



UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN

MATEMATICA

CICLO XXXV

Foundations of Modal Symbolic Learning

Coordinatrice:

Chiar.ma prof.ssa LUNARDI Alessandra

Tutore:

Prof. SCIavicco Guido

Dottorando: STAN Ionel Eduard

Anni Accademici 2019/2020–2021/2022

Abstract

Dr. Eduard Ionel STAN

Foundations of Modal Symbolic Learning

Traditional symbolic learning is the sub-field of machine learning that aims to learn symbolic models from structured data, representing propositional logic theories, and its investigation initiated with the early days of artificial intelligence. Such an approach is yet outstanding in terms of academic performance (e.g., accuracy) and industrial one (e.g., interpretability, bias, ethical concerns) over modern techniques (e.g., deep neural networks) on structured data, but it suffers to natively address the problem of learning from unstructured data (e.g., time series, images, and graphs). By systematically exploiting inductive biases, we present the mathematical framework of *modal symbolic learning* for learning symbolically from unstructured data, which is the intersection between the fields of machine learning and modal logic(s) in terms of academic discipline. We study its properties from a learning perspective, enhancing standard decision trees, the quintessential expression of conventional symbolic learning, to learn modal logic theories. We demonstrate how modal data emerges from unstructured one to conduct modal symbolic learning. We experimentally prove how models learned with this framework are more accurate, precise, and sensible than classic propositional ones and at least comparable with those learned with non-symbolic ones. In addition, we show how our approach can be generalized from trees to more complex learning techniques (e.g., fuzzy and neural-symbolic trees), and we point to several ambitious and challenging directions in which modal symbolic learning, as a field, can expand. Modal symbolic learning is still in its infancy, and investigating its foundations allows a more organic and substantial development of the matter. Nevertheless, it has already shown enormous application potential, both theoretical (giving modal logic languages a new application field and fostering the study of new ones) and practical (symbolically addressing real-world problems and thus offering interpretable models for further research).

Acknowledgements

I want to express my gratitude, first of all, by thanking prof. Guido Sciavicco, that orchestrated this thesis and, most importantly, that steered me all these years, starting from the Bachelor's degree to this work. Thank you for every bit of teaching.

Thanks to prof. Alessandra Lunardi for being such a patient, responsive, and wise coordinator.

I want to thank the reviewers for taking the necessary time and effort to review this work. All your valuable comments and suggestions helped me in improving the quality of this thesis.

I want to thank my dear friends, Alessandro, Damiano, Federica, Marco, Matteo, Nicola, Roberta, and Tommaso, for their precious support in pushing me to seek success every day.

All the great times I had during these three years of my PhD could not have been possible without my colleagues, Anna Chiara, Estrella, Francesca, Ilaria, and (my buddy) Giovanni.

Then, I want to thank all the people and friends in the Applied Computational Logic and Artificial Intelligence (ACLAI) laboratory. I saw its inception, for which I am immensely proud, and now, I see a remarkable group working together towards the same dreams.

Last but not least, thanks to my whole family and, most especially, my parents, Nuşa and Dorel, for everything. I could only have accomplished my outcomes with your support.

Contents

Abstract		iii
Acknowledgements		v
1 Introduction		1
1.1 A Brief History of Artificial Intelligence		2
1.2 Motivations of this Thesis		3
1.3 Problem Statement and Contributions		8
1.4 Organization of this Thesis		8
1.5 Published (and in Press) Results		10
2 Background		13
2.1 Propositional Logic		13
2.2 Modal Logic		14
2.3 Machine Learning		17
2.4 Taxonomy of Symbolic Learning		21
3 Modal Decision Trees		23
3.1 Propositional Decision Trees		24
3.2 Modal Decision Trees		28
3.3 Properties of Modal Decision Trees		34
3.4 Entropy-based Learning of Modal Decision Trees		40
4 Modal Logics and Modal Datasets		45
4.1 Examples of Unstructured Data		45
4.2 Modal Logics for Unstructured Data		48
4.3 A Modal Logic for (Almost) All		53
4.4 Datasets to Modal Datasets		55
5 Learning with Modal Symbolic Learning		61
5.1 Regression with Modal Decision Trees		61
5.2 Random Forests with Modal Decision Trees		61
5.3 Rules Extraction from Modal Decision Trees and Modal Random Forests		63
5.4 Multi-Frame Modal Symbolic Learning		64
5.5 Blueprint of Modal Symbolic Learning		64
5.6 Real-world Data Experiments: Temporal Data		65
5.7 Real-world Data Experiments: Spatial Data		71
6 Extensions		75
6.1 Neural-Symbolic Modal Decision Trees		75
6.2 Fuzzy Modal Decision Trees		76
6.3 Gradient-boosted Modal Decision Trees		76

6.4	Incremental Modal Decision Trees Learning	77
6.5	Geometric Modal Symbolic Learning	77
7	Related Work	79
7.1	Brief History on Propositional Decision Trees	79
7.2	Approaches for Learning from Temporal Data	80
7.3	Approaches for Learning from Spatial Data	83
7.4	Symbolic Approaches for Learning from Unstructured Data	84
8	Conclusions	87
	Bibliography	89

List of Figures

1.1	Some specializations of decision trees; r is the final result of the prediction of each model. A decision list is a set of ordered if-then rules. A random forest and boosted aggregated trees are ensembles of trees where the final result is weighted between the single trees. A neural-symbolic decision tree can have (deep) neural networks $NN_i(\cdot)$ that decide the splits, while a \mathcal{FOL} decision tree can have arbitrary relations $R_i(\cdot)$	7
2.1	Example of a Kripke model.	15
3.1	Example of a structured dataset.	25
3.2	Example of a propositional decision tree.	27
3.3	Example of a modal dataset.	29
3.4	Example of a modal decision tree.	31
4.1	Example of a multivariate time series with two measurements A_1 and A_2	46
4.2	Example of a red R , green G , and blue (B) image.	46
4.3	Example of a video with RGB images.	46
4.4	Example of an audio data.	46
4.5	Example of a textual data.	47
4.6	Example of a graph data; source Sweileh (2020).	47
4.7	Point relations and $\mathcal{LTL}_{F,P}$ modalities.	49
4.8	Allen’s interval relations and \mathcal{HS} modalities.	50
4.9	Egenhofer and Franzosa’s topological relations and \mathcal{L}_{RCC8} modalities.	52
4.10	\mathcal{HS}^d -based fragments and extensions: green-shaded require no, or at most few additional, assumptions, yellow-shaded require more assumptions, and red-shaded require even more assumptions.	55
4.11	Cases that are captured naturally by \mathcal{HS}^d and those that require more assumptions.	58
4.12	Kripke model after the application of the \mathcal{HS}_A^1 -transformer.	58
5.1	Blueprint of modal symbolic learning.	65
7.1	Literature map on the history on propositional decision trees (generated with litmaps).	80
7.2	Literature map for approaches for time series classification (generated with litmaps).	81
7.3	Literature map for approaches for image classification (generated with litmaps).	84
7.4	Literature map for symbolic approaches for learning from unstructured data (generated with litmaps).	85

List of Tables

1.1	Dimensions of AI as suggested by Russell and Norvig (2020).	2
1.2	Differences between structured and unstructured data.	3
5.1	Cross-validated results on five non-segmented datasets for COVID-19 diagnosis, using different approaches based on temporal decision trees and temporal random forests. For each approach, the average and the standard deviation across 10 repetitions of the following performance metrics are reported: overall accuracy, mean precision, mean recall, and F1 score. Results are reported in percentage points and, for each dataset, the average performance of the best decision tree and the best random forest approach is highlighted. Average training time in seconds is also reported.	69
5.2	Cross-validated results on five segmented datasets for COVID-19 diagnosis, using different approaches based on temporal decision trees and temporal random forests. For each approach, the average and the standard deviation across 10 repetitions of the following performance metrics are reported: overall accuracy, mean precision, mean recall, and F1 score. Results are reported in percentage points and, for each dataset, the average performance of the best decision tree and the best random forest approach is highlighted. Average training time in seconds is also reported.	70
5.3	Cross-validation results on five datasets for land cover classification, using different approaches based on propositional and spatial decision trees. For each approach, the average and the standard deviation across 10 repetitions of the following performance metrics are reported: κ coefficient, overall accuracy, and mean precision. Results are reported in percentage points and, for each dataset, the average performance of the best pure approach and the best derived approach is highlighted. Average training time in seconds is also reported.	73

List of Abbreviations

Generic:

AI	artificial intelligence
ML	machine learning
XAI	explainable artificial intelligence
NLP	natural language processing
LCC	land cover classification

Logics:

\mathcal{L}	generic logic
\mathcal{PL}	propositional logic
\mathcal{ML}	modal logic
\mathcal{LTL}	linear temporal logic
$\mathcal{LTL}_{F,P}$	linear temporal logic with future and past
\mathcal{MITL}	metric interval temporal logic
\mathcal{STL}	signal temporal logic
\mathcal{HS}	Halpern and Shoham's temporal logic of Allen's thirteen interval relations
\mathcal{HS}_3	Halpern and Shoham's temporal logic of Allen's three interval coarse relations
\mathcal{HS}_7	Halpern and Shoham's temporal logic of Allen's seven interval coarse relations
\mathcal{PNL}	propositional neighbourhood logic
\mathcal{CDT}	\mathcal{CDT} logic
\mathcal{DC}	duration calculi
\mathcal{CL}	compass logic
\mathcal{L}_{RCC8}	spatial logic of Egenhofer and Franzosa's eight topological relations
\mathcal{L}_{RCC5}	spatial logic of Egenhofer and Franzosa's five topological coarse relations
\mathcal{SPNL}	spatial propositional neighbourhood logic
\mathcal{WSPNL}	weak spatial propositional neighbourhood logic
\mathcal{FOL}	first order logic

Algorithms and models:

ID3	Iterative Dichotomizer 3
CART	Classification And Regression Trees
XGBoost	eXtreme Gradient Boosting
LightGBM	Light Gradient-Boosted Machines
AID	Automatic Interaction Detection
CLS	Concept Learning System
CHAID	CHi-squared Automatic Interaction Detection
THAID	THeta Automatic Interaction Detection

SVM	Support Vector Machine
SAE	Sequence Auto-Encoder
RNN	Recurrent Neural Network
CNN	Convolutional Neural Network
MLP	Multi-Layer Perceptron
FCN	Fully-Connected Network
ResNet	Residual Network
CapsNet	Capsule Network
DenseNet	Dense Network
Techniques:	
REP	Reduced Error Pruning
EWD	Equal Width Discretization
EFD	Equal Frequency Discretization
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
SVD	Singular Value Decomposition
PAA	Piecewise Aggregate Approximation
SAX	Symbolic Aggregate Approximation
TD4C	Temporal Discretization For Classification
DTW	Dynamic Time Warping
catch22	CAnonical Time series CHaracteristics
tsfresh	Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests
MFCC	Mel-Frequency Cepstral Coefficients

List of Symbols

\mathcal{A}	attributes
$A \in \mathcal{A}$	attribute
Ω	physical domain of interest
\mathbb{X}	sample space
$\mathbb{X}(\Omega, \mathcal{A})$	\mathcal{A} -valued signals on Ω
$x \in \mathbb{X}$	instance
$\mathcal{X} \subseteq \mathbb{X}$	dataset
\mathbb{Y}	label space
$y \in \mathbb{Y}$	label
\mathbb{H}	hypothesis space
$h \in \mathbb{H}$	hypothesis
\mathbb{P}	space of all sets of propositional letters
$\mathcal{P} \in \mathbb{P}$	propositional letters
$p, q, \dots, p_1, p_2, \dots \in \mathcal{P}$	proposition letters
$\Phi(\mathcal{L})$	all the formulas generated by the grammar of \mathcal{L}
$\varphi, \psi, \dots, \varphi_1, \varphi_2, \dots \in \Phi(\mathcal{L})$	formulas of \mathcal{L}
\mathbb{K}	Kripke model space
$\mathfrak{K} \in \mathbb{K}$	Kripke model
\mathbb{F}	feature extraction function space
$f \in \mathbb{F}$	feature extraction function
\mathcal{W}	worlds in a Kripke frame
$\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$	accessibility relation in a Kripke frame
\mathfrak{F}	Kripke frame
$V : \mathcal{W} \rightarrow 2^{\mathcal{P}}$	valuation function in a Kripke model
$\mathcal{I} \subseteq \mathbb{K}$	modal dataset
$\mathfrak{J} \in \mathbb{K}$	modal instance
Λ	decisions
$\lambda \in \Lambda$	decision
\mathcal{V}	nodes of a tree
\mathcal{E}	edges of a tree
$l : \mathcal{V}^\ell \rightarrow \mathbb{Y}$	leaf-labelling function
$e : \mathcal{E} \rightarrow \Lambda$	edge-labelling function
$b : \mathcal{V}^i \rightarrow \mathcal{V}^i$	back-edge function
t	decision tree
$t = (\mathcal{V}, \mathcal{E}, l, e)$	propositional decision tree
$t = (\mathcal{V}, \mathcal{E}, l, e, b)$	modal decision tree
$\mathcal{T} = \{t_1, \dots, t_k\}$	collection of k trees
$root(t)$	root of tree t
$\mathcal{V}^\ell \subseteq \mathcal{V}$	leaf nodes
$\mathcal{V}^i \subseteq \mathcal{V}$	internal (non-root and non-leaf) nodes

$v \in \mathcal{V}$	node
$\ell \in \mathcal{V}^\ell$	leaf node
$\nu^l(v)$	left child of v
$\nu^r(v)$	right child of v
$\wp(v)$	parent of v
\wp^*	transitive and reflexive closure of \wp
\wp^+	transitive closure of \wp
$leaves^t(y)$	leaves of decision tree t labelled with y
$\pi^t = v_0 \rightsquigarrow v_h$	path of length h in tree t
$\pi_1 \cdot \pi_2$	appending path π_2 to π_1
$\pi_v^t = root(t) \rightsquigarrow v$	unique path from $root(t)$ to v in tree t
$prefix(\pi^t)$	improper prefixes of π^t
π_ℓ^t	branch $root(t) \rightsquigarrow \ell$ of tree t
$\tau_{\mathcal{L}}$	\mathcal{L} -transformer

To my parents,
Nuşica and Doreluş

INTRODUCTION

*Of all things the measure is Man, of the things that are, that they are,
and of the things that are not, that they are not.*

—Protagoras of Abdera

For thousands of years, we have tried to understand the functionalities of our brain (e.g., perception, understanding, and prediction) in a world far more extensive and more complicated than itself to define (*human*) *intelligence* correctly. *Artificial intelligence* (AI) is concerned with not just understanding but also *building* machines that exhibit behaviours that (broadly) can be characterized as intelligent (Genesereth and Nilsson, 1988; Russell and Norvig, 2020). Studies periodically rank AI as one of the fast-growing and most funded fields (see, e.g., *The AI Index 2022 Annual Report* by Zhang et al., 2022).

AI is an interesting area, but there is no general *consensus* definition of what AI really *is* because there have been different versions of AI throughout history. Demis Hassabis, CEO and co-founder of Google's DeepMind, made a clear statement in a recent [interview](#):

Solve intelligence, then use that to solve everything else.

Russell and Norvig (2020) categorize the dimensions of AI that cover a significant portion of the relevant literature, illustrated in Table 1.1, as:

- *Acting humanly* based on the (famous) Turing test, proposed by Turing (1950), to answer the question “*Can a machine think?*”;
- *Thinking humanly* based on the cognitive modelling approach, which mimics human-like thinking if there is enough understanding of the human mind so that such theory can be embedded into a computer program;
- *Thinking rationally* based on the “laws of thought” approach rooted in Aristotle’s *sylogisms* whose studies, especially starting from the 19th century, gave birth to *logic* (see, e.g., Hurley, 2014 for an introduction to logic and history therein);
- *Acting rationally* based on the rational agent approach that tries to optimize the best outcome or, under uncertainty, the best-expected outcome.

Many disciplines contributed to the foundations of AI, such as, and not limited to, philosophy, mathematics, computer science, neuroscience, linguistics, economics, and psychology.

	Human-based	Rationality-based
Reasoning-based	Thinking humanly	Thinking rationally
Behaviour-based	Acting humanly	Acting rationally

TABLE 1.1: Dimensions of AI as suggested by Russell and Norvig (2020).

§1.1 A Brief History of Artificial Intelligence

To better understand the motivations and structure of this thesis, we must review some historical milestones and critiques of AI; this review is not complete by any means (e.g., see Russell and Norvig, 2020 for an in-depth history of AI).

The term AI was coined in 1956 thanks to the proposal of McCarthy et al. for a two-month, 10-person summer research conference, where they wrote (McCarthy et al., 2006):

The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.

Two attendees at the workshop, Allen Newell and Herbert A. Simon, proposed the development of models using symbolic manipulation (Newell and Simon, 1976), which later became known as *symbolic AI* paradigm. The computation in such models takes symbols that can be combined and manipulated to produce new expressions. Search and representation were the two most dominant ideas in early symbolic AI. Search techniques, adapted in the early days of AI, include graph-theoretic depth-first search and breadth-first search. Heuristic-based search techniques immediately followed to overcome the disadvantages of both (depth-first and breadth-first) search approaches. Representation expresses knowledge in a way that facilitates its use concerning the task's requirement to be handled (by AI) and mirroring the external world (in some way). Rule-based systems are examples of representations used in conjunction with search algorithms for inference purposes. Researchers also developed the *connectionist* AI paradigm (McCulloch and Pitts, 1943; Rosenblatt, 1958; Minsky and Papert, 1969) during the same period, essentially based on (artificial) neural networks. Connectionist models excel at *machine learning* (ML), the subfield of AI that learns models from input data to adapt to new circumstances. The very foundation of such a paradigm has always been learning. In contrast, symbolic AI is focused more on representation.

Those have been exciting years for the development of (either symbolic or connectionist) AI: many AI researchers were making predictions of their future successes; as such, AI received much attention, and many agencies, such as [Defense Advanced Research Projects Agency \(DARPA\)](#), have funded millions of dollars in AI research. Notwithstanding the promises and the excitement of the AI community, some years later, the British government's support in investing in AI research ended due to what AI experts know as the "*Lighthill report*" (McCarthy, 1974). The document criticized the field of AI for fulfilling the great expectations that it promised; one of the arguments against AI research in the report was the *combinatorial explosion* problem, a well-known issue in computational complexity theory. Minsky and Papert (1969) also exposed the limitations of Rosenblatt's PERCEPTRON (Rosenblatt, 1958), preventing further funding in connectionist AI for a decade. On the symbolic AI side, Dreyfus (1972) criticized the logical approach to AI, observing that humans rarely use logic when they solve problems.

	Structured data	Unstructured data
Structure	Information highly organized	No predefined structure
Data nature	Quantitative data (e.g., numbers, dates, strings)	Qualitative data (e.g., time series, images, videos, graphs)
Required memory	Less memory	More memory
Storage	Data warehouses, Relational databases	Data lakes, Non-relational databases
Analytics	Easy to analyse	Hard to analyse (without AI)
Ease of search	Easy to search	Hard to search
Percent of all existing data	5–20%	80–95%
Preferred learning models	Decision trees and the like	Neural networks and the like

TABLE 1.2: Differences between structured and unstructured data.

In the 1980s, an awakening of the symbolic AI community began with a program called *expert systems*, and knowledge representation was the mainstream research in AI. The return also of neural networks came during the same years thanks to the influential work on the *backpropagation* algorithm by Rumelhart, Hinton, and Williams (1986), which is the cornerstone for learning modern neural networks and the like. Finally, at the beginning of the 2000s, thanks to the World Wide Web and new computational power capabilities of modern computers, massive datasets were created, something called *big data* (Gandomi and Haider, 2015). Such datasets include billions of images, billions of hours of speech and video, trillions of words of text, genomic data, and social network data, among others. Such available data paved the way for developing learning algorithms specially designed to take advantage of massive datasets, specifically *deep learning* architectures (Goodfellow, Bengio, and Courville, 2016), which are the natural evolution of the connectionist AI paradigm.

§1.2 Motivations of this Thesis

Better symbolic learning models for unstructured data. *Structured* data refers to tabular data (organised in rows and columns) found in spreadsheets and relational databases, which can be easily analysed using data analytics software. It is often referred to as *quantitative* data because it is easy to count, measure, aggregate, and express in numbers. In contrast, *unstructured* data has no predefined structure, such as time series, images, audio, videos, or graphs. It is said to be *qualitative* data because it is subjective and interpretative¹. In the era of big data, unstructured data embodies 95% of all existing data (Gandomi and Haider, 2015), although other studies (are more conservative and) say it constitutes 80%. Table 1.2 points out the critical differences between structured and unstructured data. Deep neural networks are successful across various domains, including time series (Dempster, Petitjean, and Webb, 2020; Fawaz et al., 2020), images (Krizhevsky, Sutskever, and Hinton, 2012; He et al., 2016), audio (van den Oord et al., 2016), text (Devlin et al., 2019), and graphs (Scarselli et al., 2009; Bronstein et al., 2017; Zhou et al., 2020). The common denominator of such domains is the unstructured nature of data (sometimes also called *raw* data). The success of deep learning architectures boils down to the exploitation of the *inductive bias* in these data, that is, the set of assumptions that

¹The term unstructured data should be interpreted with caution. Such data *can* have a structure, but here we only refer to the fact that it does not have a well-studied data model. For instance, a graph has a structure, namely, vertices and edges between vertices, but there is no unique way to represent it.

the learner uses to predict outputs from unseen data (Mitchell, 1997), such as spatial correlations in convolutional neural networks (LeCun et al., 1989). However, sometimes deep learning models could be more effective on structured data. In contrast, decision tree models (Breiman et al., 1984; Quinlan, 1993), symbolic models by nature, and their siblings, such as random forests (Breiman, 2001) and gradient-boosted trees (Chen and Guestrin, 2016; Ke et al., 2017), are (still) dominant in such data (Shwartz-Ziv and Armon, 2022). Decision trees are among practitioner’s most commonly used algorithms for ML tasks, whereas deep learning architectures are less preferred, as a recent [Kaggle survey](#) suggests. In this work, we show how to extend symbolic models by augmenting the expressive power of their underlying logic, that is, replacing propositional logic with *modal logic(s)* (e.g., see Blackburn, de Rijke, and Venema, 2001). We aim, therefore, to perform qualitative reasoning and learn from such fast-growing and ubiquitous unstructured data to close the gap between structured and unstructured data concerning traditional symbolic learning.

Interpretability, ethics, and bias. AI and, in particular, ML models are becoming better and better every day. For example, in 2016, ALPHAGO (Silver et al., 2016), a deep learning model developed by Google’s [DeepMind](#), won against eighteen-time world champion Lee Sedol in the ancient Chinese game of Go. This very complex game was considered only human-playable using elaborate thinking and intuition². It represents a significant milestone for the AI community. However, it raises *ethical* concerns, even fears, as humankind will eventually feel that machines will take over (Coeckelbergh, 2020), reducing their economic and political power (Brynjolfsson, 2022). Deep learning models are considered *black boxes*: humans fail to understand their decision-making process. Another example of an ethical problem (in the automated decision-making process with AI) is concerned with the field of criminal justice. Automated systems proved to be *biased* against black people. For example, studies say that a black person is more than twice as likely to be arrested and five times as likely to be stopped without cause than a white person. The issue with these models is with the data they received in input: biased data leads to biased models. A further case in healthcare is where a black box AI model has systematically discriminated against black people (Obermeyer et al., 2019). Other studies show how black box ML models make wrong medical diagnoses and screening, while others show how they make lousy loan and credit decisions, to name others. If black box models are to be trained on biased data, the inability to “open” and inspect the models makes it very difficult to catch such biases. There have been many proposals to *explain* the decision-making process of neural networks, known in the literature as *explainable artificial intelligence (XAI)* (Gunning and Aha, 2019; Gunning et al., 2019), but this only perpetuates the problem. Indeed, Rudin (2019) proposed in her influential work to use *interpretable* models, that is, models that are *not* black boxes, instead of black box ML models, which, the latter, may be explained afterwards by a second (model-specific or model-agnostic) model for high-stakes decisions. There is a perplexing belief among ML practitioners that there is a trade-off between the accuracy of the models and their interpretability, as Rudin (2019) put it:

It is a myth that there is necessarily a trade-off between accuracy and interpretability.

Indeed, interpretable ML models can give new insights into the problem by inspecting the knowledge enclosed in them, which, if exploited, can improve themselves.

²See [AlphaGo - The Movie](#) | Full award-winning documentary.

