin{cases} \underset{w' \in W(\mathfrak{I}, w, \varphi)}{\zeta} T[\bar{\varphi}'] & \text{if } \mathfrak{I}, w \Vdash \Diamond\varphi \\ \top(\bowtie) & \text{otherwise,} \end{cases}$$

where $\zeta = \max$ (resp., $\min$) if $\bowtie = \geq$ (resp., $<$), $W(\mathfrak{I}, w, \varphi)$ is the set of worlds in $\mathfrak{I}$ where φ holds and are accessible from w, and H holds the value of each variable V, at each world w of each instance $\mathfrak{I}$; the missing cases can be treated in a similar way. Fixed a formula $\bar{\varphi}$ that belongs to the group of sibling formulas varying the value a for a specific propositional letter built on V and $\bowtie$ (let us denote such a formula by $\bar{\varphi}_{V,\bowtie}(v)$), it holds that:

$$\mathfrak{I}, w \Vdash \bar{\varphi}_{V,\bowtie}(v) \Leftrightarrow T[\bar{\varphi}_{V,\bowtie}] \bowtie v.$$

As noticed before, within a call to $FindBestSubTree$ the number of structurally different sub-trees of height i that are tested is at most $2^{i-1}(m+i-1)^i$. Each of the sub-trees gives rise to $2n(2nmN+4)^{i-1}$ groups of mM siblings, plus $4(2nmN+4)^{i-1}$ singletons; within each group, the formulas differ from each other by the decision taken at level i. One key observation at this point is that, in order to find the best formula (with respect to a given information measure) from a group of mN sibling formulas induced by $\bar{\varphi}_{V,\bowtie}$, we do not need to check the truth of every one of them on every instance that reached the node. In fact, we only need to check the m formulas that correspond to $\bar{\varphi}_{V,\bowtie}(v)$ with a equal to the corresponding $T[\bar{\varphi}_{V,\bowtie}]$ value of any instance $\mathfrak{I}$; this requires $\mathcal{O}(m^2N^2)$ time for pre-computing m values $T[\bar{\varphi}]$, plus $\mathcal{O}(m^2N^2)$ for performing m checks. Therefore, the overall cost becomes:

$$\mathcal{O}(\Sigma_{i=1}^{t} 2^{i-1}(m+i-1)^i(2nmN+4)^{i-1}(2n+4)m^2N^2) = \mathcal{O}(t(nm^2N)^t mN).$$

Lemma 2. *Given a dataset $\mathcal{I}$ with m instances, each with n variables and N distinct worlds, and given a lookahead amount t, the running time of ModalCART($\mathcal{I}, \Lambda, t$) is:*

$$\mathcal{O}(m^{2(t+1)+1}),$$

in the worst case, and:

$$\mathcal{O}(m^{2(t+1)}\lg(m)),$$

in the average case, assuming constant n, N, and t.

Proof. *ModalCART* is a recursive procedure whose complexity can be approached via a recurrence; even with lookahead $t > 1$, each step consists, at most, of 2 recursive calls. Assuming constant n and N corresponds to studying the complexity of *ModalCART* as the number of instances grows without changing in nature; assuming constant t is equivalent to fixing a learning parameter. In the worst case, every split of a dataset of cardinality m ends up assigning exactly one instance to a branch and exactly $m-1$ instances to the other one, so that the recurrence is:

$$\mathcal{T}(m) = \mathcal{T}(m-1) + \mathcal{O}(m^{2(t+1)}),$$

which ends up being bounded by:

$$\mathcal{T}(m) = \mathcal{O}(m^{2(t+1)+1}).$$

In the average case, however, we can assume that all splits are equally likely in terms of relative sizes. Thus the recurrence that describes the time complexity becomes:

$$\begin{aligned}\mathcal{T}(m) &= \tfrac{1}{m-1}\textstyle\sum_{i=1}^{m-1}(\mathcal{T}(i) + \mathcal{T}(m-i)) + \mathcal{O}(m^{2(t+1)})\\ &= \tfrac{2}{m-1}\textstyle\sum_{i=1}^{m-1}\mathcal{T}(i) + \mathcal{O}(m^{2(t+1)}).\end{aligned}$$