We see transparency and interpretability as key to providing predictions that are not only statistically solid but also reliable, ethical, trustworthy, and unbiased. Trained interpretable prediction models can be inspected, adjusted, and validated by domain experts. These conditions are the natural playground for symbolic methods. As a real-world example of interpretability, consider that a (symbolic) rule-based model can disagree with doctors, correct them, and *teach* them novel relational patterns. This case was discussed in 2017 by Microsoft’s researcher Rich Caruana during the [Great Artificial Intelligence Debate](#) on “*Interpretability is necessary for ML*” at the annual conference on Advances in Neural Information Processing Systems (NIPS), which is the leading conference on neural networks, where the ACM Turing award Yann LeCun refuted such claim during the debate. The model discovered (the counterintuitive result) that asthma *lowers* the probability of death from pneumonia because asthmatic people are already plugged into the healthcare system. As a result, they tend to notice pneumonia and are, thus, treated earlier than other subjects. This sort of extracted knowledge would have been difficult, even impossible, to obtain with uninterpretable models. Brynjolfsson (2022) argues that AI should not focus only on automating human tasks but should *augment* humankind by creating new capabilities, goods and services; the pneumonia diagnosis example fits well in this argument. Thus, symbolic ML models would eventually, at least partially, augment humans in this sense. Moreover, the higher interpretability degree of symbolic models over non-symbolic ones raised a political debate in the [General Data Protection Regulation \(GDPR\)](#) of the European Council and Union that highlights the need for interpretable/explainable automatic decision processes for preferring a symbolic model. Finally, the symbolic approach could also lead to a better understanding of human learning abilities.

Rebirth of logical AI through symbolic learning. Broadly, *deduction* and *induction* are the two mainstream types of scientific reasoning. In mathematical reasoning, deduction proves theorems in an axiomatic system, that is, general-to-specific; distinctively, induction summarizes specific proofs to generalized theorems, that is, specific-to-general. The logical deduction may need to be revised due to its restrictive nature; for example, the surrounding world may not always be mathematically defined, and thus, conclusions are only sometimes derivable from it. Informally, to overcome some of such difficulties, we can, in principle, first observe the behaviours of the world and then extend systems to learn new facts from it. More formally, we can formulate general theories that account for the past and predict the future by exploiting such theories (Genesereth and Nilsson, 1988). The ability to generalize from examples is an inductive process known as *learning from examples* in the context of ML. It is important to stress that deduction is *exact*, while induction is more of a *statistical approach*, and thus, it is *approximate*. Specifically, we are interested in the sub-field of ML that deals with symbolic algorithms and models, known as *symbolic learning*, which have been known for decades and successfully applied to various contexts. Standard decision trees are the quintessential expression of *propositional symbolic learning*, where the extracted theories from data are essentially represented in propositional logic. In the same years when the term AI was coined, symbolic learning started with Belson (1956) working on decision tree development that employed ML algorithms to produce *executable* rules. Decision trees are emblematic of a whole class of other symbolic models. To name a few, by studying the foundations of decision trees, we can have:

- rule-based models that can be derived from decision tree models represented as *if-then* rules, such as *decision lists* (Rivest, 1987; Clark and Niblett, 1989), where the extracted rules are *ordered*;
- bootstrap aggregation, known as *bagging* (Breiman, 1996), of parallel decision trees, such as *random forests* (Breiman, 2001), where predictions are averaged over the hypotheses;
- *boosting* (Kearns and Valiant, 1994) of multiple sequential decision trees, such as *gradient-boosted trees* (e.g., see Chen and Guestrin, 2016; Ke et al., 2017), where *weak* hypotheses are converted to *strong* hypotheses, and predictions are weighted over the strength of such hypotheses;
- *hybrid* models combining the strengths of both symbolic and connectionist methods, such as *neural-symbolic decision trees* (e.g., see Guo and Gelfand, 1992; Zhou and Chen, 2002) and *tree-based neural networks* (e.g., see Srivastava and Salakhutdinov, 2013; Kotschieder et al., 2015; Murthy et al., 2016; Murdock et al., 2016; Alaniz et al., 2021; Wan et al., 2021);
- by investigating the logical elements of propositional logic decision trees and asking ourselves if such models can be enhanced to capture complex patterns, more expressive decision trees can be designed, such as *first-order logic (FOL) decision trees* (Blockeel and De Raedt, 1998) that learn from logic programs, which are essentially a kind of unstructured data from which standard, propositional decision trees would learn awkward theories (due to their limited expressivity).

Figure 1.1 illustrates schematically such specializations. Similar to the case of \mathcal{FOL} decision trees, where \mathcal{FOL} replaces propositional logic, in this thesis, we propose to replace propositional logic with modal logic(s) for the whole symbolic learning paradigm; we call the resulting framework *modal symbolic learning*. Modal logic is a fragment of \mathcal{FOL} , both syntactically and semantically. As such, our choice may seem restrictive, but several motivations justify it:

- \mathcal{FOL} is highly expressive, meaning that one needs to explicit *all* the knowledge, which, translated in ML terms, represents the *background theory* (Genesereth and Nilsson, 1988; Muggleton, 1991; Blockeel and De Raedt, 1998); for instance, if the task is learning from temporal domains such as audio, then the temporal domain must be defined by a collection of axioms expressed in \mathcal{FOL} , which, in turn, could be, to some extent, cumbersome;
- \mathcal{FOL} extracted theories (from data) would not have the same practicability of usage of more tailored logics for the same tasks; for example, (*modal*) *temporal logics* (Pnueli, 1977; Clarke, Emerson, and Sistla, 1986; Halpern and Shoham, 1991) are more suitable for learning from temporal data;
- \mathcal{FOL} learning is more expressive than modal logic learning but pays this price in computability; for example, \mathcal{FOL} extracted theories cannot always be automatically verified, as in the case of propositional logic.

Modal (symbolic) learning does not present such restrictions. There is no need to make explicit all the knowledge since it is essentially implicit in the underlying structural domain of data. Modal logics are fragments of \mathcal{FOL} , and in some of them the satisfiability problem is decidable (e.g. Pnueli, 1977; Clarke, Emerson, and Sistla,

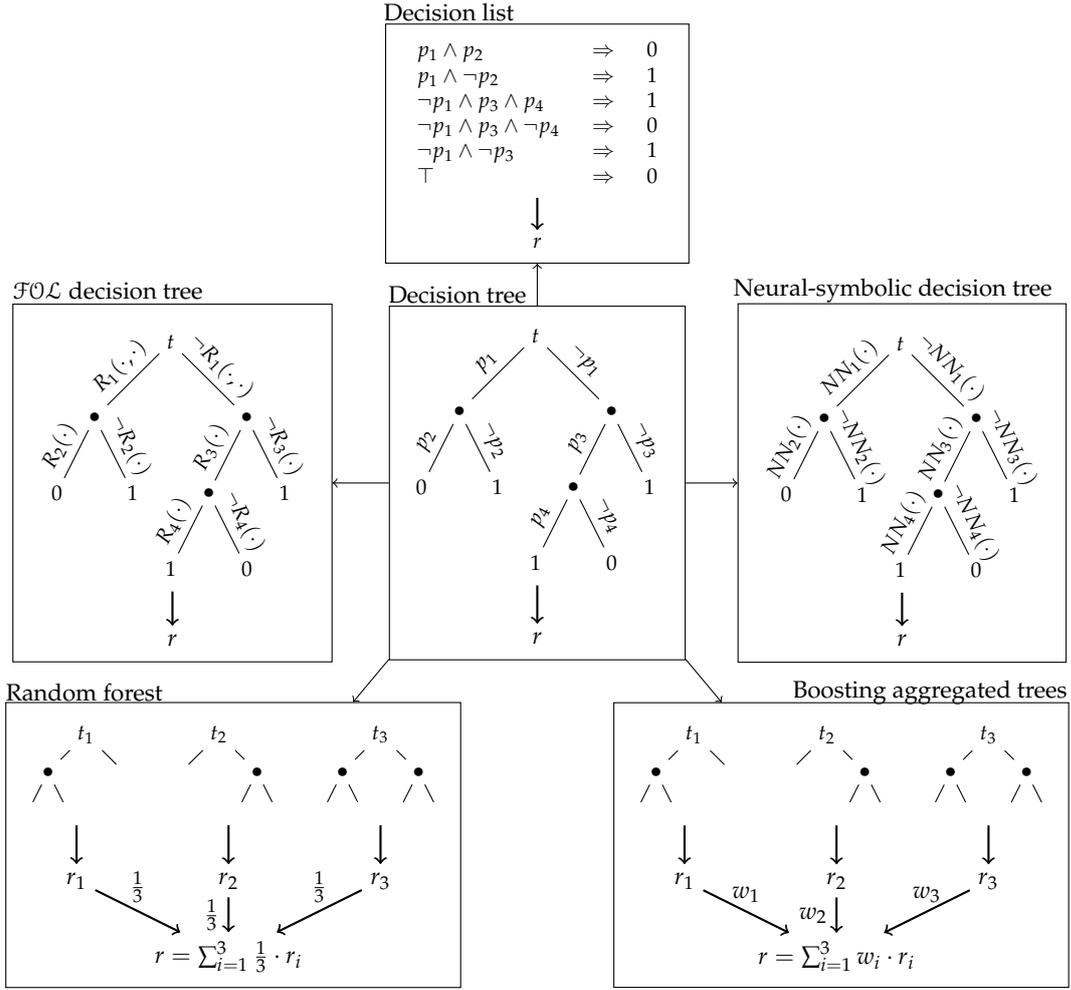


FIGURE 1.1: Some specializations of decision trees; r is the final result of the prediction of each model. A decision list is a set of ordered if-then rules. A random forest and boosted aggregated trees are ensembles of trees where the final result is weighted between the single trees. A neural-symbolic decision tree can have (deep) neural networks $NN_i(\cdot)$ that decide the splits, while a \mathcal{FOL} decision tree can have arbitrary relations $R_i(\cdot)$.

1986), in others it is undecidable (e.g., Halpern and Shoham, 1991) but can be restricted furthermore to obtain computationally efficient formalisms (e.g., Muñoz-Velasco et al., 2019). Whether the satisfiability problem is decidable, reasoning in modal logic is more straightforward to understand and implement than \mathcal{FOL} . Finally, it is important to stress that the decidability of the satisfiability problem is not the only element to consider when choosing a logical formalism. We also emphasize that our framework is integrable to all, but \mathcal{FOL} decision trees (as we motivated), singular settings in Figure 1.1, such as random forests and neural-symbolic decision trees, and many more, as we shall discuss throughout this work. We believe that symbolic learning deserves the attention of the mathematical and theoretical, but also practical, computer science researchers to push forward such an exciting and under-represented field in terms of inductive reasoning.

§1.3 Problem Statement and Contributions

In the previous section, we motivated the need for symbolic learning from challenging and enterprising perspectives. The ultimate ambition of this thesis is to bring together two essential fields of AI, namely, machine learning and modal logic(s). Instead of machine learning from unstructured data based on neural networks, our framework, called modal symbolic learning, brings together the two paradigms with the aim of enhancing symbolic learning algorithms with the ability to learn modal logic theories from different types of unstructured data, which, at this time, is not possible with canonical symbolic algorithms. It is desirable, therefore, to bridge the gap between structured and unstructured data by leveraging more expressive logics, such as modal ones discussed in this work, by exploring inductive biases at the symbolic level. The two scientific communities may join forces to embark on such an exciting field, benefiting from the known results from both sides (e.g., efficient learning algorithms and modal languages whose applications have been hampered due to undecidability that now may play a prominent role in machine learning).

The contribution of this work is many-folded. The technical contribution and novelty of the research in this thesis lie in the generalization of the well-known propositional symbolic learning approach to modal symbolic learning and showing how to learn modal logic theories from different types of unstructured data. We take standard decision trees, the quintessential representation of propositional symbolic learning, as representatives to elaborate and define the mathematical foundations of modal symbolic learning. By taking inspiration from existing taxonomies, we characterize the whole (modal) symbolic learning paradigm using inductive biases to speed up the investigation of new learning algorithms (and models). We provide a unifying view interpreting unstructured data types as Kripke structures, models of the considered modal logic formalism, by exploring such inductive biases and how many modal logics can be absorbed, in our context, by a single one. To this end, we show how to concretize the learning algorithms and the associated languages to specific modal logics to adapt to real-world scenarios. Such results are specified in precise algorithms and formulated in several theorems and lemmas. The results of a few experiments on two different real-world datasets are reported, one in the temporal and one in the spatial case, illustrating the learning capability of our framework. Moreover, we pave the way for advancing our proposal further, providing a wide vision of several research problems that go beyond the results described in this thesis. Finally, we survey the related literature to give a better perspective of the whole research field and allow assessing the merit and limitations of the proposed solutions.

§1.4 Organization of this Thesis

This thesis is organized as follows:

- **Chapter 2** This chapter provides the necessary mathematical background to understand this work. We discuss propositional and modal logic as these two logics play a crucial role in the presented work, especially the latter. We then provide general terms and definitions related to the field of ML since ML is also essential to grasp the purposes and properties of our framework. Finally, we present the taxonomy of symbolic learning that should guide every eager practitioner before starting.

- **Chapter 3** This chapter considers decision trees as representatives of the symbolic learning framework. We initially define and discuss the case of propositional decision trees. Then, we discuss the case of modal decision trees, which are the natural extension of propositional ones that learn modal logic theories. We prove the desiderata of modal decision trees: classification efficiency, correctness, and completeness. Finally, we formalize entropy-based learning and provide the algorithm for modal decision tree learning by lifting many ideas from the propositional case.
- **Chapter 4** This chapter shows how to embed unstructured data into modal datasets, where each instance is a Kripke model. We first present several examples of unstructured data to understand their complexity aiming at correctly handling such data. To do so, we discuss modal logics that fit well the considered data, and we consider several modal logics spanning from point-based to interval-based to topological-based ones. We then bring together under the same umbrella all, or at least many of, the discussed modal logics to have a tool for elegantly treating many cases as one. Finally, we define the modal logic transformer that produces Kripke models to reason with the chosen logical formalism.
- **Chapter 5** This chapter presents several ways to learn with our framework. We discuss how regression with modal decision trees is possible by simply adapting the learning schema from the propositional level. Propositional random forests are well-known for learning better models, and we discuss how to obtain modal random forests. Then, we present how to extract if-then rules from modal decision trees and random forests. We also discuss a practical situation where events are described by multiple descriptions at the same time, and we present a way to learn from them. Finally, we briefly discuss the blueprint of modal symbolic learning, before concluding with two real-world learning applications from temporal and spatial data.
- **Chapter 6** In this chapter, we highlight some extensions of our framework. By leveraging different levels of hybridization, we discuss (hybrid) neural-symbolic modal decision trees. We also discuss the need for fuzzy reasoners paving the way for fuzzy modal decision trees. Since propositional gradient-boosted trees are widely accepted and used by practitioners in daily challenges, such as Kaggle, we discuss gradient-boosted modal decision trees. We define then incremental learning and give directions on incremental learning with modal decision trees. Finally, inspired by the recent advances in geometric deep learning, we discuss the benefits of having geometric modal symbolic learning.
- **Chapter 7** We review the related works in this chapter. We briefly discuss the history of propositional decision trees. Then, we discuss approaches for learning from temporal and spatial data, which are unstructured data. Finally, we discuss symbolic approaches for learning from unstructured data.
- **Chapter 8** This chapter presents the conclusions of our work. We also discuss the future directions of our framework.

§1.5 Published (and in Press) Results

As of January 2022, I have published a series of papers, some of which are strongly related to this work, while others are less related, although they go in the same direction. In this section, from a chronological point of view, I discuss my publications briefly by grouping together those articles that are related to the same discussion.

In (Muñoz-Velasco et al., 2019), I have studied coarser fragments of \mathcal{HS} , an undecidable logic that we discuss in Section 4.2, that have better computational properties, namely, \mathcal{HS}_7 and \mathcal{HS}_3 . The former remains undecidable, while the latter is PSPACE-hard. Moreover, I have implemented a tableau system for the satisfiability problem of \mathcal{HS}_3 (Muñoz-Velasco, Sciavicco, and Stan, 2017).

As we have motivated in this chapter, we take decision trees as emblematic of the symbolic learning paradigm. I have studied the properties of modal decision trees in (Della Monica et al., 2022), which we discuss in Section 3.3. In (Brunello, Sciavicco, and Stan, 2019), I have presented the first temporal decision tree, which is a specialization of the modal decision trees that we discuss in Section 3.2, which mines patterns described in the language of \mathcal{HS} , where the assumption is that the input objects to learn from are Kripke models. In (Sciavicco and Stan, 2020), I have enhanced the previous contribution to learn from multivariate time series; we discuss in Section 4.1 what multivariate time series are, and then, in Section 4.4, we discuss how such objects can be seen as Kripke models. In real-world situations, multiple descriptions describe an event, and learning in such a multi-setting is challenging. In (Pagliarini, Sciavicco, and Stan, 2021), I have proposed a method for learning from multiple descriptions simultaneously, namely, multi-frame modal symbolic learning, which we discuss in Section 5.4. I have applied temporal decision trees and their random forest version, which we discuss in Section 5.2, in (Manzella et al., 2021) to diagnose positive from negative COVID-19 subjects; the same setting has been applied in (Manzella et al., 2022) in the multi-frame setting, whose results we discuss in Section 5.6. In (Bechini et al., 2023), I have applied temporal decision trees in the industrial domain to predict trip events (i.e., anomalous behaviours) in gas turbines, whose results we do not discuss in this thesis.

As we discuss in Section 5.1, decision trees can also apply to regression problems. Based on this principle, I have studied temporal decision trees for regression in (Lucena-Sánchez, Sciavicco, and Stan, 2020). Indeed, in (Lucena-Sánchez, Sciavicco, and Stan, 2021), I have developed a multi-objective optimization problem, solved via heuristic search, employing genetic algorithms, to mine \mathcal{HS} patterns from time series to model air quality, whose results we do not discuss in this thesis.

Rule-based systems are another essential point in the symbolic learning domain, and we briefly discuss this setting in Section 5.3. In (Stan et al., 2022), I have proposed to mine modal logic association rules from Kripke models. However, we do not discuss in-depth such a result because focusing on decision trees is enough to grasp the entire idea of modal symbolic learning. Sticking to the rule-based side, I have proposed other rule extraction methods (e.g., see Lucena-Sánchez et al., 2019; Sciavicco, Stan, and Vaccari, 2019; Kaminska et al., 2020).

We discuss in Section 6.2 that modal decision trees can also be fuzzy. Since we take a generalization of \mathcal{HS} as representative for modal symbolic learning, namely, \mathcal{HS}^d , which we discuss in Section 4.3, I have studied the fuzzy generalization of \mathcal{HS} in (Conradie et al., 2022), whose model checking problem applied to the case of multivariate time series has been studied a couple of years before in (Conradie et

al., 2020). Keeping our focus on the model checking problem, I have also studied the issue of ultimately-periodic interval temporal logic model checking in (Della Monica et al., 2020), but such a result is not presented in this thesis. Finally, in Section 6.1, we discuss the neural-symbolic hybridization of modal decision trees, and I have studied the hybridization of temporal decision trees in (Pagliarini et al., 2022).

BACKGROUND

Be kind, resourceful, beautiful, friendly, have initiative, have a sense of humour, tell right from wrong, make mistakes, fall in love, enjoy strawberries and cream, make some one fall in love with it, learn from experience, use words properly, be the subject of its own thought, have as much diversity of behaviour as a man, do something really new.

No support is usually offered for these statements. I believe they are mostly founded on the principle of scientific induction.

—Alan Turing “Computing Machinery and Intelligence” (Turing, 1950)

In this chapter, we discuss the required background to understand our work. At the end of this chapter, we present the taxonomy of symbolic learning, which shall guide every symbolic ML practitioner.

§2.1 Propositional Logic

Logic and theoretical computer science are intimately tied as logic is at the core of the birth of computer science (Davis, 2018). The starting point in this journey is the *propositional language*.

Let \mathcal{P} be a (possibly infinite, but countable) set of *proposition letters* (or, simply, *propositions*). We use $p, q, \dots, p_1, q_1, p_2, q_2, \dots$ to denote propositions. The well-formed formulas of *propositional logic* (\mathcal{PL}) are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi,$$

where $p \in \mathcal{P}$. We use $\varphi, \psi, \dots, \varphi_1, \psi_1, \varphi_2, \psi_2, \dots$ to denote formulas. The remaining propositional abbreviations are derived as usual, that is:

$$\begin{aligned} \top &\triangleq p \vee \neg p \text{ for some } p \in \mathcal{P}, \\ \perp &\triangleq \neg\top, \\ \psi_1 \wedge \psi_2 &\triangleq \neg(\neg\psi_1 \vee \neg\psi_2), \\ \psi_1 \rightarrow \psi_2 &\triangleq \neg\psi_1 \vee \psi_2, \text{ and} \\ \psi_1 \leftrightarrow \psi_2 &\triangleq (\psi_1 \rightarrow \psi_2) \wedge (\psi_2 \rightarrow \psi_1). \end{aligned}$$

Let $\Phi(\mathcal{L})$ be the smallest set that contains all the *formulas* generated by the grammar of the logic \mathcal{L} . For instance, $\Phi(\mathcal{PL})$ are the formulas generated by the grammar of \mathcal{PL} .

The semantics of \mathcal{PL} are given in terms of propositional models. A *propositional model* $\mathfrak{K} = (V)$ consists of a function $V : \mathcal{P} \rightarrow \{\top, \perp\}$ that assigns truth values to propositional letters. The (*semantical*) *truth relation* $\mathfrak{K} \models \varphi$, for a propositional model

\mathfrak{K} and a formula $\varphi \in \Phi(\mathcal{PL})$, is defined by induction on the complexity of formulas:

$$\begin{aligned} \mathfrak{K} \models p & \quad \text{iff } V(p) = \top, \text{ for all } p \in \mathcal{P}; \\ \mathfrak{K} \models \neg\psi & \quad \text{iff } \mathfrak{K} \not\models \psi \text{ (i.e., it is not the case that } \mathfrak{K} \models \psi); \\ \mathfrak{K} \models \psi_1 \vee \psi_2 & \quad \text{iff } \mathfrak{K} \models \psi_1 \text{ or } \mathfrak{K} \models \psi_2. \end{aligned}$$

§2.2 Modal Logic

We now move to propositional *modal* languages, that is, \mathcal{PL} enriched with (a set of) *modal operators* (or *modalities*). Unlike in *first-order* modal logic (Fitting and Mendelsohn, 1998), propositional modal logic operators do *not* bind variables (Blackburn, de Rijke, and Venema, 2001). Narrow speaking, modal logic was initially investigated as the logic of *necessary* and *possible* truths of judgments due to Aristotle's foresighted analysis of statements containing the words "*necessary*" and "*possible*". Having just two modalities may seem (at first glance) restrictive, but these are but two of a wide range of them: modal logic is paradigmatic for a family of related systems. Arguably, Lewis (1918) formalised propositional modal languages more than a century ago. As such, there are many works on modal logic (and variants of it); we will see more-than-modal, but still propositional, languages throughout this work.

Modal logic is an extension of classical \mathcal{PL} that allows us to characterize the validity of arguments with *modal* premises and conclusions. The well-formed formulas of (*propositional*) *modal logic* (\mathcal{ML}) are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \diamond\varphi,$$

where $p \in \mathcal{P}$. The remaining propositional abbreviations are derived as before. Just as existential (\exists) and universal (\forall) quantifiers in classical first-order languages are dual to each other (i.e., $\forall\varphi$ if and only if $\neg\exists\neg\varphi$), in \mathcal{ML} there is a dual operator called *box* (\square) for the diamond modality, defined as $\square\varphi$ if and only if $\neg\diamond\neg\varphi$. Let $\varphi \in \Phi(\mathcal{ML})$ be a formula of \mathcal{ML} , then the set of all *subformulas* of φ , denoted by $sub(\varphi)$, is defined as:

$$sub(\varphi) = \begin{cases} \{p\} & \text{if } \varphi = p \in \mathcal{P}; \\ \{\varphi\} \cup sub(\psi) & \text{if } \varphi = \neg\psi; \\ \{\varphi\} \cup sub(\psi_1) \cup sub(\psi_2) & \text{if } \varphi = \psi_1 \vee \psi_2; \\ \{\varphi\} \cup sub(\psi) & \text{if } \varphi = \diamond\psi. \end{cases}$$

and the *modal-depth* of φ , denoted by $md(\varphi)$, is defined as:

$$md(\varphi) = \begin{cases} 0 & \text{if } \varphi = p \in \mathcal{P}; \\ md(\psi) & \text{if } \varphi = \neg\psi; \\ \max\{md(\psi_1), md(\psi_2)\} & \text{if } \varphi = \psi_1 \vee \psi_2; \\ 1 + md(\psi) & \text{if } \varphi = \diamond\psi. \end{cases}$$

Moreover, the *length* of a formula φ is the number of its symbols.

Two schools of thought arose of mathematical semantics of modal language since its birth (Goldblatt, 2003). *Algebraic* semantics interprets modalities on Boolean algebras with operators (McKinsey and Tarski, 1944). *Relational* semantics, on the other hand, uses relational structures, commonly called Kripke models (named after its founder Kripke, 1963). More broadly, a relational structure is a tuple whose first

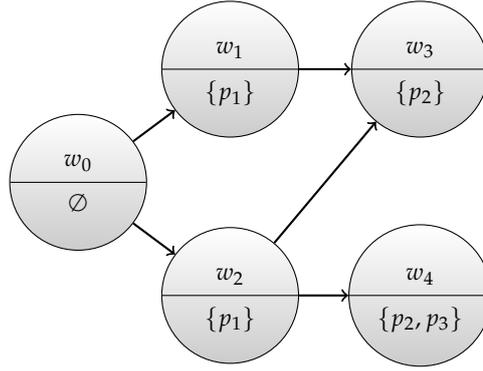


FIGURE 2.1: Example of a Kripke model.

element is a non-empty set, called *universe* (or *domain*), and whose remaining components are *relations* on the universe. A *Kripke frame* $\mathfrak{F} = (\mathcal{W}, \mathcal{R})$ consists of a non-empty (possibly infinite, but countable) set of (*possible*) *worlds* \mathcal{W} , and an *accessibility relation* over worlds $\mathcal{R} \subseteq \mathcal{W} \times \mathcal{W}$. A *Kripke model* $\mathfrak{K} = (\mathcal{W}, \mathcal{R}, V)$, over \mathcal{P} , is a Kripke frame $\mathfrak{F} = (\mathcal{W}, \mathcal{R})$ enriched with a *valuation function* $V : \mathcal{W} \rightarrow 2^{\mathcal{P}}$, which associates each world w with the set of propositions $V(w) \subseteq \mathcal{P}$ that are true on it, while the remaining propositions in $\mathcal{P} \setminus V(w)$ are false on it. Figure 2.1 illustrates an example of a Kripke model. Frames and models are relational structures based on the same universe (i.e., set of worlds), but a model is a frame augmented with a set of unary relations \mathcal{V}_w interpreted as $V(w)$, for all $w \in \mathcal{W}$, given by the valuation function V . What is more, the (Kripke) frame provides the *structure* of the model, and the valuation function provides the *semantical meaning* of which propositional letters are true relative to worlds; one could also define the valuation function as $V : \mathcal{W} \times \mathcal{P} \rightarrow \{\perp, \top\}$, which assigns truth values to each pair of world $w \in \mathcal{W}$ and propositional letter $p \in \mathcal{P}$, but, to keep things simple, we prefer the former version. The truth relation $\mathfrak{K}, w \Vdash \varphi$, for a (Kripke) model \mathfrak{K} , a world w (in that model) and a formula $\varphi \in \Phi(\mathcal{ML})$ (to be interpreted on that model), is defined by induction on the complexity of the formulas¹:

$$\begin{array}{ll}
 \mathfrak{K}, w \Vdash p & \text{iff } p \in V(w), \text{ for all } p \in \mathcal{P}; \\
 \mathfrak{K}, w \Vdash \neg\psi & \text{iff } \mathfrak{K}, w \not\Vdash \psi \text{ (i.e., it is not the case that } \mathfrak{K}, w \Vdash \psi); \\
 \mathfrak{K}, w \Vdash \psi_1 \vee \psi_2 & \text{iff } \mathfrak{K}, w \Vdash \psi_1 \text{ or } \mathfrak{K}, w \Vdash \psi_2; \\
 \mathfrak{K}, w \Vdash \diamond\psi & \text{iff there exists } w' \text{ s.t. } w\mathcal{R}w' \text{ and } \mathfrak{K}, w' \Vdash \psi.
 \end{array}$$

We write $\mathfrak{K} \Vdash \varphi$ as an abbreviation for $\mathfrak{K}, w_0 \Vdash \varphi$, where w_0 is the *initial world* of \mathfrak{K} . Observe that the definition of truth relation is intrinsically *internal* and *local*: formulas are evaluated *inside* models at some particular world (i.e., *current state*), and $\diamond\psi$ *locally* scans the \mathcal{R} -accessible worlds searching for one possible world where ψ is true; similarly, $\square\psi$ *locally* scans the \mathcal{R} -accessible worlds to assess that ψ is true on them, if any, that is:

$$\mathfrak{K}, w \Vdash \square\psi \quad \text{iff for all } w', \text{ if } (w, w') \in \mathcal{R}, \text{ then } \mathfrak{K}, w' \Vdash \psi.$$

In linguistics terms, $\diamond\psi$ is read as “*possibly* ψ ”, and $\square\psi$ is read as “*necessarily* ψ ”. This reading is often attributed to Leibniz due to his studies on modal metaphysics

¹It is common to use the symbol \Vdash for non-classical logics truth relation, such as modal logics. In contrast, it is common to use the symbol \models for classical logics truth relation, such as propositional logic and first-order logic.

that “possibility” means “truth in some possible world”, and “necessity” means “truth in all possible worlds”. The following definitions hold for any modal system. Given a model \mathfrak{K} and a formula φ , we say that \mathfrak{K} *satisfies* φ if there exists a world w such that $\mathfrak{K}, w \Vdash \varphi$. A formula φ is *satisfiable* if there exists a model and a world that satisfies it. A formula φ is *valid in a model* \mathfrak{K} if every world in \mathfrak{K} satisfies it, that is, $\mathfrak{K}, w \Vdash \varphi$, for all w . Moreover, a formula φ is *valid* in a modal system (e.g., \mathbf{K}) if it is valid in every model of the modal system.

\mathcal{ML} arises from philosophical inquiry. Extensions of (basic) \mathcal{ML} include more-than-one accessibility relations and constraints on such relations, among others. From the theoretical point of view, \mathcal{ML} is paradigmatic for such logics, and, from an application point of view, expressive enough that it has attracted mathematical and scientific inquiry in the field of deductive reasoning. Among the many interesting mathematical problems studied over the years in the field of \mathcal{ML} s is *model checking* (Clarke et al., 2018); indeed, in 2007 Edmund M. Clarke, E. Allen Emerson, and Joseph Sifakis won the ACM Turing award for their roles in developing model checking.

Verifying the correctness of hardware and software systems is of utmost importance as their applications are ubiquitous in our daily lives, where failure is critical and should be avoided. Verification is more appreciated in *safety-critical* systems (e.g., e-health), *commercially critical* systems (e.g., e-commerce), and *mission-critical* systems (e.g., space missions), among others. The principal tools for assessing the correctness of complex systems are simulation, testing, deductive verification, and model checking. *Simulation* and *testing* involve *conducting experiments* before the actual deployment of the system in production. Simulation is performed on an abstraction, or model, of the system, while testing is performed on the original system. Simulation and testing are cost-effective approaches to finding many errors; however, checking *all* the possible interactions and pitfalls is rarely possible. *Deductive verification* uses *axioms* and *proof rules* to check the correctness of (possible infinite state) systems. Deductive verification is time-consuming as it can be performed only by educated experts to logical reasoning. *Model checking* is a technique for automatically verifying finite state concurrent systems (Clarke et al., 2018). Usually, the model checker *exhaustively* searches through the finite state space of the system to assess if some specification (i.e., property of the system) is true or not. An excellent characteristic of model checkers is that they produce a *counterexample* that proves the wrong behaviour of the system when the system fails to satisfy the desired property, which is very useful for debugging. The *theory of computability* provides limitations to what can, or cannot, be decided by an algorithm (Papadimitriou, 1994; Sipser, 2013), and this is the case also for the model checking problem. Therefore, restrictions on systems and on properties to be verified must be taken into account whenever we aim to develop tools for automatic verification.

The model checking process encompasses three parts (Huth and Ryan, 2004): modelling, specification, and verification. *Modelling* converts the design of a system into an *abstract model*, which is accepted by a model checking tool and which should eliminate irrelevant details of the design; it is standard to use Kripke models to model the behaviour of systems. *Specification* states the property, or properties, that a system must satisfy, usually specified in propositional modal languages (e.g., temporal logics). *Verification* establishes whether the description of a system satisfies the specification(s). Formally, the *model checking problem* is the process of establishing if:

$$\mathfrak{K}, w \Vdash \varphi,$$

where $\mathfrak{K} = (\mathcal{W}, \mathcal{R}, V)$ is a Kripke model, $w \in \mathcal{W}$ is a world of \mathfrak{K} , and $\varphi \in \Phi(\mathcal{ML})$ is a formula of \mathcal{ML} . Canonically, model checking is the problem of verifying temporal logic, that is, \mathcal{ML} customized to reason over temporal domains, properties on *infinite state, finitely represented*, abstract models (i.e., Kripke models) of concrete ones (e.g., reactive systems). Depending on the logical formalism, model checking may not be a trivial task; for example, Sistla and Clarke (1985) showed that the *infinite* model checking for *linear temporal logic* (\mathcal{LTL}) (Pnueli, 1977) formulas is PSPACE-complete. The common denominator of the ML logical approaches is that the kind of model checking, which is *crucial* for the entire learning process, is, in fact, *finite*. The fact that model checking is finite for learning trivializes, to some extent, the problem itself, which generally becomes PTIME; nevertheless, it still raises many difficulties that must be addressed with mathematical rigour. For example, learning from *sparse* inputs (i.e., with a lot of missing values, or better, in logical terms, with many states in the Kripke models without any propositional letters) may lead to an *exponential* blow-up in the size of the input when performing model checking; Della Monica et al. (2017) proposed a bisimulation algorithm between a sparse and a non-sparse interval temporal (Kripke) models, so that model checking on the non-sparse model remains PTIME. A further example, which is of interest in this work, emerges in the context of model checking *multiple* models against *multiple* formulas; this generalization is needed for the entire inductive process that learns a general theory (seen as multiple formulas) from data (seen as various models).

Algorithm 1 illustrates the *finite* model checking algorithm for \mathcal{ML} . The algorithm computes a mapping $\ell : \mathcal{W} \rightarrow 2^{\Phi(\mathcal{ML})}$ which intuitively labels each world $w \in \mathcal{W}$ with the subformulas $\psi \in \text{sub}(\varphi)$ that are true on it. The entire process is a big loop on all the subformulas of the input formula φ by increasing length; this is because model checking is performed bottom-up on the syntax tree of the formula φ . The main loop, thus, runs for $|\text{sub}(\varphi)|$ times. Each subformula $\psi \in \text{sub}(\varphi)$ can be a proposition letter $p \in \mathcal{P}$, a negation $\neg\psi_1$, a disjunction $\psi_1 \vee \psi_2$, or a formula prepended with a diamond $\diamond\psi_1$. The Boolean cases cost $O(|\mathcal{W}|)$ and the modal case costs $O(|\mathcal{W}| + |\mathcal{R}|)$. Hence, the model checker for \mathcal{ML} runs in time $O(|\text{sub}(\varphi)| \cdot (|\mathcal{W}| + |\mathcal{R}|))$ in the worst-case, which is linear in the product of the length of the formula φ and the size of the Kripke model \mathfrak{K} .

§2.3 Machine Learning

ML focuses on building computer programs that automatically improve with experience (Mitchell, 1997). Data represents experience, and the main ML task is to develop *learning algorithms* that build *models* from data (Zhou, 2021). Thus, ML techniques are the primary approach in the era of big data.

In ML tasks, we have a *dataset* $\mathcal{X} = \{x_1, \dots, x_m\}$ of m *instances* (or *samples*) each containing the description of an event or an object. The space from which instances are drawn is called *sample space* (or *input space*) denoted by \mathbb{X} . *Learning* (or *training*) is the process of using ML algorithms to build models from data. In the training phase, the used dataset is called *training data*, where each instance is called *training instance*, and the set of all training instances is called *training set*. If the instances are associated with a *target* (or *response*) *variable*, then the learning task is called *supervised learning*; otherwise, it is called *unsupervised learning*. Within the supervised learning paradigm, based on the type of target variable, we have:

ALGORITHM 1: Model checking for \mathcal{ML} .

```

1 function Check( $\mathfrak{K}, \varphi$ ):
  input : A Kripke model  $\mathfrak{K} = (\mathcal{W}, \mathcal{R}, V)$  and an  $\mathcal{ML}$  formula  $\varphi$ .
  output: A mapping  $\ell : \mathcal{W} \rightarrow 2^{\Phi(\mathcal{ML})}$ 
2  foreach  $\psi \in \text{sub}(\varphi)$  ordered by increasing length do
3    if  $\psi = p \in \mathcal{P}$  then
4      foreach  $w \in \mathcal{W}$  do
5        if  $p \in V(w)$  then
6           $\ell(w) \leftarrow \{p\}$ 
7        end
8      end
9    end
10   if  $\psi = \neg\psi_1$  then
11     foreach  $w \in \mathcal{W}$  do
12       if  $\psi_1 \notin \ell(w)$  then
13          $\ell(w) \leftarrow \ell(w) \cup \{\psi\}$ 
14       end
15     end
16   end
17   if  $\psi = \psi_1 \vee \psi_2$  then
18     foreach  $w \in \mathcal{W}$  do
19       if  $\psi_1 \in \ell(w)$  or  $\psi_2 \in \ell(w)$  then
20          $\ell(w) \leftarrow \ell(w) \cup \{\psi\}$ 
21       end
22     end
23   end
24   if  $\psi = \diamond\psi_1$  then
25     foreach  $w \in \mathcal{W}$  do
26       if  $\exists w'$  such that  $(w, w') \in \mathcal{R}$  and  $\psi_1 \in \ell(w')$  then
27          $\ell(w) \leftarrow \ell(w) \cup \{\psi\}$ 
28       end
29     end
30   end
31 end
32 return  $\ell$ 
33 end

```

- *classification problems* if the target variable, also known as *class variable* (or, simply, *class*), is categorical;
- *regression problems* if the target variable is numerical.

For supervised tasks, a dataset \mathcal{X} is called *labelled* if each instance is labelled with a *label* from a set \mathbb{Y} called *label space* (or *output space*); for classification tasks, the labels are also called *classes*. Let $\mathbb{Y} = \{0, 1\}$ for *binary* classification problems, $|\mathbb{Y}| > 2$ for *multiclass* classification problems, that is, more than two classes are present, and $\mathbb{Y} = \mathbb{R}$ for regression problems, where \mathbb{R} is the set of real numbers. In general, we denote the i -th labelled instance as (x_i, y_i) , where $x_i \in \mathbb{X}$ and $y_i \in \mathbb{Y}$ is the label for such sample; therefore, a labelled dataset is $\mathcal{X} = \{(x_1, y_1), \dots, (x_m, y_m)\}$.

To understand the general ML terminology, it is canonical to consider the supervised learning paradigm. In supervised learning, the objective is to learn a function h called *hypothesis* drawn from a *hypothesis space* \mathbb{H} of possible functions that approximates with a reasonable margin of error the relationship between the input space and the output space. In alternative terms, h is a *model* and \mathbb{H} is the *model space*. The *best predictive model* h^* , also known as *best-fit model*, is the one that minimises the

risk (Gambella, Ghaddar, and Naoum-Sawaya, 2021):

$$\mathbb{E}_p[\text{loss}(h(x), y)] = \int_{\mathbb{Y}} \int_{\mathbb{X}} p(x, y) \text{loss}(h(x), y) dx dy,$$

where $\text{loss}(h(x), y)$ is the *loss function* that measures the accuracy of a prediction, and $p(x, y)$ is the *probability of observing* (x, y) . In practice, $p(x, y)$ is unknown. However, since the given instances in a labelled dataset are assumed to be *independent and identically distributed*, h^* is obtained by minimising the *empirical risk*:

$$h^* = \arg \min_{h \in \mathbb{H}} \frac{1}{m} \sum_{i=1}^m \text{loss}(h(x_i), y_i).$$

We can make predictions with h^* , a process called *testing*, and the instances to be predicted are the *testing instances*; moreover, the set of testing instances is called *testing set*. For example, the (predicted) label \hat{y} of the testing instance x can be obtained with the learned model $\hat{y} = h^*(x)$. The ability to predict new, unseen before, instances is called *generalization ability*, and should work well on the whole sample space \mathbb{X} .

We need to define performance measures, which depend on the ML task (e.g., classification), to quantify the generalization ability. Let $\mathcal{X} = \{(x_1, y_1), \dots, (x_m, y_m)\}$ be a labelled dataset and let h be an hypothesis. To evaluate the performance of h , we compare its predictions $h(x_i) = \hat{y}_i$ with the ground truth y_i , for all $1 \leq i \leq m$. For binary classification tasks, the generic performance of h on \mathcal{X} can be measured in terms of its *confusion matrix*, which, for each given instance, expresses one of four mutually exclusive indicators, namely, *true positive*, *true negative*, *false positive*, and *false negative*, by comparing y with \hat{y} as:

	$y = 0$	$y = 1$
$\hat{y} = 0$	true negative (TN)	false negative (FN)
$\hat{y} = 1$	false positive (FP)	true positive (TP)

Accuracy, which measures the number of correctly classified instances, is defined as:

$$\text{acc} = \frac{TN + TP}{TN + FN + FP + TP}.$$

In case of unbalanced datasets, other measures are preferred, such as precision, recall and F1 score. *Precision*, also called *positive predictive value*, is defined as:

$$\text{prec} = \frac{TP}{TP + FP},$$

recall, also called *sensitivity*, is defined as:

$$\text{rec} = \frac{TP}{TP + FN},$$

and *F1 score*, which is the harmonic mean of precision and recall, is defined as:

$$F1 = 2 \cdot \frac{\text{prec} \cdot \text{rec}}{\text{prec} + \text{rec}}.$$

Similar metrics can be defined for regression tasks, but such a discussion goes beyond the scope of this work; indeed, in our experiments, we address classification tasks.

The *model complexity* must be considered when learning a model, which is measured in terms of its *size*. *Overfitting* is the problem when the learned model fits the training set very well but performs poorly on the testing set. The minimisation of the empirical risk often leads to overfitted models, and hence, has a poor generalization ability. Therefore, a better predictive model can be obtained by minimising the *regularised empirical risk* (Russell and Norvig, 2020):

$$h^* = \arg \min_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m \text{loss}(h(x_i), y_i) + \chi \cdot \text{size}(h),$$

where $\text{size}(h)$ is the size/complexity of the hypothesis h and the parameter χ is the error one would sacrifice to have one fewer term in the model. In other words, regularisation penalizes complex hypotheses to generalize better (on future samples).

The minimisation of the (regularised) empirical risk forces the learner to choose a hypothesis from the set of hypothesis space; that is, the learner is *biased* toward a particular class of predictive models. Such restriction, also known as *inductive bias*, is imposed regardless of the training set and ideally should be based on some prior domain-expert knowledge about the task to be handled (by ML). For example, the class of linear functions induces a strong bias since the resulting best hypothesis h^* will result in a straight line limited to capture only linear separable patterns, which may not always be the case for noisy data. Another essential way to analyse hypothesis spaces is in terms of the *variance* they produce, which measures the amount of change in the hypothesis due to fluctuation in the training set. Broadly, choosing a more restrictive inductive bias prevents the resulting model from overfitting, but at the same time, it reduces its variance. Formally, choosing between simpler, low-variance hypotheses with better generalization ability and more complex, low-bias hypotheses that fit the training data well is known as the *bias-variance trade-off*. *Ockham's razor* principle says we should choose the simplest hypothesis matching the data.

The statistical learning approaches can be separated between *parametric* and *non-parametric* (James et al., 2013), and this is also the case for ML approaches (Russell and Norvig, 2020). Parametric learning is the process of summarizing data with a set of *parameters* of fixed size that do not grow as the training set grows (i.e., they are independent of the number of training instances), spanning from simple methods (e.g., linear regression), to probabilistic ones (e.g., naive Bayes), to more complex ones (e.g., deep neural networks). Non-parametric learning, on the other hand, cannot summarize data with a fixed set of parameters, spanning from tree-like methods (e.g., decision trees), to instance-based ones (e.g., k -nearest neighbours), and others (e.g., support vector machines). Parametric models are constrained as they choose the shape of the hypothesis space, which in turn may *underfit* (i.e., the opposite of overfit) the training set. Deep neural networks are *universal function approximators* (Hornik, Stinchcombe, and White, 1989; Hornik, 1991), that is, they can approximate any mathematical function by increasing the numbers of parameters, which can be millions (Vaswani et al., 2017) or even trillions (Fedus, Zoph, and Shazeer, 2022) in modern architectures, to overcome underfitting at the expenses of computational resources. Non-parametric models make no assumptions about the underlying hypothesis space, and this is of greater benefit for simple parametric models (i.e., non-deep neural networks) as they can learn a vast number of mathematical

functions, but they tend to overfit the training set; in such cases, regularisation techniques are generally preferred. Another fundamental separation between the subfields of ML, which is of interest in this work, is the one between *symbolic* and *non-symbolic* learning. Symbolic learning is learning a *logical description* that represents the theory underlying a particular phenomenon, such as decision trees or rule-based classifiers. Non-symbolic learning is learning a *non-logical description* representing that phenomenon, such as deep neural networks or naive Bayes. Therefore, decision trees, which we use as a guiding example to present the modal symbolic learning framework, are non-parametric and symbolic models.

§2.4 Taxonomy of Symbolic Learning

Biases characterize ML algorithms. We can, thus, characterize symbolic learning algorithms along four dimensions:

- *conceptual bias* which defines the vocabulary (i.e., propositional letters) of formulas,
- *logical bias* that defines the logical form (i.e., grammar) of formulas,
- *interpretation bias* which defines how concepts (i.e., formulas) are evaluated in semantical terms, and
- *search bias* that refers to how the hypothesis space is explored.

Many important considerations must be made of the proposed taxonomy. Conceptual and logical biases were defined by Genesereth and Nilsson (1988). Fürnkranz (1999) introduced *language bias* encompassing both, conceptual bias and logical bias, under the same umbrella; we prefer to distinguish between the two in our framework as many assumptions can be made about each of them individually. Moreover, Fürnkranz (1999) also introduced search bias and *overfitting avoidance bias* which the author claims to be a type of search bias; we prefer to collapse both definitions in search bias.

In linguistics, syntax (i.e., tokens and their composition), semantics (i.e., the meaning of sentences), and pragmatics (i.e., how constructs and features of the language are used to produce new sound sentences) are used to characterize languages, and this is also the case for formal languages, such as logics. In deductive reasoning, theorems can be proved in an axiomatic system (i.e., pragmatics), given the language (i.e., syntax) along with its meaning (i.e., semantics); indeed, for example, the satisfiability problem is among the most addressed problems in mathematical inquiry. Inspired by the duality between deduction and induction processes, in inductive reasoning, on the other hand, conceptual bias and logical bias represent the syntax, interpretation bias represents the semantics, and search bias represents the pragmatics. In particular, interpretation bias (which, to the best of our knowledge, is introduced here for the first time) is motivated by the duality between crisp and fuzzy logics; for example, propositional letters could be interpreted with Boolean truth values (i.e., crisp) and the remaining connectives of the grammar with multiple truth values (i.e., fuzzy). Thus, in their endeavours, symbolic learning researchers and practitioners must make hypotheses on each dimension of the taxonomy to design symbolic learning algorithms.

MODAL DECISION TREES

As a matter of fact, logic has turned out to be significantly more effective in computer science than it has been in mathematics. This is quite remarkable, especially since much of the impetus for the development of logic during the past one hundred years came from mathematics.

—Joseph Yehuda Halpern et al. “On the Unusual Effectiveness of Logic in Computer Science” (Halpern et al., 2001)

We systematically investigate the framework of modal symbolic learning. As we have discussed previously, decision trees are emblematic of the class of symbolic learning, and this is also the case for the modal symbolic ones. Natural extensions can be formulated based on modal decision trees. For the sake of simplicity, we restrict our attention to the case of binary decision trees (the natural choice for numerical attributes) for binary classification, both in the propositional and modal case. Nonetheless, it is important to stress that binary splits are purely arbitrary: more-than-binary splits could be performed on numerical attributes, and binary splits could be performed on categorical ones. Admitting more-than-binary splits also has consequences on the choices of the locally optimal splits, as observed by Quinlan (1986). Therefore, generalizing our approach to the case of general trees and multiple classes is immediate.