We claim that $\mathcal{T}(m) = \mathcal{O}(m^{2(t+1)}\lg(m))$, and we prove it by substitution, that is, by proving that there exists a constant α such that $\mathcal{T}(m) \leq \alpha m^{2(t+1)}\lg(m)$ for large enough values of m. Let us fix $k = 2(t+1)$. Then:

$$\begin{aligned}\mathcal{T}(m) &= \tfrac{2}{m-1}\textstyle\sum_{i=1}^{m-1}\mathcal{T}(i) + \mathcal{O}(m^k)\\ &\leq \tfrac{2}{m-1}\textstyle\sum_{i=1}^{m-1}\alpha i^k\lg(i) + \mathcal{O}(m^k)\\ &\leq \tfrac{2}{m-1}\alpha\lg(m)\textstyle\sum_{i=1}^{m-1} i^k + \mathcal{O}(m^k)\\ &\leq \tfrac{2\alpha\lg(m)}{m-1}\left(\tfrac{(m-1)^{k+1}}{k+1} + \tfrac{(m-1)^k}{2} + \tfrac{k(m-1)^{k-1}}{12}\right) + \mathcal{O}(m^k)\\ &= \tfrac{2\alpha\lg(m)}{(k+1)(m-1)}\left((m-1)^{k+1} + \tfrac{(k+1)(m-1)^k}{2} + \tfrac{(k+1)k(m-1)^{k-1}}{12}\right) + \mathcal{O}(m^k)\\ &\leq \tfrac{2\alpha\lg(m)}{(k+1)(m-1)}\left((m-1)^{k+1} + \tfrac{(m-1)^{k+1}}{2} + \tfrac{(m-1)^{k+1}}{12}\right) + \mathcal{O}(m^k)\\ &= \tfrac{2\alpha\lg(m)(m-1)^k}{(k+1)}(1 + \tfrac{1}{2} + \tfrac{1}{12}) + \mathcal{O}(m^k).\end{aligned}$$

using (a further bounded version of) the Faulhaber formula [20], and taking into account that assuming constant t implies $k \leq m-2$, and therefore, $k < k+1 \leq m-1$. For large enough values of m, all of the above amounts to proving that:

$$\tfrac{19}{6(k+1)}\alpha\lg(m)(m-1)^k \leq \alpha\lg(m)m^k,$$

which is implied by:

$$\tfrac{19}{6(k+1)}m^k \leq m^k,$$

which is true for $k \geq \frac{13}{6}$ that is, $k \geq 3$, which is always true as $t \geq 1$.

Theorem 2. *The family $\mathcal{MDT}$ of modal decision trees is weakly efficient.*

4 Conclusions

In the past few years, modal symbolic learning in general, and modal decision trees in particular, have proven themselves to be a very useful tool for extracting complex knowledge from non-tabular data, in both the temporal and the spatial case. Admittedly, however, current implementations of modal decision trees could not guarantee the completeness of the approach in the logical sense. Building upon a very recent result in which a complete version of modal decision trees has been proposed, in this paper we provided an assessment of their computational complexity, proving that, in fact, modal decision trees are both complete and efficient. Modal decision trees can be seen as prototypical of a large family of symbolic learning tools based on more-than-propositional logics; their development can now rest on a more solid theoretical background. Future work includes the empirical evaluation of their behaviour.

Acknowledgements. We acknowledge the support of the INDAM-GNCS project *Symbolic and Numerical Analysis of Cyberphysical Systems*, founded by INDAM (code CUP_E53C22001930001), as well as that of the FIRD project *Symbolic Geometric Learning*, founded by the University of Ferrara.

References

1. Aiello, M., van Benthem, J.: A modal walk through space. J. Appl. Non-Class. Log. **12**(3–4), 319–364 (2002)
2. Alaniz, S., Marcos, D., Schiele, B., Akata, Z.: Learning decision trees recurrently through communication. In: Proceedings of the 34th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13518–13527 (2021)
3. Belson, W.: A technique for studying the effects of television broadcast. J. R. Stat. Soc. **5**(3), 195–202 (1956)
4. Bienvenu, M.: Prime implicates and prime implicants: from propositional to modal logic. J. Artif. Intell. Res. **36**, 71–128 (2009)
5. Blockeel, H., De Raedt, L.: Top-down induction of first-order logical decision trees. Artif. Intell. **101**(1–2), 285–297 (1998)