Binary decision trees, typical classifiers, are binary trees whose leaves and edges are labelled. Leaf-labels identify the different classes an instance can belong to, while edge-labels are logical atomic elements that are then composed to obtain complex formulas in the considered logical formalism (e.g., in \mathcal{PL} , edge-labels are literals and formulas are Boolean combinations). A tree associates a formula to every class it features (i.e., every label occurring in a leaf) and classifies an instance into a class if and only if the instance satisfies the formula corresponding to that class. As there can be exponentially many leaves in a tree, the classification process may require verifying an instance’s satisfaction against exponentially many formulas. However, decision trees provide an efficient mechanism for classifying an instance that does not explore the entire tree: for every node, starting from the root and going down towards the leaves, the truth of the formula associated with that node is checked against the instance to be classified and, depending on the outcome, the instance is passed to the right or the left child, and the process is repeated. When a leaf is reached, the instance is classified into the class that labels that leaf. Summing up, the desired properties for a family of decision trees include:

- *correctness*, that is, every tree classifies any given instance into precisely one class,

- *completeness*, that is, for every formula $\varphi \in \Phi(\mathcal{L})$ in the considered formalism \mathcal{L} , there is a decision tree t of the supposed family of decision trees that realizes φ , and
- *classification efficiency*, that is, a decision tree t of height h must be able to classify an instance by checking the truth of, at most, a number of formulas polynomial in h .

§3.1 Propositional Decision Trees

We first introduce some general concepts for trees. Let $t = (\mathcal{V}, \mathcal{E})$ be a *full directed binary tree* with *nodes* in \mathcal{V} and *edges* in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We denote by $\text{root}(t)$ the *root* of t , by $\mathcal{V}^\ell \subseteq \mathcal{V}$ the set of its *leaf nodes* (or, simply, *leaves*), and by $\mathcal{V}^i = \mathcal{V} \setminus \mathcal{V}^\ell$ the set of its *internal nodes* (i.e., non-root and non-leaf nodes). We also denote nodes (either root, internal, or leaf) by $v, v', \dots, v_1, v_2, \dots$ and leaves by $\ell, \ell', \dots, \ell_1, \ell_2, \dots$. Each non-leaf node v of a tree t has a *left child* $\varepsilon(v)$ and a *right child* $\varepsilon_r(v)$, and each non-root node v has a *parent* $\varepsilon(v)$. For a node v , the set of its *ancestors* (v included) is denoted by $\varepsilon^*(v)$, where ε^* is the transitive and reflexive closure of ε ; we also define $\varepsilon^+(v) = \varepsilon^*(v) \setminus \{v\}$. For every $y \in \mathbb{Y}$, where \mathbb{Y} is the label space, we denote by $\text{leaves}^t(y)$ the set of *leaves* of t labelled with y . A *path* $\pi^t = v_0 \rightsquigarrow v_h$ in t $h \geq 0$ between two nodes v_0 and v_h is a finite sequence of $h + 1$ nodes such that $v_i = \varepsilon(v_{i+1})$, for each $i = 0, \dots, h - 1$. We denote by $\pi_1 \cdot \pi_2$ the operation of *appending* the path π_2 to path π_1 . We also say that a path $v_0 \cdot v_1 \rightsquigarrow v_h$ is *left* (resp., *right*) if $v_1 = \varepsilon(v_0)$ (resp., $v_1 = \varepsilon_r(v_0)$). For a path π^t and for a node v in t , π_v^t denotes the unique path $\text{root}(t) \rightsquigarrow v$. Moreover, for a path π^t , the set of its *improper prefixes* is denoted by $\text{prefix}(\pi^t)$. Finally, a *branch* of t is a path π_ℓ^t , for some $\ell \in \mathcal{V}^\ell$. We omit the superscript notation, \cdot^t , if t is clear from the context.

Propositional decision trees are defined for structured datasets, which are the classic type of datasets to which one is used, typically presented in tabular form, where each row corresponds to an instance.

Definition 3.1: Structured datasets

Let \mathbb{X} be the sample space and \mathbb{Y} the label space. Then, a *structured dataset* is a set $\mathcal{X} = \{x_1, \dots, x_m\}$ of m *instances*, where $x_i \in \mathbb{X}$, for all $1 \leq i \leq m$, defined over a vector space $\mathcal{A} = \{A_1, \dots, A_n\}$ whose n dimensions are called *attributes*. The dataset \mathcal{X} is called *labelled* if each instance is labelled with an element from \mathbb{Y} , that is, $\mathcal{X} = \{(x_1, y_1), \dots, (x_m, y_m)\}$, where $y_i \in \mathbb{Y}$, for all $1 \leq i \leq m$. A *label function* $Y : \mathbb{X} \rightarrow \mathbb{Y}$ is a function that associates each labelled instance to its true label.

Without loss of generality, we assume that all attributes are numerical because the vast majority of real-world datasets are indeed numerical. For each instance x , we denote by $x[i]$ the value of x in the i th component, that is, the value of x associated to A_i . Let $\text{dom}(A)$ be the *domain* of the attribute A , namely, the set of values that A has in dataset \mathcal{X} .

	A_1	A_2	Y
x_1	10	3.9	0
x_2	7	3.5	1
x_3	6	1.3	0
x_4	5	2.7	1

FIGURE 3.1: Example of a structured dataset.

Example 1 (Structured dataset). *Figure 3.1 illustrates an example of labelled structured dataset, where $\mathcal{X} = \{(x_1, 0), (x_2, 1), (x_3, 0), (x_4, 1)\}$, $\mathcal{A} = \{A_1, A_2\}$, $\text{dom}(A_1) = \{10, 7, 6, 5\}$, $\text{dom}(A_2) = \{3.9, 3.5, 1.3, 2.7\}$, $\mathbb{Y} = \{0, 1\}$, $x_4[1] = 5$, $x_1[2] = 3.9$, $Y(x_1) = Y(x_3) = 0$, and $Y(x_2) = Y(x_4) = 1$.*

A structured dataset induces a set of propositional letters.

Definition 3.2: Induced propositional letters from structured datasets

Let \mathcal{X} be a structured dataset defined on attributes $\mathcal{A} = \{A_1, \dots, A_n\}$. Then, the set of *induced propositional letters* \mathcal{P} from \mathcal{X} is defined as:

$$\mathcal{P} = \{A_i \bowtie a \mid A_i \in \mathcal{A}, \bowtie \in \{<, \leq, =, \neq, \geq, >\}, a \in \text{dom}(A)\}.$$

Observe that each instance of a structured dataset is, in fact, a propositional model; indeed, in a (derived) propositional model $\mathfrak{K} = (V)$, we have that $V(A_i \bowtie a) = \top$ if and only if $x[i] \bowtie a$, where x is the original instance of the structured dataset. A set of induced propositional letters \mathcal{P} could be closed under negation or not, that is, for all $p \in \mathcal{P}$ there is $\neg p \in \mathcal{P}$, and vice versa; the domains $\text{dom}(A)$ could be too huge, and it should be better to consider smaller ones also to avoid overfitting (since the resulting symbolic model would fit too well in training on specific constants $a \in \text{dom}(A)$, but such constants may not be seen in testing instances); \mathcal{P} may be defined differently depending on the application-domain. What is more, a smaller set of induced propositional letters reduces the required time to learn (i.e., the search space is smaller).

Example 2 (Induced propositional decisions from a structured dataset). *Consider the labelled structured dataset in Figure 3.1. Then, if $\bowtie \in \{<, \geq\}$, the induced propositional letters are:*

$$A_1 < 10, \quad A_1 < 7, \quad A_1 < 6, \quad A_1 < 5, \quad A_2 < 3.9, \quad A_2 < 3.5, \quad A_2 < 1.3, \quad A_2 < 2.7, \\ A_1 \geq 10, \quad A_1 \geq 7, \quad A_1 \geq 6, \quad A_1 \geq 5, \quad A_2 \geq 3.9, \quad A_2 \geq 3.5, \quad A_2 \geq 1.3, \quad A_2 \geq 2.7.$$

We are ready to define propositional decision trees.

Definition 3.3: Propositional decision trees

Let $\mathcal{X} = \{(x_1, y_1), \dots, (x_m, y_m)\}$ be a labelled structured dataset defined over a vector space $\mathcal{A} = \{A_1, \dots, A_n\}$, \mathbb{Y} the label space, and \mathcal{P} the set of induced propositional letters induced from \mathcal{X} . Then, a *propositional decision tree* t is a structure defined as:

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

where:

- $(\mathcal{V}, \mathcal{E})$ is a full directed binary tree,
- $l : \mathcal{V}^\ell \rightarrow \mathbb{Y}$ is a *leaf-labelling function* that assigns a label from \mathbb{Y} to each leaf node in \mathcal{V}^ℓ ,
- $e : \mathcal{E} \rightarrow \mathcal{P}$ is a *edge-labelling function* that assigns a propositional letter from \mathcal{P} to each edge in \mathcal{E} ,

and the following conditions hold:

1. $e(v, v') = \neg e(v, v'')$, for all $(v, v'), (v, v'') \in \mathcal{E}$,
2. $l(\ell) \neq l(\ell')$, for all $\ell, \ell' \in \mathcal{V}^\ell$ such that $\mathfrak{z}(\ell) = \mathfrak{z}(\ell')$.

Condition 1 says that the edge-labels of each pair of outgoing edges, (v, v') and (v, v'') , from the same node, v , must be one the (logical) negation of the other, $e(v, v') = \neg e(v, v'')$. Condition 2 says that the leaf-labels of each pair of leaf nodes, ℓ and ℓ' , that share the same parent, $\mathfrak{z}(\ell) = \mathfrak{z}(\ell')$, node must be different, $l(\ell) \neq l(\ell')$.

Example 3 (Propositional decision tree). Let $t = (\mathcal{V}, \mathcal{E}, l, e)$ be the propositional decision tree in Figure 3.2. Then, we have that:

$$\begin{aligned} \mathcal{V} &= \{\text{root}(t), v_1, \ell_1, \ell_2, \ell_3\}, \\ \mathcal{E} &= \{(\text{root}(t), v_1), (\text{root}(t), \ell_3), (v_1, \ell_1), (v_1, \ell_2)\}, \\ l &= \{\ell_1 \mapsto y_2, \ell_2 \mapsto y_1, \ell_3 \mapsto y_1\}, \\ e &= \{(\text{root}(t), v_1) \mapsto p_1, (\text{root}(t), \ell_3) \mapsto \neg p_1, (v_1, \ell_1) \mapsto p_2, (v_1, \ell_2) \mapsto \neg p_2\}. \end{aligned}$$

Moreover, we have that:

$$\begin{aligned} \text{leaves}^t(y_1) &= \{\ell_2, \ell_3\}, \\ \text{leaves}^t(y_2) &= \{\ell_1\}. \end{aligned}$$

The following definition defines path-, leaf-, and class-formulas of propositional decision trees.

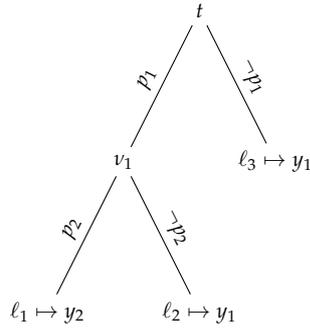


FIGURE 3.2: Example of a propositional decision tree.

Definition 3.4: Path-, leaf-, and class-formulas of propositional decision trees

Let $t = (\mathcal{V}, \mathcal{E}, l, e)$ be a propositional decision tree and \mathbb{Y} the label space. Then, for each path $\pi^t = v_0 \rightsquigarrow v_h$ in t , the *path-formula* φ_{π}^t is defined inductively as:

$$\varphi_{\pi}^t = \begin{cases} \top & \text{if } h = 0; \\ e(v_0, v_1) \wedge \varphi_{v_1 \rightsquigarrow v_h}^t & \text{if } h > 0. \end{cases}$$

Moreover, for each leaf $\ell \in \mathcal{V}^l$, the *leaf-formula* φ_{ℓ}^t is defined as:

$$\varphi_{\ell}^t = \varphi_{\pi_{\ell}}^t.$$

Finally, for each class $y \in \mathbb{Y}$, the *class-formula* φ_y^t is defined as:

$$\varphi_y^t = \bigvee_{\ell \in \text{leaves}^t(y)} \varphi_{\ell}^t.$$

Path- and leaf-formulas are conjunctions of propositional letters, and class-formulas are disjunctions of conjunctions. Therefore, class-formulas are formulas of \mathcal{PL} in *disjunctive normal form*; this means that, theoretically, each formula $\varphi \in \Phi(\mathcal{PL})$ can be expressed by a propositional decision tree. We omit the superscript notation, \cdot^t , if t is clear from the context.

Example 4 (Path-, leaf-, and class-formulas of propositional decision tree). Consider the propositional decision tree $t = (\mathcal{V}, \mathcal{E}, l, e)$ in Figure 3.2. Then, we have that:

$$\begin{aligned} \varphi_{\text{root}(v) \rightsquigarrow v_1}^t &= p_1, \\ \varphi_{\text{root}(v) \rightsquigarrow l_1}^t &= p_1 \wedge p_2, \\ \varphi_{\text{root}(v) \rightsquigarrow l_2}^t &= p_1 \wedge \neg p_2, \\ \varphi_{\text{root}(v) \rightsquigarrow l_3}^t &= \neg p_1, \\ \varphi_{l_1}^t &= \varphi_{\text{root}(v) \rightsquigarrow l_1}, \\ \varphi_{l_2}^t &= \varphi_{\text{root}(v) \rightsquigarrow l_2}, \\ \varphi_{l_3}^t &= \varphi_{\text{root}(v) \rightsquigarrow l_3}, \\ \varphi_{y_1}^t &= \varphi_{l_2} \vee \varphi_{l_3}, \\ \varphi_{y_2}^t &= \varphi_{l_1}. \end{aligned}$$

Finally, an instance is classified by a propositional decision tree as follows.

Definition 3.5: Run of propositional decision trees

Let $t = (\mathcal{V}, \mathcal{E}, l, e)$ be a propositional decision tree, v a node in t , and x an instance of a structured dataset \mathcal{X} interpreted as a propositional model. Then, the *run of t on x from v* is defined as:

$$t(x, v) = \begin{cases} l(v) & \text{if } v \in \mathcal{V}^\ell; \\ t(x, \varepsilon^r(v)) & \text{if } x \models \varphi_{\pi^t_{\varepsilon^r(v)}}; \\ t(x, \varepsilon^{\neg}(v)) & \text{if } x \models \varphi_{\pi^t_{\varepsilon^{\neg}(v)}}. \end{cases}$$

Finally, the *run of t on x* , denoted by $t(x)$, is defined as $t(x, \text{root}(t))$.

Example 5 (Run of propositional decision trees). Consider the propositional decision tree $t = (\mathcal{V}, \mathcal{E}, l, e)$ in Figure 3.2, and the labelled structured dataset \mathcal{X} in Figure 3.1. Then, we have that:

$$\begin{aligned} t(x_4, \text{root}(t)) &= t(x_4, v_1) \quad (x_4 \models p_1) \\ &= l(\ell_1) \quad (x_4 \models p_1 \wedge p_2) \\ &= y_2, \end{aligned}$$

where $p_1 \triangleq A_1 \leq 7$ and $p_2 \triangleq A_2 > 1.3$.

Correctness, completeness, and classification efficiency could be discussed at this point. However, they are well-known (although they may not have been completely formalized in the literature); moreover, modal decision trees include propositional decision trees as a particular case, and the properties, which we shall discuss in the modal case, transfer from the latter to the propositional case.

§3.2 Modal Decision Trees

We now introduce the notion of modal dataset which is central to the entire modal symbolic learning framework.

Definition 3.6: Modal dataset

Let \mathbb{K} be the Kripke model space, \mathbb{Y} the label space, and \mathcal{P} a set of proposition letters. Then, a *modal dataset* $\mathcal{I} = \{\mathcal{J}_1, \dots, \mathcal{J}_m\}$ is a set of m instances, where $\mathcal{J}_i \in \mathbb{K}$, for all $1 \leq i \leq m$, defined over \mathcal{P} . The dataset \mathcal{I} is called *labelled* if each instance is labelled with an element from \mathbb{Y} , that is, $\mathcal{I} = \{(\mathcal{J}_1, y_1), \dots, (\mathcal{J}_m, y_m)\}$, where $y_i \in \mathbb{Y}$, for all $1 \leq i \leq m$. A *label function* $Y : \mathbb{K} \rightarrow \mathbb{Y}$ is a function that associates each labelled instance to its true label.

If each Kripke model \mathcal{J} in a modal dataset is a (trivial) model with a single world (and thus, no accessibility relation), namely, $\mathcal{J} = \langle \{w\}, \emptyset, V \rangle$, then such dataset is called

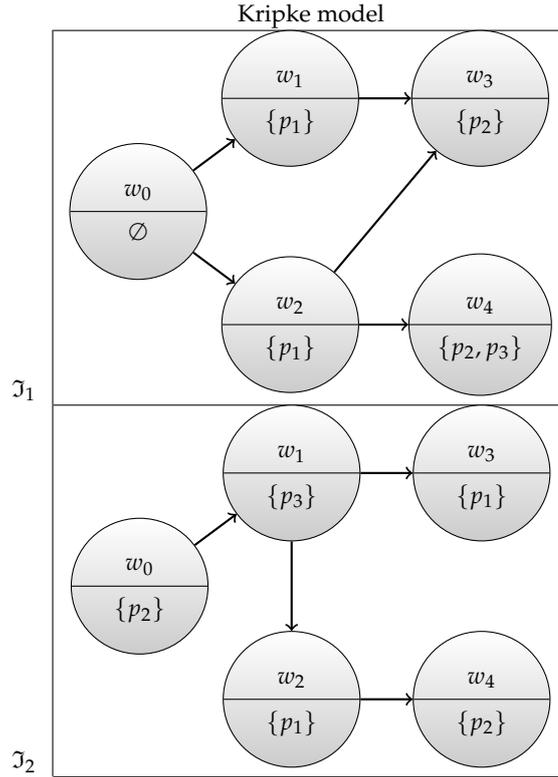


FIGURE 3.3: Example of a modal dataset.

propositional. It is immediate to observe that structured datasets can be logically viewed as propositional datasets, and \mathcal{PL} learning algorithms can be applied to the latter (e.g., propositional decision trees).

Example 6 (Modal dataset). Consider the modal dataset in Figure 3.3. In this example, $\mathcal{I} = \{\mathcal{J}_1, \mathcal{J}_2\}$, where each instance is a Kripke model, and $\mathcal{P} = \{p_1, p_2, p_3\}$.

As we have discussed, decision trees recursively split samples, and this is also the case of modal decision trees that split (Kripke-like) instances based on the following decisions.

Definition 3.7: Modal decisions

Let \mathcal{P} be a set of propositional letters. Then, the set of *modal split-decisions* Λ (or, for brevity, *decisions*) is defined as:

$$\Lambda = \{\top, \perp, p, \neg p, \diamond\top, \square\perp \mid p \in \mathcal{P}\}.$$

We say that p and $\neg p$ are *propositional* decisions, while $\diamond\top$ (resp., $\square\perp$) are *modal existential* (resp., *modal universal*) ones. For a decision $\lambda \in \Lambda$, the decision that corresponds to its logical negation $\neg\lambda$ is univocally identified; thus, when $\lambda = \top$ (resp., $p, \diamond\top$), we use $\neg\lambda$ to denote \perp (resp., $\neg p, \square\perp$), and vice versa. Moreover, $\diamond\top$ and

$\Box\perp$ may seem unusual, but it allows us, combined with the other connectives, to obtain all the formulas with respect to \mathcal{ML} (we prove such a result in the next section).

Modal decision trees are defined as follows.

Definition 3.8: Modal decision trees

Let $\mathcal{I} = \{(\mathcal{J}_1, y_1), \dots, (\mathcal{J}_m, y_m)\}$ be a labelled modal dataset, \mathbb{Y} the label space, and Λ a set of decisions. Then, a *modal decision tree* is a structure:

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

where:

- $(\mathcal{V}, \mathcal{E})$ is a full directed binary tree,
- $l : \mathcal{V}^\ell \rightarrow \mathbb{Y}$ is a *leaf-labelling function* that assigns a label from \mathbb{Y} to each leaf node in \mathcal{V}^ℓ ,
- $e : \mathcal{E} \rightarrow \Lambda$ is a *edge-labelling function* that assigns a decision from Λ to each edge in \mathcal{E} ,
- $b : \mathcal{V}^i \rightarrow \mathcal{V}^i$ is a *back-edge function* that links an internal node to one of its ancestors,

and the following conditions hold:

1. $e(v, v') = \neg e(v, v'')$, for all $(v, v'), (v, v'') \in \mathcal{E}$,
2. $l(\ell) \neq l(\ell')$, for all $\ell, \ell' \in \mathcal{V}^\ell$ such that $\mathfrak{z}(\ell) = \mathfrak{z}(\ell')$,
3. if $b(v) = v'$, then $v' \in \mathfrak{z}^*(v)$, for all $v, v' \in \mathcal{V}^i$,
4. if $b(v) \neq v$ and $b(v') \neq v'$, then $b(v) \neq b(v')$, for all $v, v' \in \mathcal{V}^i$,
5. if $b(v) = v', v' \in \mathfrak{z}^+(v'')$, and $v'' \in \mathfrak{z}^+(v)$, then $v' \in \mathfrak{z}^+(b(v''))$, for all $v, v', v'' \in \mathcal{V}^i$,
6. if $e(v, v') \in \{\perp, \Box\perp\}$ and $v' \notin \mathcal{V}^\ell$, then $b(v') \neq v'$, for all $(v, v') \in \mathcal{E}$.

A propositional decision tree is a modal decision tree in which edges are labelled with propositional decisions (instead of modal decisions), and the back-edge function plays no role; thus, propositional decision trees are a particular case of modal ones. Since any modal decision (sub)tree rooted in a node, v , is also a modal decision tree representing a collection of formulas (similar to the propositional case), the back-edge, $b(v)$, allows a modal decision tree to add (sub)formulas (those related to the tree rooted in v) at any depth of the syntax tree of the formulas that are to be built/learned. As we shall see, adding the back-edges allows us to build (weakly) complete modal decision trees. Conditions 1 and 2 are as in the case of propositional decision trees. Condition 3 says that if v' is the back-edge link of v , then v' must be an ancestor of v . Condition 4 says that if the back-edge links of v and v' are not self-loops, then such links are not equal. Condition 5 says that if v' is the back-edge link of v , v'' is an ancestor of v (excluding v) and v' is an ancestor of v'' (excluding v''), then v' must be an ancestor of $b(v'')$ (excluding $b(v'')$), that is, $b(v'')$ must be a node on the path $v' \rightsquigarrow v''$. Finally, condition 6 says that, for any edge (v, v') , if the

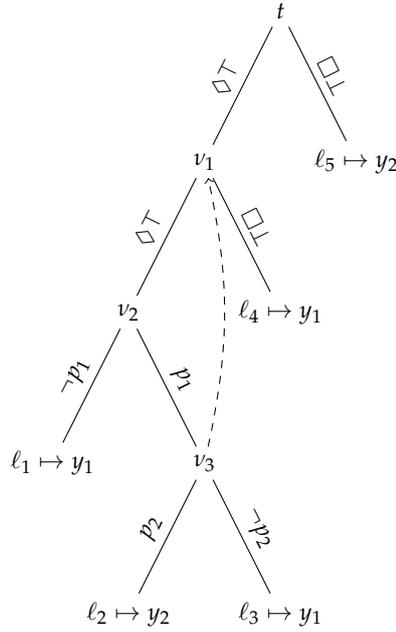


FIGURE 3.4: Example of a modal decision tree.

edge-label is either \perp or $\Box\perp$, where v' is not a leaf, then the back-edge link of v' cannot be v' itself.

Example 7 (Modal decision tree). Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be the modal decision tree in Figure 3.4. Then:

$$\begin{aligned}
 \mathcal{V} &= \{\text{root}(t), v_1, v_2, v_3, l_1, l_2, l_3, l_4, l_5\}, \\
 \mathcal{E} &= \{(\text{root}(t), v_1), (\text{root}(t), l_5), (v_1, v_2), (v_1, l_4), (v_2, l_1), (v_2, v_3), (v_3, l_2), (v_3, l_3)\}, \\
 l &= \{l_1 \mapsto y_1, l_2 \mapsto y_2, l_3 \mapsto y_1, l_4 \mapsto y_1, l_5 \mapsto y_2\}, \\
 e &= \{(\text{root}(t), v_1) \mapsto \Diamond\top, (\text{root}(t), l_5) \mapsto \Box\perp, (v_1, v_2) \mapsto \Diamond\top, (v_1, l_4) \mapsto \Box\perp, \\
 &\quad (v_2, l_1) \mapsto \neg p_1, (v_2, v_3) \mapsto p_1, (v_3, l_2) \mapsto p_2, (v_3, l_3) \mapsto \neg p_2\}, \\
 b &= \{v_3 \mapsto v_1\}.
 \end{aligned}$$

Moreover, we have that:

$$\begin{aligned}
 \text{leaves}^t(y_1) &= \{l_1, l_3, l_4\}, \\
 \text{leaves}^t(y_2) &= \{l_2, l_5\}.
 \end{aligned}$$

We now show how a modal decision tree defines a modal formula for each of its classes. \mathcal{ML} does not have a normal form that allows one to bound the nesting of modal operators, and this makes the construction of formulas more complicated. Let us fix the following concepts.

Definition 3.9: Contributor

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree and $\pi^t = v_0 \rightsquigarrow v_h$ a path in t , with $h > 1$. Then, the *contributor* of π^t , denoted by $\text{ctr}(\pi^t)$, is defined as the only node $v_i \in \pi^t$ such that $v_i \neq v_1$, with $0 < i < h$, and $b(v_i) = v_1$, if it exists, and

v_1 , otherwise.

Example 8 (Contributor). Consider the modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ in Figure 3.4. Let $\pi_1^t = \text{root}(t) \rightsquigarrow \ell_3$ and $\pi_2^t = \text{root}(t) \rightsquigarrow \ell_1$. Then, we have that:

$$\begin{aligned} \text{ctr}(\pi_1^t) &= v_3, \\ \text{ctr}(\pi_2^t) &= v_1. \end{aligned}$$

Definition 3.10: Node agreement

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree and $\pi^t = v_0 \rightsquigarrow v_h$ a path in t , with $h > 1$. Then, given two nodes $v_i, v_j \in \pi^t$, with $i, j < h$, we say that they *agree*, denoted by $A(v_i, v_j)$, if $v_{i+1} = \varepsilon^r(v_i)$ (resp., $v_{i+1} = \varepsilon^s(v_i)$) and $v_{j+1} = \varepsilon^r(v_j)$ (resp., $v_{j+1} = \varepsilon^s(v_j)$); otherwise, we say that they *disagree*, denoted by $D(v_i, v_j)$.

Example 9 (Node agreement). Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be the modal decision tree in Figure 3.4, and consider the path $\pi^t = \text{root}(t) \rightsquigarrow \ell_2$. Then, we have that $A(v_1, v_3)$ and $D(v_2, v_3)$.

Definition 3.11: Implicative formulas

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*.

Thanks to the above definitions, we are ready to define path-, leaf-, and class-formulas of modal decision trees.

Definition 3.12: Path-, leaf-, and class-formulas of modal decision trees

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree and \mathbb{Y} the label space. Then, for each path $\pi^t = v_0 \rightsquigarrow v_h$ in t , the *path-formula* φ_π^t is defined inductively as:

- if $h = 0$, then $\varphi_\pi^t = \top$;
- if $h = 1$, then $\varphi_\pi^t = e(v_0, v_1)$;

- if $h > 1$, let $\lambda = e(v_0, v_1)$, $\pi_1^t = v_1 \rightsquigarrow \text{ctr}(\pi^t)$, and $\pi_2^t = \text{ctr}(\pi^t) \rightsquigarrow v_h$, then

$$\varphi_{\pi}^t = \begin{cases} \lambda \wedge (\varphi_{\pi_1}^t \wedge \varphi_{\pi_2}^t) & \text{if } \lambda \neq \diamond\top, A(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \notin \text{Im}, \\ & \text{or } \lambda \neq \diamond\top, D(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \in \text{Im}; \\ \lambda \rightarrow (\varphi_{\pi_1}^t \rightarrow \varphi_{\pi_2}^t) & \text{if } \lambda \neq \diamond\top, D(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \notin \text{Im}, \\ & \text{or } \lambda \neq \diamond\top, A(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \in \text{Im}; \\ \diamond(\varphi_{\pi_1}^t \wedge \varphi_{\pi_2}^t) & \text{if } \lambda = \diamond\top, A(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \notin \text{Im}, \\ & \text{or } \lambda = \diamond\top, D(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \in \text{Im}; \\ \square(\varphi_{\pi_1}^t \rightarrow \varphi_{\pi_2}^t) & \text{if } \lambda = \diamond\top, D(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \notin \text{Im}, \\ & \text{or } \lambda = \diamond\top, A(v_0, \text{ctr}(\pi^t)), \text{ and } \varphi_{\pi_2} \in \text{Im}. \end{cases}$$

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

$$\varphi_\ell^t = \bigwedge_{\pi \in \text{prefix}(\pi_\ell^t)} \varphi_\pi^t.$$

Finally, for each class $y \in \mathbb{Y}$, the *class-formula* φ_y^t is defined as:

$$\varphi_y^t = \bigvee_{\ell \in \text{leaves}^t(y)} \varphi_\ell^t.$$

Again, we omit the superscript notation, \cdot^t , if t is clear from the context.

Example 10 (Path-, leaf-, and class-formula of modal decision trees). *Consider the modal decision tree in Figure 3.4. Then, we have that:*

$$\begin{aligned} \varphi_{\{v\} \rightsquigarrow v}^t &= e(\{v\}, v), \text{ for all } v \in \mathcal{V} \setminus \{\text{root}(t)\}, \\ \varphi_{v_1 \rightsquigarrow \ell_1}^t &= \diamond(\top \wedge \neg p_1), \\ \varphi_{\text{root}(t) \rightsquigarrow v_2}^t &= \diamond(\top \wedge \diamond\top), \\ \varphi_{\text{root}(t) \rightsquigarrow \ell_1}^t &= \diamond(\top \wedge \diamond(\top \wedge \neg p_1)), \\ \varphi_{v_1 \rightsquigarrow v_3}^t &= \square(\top \rightarrow p_1), \\ \varphi_{\text{root}(t) \rightsquigarrow v_3}^t &= \square(\top \rightarrow \square(\top \rightarrow p_1)), \\ \varphi_{\text{root}(t) \rightsquigarrow \ell_2}^t &= \diamond(\square(\top \rightarrow p_1) \wedge p_2), \\ \varphi_{\text{root}(t) \rightsquigarrow \ell_3}^t &= \square(\square(\top \rightarrow p_1) \rightarrow \neg p_2), \\ \varphi_{\text{root}(t) \rightsquigarrow \ell_4}^t &= \square(\top \rightarrow \square\perp), \\ \varphi_\ell^t &= \bigwedge_{\pi \in \text{prefix}(\pi_\ell^t)} \varphi_\pi^t, \\ \varphi_{y_1}^t &= \varphi_{\ell_1}^t \vee \varphi_{\ell_3}^t \vee \varphi_{\ell_4}^t, \\ \varphi_{y_2}^t &= \varphi_{\ell_2}^t \vee \varphi_{\ell_5}^t. \end{aligned}$$

Similarly to the propositional case, an instance is classified by a modal decision tree as follows.

Definition 3.13: Run of modal decision trees

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree, v a node in t , and \mathcal{J} an instance of a modal dataset \mathcal{I} . Then, the *run of t on \mathcal{J} from v* is defined as:

$$t(\mathcal{J}, v) = \begin{cases} l(v) & \text{if } v \in \mathcal{V}^\ell; \\ t(\mathcal{J}, \varrho^r(v)) & \text{if } \mathcal{J} \Vdash \varphi_{\pi_{\varrho^r(v)}^t}; \\ t(\mathcal{J}, \varrho^s(v)) & \text{if } \mathcal{J} \Vdash \varphi_{\pi_{\varrho^s(v)}^t}. \end{cases}$$

Finally, the *run of t on \mathcal{J}* , denoted by $t(\mathcal{J})$, is defined as $t(\mathcal{J}, \text{root}(t))$.

Following the above definition, a modal decision tree classifies an instance using its class-formulas, and does so by checking, progressively, the path-formulas that contribute to building a leaf-formula, which, in turn, is one of the disjuncts that take part in a class-formula. Observe that, inter alia, this implies that propositional decision trees can be seen as particular cases of modal decision trees even from a semantic point of view: formulas of the type $\varphi_1 \wedge \varphi_2$ behave exactly as in the propositional case, while those of the type $\varphi_1 \rightarrow \varphi_2$, are such that their antecedent is always included as a conjunct in their corresponding leaf-formula, effectively reducing it to a conjunction, as in the propositional case.

Example 11 (Run of modal decision trees). Consider the modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ in Figure 3.4, and the modal dataset in Figure 3.3. Then, we have that:

$$\begin{aligned} t(\mathcal{J}_2, \text{root}(t)) &= t(\mathcal{J}_2, v_1) \quad (\mathcal{J}_2 \Vdash \diamond \top) \\ &= t(\mathcal{J}_2, v_2) \quad (\mathcal{J}_2 \Vdash \diamond(\top \wedge \diamond \top)) \\ &= t(\mathcal{J}_2, v_3) \quad (\mathcal{J}_2 \Vdash \square(\top \rightarrow \square(\top \rightarrow p_1))) \\ &= l(\ell_3) \quad (\mathcal{J}_2 \Vdash \square(\square(\top \rightarrow p_1) \rightarrow \neg p_2)) \\ &= y_1. \end{aligned}$$

§3.3 Properties of Modal Decision Trees

As we have stated at the beginning of the chapter, the desiderata of decision trees are correctness, completeness, and classification efficiency. While such properties at the propositional level may seem trivial, in the case of modal decision trees they are not, and we must formally define each property.

We start discussing correctness of modal decision trees.

Definition 3.14: Correctness

A decision tree t is *correct* if and only if, for every dataset \mathcal{I} and every instance $\mathcal{J} \in \mathcal{I}$, it is the case that \mathcal{J} satisfies exactly one of its class-formulas. A class of decision trees is *correct* if and only if all of its decision trees are correct.

Lemma 3.1

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree, and let $\pi_1 = v_0 \rightsquigarrow v_{h-1} \cdot \varepsilon^r(v_{h-1})$ and $\pi_2 = v_0 \rightsquigarrow v_{h-1} \cdot \varepsilon^l(v_{h-1})$ be two paths in t . Then, $\varphi_{\pi_1} \leftrightarrow \neg\varphi_{\pi_2}$ is valid.

PROOF. We prove a stronger statement, that is, we prove that for each pair of paths $\pi_1 = v_0 \rightsquigarrow v_{h-1} \cdot \varepsilon^r(v_{h-1})$ and $\pi_2 = v_0 \rightsquigarrow v_{h-1} \cdot \varepsilon^l(v_{h-1})$ it holds that:

1. $\varphi_{\pi_1} \leftrightarrow \neg\varphi_{\pi_2}$ is valid, and
2. if $h \geq 2$, then $\varphi_{\pi_1}^t \in Im$ and $\varphi_{\pi_2}^t \notin Im$, or the other way around.

We proceed by induction on h :

- $\boxed{h = 1}$ In this case, $\pi = v_0, \pi_1 = v_0 \cdot \varepsilon^r(v_0)$, and $\pi_2 = v_0 \cdot \varepsilon^l(v_0)$. By definition, it must be the case that $e(v_0, \varepsilon^r(v_0)) = \lambda$ and $e(v_0, \varepsilon^l(v_0)) = \neg\lambda$, for some decision λ . By construction, therefore, $\varphi_{\pi_1} \leftrightarrow \neg\varphi_{\pi_2}$ is valid, as wanted.
- $\boxed{h = 2}$ In this case, $\pi = v_0 \rightsquigarrow v_1, \pi_1 = \pi \cdot \varepsilon^r(v_1)$, and $\pi_2 = \pi \cdot \varepsilon^l(v_1)$. It holds that $\varphi_{v_1 \rightsquigarrow \varepsilon^r(v_1)}, \varphi_{v_1 \rightsquigarrow \varepsilon^l(v_1)} \notin Im$. There are four cases to consider depending on the relationship between v_0 and v_1 that can be left or right, and on the value of $e(v_0, v_1)$ that can be equal to or not to $\diamond\top$. Let $v_1 = \varepsilon^r(v_0)$ and $e(v_0, v_1) = \lambda \neq \diamond\top$; the other cases are similar. Observe that $ctr(\pi_1) = ctr(\pi_2) = v_1$. Since $A(v_0, ctr(\pi_1))$, following Definition 3.12, $\varphi_{\pi_1} = \lambda \wedge (\varphi_{v_1 \rightsquigarrow v_1} \wedge \varphi_{v_1 \rightsquigarrow \varepsilon^r(v_1)})$. Since $D(v_0, ctr(\pi_2))$, following Definition 3.12, $\varphi_{\pi_2} = \lambda \rightarrow (\varphi_{v_1 \rightsquigarrow v_1} \rightarrow \varphi_{v_1 \rightsquigarrow \varepsilon^l(v_1)})$. Therefore, $\varphi_{\pi_1} \notin Im$, while $\varphi_{\pi_2} \in Im$, and since $\varphi_{v_1 \rightsquigarrow \varepsilon^r(v_1)} \leftrightarrow \neg\varphi_{v_1 \rightsquigarrow \varepsilon^l(v_1)}$ is valid by inductive hypothesis, $\varphi_{\pi_1} \leftrightarrow \neg\varphi_{\pi_2}$ must be valid, as we wanted.
- $\boxed{h > 2}$ In this case, $\pi = v_0 \rightsquigarrow v_{h-1}, \pi_1 = \pi \cdot \varepsilon^r(v_{h-1})$, and $\pi_2 = \pi \cdot \varepsilon^l(v_{h-1})$. Observe that $ctr(\pi_1) = ctr(\pi_2) = v_j$, for some $j \leq h-1$, and consider the paths $v_j \rightsquigarrow \varepsilon^r(v_{h-1})$ and $v_j \rightsquigarrow \varepsilon^l(v_{h-1})$. By inductive hypothesis, $\varphi_{v_j \rightsquigarrow \varepsilon^r(v_{h-1})} \leftrightarrow \neg\varphi_{v_j \rightsquigarrow \varepsilon^l(v_{h-1})}$ is valid. If $j = h-1$, then there are four cases to consider depending on the relationship between v_0 and v_1 that can be left or right, and on the value of $e(v_0, v_1)$ that can be equal to or not to $\diamond\top$. Let $v_1 = \varepsilon^r(v_0)$ and $e(v_0, v_1) = \lambda \neq \diamond\top$; the other cases are similar. Since $A(v_0, ctr(\pi_1))$, following Definition 3.12, $\varphi_{\pi_1} = \lambda \wedge (\varphi_{v_1 \rightsquigarrow v_j} \wedge \varphi_{v_j \rightsquigarrow \varepsilon^r(v_{h-1})})$. Since $D(v_0, ctr(\pi_2))$, following Definition 3.12, $\varphi_{\pi_2} = \lambda \rightarrow (\varphi_{v_1 \rightsquigarrow v_j} \rightarrow \varphi_{v_j \rightsquigarrow \varepsilon^l(v_{h-1})})$. Therefore, $\varphi_{\pi_1} \notin Im$, while $\varphi_{\pi_2} \in Im$, and since $\varphi_{v_j \rightsquigarrow \varepsilon^r(v_{h-1})} \leftrightarrow \neg\varphi_{v_j \rightsquigarrow \varepsilon^l(v_{h-1})}$ is valid by inductive hypothesis, $\varphi_{\pi_1} \leftrightarrow \neg\varphi_{\pi_2}$ must be valid, as we wanted. If, on the other hand, $j < h-1$, then we need to consider eight cases depending on the relationship between v_0 and v_1 that can be left or right, the value of $e(v_0, v_1)$ that can be equal or not to $\diamond\top$, and on the relationship between v_j and v_{j+1} that can be left or right. Let $v_1 = \varepsilon^r(v_0), e(v_0, v_1) = \lambda \neq \diamond\top$, and $v_{j+1} = \varepsilon^r(v_j)$; as before, the other cases are similar. By inductive hypothesis, either $\varphi_{v_j \rightsquigarrow \varepsilon^r(v_{h-1})} \notin Im$ and $\varphi_{v_j \rightsquigarrow \varepsilon^l(v_{h-1})} \in Im$, or the other way around; without loss of generality, assume the former. Since $A(v_0, ctr(\pi_1))$, following Definition 3.12, $\varphi_{\pi_1} = \lambda \wedge (\varphi_{v_1 \rightsquigarrow v_j} \wedge \varphi_{v_j \rightsquigarrow \varepsilon^r(v_{h-1})})$. Since $A(v_0, ctr(\pi_2))$, following Definition 3.12, $\varphi_{\pi_2} = \lambda \rightarrow (\varphi_{v_1 \rightsquigarrow v_j} \rightarrow \varphi_{v_j \rightsquigarrow \varepsilon^l(v_{h-1})})$. Therefore, $\varphi_{\pi_1} \notin Im$, while $\varphi_{\pi_2} \in Im$, and since $\varphi_{v_j \rightsquigarrow \varepsilon^r(v_{h-1})} \leftrightarrow \neg\varphi_{v_j \rightsquigarrow \varepsilon^l(v_{h-1})}$ is valid by inductive hypothesis, $\varphi_{\pi_1} \leftrightarrow \neg\varphi_{\pi_2}$ must be valid, as we wanted.



We want to prove correctness of modal decision trees.

Theorem 3.1: Correctness of modal decision trees

Modal decision trees are correct.

PROOF. Consider a modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$, a modal dataset \mathcal{I} , and an instance $\mathfrak{J} \in \mathcal{I}$.

We want to prove the correctness of t with respect to \mathfrak{J} . We want to prove, first, that, for every pair of classes $y_1, y_2 \in \mathbb{Y}$, with $y_1 \neq y_2$, it is not the case that $\mathfrak{J} \Vdash \varphi_{y_1} \wedge \varphi_{y_2}$. Suppose, by way of contradiction, that this is the case. Therefore, for some $\ell_1, \ell_2 \in \mathcal{V}^\ell$ such that $\ell_1 \in \text{leaves}(y_1)$ and $\ell_2 \in \text{leaves}(y_2)$, it must be the case that $\mathfrak{J} \Vdash \varphi_{\ell_1} \wedge \varphi_{\ell_2}$. Consider the branches π_{ℓ_1} and π_{ℓ_2} . Let v_i be the lowest node common to π_{ℓ_1} and π_{ℓ_2} . Since v_i is not a leaf, let $v_{i+1} = \varepsilon^\ell(v_i)$ and $v'_{i+1} = \varepsilon^r(v_i)$. By Lemma 3.1, $\varphi_{\pi_{v_{i+1}}} \leftrightarrow \neg \varphi_{\pi_{v'_{i+1}}}$ is valid, but, by definition, $\varphi_{\ell_1} \rightarrow \varphi_{\pi_{v_{i+1}}}$ and $\varphi_{\ell_2} \rightarrow \varphi_{\pi_{v'_{i+1}}}$ are also valid, which leads to a contradiction.

Second, we want to prove that, for at least one class $y \in \mathbb{Y}$ it is the case that $\mathfrak{J} \Vdash \varphi_y$, which is equivalent to say that there exists a leaf $\ell \in \text{leaves}(y)$ such that $\mathfrak{J} \Vdash \varphi_\ell$. We prove the following stronger statement: for all $h' \leq h$, where h is the height of t , if there exists a path $\pi_{v_{h'}}$ such that $\mathfrak{J} \Vdash \varphi_\pi$, for every $\pi \in \text{prefix}(\pi_{v_{h'}})$, then $v_{h'}$ is a leaf, or $\mathfrak{J} \Vdash \varphi_{\pi_{v_{h'} \cdot \varepsilon^\ell(v_{h'})}}$, or $\mathfrak{J} \Vdash \varphi_{\pi_{v_{h'} \cdot \varepsilon^r(v_{h'})}}$. We proceed by induction:

- $h' = 0$ The result is immediate.
- $h' > 0$ Let $\pi_{v_{h'}} = v_0 \rightsquigarrow v_{h'}$ be the path identified by the inductive hypothesis. If $v_{h'}$ is a leaf, the result is immediate. Otherwise, observe that, by Lemma 3.1, $\varphi_{\pi_{v_{h'} \cdot \varepsilon^\ell(v_{h'})}} \leftrightarrow \neg \varphi_{\pi_{v_{h'} \cdot \varepsilon^r(v_{h'})}}$ is valid, which means that we can take $\pi_{v_{h'+1}} = \pi_{v_{h'} \cdot \varepsilon^\ell(v_{h'})}$ or $\pi_{v_{h'+1}} = \pi_{v_{h'} \cdot \varepsilon^r(v_{h'})}$, and we have the result.

By taking $h' = h$, this immediately leads to the conclusion that $\mathfrak{J} \Vdash \varphi_\ell$, for some $\ell \in \mathcal{V}^\ell$, that is, $\mathfrak{J} \Vdash \varphi_y$, for some class $y \in \mathbb{Y}$, such that $\ell \in \text{leaves}(y)$, as we wanted. ■

Corollary 3.1: Correctness of propositional decision trees

Propositional decision trees are correct.

We now discuss the completeness of modal decision trees with respect to \mathcal{ML} .

Definition 3.15: Completeness

A family of decision trees is *strongly complete* for a logical formalism if and only if, for each of its formula φ , there is a decision tree t and a class $y \in \mathbb{Y}$ such that $\varphi_y^t \leftrightarrow \varphi$ is valid. Moreover, a family of decision trees is *weakly complete* for a logical formalism if and only if, for each of its formula φ , there is a decision tree t and two classes $y, \bar{y} \in \mathbb{Y}$ such that $\varphi_y^t \rightarrow \varphi$ and $\varphi_{\bar{y}}^t \rightarrow \neg \varphi$ are both valid.

It is worth discussing how the above definition relates to the purpose of a decision tree model. From a practical point of view, (modal) decision trees are learned from labelled (modal) datasets via approximation algorithms which are incomplete by design; in other words, decision trees are never used as top-down model checkers for specific formulas. While decision trees from a strongly complete family guarantee the correctness of classification for a formula, those from weakly complete ones allow one to only partially identify the hypothetical formula that defines a class. However, the fundamental desideratum is expressing such a formula as a decision tree. As it turns out, both weakly and strongly complete families of decision trees can do so.

Lemma 3.2

Let $\varphi \in \Phi(\mathcal{ML})$. Then, there exists a modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ and two leaves $\ell_y, \ell_{\bar{y}} \in \mathcal{V}^\ell$ such that $\varphi_{\ell_y} \leftrightarrow \varphi$ and $\varphi_{\ell_{\bar{y}}} \leftrightarrow \neg\varphi$.

PROOF. For the purpose of this proof, let us fix $\mathbb{Y} = \{y, \bar{y}, y^*\}$. We prove a stronger statement, that is, given $\varphi \in \Phi(\mathcal{ML})$, there exists a modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ over \mathbb{Y} , and two leaves $\ell_y, \ell_{\bar{y}} \in \mathcal{V}^\ell$ such that $\ell_y \in \text{leaves}(y)$, $\ell_{\bar{y}} \in \text{leaves}(\bar{y})$, and that:

1. $\varphi_{\ell_y} \leftrightarrow \varphi$ is valid, and either $\varphi_{\pi_{\ell_y}} \notin \text{Im}$ and π_{ℓ_y} is right or $\varphi_{\pi_{\ell_y}} \in \text{Im}$ and π_{ℓ_y} is left, and
2. $\varphi_{\ell_{\bar{y}}} \leftrightarrow \neg\varphi$ is valid, and either $\varphi_{\pi_{\ell_{\bar{y}}}} \notin \text{Im}$ and $\pi_{\ell_{\bar{y}}}$ is left or $\varphi_{\pi_{\ell_{\bar{y}}}} \in \text{Im}$ and $\pi_{\ell_{\bar{y}}}$ is right.