6. Bombara, G., Vasile, C., Penedo, F., Yasuoka, H., Belta, C.: A decision tree approach to data classification using signal temporal logic. In: Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control (HSCC), pp. 1–10 (2016)
7. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
8. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001)
9. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and regression trees. Wadsworth Publishing Company (1984)
10. Brunello, A., Sciavicco, G., Stan, I.E.: Interval temporal logic decision tree learning. In: Calimeri, F., Leone, N., Manna, M. (eds.) JELIA 2019. LNCS (LNAI), vol. 11468, pp. 778–793. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19570-0_50
11. Caselli, E., et al.: Towards an objective theory of subjective liking: a first step in understanding the sense of beauty. PLoS ONE **8**(6), 1–20 (2023)
12. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 785–794 (2016)
13. Clark, P., Niblett, T.: The CN2 induction algorithm. Mach. Learn. **3**, 261–283 (1989)
14. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logic of Programs 1981. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982). https://doi.org/10.1007/BFb0025774
15. Clarke, E., Emerson, E., Sistla, A.: Automatic verification of finite-state concurrent systems using temporal logic specifications. ACM Trans. Program. Lang. Syst. **8**(2), 244–263 (1986)
16. Coccagna, M., Manzella, F., Mazzacane, S., Pagliarini, G., Sciavicco, G.: Statistical and symbolic neuroaesthetics rules extraction from EEG signals. In: Ferrandez Vicente, J.M., Alvarez-Sanchez, J.R., de la Paz Lopez, F., Adeli, H. (eds.) Artificial Intelligence in Neuroscience: Affective Analysis and Health Applications. IWINAC 2022. LNCS, vol. 13258, pp. 536–546. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06242-1_53
17. Console, L., Picardi, C., Dupré, D.: Temporal decision trees: model-based diagnosis of dynamic systems on-board. J. Artif. Intell. Res. **19**, 469–512 (2003)
18. de Ville, B.: Decision trees. WIREs Comput. Stat. **5**(6), 448–455 (2013)
19. Della Monica, D., Pagliarini, G., Sciavicco, G., Stan, I.: Decision trees with a modal flavor. In: Proceedings of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA). in press (2022)
20. Gnewuch, M., Pasing, H., Weiss, C.: A generalized faulhaber inequality, improved bracketing covers, and applications to discrepancy. Math. Comput. **90**, 2873–2898 (2021)
21. Guo, H., Gelfand, S.: Classification trees with neural network feature extraction. IEEE Trans. Neural Netw. **3**(6), 923–933 (1992)
22. Halpern, J., Shoham, Y.: A propositional modal logic of time intervals. J. ACM **38**(4), 935–962 (1991)
23. Kass, G.V.: An exploratory technique for investigating large quantities of categorical data. J. R. Stat. Soc. **29**(2), 119–127 (1980)
24. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Proceedings of the 31st Advances in Neural Information Processing Systems (NIPS), pp. 3146–3154 (2017)

25. Kearns, M., Valiant, L.: Cryptographic limitations on learning Boolean formulae and finite automata. J. ACM **41**(1), 67–95 (1994)
26. Kontschieder, P., Fiterau, M., Criminisi, A., Rota Bulò, S.: Deep neural decision forests. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1467–1475 (2015)
27. Loh, W.: Classification and regression trees. WIREs Data Min. Knowl. Discov. **1**(1), 14–23 (2011)
28. Loh, W.: Fifty years of classification and regression trees. Int. Stat. Rev. **82**(3), 329–348 (2014)
29. Lutz, C., Wolter, F.: Modal logics of topological relations. Log. Methods Comput. Sci. **2**(2), 1–41 (2006)
30. Manzella, F., Pagliarini, G., Sciavicco, G., Stan, I.: The voice of COVID-19: breath and cough recording classification with temporal decision trees and random forests. Artif. Intell. Med. **137**, 102486 (2023)
31. Messenger, R., Mandell, L.: A modal search technique for predictive nominal scale multivariate analysis. J. Am. Stat. Assoc. **67**(340), 768–772 (1972)
32. Morgan, J., Sonquist, J.: Problems in the analysis of survey data, and a proposal. J. Am. Stat. Assoc. **58**(302), 415–434 (1963)
33. Murdock, C., Li, Z., Zhou, H., Duerig, T.: Blockout: dynamic model selection for hierarchical deep networks. In: Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2583–2591 (2016)
34. Murthy, V., Singh, V., Chen, T., Manmatha, R., Comaniciu, D.: Deep Decision network for multi-class image classification. In: Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2240–2248 (2016)
35. Pagliarini, G., Sciavicco, G.: Decision tree learning with spatial modal logics. In: Proceedings of the 12th International Symposium on Games, Automata, Logics, and Formal Verification (GANDALF). EPTCS, vol. 346, pp. 273–290 (2021)
36. Pnueli, A.: The temporal logic of programs. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS), pp. 46–57 (1977)
37. Quinlan, J.: Induction of decision trees. Mach. Learn. **1**, 81–106 (1986)
38. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, Burlington (1993)
39. Quinlan, J.: Simplifying decision trees. Int. J. Hum. Comput. Stud. **51**(2), 497–510 (1999)
40. Rivest, R.: Learning decision lists. Mach. Learn. **2**(3), 229–246 (1987)
41. Sciavicco, G., Stan, I.: Knowledge extraction with interval temporal logic decision trees. In: Proceedings of the 27th International Symposium on Temporal Representation and Reasoning (TIME). LIPIcs, vol. 178, pp. 9:1–9:16 (2020)
42. Srivastava, N., Salakhutdinov, R.: Discriminative transfer learning with tree-based priors. In: Proceedings of the 26th Advances in Neural Information Processing Systems (NIPS), pp. 2094–2102 (2013)
43. Wan, A., et al.: NBDT: neural-backed decision tree. In: Proceedings of the 9th International Conference on Learning Representations (ICLR), pp. 1–12 (2021)
44. Zhou, Z., Chen, Z.: Hybrid decision tree. Knowl.-Based Syst. **15**(8), 515–528 (2002)