We build, now, by induction on the complexity of φ , a modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ over \mathbb{Y} for which items 1 and 2 hold:

- $\boxed{\varphi = p}$ We build $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ as follows:

- $\mathcal{V} = \{v, v', v''\}$,
- $\mathcal{E} = \{(v, v'), (v, v'')\}$,
- $l = \{v' \mapsto y, v'' \mapsto \bar{y}\}$,
- $e = \{(v, v') \mapsto p, (v, v'') \mapsto \neg p\}$,
- $b = \emptyset$,

and we impose that $v' = \imath_y(v)$ and $v'' = \imath_{\bar{y}}(v)$. Clearly, $v' \in \text{leaves}(y)$, and since $\varphi_{v'} = p$, $\varphi_{v'} \leftrightarrow \varphi$ is valid, $\varphi_{\pi_{v'}} \notin \text{Im}$ and $\pi_{v'}$ is right, then item 1 holds for $\ell_y = v'$. Similarly, $v'' \in \text{leaves}(\bar{y})$, and since $\varphi_{v''} = \neg p$, $\varphi_{v''} \leftrightarrow \neg\varphi$ is valid, $\varphi_{\pi_{v''}} \notin \text{Im}$ and $\pi_{v''}$ is left, then item 2 holds for $\ell_{\bar{y}} = v''$.

- $\boxed{\varphi = \neg\varphi_1}$ By inductive hypothesis, there exists $t_1 = (\mathcal{V}_1, \mathcal{E}_1, l_1, e_1, b_1)$ such that item 1 holds for some leaf $\ell_y^1 \in \mathcal{V}_1$ and item 2 holds for some leaf $\ell_{\bar{y}}^1 \in \mathcal{V}_1$, with respect to φ_1 . Informally, we obtain t by producing the mirror image of t_1 , which consists of switching labels y with \bar{y} of ℓ_y^1 and $\ell_{\bar{y}}^1$, respectively, and swapping right and left children of every node. Formally, we build $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ as follows:

- $\mathcal{V} = \mathcal{V}_1$,
- $\mathcal{E} = \mathcal{E}_1$,

- $l = (l_1 \setminus \{\ell_y^1 \mapsto y, \ell_{\bar{y}}^1 \mapsto \bar{y}\}) \cup \{\ell_y^1 \mapsto \bar{y}, \ell_{\bar{y}}^1 \mapsto y\}$,
- $e = e_1$,
- $b = b_1$,

and we impose, for every triple of nodes $v, v', v'' \in \mathcal{V}$, that $v' = \varepsilon'(v)$ (resp., $v'' = \varepsilon_{\downarrow}(v)$) in t if and only if $v' = \varepsilon_{\downarrow}(v)$ (resp., $v'' = \varepsilon'(v)$) in t_1 . Let us set $\ell_{\bar{y}} = \ell_{\bar{y}}^1$. It is immediate to see that $\pi_{\ell_{\bar{y}}}^t$ is right (resp., left) if and only if $\pi_{\ell_{\bar{y}}}^{t_1}$ is left (resp., right), and that $\varphi_{\pi_{\ell_{\bar{y}}}^t}^t = \varphi_{\pi_{\ell_{\bar{y}}}^{t_1}}^{t_1}$ which means $\varphi_{\pi_{\ell_{\bar{y}}}^t}^t \in Im$ if and only if $\varphi_{\pi_{\ell_{\bar{y}}}^{t_1}}^{t_1} \in Im$; observe also that $\ell_{\bar{y}} \in leaves^t(\bar{y})$. Thus, in t , item 2 holds. Let us also set $\ell_y = \ell_y^1$. It is immediate to see that $\pi_{\ell_y}^t$ is right (resp., left) if and only if $\pi_{\ell_y}^{t_1}$ is left (resp., right), and that $\varphi_{\pi_{\ell_y}^t}^t = \varphi_{\pi_{\ell_y}^{t_1}}^{t_1}$, which means $\varphi_{\pi_{\ell_y}^t}^t \in Im$ if and only if $\varphi_{\pi_{\ell_y}^{t_1}}^{t_1} \in Im$; observe also that $\ell_y \in leaves^t(y)$. Thus, in t , item 1 holds as well.

- $\boxed{\varphi = \varphi_1 \wedge \varphi_2}$ By inductive hypothesis, there exists $t_1 = (\mathcal{V}_1, \mathcal{E}_1, l_1, e_1, b_1)$ such that item 1 holds for some leaf $\ell_y^1 \in leaves^{t_1}(y)$ and item 2 holds for some leaf $\ell_{\bar{y}}^1 \in leaves^{t_1}(\bar{y})$, with respect to φ_1 , and $t_2 = (\mathcal{V}_2, \mathcal{E}_2, l_2, e_2, b_2)$ such that item 1 holds for some leaf $\ell_y^2 \in leaves^{t_2}(y)$ and item 2 holds for some leaf $\ell_{\bar{y}}^2 \in leaves^{t_2}(\bar{y})$, with respect to φ_2 . Informally, we obtain t by:

- appending t_2 to the branch of t_1 ending in ℓ_y^1 ,
- prepending to t_1 a new node v' (the root of the new tree t) whose right child is the root of t_1 , whose left child is a new node v'' , such that the edge $(v', root(t_1))$ (resp., (v', v'')) is labelled with \top (resp., \perp), and
- adding a back-edge from the root of t_2 to the root of t_1 and from root of t_1 to itself.

Let $v_1 = \exists(\ell_y^1)$, $\lambda = e(v_1, \ell_y^1)$, and $v', v'' \notin \mathcal{V}_1 \cup \mathcal{V}_2$. Formally, we build $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ as follows:

- $\mathcal{V} = (\mathcal{V}_1 \setminus \{\ell_y^1\}) \cup \mathcal{V}_2 \cup \{v', v''\}$,
- $\mathcal{E} = (\mathcal{E}_1 \setminus \{(v_1, \ell_y^1)\}) \cup \{(v_1, root(t_2)), (v', root(t_1)), (v', v'')\} \cup \mathcal{E}_2$,
- $l = (l_1 \setminus \{\ell_y^1 \mapsto y, \ell_{\bar{y}}^1 \mapsto \bar{y}\}) \cup \{v'' \mapsto y^*, \ell_{\bar{y}}^1 \mapsto y^*\} \cup l_2$,
- $e = (e_1 \setminus \{(v_1, \ell_y^1) \mapsto \lambda\}) \cup \{(v_1, root(t_2)) \mapsto \lambda, (v', root(t_1)) \mapsto \top, (v', v'') \mapsto \perp\} \cup e_2$,
- $b = b_1 \cup \{(root(t_2), root(t_1)), (root(t_1), root(t_1))\} \cup b_2$,

and we impose $root(t_1) = \varepsilon_{\downarrow}(v')$ and $v'' = \varepsilon'(v')$. Now, we show that t satisfies item 1 and item 2 with ℓ_y^2 and $\ell_{\bar{y}}^2$, respectively, with respect to $\varphi_1 \wedge \varphi_2$. Consider, first, $\ell_y^2 \in leaves^{t_2}(y)$. Clearly, $\ell_y^2 \in leaves^t(y)$ as well. Observe that, by construction, $ctr(\pi_{\ell_y^2}^{t_2}) = root(t_2)$. Two cases arise: if $\pi_{\ell_y^2}^{t_2}$ is right, then $\varphi_{\pi_{\ell_y^2}^{t_2}}^{t_2} \notin Im$ and $A(root(t), ctr(\pi_{\ell_y^2}^t))$; if, on the other hand, $\pi_{\ell_y^2}^{t_2}$ is left, then $\varphi_{\pi_{\ell_y^2}^{t_2}}^{t_2} \in Im$ and $D(root(t), ctr(\pi_{\ell_y^2}^t))$. Either way, $\varphi_{\pi_{\ell_y^2}^t}^t = \top \wedge (\varphi_{\pi_{\ell_y^1}^{t_1}}^{t_1} \wedge \varphi_{\pi_{\ell_y^2}^{t_2}}^{t_2})$. Now, it is immediate to see that $\mathfrak{J}, w \Vdash \varphi_{\pi_{\ell_y^2}^t}^t$ if and only if $\mathfrak{J}, w \Vdash \varphi_{\pi_{\ell_y^1}^{t_1}}^{t_1}$ and $\mathfrak{J}, w \Vdash \varphi_{\pi_{\ell_y^2}^{t_2}}^{t_2}$, that is, by inductive hypothesis, if and only if $\mathfrak{J}, w \Vdash \varphi_1$ and $\mathfrak{J}, w \Vdash \varphi_2$, that is,

if and only if $\mathfrak{J}, w \Vdash \varphi$. Thus, item 1 holds for $\ell_y = \ell_y^2$. Consider, now, $\ell_y^2 \in \text{leaves}^{t_2}(\bar{y})$. Clearly, $\ell_y^2 \in \text{leaves}^t(\bar{y})$ as well. Observe that, by construction, $\text{ctr}(\pi_{\ell_y^2}^t) = \text{root}(t_2)$. Two cases arise: if $\pi_{\ell_y^2}^t$ is right, then $\varphi_{\pi_{\ell_y^2}^t}^t \in \text{Im}$ and $A(\text{root}(t), \text{ctr}(\pi_{\ell_y^2}^t))$; if, on the other hand, $\pi_{\ell_y^2}^t$ is left, then $\varphi_{\pi_{\ell_y^2}^t}^t \notin \text{Im}$ and $D(\text{root}(t), \text{ctr}(\pi_{\ell_y^2}^t))$. Either way, $\varphi_{\pi_{\ell_y^2}^t}^t = \top \rightarrow (\varphi_{\pi_{\ell_y^1}^t}^t \rightarrow \varphi_{\pi_{\ell_y^2}^t}^t)$. Now, it is immediate to see that $\mathfrak{J}, w \Vdash \varphi_{\pi_{\ell_y^2}^t}^t$ if and only if $\mathfrak{J}, w \not\Vdash \varphi_{\pi_{\ell_y^1}^t}^t$ or $\mathfrak{J}, w \Vdash \varphi_{\pi_{\ell_y^2}^t}^t$, that is, by inductive hypothesis, if and only if $\mathfrak{J}, w \not\Vdash \varphi_1$ or $\mathfrak{J}, w \Vdash \varphi_2$, that is, if and only if $\mathfrak{J}, w \not\Vdash \varphi$. Thus, item 2 holds for $\ell_{\bar{y}} = \ell_{\bar{y}}^2$.

- $\boxed{\varphi = \diamond\varphi_1}$ By inductive hypothesis, there exists $t_1 = (\mathcal{V}_1, \mathcal{E}_1, l_1, e_1, b_1)$ such that item 1 holds for some leaf $\ell_y^1 \in \mathcal{V}_1$ and item 2 holds for some leaf $\ell_{\bar{y}}^1 \in \mathcal{V}_1$, with respect to φ_1 . Informally, we obtain t by prepending t_1 to a new node v' (the root of the new tree t) whose right child is the root of t_1 , whose left child is a new node v'' , such that the edge $(v', \text{root}(t_1))$ (resp., (v', v'')) is labelled with $\diamond\top$ (resp., $\square\perp$). Let $v', v'' \notin \mathcal{V}_1$. Formally, we build $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ as follows:

- $\mathcal{V} = \mathcal{V}_1 \cup \{v', v''\}$,
- $\mathcal{E} = \mathcal{E}_1 \cup \{(v', \text{root}(t_1)), (v', v'')\}$,
- $l = l_1 \cup \{v'' \mapsto y^*\}$,
- $e = e_1 \cup \{(v', \text{root}(t_1)) \mapsto \diamond\top, (v', v'') \mapsto \square\perp\}$,
- $b = b_1 \cup \{(\text{root}(t_1), \text{root}(t_1))\}$,

and we impose $\text{root}(t_1) = \surd(v')$ and $v'' = \surd(v')$. Now, we show that t satisfies item 1 and item 2 with ℓ_y^1 and $\ell_{\bar{y}}^1$, respectively, with respect to $\diamond\varphi_1$. Consider, first, $\ell_y^1 \in \text{leaves}^{t_1}(y)$. Clearly, $\ell_y^1 \in \text{leaves}^t(y)$ as well. Observe that, by construction, $\text{ctr}(\pi_{\ell_y^1}^t) = \text{root}(t_1)$. Two cases arise: if $\pi_{\ell_y^1}^t$ is right, then $\varphi_{\pi_{\ell_y^1}^t}^t \notin \text{Im}$ and $A(\text{root}(t), \text{ctr}(\pi_{\ell_y^1}^t))$; if, on the other hand, $\pi_{\ell_y^1}^t$ is left, then $\varphi_{\pi_{\ell_y^1}^t}^t \in \text{Im}$ and $D(\text{root}(t), \text{ctr}(\pi_{\ell_y^1}^t))$. Either way, $\varphi_{\pi_{\ell_y^1}^t}^t = \diamond(\top \wedge \varphi_{\pi_{\ell_y^1}^t}^t)$. Now, it is immediate to see that $\mathfrak{J}, w \Vdash \varphi_{\pi_{\ell_y^1}^t}^t$ if and only if there exists w' such that $(w, w') \in \mathcal{R}$ and $\mathfrak{J}, w' \Vdash \varphi_{\pi_{\ell_y^1}^t}^t$, that is, by inductive hypothesis, if and only if $\mathfrak{J}, w' \Vdash \varphi_1$, that is, if and only if $\mathfrak{J}, w \Vdash \varphi$. Thus, item 1 holds for $\ell_y = \ell_y^1$. Consider, now, $\ell_{\bar{y}}^1 \in \text{leaves}^{t_1}(\bar{y})$. Clearly, $\ell_{\bar{y}}^1 \in \text{leaves}^t(\bar{y})$ as well. Observe that, by construction, $\text{ctr}(\pi_{\ell_{\bar{y}}^1}^t) = \text{root}(t_1)$. Two cases arise: if $\pi_{\ell_{\bar{y}}^1}^t$ is right, then $\varphi_{\pi_{\ell_{\bar{y}}^1}^t}^t \in \text{Im}$ and $A(\text{root}(t), \text{ctr}(\pi_{\ell_{\bar{y}}^1}^t))$; if, on the other hand, $\pi_{\ell_{\bar{y}}^1}^t$ is left, then $\varphi_{\pi_{\ell_{\bar{y}}^1}^t}^t \notin \text{Im}$ and $D(\text{root}(t), \text{ctr}(\pi_{\ell_{\bar{y}}^1}^t))$. Either way, $\varphi_{\pi_{\ell_{\bar{y}}^1}^t}^t = \square(\top \rightarrow \varphi_{\pi_{\ell_{\bar{y}}^1}^t}^t)$. Now, it is immediate to see that $\mathfrak{J}, w \Vdash \varphi_{\pi_{\ell_{\bar{y}}^1}^t}^t$ if and only if for every w' such that $(w, w') \in \mathcal{R}$ it is the case that $\mathfrak{J}, w' \Vdash \varphi_{\pi_{\ell_{\bar{y}}^1}^t}^t$, that is, by inductive hypothesis, if and only if $\mathfrak{J}, w' \not\Vdash \varphi_1$, that is, if and only if $\mathfrak{J}, w \not\Vdash \varphi$. Thus, item 2 holds for $\ell_{\bar{y}} = \ell_{\bar{y}}^1$. ■

Modal decision trees are complete with respect to \mathcal{PL} by definition, and weakly complete with respect to \mathcal{ML} .

Theorem 3.2: Completeness of modal decision trees

Modal decision trees are strongly complete for \mathcal{PL} and weakly complete for \mathcal{ML} .

PROOF. Strong completeness for \mathcal{PL} comes trivially from the definitions. Let $\varphi \in \Phi(\mathcal{ML})$. By Lemma 3.2, there exists a modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ and two leaves $\ell_y, \ell_{\bar{y}} \in \mathcal{V}^\ell$ such that $\varphi_{\ell_y} \leftrightarrow \varphi$ and $\varphi_{\ell_{\bar{y}}} \leftrightarrow \neg\varphi$ are both valid. Therefore, $\varphi_y \rightarrow \varphi$ and $\varphi_{\bar{y}} \rightarrow \neg\varphi$ are both valid, as required. ■

Propositional decision trees are strongly complete for \mathcal{PL} , which is a known result (although not discussed in these terms in the literature), but it is not a corollary of the above result. Indeed, such a result should be proved separately, whose proof technique is similar to the above, but we decided to omit it.

Theorem 3.3: Completeness of propositional decision trees

Propositional decision trees are strongly complete for \mathcal{PL} .

Finally, we discuss classification efficiency.

Definition 3.16: Classification efficiency

A decision tree t of height h is an *efficient classifier* if and only if, for every dataset \mathcal{I} and every instance $\mathcal{J} \in \mathcal{I}$, it is the case that its run $t(\mathcal{J})$ can be computed in polynomial time with respect to h and the size of \mathcal{J} . A class of decision trees is *classification efficient* if and only if all of its decision trees are efficient classifiers.

The following result holds due the fact that model checking against an \mathcal{ML} formula against a Kripke structure can be done in polynomial time in terms of structure size and formula size (Clarke et al., 2018) (see Algorithm 1).

Theorem 3.4: Classification efficiency of modal decision trees

Modal decision trees are classification efficient.

PROOF. Immediate by the definition of run of modal decision trees. ■

Corollary 3.2: Classification efficiency of propositional decision trees

Propositional decision trees are classification efficient.

§3.4 Entropy-based Learning of Modal Decision Trees

Propositional decision trees are well-known constructs that can be used for classification and regression tasks; their popularity is due to their intrinsic simplicity,

versatility, and interpretability. It is known that the problem of extracting the *optimal* decision tree from a structured dataset is NPTIME-hard (Hyafil and Rivest, 1976), where optimality is expressed as the relation between the height and the performance of the tree, which justifies the use of sub-optimal approaches for practical applications, such as *Iterative Dichotomizer 3 (ID3)* (Quinlan, 1986), *C4.5* (Quinlan, 1993), and *Classification And Regression Trees (CART)* (Breiman et al., 1984). In general, the approaches for sub-optimal learning are divided into deterministic and non-deterministic; while it is true that decision trees can be extracted by a non-deterministic method, in the context of an optimization problem, the greedy, deterministic approach is a standard *de facto* in the case of decision trees. The most typical approach to sub-optimal decision tree learning schema is simple: starting from the root that has the entire labelled (structured) dataset \mathcal{X} , recursively partition (or, in decision tree terms, split) \mathcal{X} into k subsets $\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_k$, that contain progressively similar intra-node target values and dissimilar inter-node target values at any given level of the tree. The process continues until a stopping criterion based on the *purity* of the node (i.e., how is the distribution of values of the target variable in the node) is met, in which case the node is called *leaf*. Stopping conditions, for instance, all the samples in the node having the same class (for classification problems) and the variance of the observations being below a certain threshold (for regression problems).

Now, we lift the same greedy approach to the modal case. For a labelled modal dataset $\mathcal{I} = \{(\mathcal{J}_1, y_1), \dots, (\mathcal{J}_m, y_m)\}$, let P_i be the *fraction* of instances in \mathcal{I} labelled with y_i , that is:

$$P_i = \frac{|\{\mathcal{J} \in \mathcal{I} \mid Y(\mathcal{J}) = y_i\}|}{|\mathcal{I}|}.$$

Then, the *information conveyed by* (or, specifically, *entropy of*) \mathcal{I} is defined as:

$$Info(\mathcal{I}) = - \sum_i P_i \cdot \log P_i.$$

Intuitively, the entropy is inversely proportional to the purity degree of \mathcal{I} with respect to the class values. Splitting, the main greedy operation in learning a propositional decision tree, is performed over a specific decision $\lambda \in \Lambda$.

Definition 3.17: Associated dataset to a node

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree, v a node of t , and \mathcal{I} a modal dataset. Then, the v -associated dataset is defined as:

$$\mathcal{I}_v = \{\mathcal{J} \in \mathcal{I} \mid \mathcal{J} \Vdash \varphi_v^t\}.$$

Propositional decision tree algorithms recursively split the dataset associated with node v over the attribute A , relation \bowtie , and value a of the domain of A (i.e., a propositional decision) that guarantee the greatest information gain until a specific stopping criterion applies. When non-binary splits are allowed, the concept of split information must be slightly modified, but the underlying ideas remain. As observed by Quinlan (1986), in the case when attributes are categorical, the information gain tends to be biased towards attributes having a high number of values, that is, in our terms, higher domains, and, in some sense, the heuristic split tends to overfit the dataset. To overcome such bias, Quinlan proposed an alternative splitting criterion, called *information gain ratio*, which makes less sense for binary splits.

A modal decision tree t splits a v -associated datasets \mathcal{I}_v based on a decision $\lambda \in \Lambda$ which, together with the back-edge function b , forms two path-formulas $\varphi_{\mathcal{I}_{\lambda^r}(v)}^t$ and $\varphi_{\mathcal{I}_{\lambda^l}(v)}^t$. Hence, we have the following definitions.

Definition 3.18: Split

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree, v a node of t , \mathcal{I} a modal dataset, and \mathcal{I}_v the v -associated dataset. Then, the (binary) split of \mathcal{I}_v is the pair defined as:

$$(\mathcal{I}_{\mathcal{I}_{\lambda^r}(v)}, \mathcal{I}_{\mathcal{I}_{\lambda^l}(v)}).$$

Definition 3.19: Split information of modal decision trees

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree, v a node of t , \mathcal{I} a labelled modal dataset, \mathcal{I}_v the v -associated dataset, and $(\mathcal{I}_{\mathcal{I}_{\lambda^r}(v)}, \mathcal{I}_{\mathcal{I}_{\lambda^l}(v)})$ the split of \mathcal{I}_v . Then, the (binary) split information of \mathcal{I}_v on $(\mathcal{I}_{\mathcal{I}_{\lambda^r}(v)}, \mathcal{I}_{\mathcal{I}_{\lambda^l}(v)})$ is defined as:

$$\text{InfoSplit}(\mathcal{I}_v, \mathcal{I}_{\mathcal{I}_{\lambda^r}(v)}, \mathcal{I}_{\mathcal{I}_{\lambda^l}(v)}) = \frac{|\mathcal{I}_{\mathcal{I}_{\lambda^r}(v)}|}{|\mathcal{I}|} \cdot \text{Info}(\mathcal{I}_{\mathcal{I}_{\lambda^r}(v)}) + \frac{|\mathcal{I}_{\mathcal{I}_{\lambda^l}(v)}|}{|\mathcal{I}|} \cdot \text{Info}(\mathcal{I}_{\mathcal{I}_{\lambda^l}(v)}).$$

Definition 3.20: Information gain of modal decision trees

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ be a modal decision tree, v a node of t , \mathcal{I} a labelled modal dataset, \mathcal{I}_v the v -associated dataset, and $\lambda \in \Lambda$ a decision. Then, the information gain on \mathcal{I}_v by decision λ and back-edge function b is defined as:

$$\text{InfoGain}(\mathcal{I}_v, \lambda) = \text{Info}(\mathcal{I}_v) - \text{InfoSplit}(\mathcal{I}_v, \mathcal{I}_{\mathcal{I}_{\lambda^r}(v)}, \mathcal{I}_{\mathcal{I}_{\lambda^l}(v)}).$$

It is interesting to discuss the computational complexity of Algorithm 2. We do so under the hypothesis of having m instances described, each, by a Kripke frame with at most N distinct worlds, each containing the truth values of $|\mathcal{P}|$ different propositional letter. We study the algorithm's behaviour as m grows, considering N as a constant.

Theorem 3.5: Time complexity of learning modal decision trees

Algorithm 2 on a modal dataset \mathcal{I} with m instances, each described by a Kripke frame with at most N distinct worlds containing the truth values of $|\mathcal{P}|$ propositional letters, runs in $O(m^5)$ in the worst case and $O(m^4 \log(m))$ in the average case if N is considered a constant.

PROOF. The cardinality of the modal decisions Λ is bounded by the cardinality of the propositional letters \mathcal{P} , which is $O(m)$ because each instance can induce new propositions, but N is a constant. The number of back-edges starting at a given node is bounded by the tree's height, which, in turn, is bounded by the number of instances. Checking each modal decision consists of model checking two modal formulas whose length is bounded by $O(m)$ against every instance and model checking

ALGORITHM 2: Learning modal decision trees.

```

1 function ModalDecisionTree( $\mathcal{I}, \Lambda$ ):
  input : A labelled modal dataset  $\mathcal{I}$ , and a set of decisions  $\Lambda$ .
  output: A modal decision tree  $t$ .
2    $t \leftarrow \text{Initialise}()$ 
3    $\text{Learn}(\mathcal{I}, \Lambda, \text{root}(t), t)$ 
4   return  $t$ 
5 end
6 function Learn( $\mathcal{I}, \Lambda, v, t$ ):
7   if a stopping condition applies then return MakeLeafNode( $\mathcal{I}, v$ )
8    $v.\text{left} \leftarrow \text{CreateNode}(v)$ 
9    $v.\text{right} \leftarrow \text{CreateNode}(v)$ 
10   $\mathcal{I}^{\leftarrow}(v) \leftarrow v.\text{left}$ 
11   $\mathcal{I}^{\rightarrow}(v) \leftarrow v.\text{right}$ 
12   $\mathcal{I}^{\leftarrow}(v) \leftarrow v$ 
13   $\mathcal{I}^{\rightarrow}(v) \leftarrow v$ 
14   $(\mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}, \mathcal{I}_{\mathcal{I}^{\rightarrow}(v)}, v') \leftarrow \text{BestSplit}(\mathcal{I}, \Lambda, v, t)$ 
15   $b(v) \leftarrow v'$ 
16   $\text{Learn}(\mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}, \Lambda, v.\text{left}, t)$ 
17   $\text{Learn}(\mathcal{I}_{\mathcal{I}^{\rightarrow}(v)}, \Lambda, v.\text{right}, t)$ 
18  return
19 end
20 function BestSplit( $\mathcal{I}, \Lambda, v, t$ ):
21   $g^* \leftarrow 0$ 
22  foreach  $v' \in \mathcal{I}^*(v)$  do
23     $b(v) \leftarrow v'$ 
24    foreach  $\lambda \in \Lambda$  do
25       $(\varphi_{\mathcal{I}^{\leftarrow}(v)}^t, \varphi_{\mathcal{I}^{\rightarrow}(v)}^t) \leftarrow \text{BuildFormulas}(t, v, \lambda)$ 
26       $g \leftarrow \text{Info}(\mathcal{I}_v) - \text{InfoSplit}(\mathcal{I}_v, \mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}, \mathcal{I}_{\mathcal{I}^{\rightarrow}(v)})$ 
27      if  $g \geq g^*$  then
28         $(\mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}^*, \mathcal{I}_{\mathcal{I}^{\rightarrow}(v)}^*, v'^*) \leftarrow (\mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}, \mathcal{I}_{\mathcal{I}^{\rightarrow}(v)}, v')$ 
29         $g^* \leftarrow g$ 
30      end
31    end
32  end
33  return  $(\mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}^*, \mathcal{I}_{\mathcal{I}^{\rightarrow}(v)}^*, v'^*)$ 
34 end

```

a single formula against a single Kripke structure is linear in the size of the formula and the structure, which, in turn, is bounded by $O(N)$. Thus, the cost of finding the best split is bounded by $O(m^4)$.

In the worst-case, at each node v , the split is such that one among $\mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}$ and $\mathcal{I}_{\mathcal{I}^{\rightarrow}(v)}$ is a singleton so that the recurrence that describes the time complexity is:

$$\mathfrak{T}(m) = \mathfrak{T}(1) + \mathfrak{T}(m-1) + O(m^4),$$

from which we can conclude that:

$$\mathfrak{T} = O(m^5).$$

In the average case, however, we can assume that all splits are equally likely in terms of relative sizes of $\mathcal{I}_{\mathcal{I}^{\leftarrow}(v)}$ and $\mathcal{I}_{\mathcal{I}^{\rightarrow}(v)}$. Thus, the recurrence that describes the

time complexity becomes:

$$\begin{aligned}\mathfrak{T}(m) &= \frac{1}{m-1} \sum_{i=1}^{m-1} (\mathfrak{T}(i) + \mathfrak{T}(m-i)) + O(m^4) \\ &= \frac{2}{m-1} \sum_{i=1}^{m-1} \mathfrak{T}(i) + O(m^4).\end{aligned}$$

We prove by substitution that there exists a constant a such that $\mathfrak{T}(m) \leq am^4 \log(m)$ for m sufficiently large, from which we can conclude that $\mathfrak{T}(m) = O(m^4 \log(m))$. Let $k = 4$, then we have that:

$$\begin{aligned}\mathfrak{T}(m) &= \frac{2}{m-1} \sum_{i=1}^{m-1} \mathfrak{T}(i) + O(m^k) \\ &\leq \frac{2}{m-1} \sum_{i=1}^{m-1} ai^k \log(i) + O(m^k) \\ &\leq \frac{2a \log(m)}{m-1} \sum_{i=1}^{m-1} i^k + O(m^k).\end{aligned}$$

We leverage the generalized Faulhaber formula (Gnewuch, Pasing, and Weiß, 2021) to bound the above summation:

$$\begin{aligned}\mathfrak{T}(m) &\leq \frac{2a \log(m)}{m-1} \left(\frac{(m-1)^{k+1}}{k+1} + \frac{(m-1)^k}{2} + \frac{k(m-1)^{k-1}}{12} \right) + O(m^k) \\ &= \frac{2a \log(m)}{(k+1)(m-1)} \left((m-1)^{k+1} \frac{(k+1)(m-1)^k}{2} + \right. \\ &\quad \left. + \frac{k(k+1)(m-1)^{k-1}}{12} \right) + O(m^k).\end{aligned}$$

Since we assumed m sufficiently large, we have that $k \leq m-2$, that is, $k < k+1 \leq m-1$. Therefore, we have that:

$$\begin{aligned}\mathfrak{T}(m) &\leq \frac{2a \log(m)}{(k+1)(m-1)} \left((m-1)^{k+1} + \frac{(m-1)^{k+1}}{2} + \frac{(m-1)^{k+1}}{12} \right) + O(m^k) \\ &= \frac{2a(m-1)^k \log(m)}{(k+1)} \left(1 + \frac{1}{2} + \frac{1}{12} \right) + O(m^k).\end{aligned}$$

For large m , this means that we must prove that:

$$\frac{19}{6(k+1)} a(m-1)^k \log(m) \leq am^k \log(m),$$

which is implied by:

$$\frac{19}{6(k+1)} m^k \leq m^k.$$

The latter is true for $k \geq \frac{13}{6}$, that is, $k \geq 3$. The proof is completed. ■

MODAL LOGICS AND MODAL DATASETS

If you torture data long enough, it will confess to anything.

—Ronald H. Coase

The piece that needs to be added to the picture of modal symbolic learning is how modal datasets emerge from real-world data. This chapter shows how to interpret unstructured datasets as modal ones in several practical, real-world cases.

§4.1 Examples of Unstructured Data

The proliferation of digital devices, such as smartphones, smartwatches, and sensors, together with the advances in storage and computations, have enabled the era of big data (Gandomi and Haider, 2015). Consequently, large datasets have been created, and there is an academic, industrial, and political demand to manage and analyse such ever-growing sources of information. In the following, we give some examples of real-world data.

Time series are observations interpreted over linear orders; they are series of temporally ordered observations. Observations can be *univariate* if there is only one measurement or *multivariate* if there is more than one. Moreover, data types of observations can be numerical or categorical. Thus, a univariate time series is a single measurement evolving through time, while a multivariate time series are multiple measurements that evolve. A *temporal sequence* is a (multivariate) time series with categorical measurements; otherwise, we refer to such objects as time series. Figure 4.1 illustrates an example of time series, which represents the evolution of the temperature (A_1) and blood pressure (A_2) of a patient in a medical domain over 10 timestamps. Typical learning problems from time series are classification and regression.

Image datasets are another crucial element in big data. Massive datasets, such as ImageNet (Deng et al., 2009), have been collected to accelerate computer vision research by dealing with enormous sets of images. Images are interpreted over a 2D geometrical space (e.g., the Euclidean plane), where observations are attached to each point. Figure 4.2 illustrates an image representing the tomography of the brain as a red (R), green (G), and blue (B) image. Typical learning problems from images are image retrieval, restoration, segmentation, and classification.

Video data, by definition, are temporally ordered images, that is, videos are interpreted over a 3D geometrical space, and, as before, observations are attached to each point of the space. As such, time series and images are emblematic of studying video data. Figure 4.3 depicts an example of video, which represents a functional

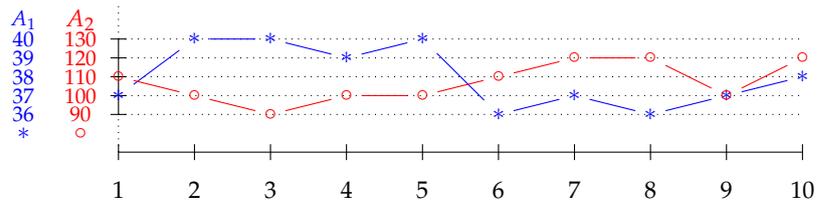


FIGURE 4.1: Example of a multivariate time series with two measurements A_1 and A_2 .

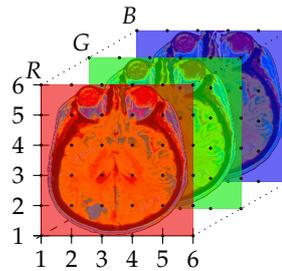


FIGURE 4.2: Example of a red R , green G , and blue (B) image.

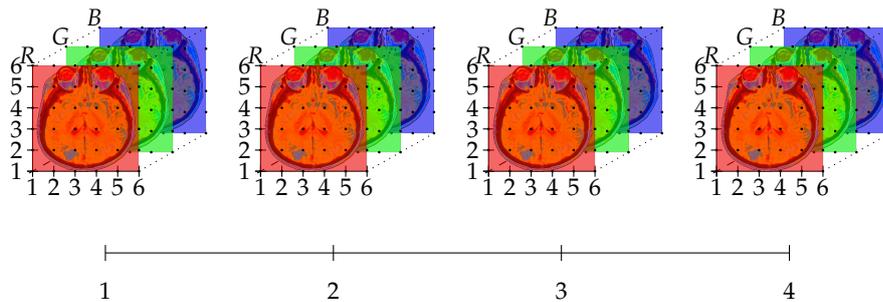


FIGURE 4.3: Example of a video with RGB images.



FIGURE 4.4: Example of an audio data.

magnetic resonance image (fMRI) evolving over 4 timestamps of RGB images. Video learning problems include video restoration, scene generation, video classification, and object detection.

Audio, or digital sound, data is the digitalisation of analogue sound by taking samples at a repeated rate, called *sampling rate*, over time. Thus, to some extent, such data can be viewed as time series and treated as such. Call centres and healthcare are the primary application areas of audio data. Call centres analyse hours and hours of recorded calls to enhance customer experience. In the clinical domain, on the other hand, breath and cough recordings can be used to diagnose COVID-19 positive subjects from negative ones. Figure 4.4 illustrates an audio recording of a COVID-19 positive subject.

Survey responses, interview transcripts, journal articles, emails, blogs, call centre

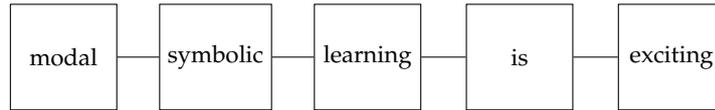


FIGURE 4.5: Example of a textual data.

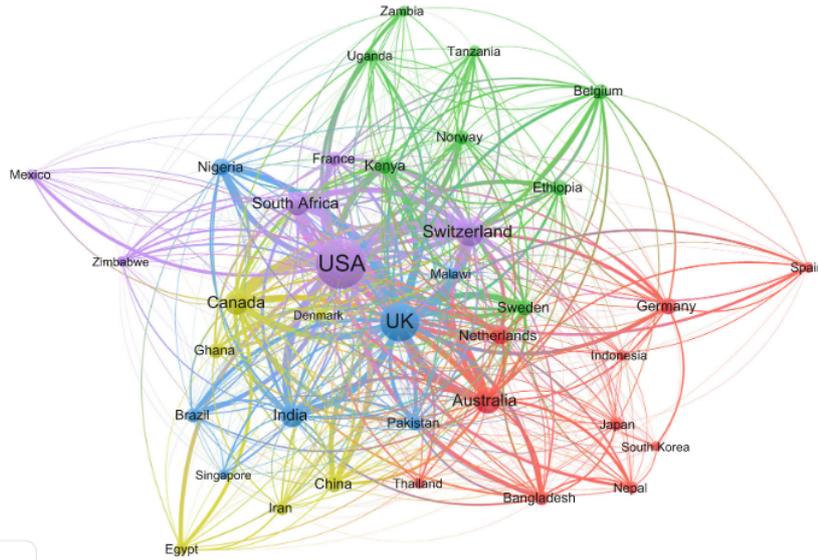


FIGURE 4.6: Example of a graph data; source Sweileh (2020).

logs, news feeds, and text documents are all examples of *textual* data. Machines cannot comprehend raw, human-generated text unless properly represented, usually in numerical form. *Natural language processing (NLP)*, which evolves from computational linguistics, uses computer science, AI, linguistics, and data science methods to enable computers to *understand* human language. NLP is an end-to-end task between the system and the human that spans from understanding the information to making decisions while interacting. Therefore, after suitable NLP preprocessing, textual data represents another form of unstructured data, which can take different forms, including, but not limited to, graphs. Figure 4.5 depicts an example of textual data, where the text is “*modal symbolic learning is exciting*” represented as a graph.

As a last example, *graph* data is a broad term that encompasses real-world data such as those emerging from social networks, collaboration networks, protein structure networks, and semantic networks, among many others. Figure 4.6 illustrates an international collaboration graph between countries.

Each of the above examples is a category on its own, and there is a broad literature about knowledge extraction from each via classification, regression, clustering, and other ML tasks. Modal symbolic learning is a step towards a unifying view, which starts with showing that most unstructured datasets can be seen, in a way, as modal ones. Let Ω be a *physical domain* of interest which can have additional structure. The *space of \mathcal{A} -valued signals on Ω* defined as:

$$\mathbb{X}(\Omega, \mathcal{A}) = \{x : \Omega \rightarrow \mathcal{A}\},$$

is a function space, where $\mathcal{A} = \{A_1, \dots, A_n\}$ is a vector space with n dimensions called *attributes*.

Definition 4.1: Datasets

Let $\mathbb{X}(\Omega, \mathcal{A})$ be the space of \mathcal{A} -valued signals on Ω and \mathbb{Y} the label space. Then, a *dataset* is a set $\mathcal{X} = \{x_1, \dots, x_m\}$ of m instances, where $x_i \in \mathbb{X}(\Omega, \mathcal{A})$, for all $1 \leq i \leq m$. The dataset \mathcal{X} is called *labelled* if each instance is labelled with an element from \mathbb{Y} , that is, $\mathcal{X} = \{(x_1, y_1), \dots, (x_m, y_m)\}$, where $y_i \in \mathbb{Y}$, for all $1 \leq i \leq m$. A *label function* $Y : \mathbb{X}(\Omega, \mathcal{A}) \rightarrow \mathbb{Y}$ is a function that associates each labelled instance to its true label.

For a signal $x \in \mathbb{X}(\Omega, \mathcal{A})$, let $x(w)[i]$ denote the value of the signal x at point w in the i th component, that is, the value of x at point w associated to A_i . Observe that, if Ω is a *singleton*, then structured datasets are a particular case of the above definition, and we use the notation $x[i]$ (instead of $x(w)[i]$ since w is the only element in Ω).

Unstructured data can be represented in terms of datasets.

Example 12 (Time series). Consider the time series in Figure 4.1. We can treat such object as a signal $x \in \mathbb{X}(\Omega, \mathcal{A})$, where $\Omega = \{1, 2, \dots, 10\}$ is a subset of the natural numbers, and $\mathcal{A} = \{A_1, A_2\}$. Moreover, we have that $x(2)[1] = 40$ and $x(7)[2] = 120$.

Example 13 (Images). Consider the image in Figure 4.2. We can treat such object as a signal $x \in \mathbb{X}(\Omega, \mathcal{A})$, where $\Omega = \{1, 2, \dots, 6\} \times \{1, 2, \dots, 6\}$ is a 2D Euclidean plane, and $\mathcal{A} = \{R, G, B\}$.

§4.2 Modal Logics for Unstructured Data

A specific modal logic is the second essential ingredient needed to see a dataset as a modal one. As we have recalled in Section 2.2, most temporal, spatial, and spatial-temporal logics are, in fact, modal. We use familiar logics because it is easier to interpret the formulas. After all, we can do qualitative reasoning over unstructured objects.

Suppose that we are interested in extracting point-based temporal knowledge from temporal data that would be beneficial in an e-commerce domain where users periodically buy goods (e.g., books, clothes, etc.), such as the marketplace of [Amazon](#). In this case, an example of temporal patterns is “if the user bought (in his/her past) an AI book, then it will buy (in his/her future) an ML one.”

From a modal symbolic learning point of view, the most emblematic temporal logic that interprets time as a set of sequences of time instants is *linear temporal logic* (Pnueli, 1977) (\mathcal{LTL}). We consider, however, a fragment of \mathcal{LTL} which encompasses only two modalities, namely, *future* (F) and *past* (P), called \mathcal{LTL} with future and past ($\mathcal{LTL}_{F,P}$). Let \mathcal{D} be a finite linearly ordered set. If we exclude the equality, there are two different binary ordering relations between two time instants on a linear order, depicted in Figure 4.7. Let \mathcal{P} be a set of propositional letters. The well-formed formulas of $\mathcal{LTL}_{F,P}$ are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid F\varphi \mid P\varphi,$$

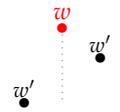
$\mathcal{LTL}_{F,P}$ modality	Definition w.r.t. the point structure	Example
F	$w\mathcal{R}_<w'$ iff $w < w'$	
P	$w'\mathcal{R}_<w$ iff $w' < w$	

FIGURE 4.7: Point relations and $\mathcal{LTL}_{F,P}$ modalities.

where $p \in \mathcal{P}$. The remaining connectives can be derived as before with particular attention to $G\varphi$ if and only if $\neg F\neg\varphi$, which stands for “it will always (in the future) be the case that φ ”, and to $H\varphi$ if and only if $\neg P\neg\varphi$, which stands for “it has always (in the past) been the case that φ ”.

The semantics of $\mathcal{LTL}_{F,P}$ are given in terms of Kripke models:

$$\mathfrak{K} = (\mathcal{W}, \mathcal{R}_<, V),$$

where \mathcal{W} is the set of time instants over \mathcal{D} , $\mathcal{R}_<$ is the transitive closure of the successor relation between time instants, and V is a valuation function $V : \mathcal{W} \rightarrow 2^{\mathcal{P}}$ which assigns to each time instant w the set of propositional letters $V(w) \subseteq \mathcal{P}$ that are true on it. The truth relation $\mathfrak{K}, w \Vdash \varphi$, for a model \mathfrak{K} , a time instant w and a formula $\varphi \in \Phi(\mathcal{LTL}_{F,P})$, is defined by induction on the complexity of formulas:

$$\begin{aligned} \mathfrak{K}, w \Vdash p & \quad \text{iff } p \in V(w), \text{ for all } p \in \mathcal{P}; \\ \mathfrak{K}, w \Vdash \neg\psi & \quad \text{iff } \mathfrak{K}, w \not\Vdash \psi \text{ (i.e., it is not the case that } \mathfrak{K}, w \Vdash \psi); \\ \mathfrak{K}, w \Vdash \psi_1 \vee \psi_2 & \quad \text{iff } \mathfrak{K}, w \Vdash \psi_1 \text{ or } \mathfrak{K}, w \Vdash \psi_2; \\ \mathfrak{K}, w \Vdash F\psi & \quad \text{iff there exists } w' \text{ s.t. } w\mathcal{R}_<w' \text{ and } \mathfrak{K}, w' \Vdash \psi; \\ \mathfrak{K}, w \Vdash P\psi & \quad \text{iff there exists } w' \text{ s.t. } w'\mathcal{R}_<w \text{ and } \mathfrak{K}, w' \Vdash \psi. \end{aligned}$$

\mathcal{LTL} and its variants have been studied for years and successfully applied to learning. *Metric interval temporal logic* (\mathcal{MITL}) (Alur, Feder, and Henzinger, 1996), a temporal logic based on \mathcal{LTL} , properties can be learned to discriminate with a high probability between good and bad time traces (Bartocci, Bortolussi, and Sanguinetti, 2014). *Signal temporal logic* (\mathcal{STL}) (Maler and Nickovic, 2004), an extension of \mathcal{MITL} that deals with real-valued signals, formulas can be learned using decision trees (Bombara et al., 2016) or optimization-based procedures (Jones, Kong, and Belta, 2014; Kong et al., 2014).

Time series represent continuous processes, and it makes little sense to model each time point on its own: time is better modelled with periods of time. When continuous signals are brought into a computer, they must be digitalised or discretized, but one should always consider the nature of signals. In the previous e-commerce example, we have seen how time series can be treated as discrete objects; now, however, we discuss how time series can be treated as continuous objects, that is, using periods of time, and examples, to name a few, span from the predictive maintenance of industrial machines based on sensors, to the study of audio signals, and the knowledge extraction from biological signals. In this case, an example of interval temporal pattern is “if the patient had (in his/her past) a period of therapy starting with some new type of medicine, then the patient will (in his/her future) need some period of monitoring to assess the clinical condition.”

While several different interval temporal logics have been proposed in the recent literature (Goranko, Montanari, and Sciavico, 2004), *Halpern and Shoham’s interval temporal logic* (\mathcal{ITS}) (Halpern and Shoham, 1991), of interest in this thesis,

\mathcal{HS} modality	Definition w.r.t. the interval structure	Example
$\langle A \rangle$	$[w, v] \mathcal{R}_A [w', v']$ iff $v = w'$	
$\langle L \rangle$	$[w, v] \mathcal{R}_L [w', v']$ iff $v < w'$	
$\langle B \rangle$	$[w, v] \mathcal{R}_B [w', v']$ iff $w = w' \wedge v' < v$	
$\langle E \rangle$	$[w, v] \mathcal{R}_E [w', v']$ iff $v = v' \wedge w < w'$	
$\langle D \rangle$	$[w, v] \mathcal{R}_D [w', v']$ iff $w < w' \wedge v' < v$	
$\langle O \rangle$	$[w, v] \mathcal{R}_O [w', v']$ iff $w < w' < v < v'$	
$\langle \bar{A} \rangle$	$[w, v] \mathcal{R}_{\bar{A}} [w', v']$ iff $[w', v'] \mathcal{R}_A [w, v]$	
$\langle \bar{L} \rangle$	$[w, v] \mathcal{R}_{\bar{L}} [w', v']$ iff $[w', v'] \mathcal{R}_L [w, v]$	
$\langle \bar{B} \rangle$	$[w, v] \mathcal{R}_{\bar{B}} [w', v']$ iff $[w', v'] \mathcal{R}_B [w, v]$	
$\langle \bar{E} \rangle$	$[w, v] \mathcal{R}_{\bar{E}} [w', v']$ iff $[w', v'] \mathcal{R}_E [w, v]$	
$\langle \bar{D} \rangle$	$[w, v] \mathcal{R}_{\bar{D}} [w', v']$ iff $[w', v'] \mathcal{R}_D [w, v]$	
$\langle \bar{O} \rangle$	$[w, v] \mathcal{R}_{\bar{O}} [w', v']$ iff $[w', v'] \mathcal{R}_O [w, v]$	

FIGURE 4.8: Allen's interval relations and \mathcal{HS} modalities.

is certainly the formalism that received the most attention, being the most natural logic for time intervals. From a logical point of view, \mathcal{HS} and its fragments have been studied on the most important classes of linearly ordered sets, from the class of all linear orders to the classes of linear orders that can be built on classical sets such as \mathbb{N} (natural numbers), \mathbb{Q} (rational numbers) and \mathbb{R} (real numbers) (Halpern and Shoham, 1991; Bresolin et al., 2014; Bresolin et al., 2019). Let \mathcal{D} be a finite linearly ordered set. An *interval* over \mathcal{D} is an ordered pair $[w, v]$ starting from w and ending in v (both included), where $w, v \in \mathcal{D}$ and $w \leq v$. An interval is called *point interval* if $w = v$, and *strict interval* if $w < v$. If we exclude the equality relation, there are twelve different binary ordering relations between two strict intervals on a linear order, often called *Allen's interval relations* (Allen, 1983): the six relations \mathcal{R}_A (*adjacent to*), \mathcal{R}_L (*later than*), \mathcal{R}_B (*begins*), \mathcal{R}_E (*ends*), \mathcal{R}_D (*during*), \mathcal{R}_O (*overlaps*), and their six *inverses*, that is, $\mathcal{R}_{\bar{X}} = (\mathcal{R}_X)^{-1}$, for each $X \in \{A, L, B, E, D, O\}$, depicted in Figure 4.8. Thus, we associate an *existential modality* $\langle X \rangle$ with each Allen's relation \mathcal{R}_X . Let \mathcal{P} be a set of propositional letters. \mathcal{HS} formulas are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi,$$

where $p \in \mathcal{P}$ and $X \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}\}$.

The strict semantics of \mathcal{HS} is given in terms of *timelines* (or, more commonly, *interval models*):

$$\mathfrak{K} = (\mathcal{W}, \{\mathcal{R}_X\}_{X \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}\}}, V),$$

where \mathcal{W} is the set of *strict intervals* over \mathcal{D} , \mathcal{R}_X are Allen's interval relations, and V is a valuation function $V : \mathcal{W} \rightarrow 2^{\mathcal{P}}$ which assigns to every interval $[w, v]$ the set of proposition letters $V([w, v]) \subseteq \mathcal{P}$ that are true on it. The truth relation $\mathfrak{K}, [w, v] \Vdash \varphi$, for an interval model \mathfrak{K} , an interval $[w, v]$ and a formula $\varphi \in \Phi(\mathcal{HS})$, is defined by

structural induction on the complexity of formulas:

$$\begin{aligned}
\mathfrak{R}, [w, v] \Vdash p & \quad \text{iff } p \in V([w, v]), \text{ for all } p \in \mathcal{P}; \\
\mathfrak{R}, [w, v] \Vdash \neg\psi & \quad \text{iff } \mathfrak{R}, [w, v] \not\Vdash \psi \text{ (i.e., it is not the case that } \mathfrak{R}, [w, v] \Vdash \psi); \\
\mathfrak{R}, [w, v] \Vdash \psi_1 \vee \psi_2 & \quad \text{iff } \mathfrak{R}, [w, v] \Vdash \psi_1 \text{ or } \mathfrak{R}, [w, v] \Vdash \psi_2; \\
\mathfrak{R}, [w, v] \Vdash \langle X \rangle \psi & \quad \text{iff there exists } [w', v'] \text{ s.t. } [w, v] \mathcal{R}_X [w', v'] \text{ and } \mathfrak{R}, [w', v'] \Vdash \psi,
\end{aligned}$$

where $X \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}\}$.

Interval temporal logics have been studied in the literature from a deductive point of view. Recall that satisfiability for \mathcal{HS} is undecidable (Halpern and Shoham, 1991) and various fragments have been considered in the literature to define fragments or variants of \mathcal{HS} with better computational behaviour. These include restricting the set of modal operators (Aceto et al., 2016; Bresolin et al., 2014), constraining the underlying temporal structure (Montanari, Sciavicco, and Vitacolonna, 2002), restricting the propositional power of the languages (Bresolin et al., 2017), and considering *coarser* interval temporal logics based on interval relations that describe a less precise relationship between intervals (similar to what topological relations do) (Muñoz-Velasco et al., 2019). $\mathcal{LTL}_{F,P}$ is a particular case of \mathcal{HS} with point intervals and two modalities, namely, $\langle L \rangle$ and $\langle \bar{L} \rangle$; \mathcal{HS}_3 and \mathcal{HS}_7 are coarse fragments of \mathcal{HS} with only three and seven modalities, respectively (Muñoz-Velasco et al., 2019); *propositional neighbourhood logic* (\mathcal{PNL}) is the fragment of \mathcal{HS} having only $\langle A \rangle$ and $\langle \bar{A} \rangle$ as modalities (Bresolin et al., 2014); *duration calculus* (\mathcal{DC}) is an interval temporal logic for real-time systems (Chaochen, Hoare, and Ravn, 1991); \mathcal{CDT} is a \mathcal{HS} -based modal logic with three binary modalities connected with a ternary accessibility relation (Venema, 1991).

Consider now, as a last scenario, the case where we are interested in extracting spatial knowledge from image data. There are many applications where such knowledge would be beneficial. For example, knowledge extracted from images can be used to classify archaeological, industrial, agricultural, and military, among many others, sites.

From a modal logic point of view, spatial logics are less studied than temporal ones. Among the known logics for space, probably the best one is the *topological* logic developed by Lutz and Wolter (2006), called \mathcal{L}_{RCC8} . Let $\mathfrak{T} = (\mathfrak{D}, \mathbb{I}^\circ)$ be a *topological space*, where \mathfrak{D} is a finite set and \mathbb{I}° is an *interior operator* on \mathfrak{D} , that is, for all $w, w' \subseteq \mathfrak{D}$, we have that:

$$\begin{aligned}
\mathbb{I}^\circ(\mathfrak{D}) & \quad = \mathfrak{D}, \\
\mathbb{I}^\circ(w) & \quad \subseteq w, \\
\mathbb{I}^\circ(w) \cap \mathbb{I}^\circ(w') & \quad = \mathbb{I}^\circ(w \cap w'), \\
\mathbb{I}^\circ \mathbb{I}^\circ(w) & \quad = \mathbb{I}^\circ(w).
\end{aligned}$$

In other terms, $\mathbb{I}^\circ(w)$ is the largest open set contained in w . The *closure* $\bar{\mathbb{I}}(w)$ of w is $\bar{\mathbb{I}}(w) = \mathfrak{D} \setminus \mathbb{I}^\circ(\mathfrak{D} \setminus w)$, that is, the smallest closed set containing w . A subset $w \subseteq \mathfrak{D}$ is called *regular closed* if $\bar{\mathbb{I}}\mathbb{I}^\circ(w) = w$, also called *region*. If we exclude the equality relation, there are eight different relations between two regions in a topological space, often called *Egenhofer and Franzosa's topological relations* (Egenhofer and Franzosa, 1991): \mathcal{R}_{DC} (*disconnected*), \mathcal{R}_{EC} (*externally connected*), \mathcal{R}_{PO} (*partially overlap*), \mathcal{R}_{TPP} (*tangential proper part*), \mathcal{R}_{NTPP} (*non-tangential proper part*), \mathcal{R}_{TPPI} (*inverse of tangential proper part*), and \mathcal{R}_{NTPPI} (*inverse of non-tangential proper part*), depicted in Figure 4.9, where the example column is interpreted over \mathbb{R}^2 . We associate an existential modality $\langle X \rangle$, for each Egenhofer and Franzosa's relation \mathcal{R}_X . Let \mathcal{P} be a set of proposition

\mathcal{L}_{RCC8} modality	Definition w.r.t. the topological structure	Example
$\langle DC \rangle$	$w\mathcal{R}_{DC}w'$ iff $w \cap w' = \emptyset$	
$\langle EC \rangle$	$w\mathcal{R}_{EC}w'$ iff $\mathbb{I}^\circ(w) \cap \mathbb{I}^\circ(w') = \emptyset \wedge w \cap w' \neq \emptyset$	
$\langle PO \rangle$	$w\mathcal{R}_{PO}w'$ iff $\mathbb{I}^\circ(w) \cap \mathbb{I}^\circ(w') \neq \emptyset \wedge w \not\subseteq w' \wedge w' \not\subseteq w$	
$\langle TPP \rangle$	$w\mathcal{R}_{TPP}w'$ iff $w \subseteq w' \wedge w \not\subseteq \mathbb{I}^\circ(w') \wedge w \neq w'$	
$\langle NTPP \rangle$	$w\mathcal{R}_{NTPP}w'$ iff $w \subseteq \mathbb{I}^\circ(w') \wedge w \neq w'$	
$\langle TPPI \rangle$	$w\mathcal{R}_{TPPI}w'$ iff $w'\mathcal{R}_{TPP}w$	
$\langle NTPPI \rangle$	$w\mathcal{R}_{NTPPI}w'$ iff $w'\mathcal{R}_{NTPP}w$	

FIGURE 4.9: Egenhofer and Franzosa's topological relations and \mathcal{L}_{RCC8} modalities.

letters. \mathcal{L}_{RCC8} formulas are generated by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi,$$

where $p \in \mathcal{P}$ and $X \in \{DC, EC, PO, TPP, NTPP, TPPI, NTPPI\}$.

The semantics of \mathcal{L}_{RCC8} is given in terms of *region structures*:

$$\mathfrak{R} = (\mathcal{W}, \{\mathcal{R}_X\}_{X \in \{DC, EC, PO, TPP, NTPP, TPPI, NTPPI\}}, V),$$

where \mathcal{W} is the set of all regions over $\mathfrak{T} = (\mathfrak{D}, \mathbb{I}^\circ)$, \mathcal{R}_X are Egenhofer and Franzosa's relations, and V is a valuation function $V : \mathcal{W} \rightarrow 2^{\mathcal{P}}$ which assigns to every region w the set of proposition letters $V(w) \subseteq \mathcal{P}$ that are true on it. The truth relation $\mathfrak{R}, w \Vdash \varphi$, for a region structure \mathfrak{R} , a region w and a formula $\varphi \in \Phi(\mathcal{L}_{RCC8})$, is defined by structural induction on the complexity of formulas:

$$\begin{aligned} \mathfrak{R}, w \Vdash p & \quad \text{iff } p \in V(w), \text{ for all } p \in \mathcal{P}; \\ \mathfrak{R}, w \Vdash \neg\psi & \quad \text{iff } \mathfrak{R}, w \not\Vdash \psi \text{ (i.e., it is not the case that } \mathfrak{R}, w \Vdash \psi); \\ \mathfrak{R}, w \Vdash \psi_1 \vee \psi_2 & \quad \text{iff } \mathfrak{R}, w \Vdash \psi_1 \text{ or } \mathfrak{R}, w \Vdash \psi_2; \\ \mathfrak{R}, w \Vdash \langle X \rangle \psi & \quad \text{iff there exists } w' \text{ s.t. } w\mathcal{R}_X w' \text{ and } \mathfrak{R}, w' \Vdash \psi, \end{aligned}$$

where $X \in \{DC, EC, PO, TPP, NTPP, TPPI, NTPPI\}$.

Spatial logics have been also studied from a deductive point of view. \mathcal{L}_{RCC8} is undecidable (Lutz and Wolter, 2006). Lutz and Wolter (2006) proposed a coarse fragment of \mathcal{L}_{RCC8} , called \mathcal{L}_{RCC5} , having only five modalities, that remains undecidable. Topological relations are not the only ones to be adopted in spatial reasoning. In fact, *directional* relations can be exploited instead, and this immediately gives a classification of spatial reasoning in topological-based and directional-based, depending on the considered type of relations. Topological relations, as we have recalled them, can be defined between objects (viewed as a set of points) without referring to their shape or their mutual position, while directional-based spatial reasoning is

closely related to the shape of the considered object, and the reference system becomes important for the choice of the set of relations. As for the directional-based spatial logics, Morales, Navarrete, and Sciavicco (2007) proposed *spatial neighbourhood logic* (\mathcal{SPNL}) as a two-dimensional spatial logic, whose *weak* variant, called *weak spatial neighbourhood logic* (\mathcal{WSPNL}), has been studied some years later (Bresolin et al., 2009).

§4.3 A Modal Logic for (Almost) All

From the above examples, we can observe that most (unstructured) data is balanced by a variety of modal logics that can be used to describe patterns in such data. Modal symbolic learning is a step towards a unifying view, that is, many datasets can be seen as modal ones, and many modal logics can be absorbed, in our context, by a single one. The resulting logic, \mathcal{HS}^d , does not include all the relevant modal logics as particular cases, but it includes many; however, those that are not included can be implemented with slight variations of it.

We can lift \mathcal{HS} to the case where worlds are regular extended objects with axes-parallel sides, which, as we shall see in the next section, produces modal datasets that can be elegantly dealt with the same family of modal logics. Let \mathcal{D} be a finite linearly ordered set, and let \mathcal{D}^d be a finite d -dimensional Euclidean space, where $d \in \mathbb{N}$, with $d \geq 1$. Elements of \mathcal{D}^d are called points denoted by (w_1, \dots, w_d) . In analogy with interval temporal logic, an *hyperrectangle* in \mathcal{D}^d is an object of the type:

$$[(w_1, v_1), (w_2, v_2), \dots, (w_d, v_d)],$$

where $w_i \leq v_i$, for each $1 \leq i \leq d$. Hyperrectangles are essentially the extension of intervals in a higher dimensional space: in the 1-dimensional case hyperrectangles are just intervals in their familiar notations, that is, $[w_1, v_1]$, obtained by simply omitting the inner brackets. In the multi-dimensional generalization, we represent any Allen's interval relation as a tuple of d single-dimensional relations, that is, $(\mathcal{R}_{X_1}, \dots, \mathcal{R}_{X_d})$, where $X_i \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}, EQ\}$, for all $1 \leq i \leq d$, where EQ is the equality relation, with the additional constraint that they cannot be all the equality; this leads to $13^d - 1$ distinct relations between any two hyperrectangles in a d -dimensional space. Let \mathcal{P} be a set of propositional letters. *Halpern and Shoham's d -dimensional hyperrectangle logic* (\mathcal{HS}^d) formulas are obtained by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X_1, \dots, X_d \rangle \varphi,$$

where $p \in \mathcal{P}$ and $X_i \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}, EQ\}$, for each $1 \leq i \leq d$.

Formulas of \mathcal{HS}^d are interpreted in a *d -dimensional model*:

$$\mathfrak{K} = (\mathcal{W}, \{(\mathcal{R}_{X_1}, \dots, \mathcal{R}_{X_d})\}_{X_i \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}, EQ\}} \setminus \overbrace{\{(\mathcal{R}_{EQ}, \dots, \mathcal{R}_{EQ})\}}^{d \text{ times}}, V),$$

where \mathcal{W} is the set of all hyperrectangles of the type $[(w_1, v_1), \dots, (w_d, v_d)]$, with $w_i \leq v_i$, for all $1 \leq i \leq d$, that can be formed on \mathcal{D}^d , $(\mathcal{R}_{X_1}, \dots, \mathcal{R}_{X_d})$ are d -dimensional Allen's relations, and V is a valuation function $V : \mathcal{W} \rightarrow 2^{\mathcal{P}}$ which assigns to every hyperrectangle w the set of proposition letters $V(w) \subseteq \mathcal{P}$ that are true on it. The truth relation $\mathfrak{K}, w \Vdash \varphi$, for a d -dimensional spatial model \mathfrak{K} , a hyperrectangle w

and a formula $\varphi \in \Phi(\mathcal{HS}^d)$, is defined by structural induction on the complexity of formulas:

$$\begin{aligned} \mathfrak{K}, w \Vdash p & \quad \text{iff } p \in V(w), \text{ for all } p \in \mathcal{P}; \\ \mathfrak{K}, w \Vdash \neg\psi & \quad \text{iff } \mathfrak{K}, w \not\Vdash \psi \text{ (i.e., it is not the case that } \mathfrak{K}, w \Vdash \psi); \\ \mathfrak{K}, w \Vdash \psi_1 \vee \psi_2 & \quad \text{iff } \mathfrak{K}, w \Vdash \psi_1 \text{ or } \mathfrak{K}, w \Vdash \psi_2; \\ \mathfrak{K}, w \Vdash \langle X_1, \dots, X_d \rangle \psi & \quad \text{iff there exists } w' \text{ s.t. } w(\mathcal{R}_{X_1}, \dots, \mathcal{R}_{X_d})w' \text{ and } \mathfrak{K}, w' \Vdash \psi, \end{aligned}$$

where $X_i \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}, EQ\}$, for each $1 \leq i \leq d$. We denote by \mathcal{HS}^0 the degenerate case when the resulting logic is simply \mathcal{PL} .

Clearly, d -dimensional models are Kripke models. Although \mathcal{HS}^d has not been studied *per se*, the literature that concerns its fragments is very wide. We have already discussed the case of \mathcal{HS} , the 1-dimensional case of \mathcal{HS}^d , in the previous section. In the case of \mathcal{HS}^2 , only a few sub-languages have been studied (Lutz and Wolter, 2006), and their satisfiability problem has been shown to be undecidable as well, even under very simple assumptions, or they can be proven so by exploiting the results on the 1-dimensional case. In general, one can expect deduction in \mathcal{HS}^d to be a computationally hard problem even under very restrictive assumptions, such as finite domains. Although, in the spirit of existing works for interval temporal logic, one can imagine exploring fragments of \mathcal{HS}^d ; here, we are interested in induction problems, for which expressive power and the possibility of describing patterns are more important desiderata. Figure 4.10 schematically illustrates some extensions and fragments of \mathcal{HS}^d ; we briefly discuss some scenarios to better grasp the general idea. First, consider the formalisms that are green-shaded in Figure 4.10. Then, \mathcal{HS} is just \mathcal{HS}^1 with strict intervals, \mathcal{PNL} is just \mathcal{HS}^1 with two modalities (i.e., $\langle A \rangle$ and $\langle \bar{A} \rangle$), and $\mathcal{LTL}_{F,P}$ is \mathcal{HS}^1 with only point intervals and only two modalities (i.e., $\langle L \rangle$ and $\langle \bar{L} \rangle$); thus, in a sense, we can classify these logics as particular cases of \mathcal{HS}^d , obtainable with minimal effort. Now, consider the logics represented in yellow in Figure 4.10. Then, \mathcal{HS}^d is similar to \mathcal{L}_{RCC8} with regular extended objects with axes-parallel sides having eight modalities, some of which are *coarse*; for example, $w\mathcal{R}_{NTPP}w'$ in \mathcal{L}_{RCC8} if and only if $w(\overbrace{\mathcal{R}_D, \dots, \mathcal{R}_D}^{d \text{ times}})w'$ in \mathcal{HS}^d . Finally, consider the red cases in Figure 4.10. Then, these logics share their modal nature with \mathcal{ML} and their temporal nature with \mathcal{HS} , of which \mathcal{HS}^d is a generalization. The binary operators, as well as the duration constraint, may require non-trivial modifications of the modal symbolic learning machinery; nonetheless, the driving ideas would be the same.

We can, thus, choose \mathcal{HS}^d as a representative for modal symbolic learning. The need of having a unifying view of many real-world situations that can be captured, at least partially, by a single logic justifies our commitment. We must, however, discuss model checking for it since it is at the core of any modal symbolic learning procedure. Algorithm 3 is just the adaptation of Algorithm 1 to \mathcal{HS}^d from \mathcal{ML} , and it has the same asymptotic time complexity, given in terms of modal datasets. As we shall see in the next section, a dataset that is transformed to a modal one has a polynomial blow-up in the size of the original input (e.g., a time series treated with an interval-based formalism, such as \mathcal{HS} , can be seen as a Kripke model which is quadratic in the size of the original time series due to the possible intervals over such time series), which affects the performances of the model checking in practice.

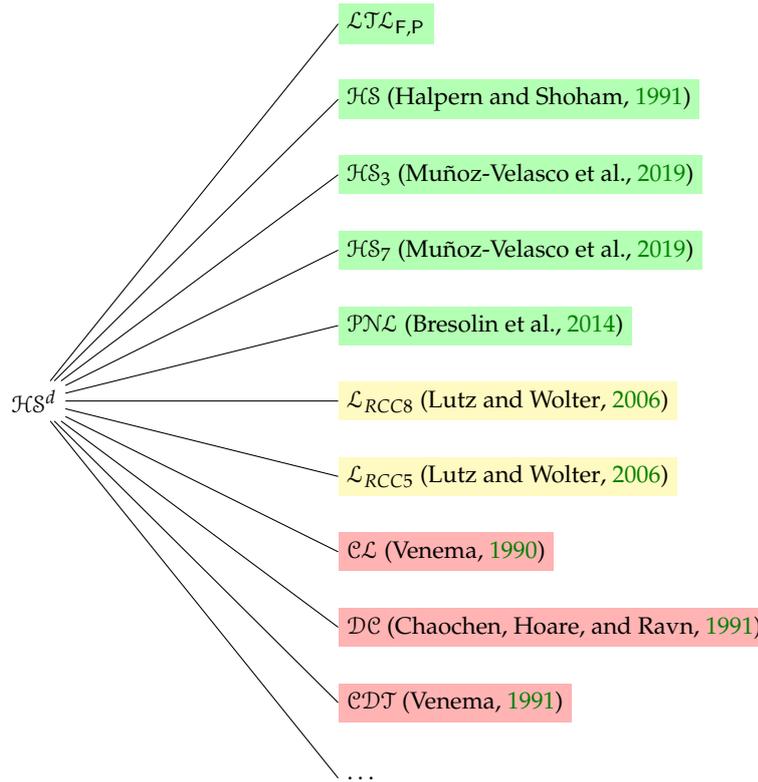


FIGURE 4.10: $\mathcal{H}\mathcal{S}^d$ -based fragments and extensions: green-shaded require no, or at most few additional, assumptions, yellow-shaded require more assumptions, and red-shaded require even more assumptions.

§4.4 Datasets to Modal Datasets

In the process of transforming a dataset into a modal one, each resulting instance is associated with a Kripke model; as it turns out, however, there is no unique way to identify the Kripke frame (and, thus, the model).

To present the general idea, consider the simple case of signals on $\mathbb{X}(\mathcal{D}, \mathcal{A})$, where \mathcal{D} is a finite linearly ordered set. Propositional decision trees, or other off-the-shelf models, can be learned from such signals by transforming an unstructured dataset into a structured one as follows. Consider the time series case as an example. For a signal $x \in \mathbb{X}(\mathcal{D}, \mathcal{A})$, let $x(w : v)[i]$ be the vector having exactly $v - w + 1$ values of $A_i \in \mathcal{A}$ contained from point w to point v (both included). Observe that, $x(1 : |\mathcal{D}|)[i]$ is the vector of $|\mathcal{D}|$ values of A_i , that is, the original values relative to A_i on \mathcal{D} . Therefore, intuitively, $x(w : v)[i]$ represents an *interval-slice* $[w, v]$ of the original signal projected on A_i . To obtain a structured object x' from the unstructured one x , we must devise a set of functions that can be applied to vectors (or, in general, matrices) that return a single value that can be compared (through \boxtimes) with some other value a (see Definition 3.2). Assume that such set of functions contains only the maximum function of a collection of numbers. Now, we can compute $\max(x(1 : |\mathcal{D}|)[i])$, for all $1 \leq i \leq n$, to obtain a n -dimensional vector representation x' of x , and a propositional decision tree algorithm can learn from x' . The important observation here is that one could apply \max to any other slice that is contained in $x(1 : |\mathcal{D}|)[i]$. The same reasoning holds for more-than-one dimension, that is, for $\Omega = \mathcal{D}^d$. In fact, let $x(w_1 : v_1, \dots, w_d : v_d)[i]$ be the $v_1 - w_1 + 1 \times \dots \times v_d - w_d + 1$

ALGORITHM 3: Model checking for $\mathcal{H}\mathcal{S}^d$.

```

1 function Check( $\mathfrak{R}, \varphi$ ):
  input : A  $d$ -dimensional model
           $\mathfrak{R} = (\mathcal{W}, \{\mathcal{R}_{X_1}, \dots, \mathcal{R}_{X_d}\})_{X_i \in \{A, L, B, E, D, O, \bar{A}, \bar{L}, \bar{B}, \bar{E}, \bar{D}, \bar{O}, EQ\}, V}$  and a  $\mathcal{H}\mathcal{S}^d$  formula  $\varphi$ .
  output: A mapping  $\ell : \mathcal{W} \rightarrow 2^{\Phi(\mathcal{H}\mathcal{S}^d)}$ 
  foreach  $\psi \in \text{sub}(\varphi)$  ordered by increasing length do
2     if  $\psi = p \in \mathcal{P}$  then
3         foreach  $w \in \mathcal{W}$  do
4             if  $p \in V(w)$  then
5                  $\ell(w) \leftarrow \{p\}$ 
6             end
7         end
8     end
9     if  $\psi = \neg\psi_1$  then
10        foreach  $w \in \mathcal{W}$  do
11            if  $\psi_1 \notin \ell(w)$  then
12                 $\ell(w) \leftarrow \ell(w) \cup \{\psi\}$ 
13            end
14        end
15    end
16    if  $\psi = \psi_1 \vee \psi_2$  then
17        foreach  $w \in \mathcal{W}$  do
18            if  $\psi_1 \in \ell(w)$  or  $\psi_2 \in \ell(w)$  then
19                 $\ell(w) \leftarrow \ell(w) \cup \{\psi\}$ 
20            end
21        end
22    end
23    if  $\psi = \langle X_1, \dots, X_d \rangle \psi_1$  then
24        foreach  $w \in \mathcal{W}$  do
25            if  $\exists w'$  such that  $(w, w') \in (\mathcal{R}_{X_1}, \dots, \mathcal{R}_{X_d})$  and  $\psi_1 \in \ell(w')$  then
26                 $\ell(w) \leftarrow \ell(w) \cup \{\psi\}$ 
27            end
28        end
29    end
30    end
31    return  $\ell$ 
32 end
33 end

```

matrix of values of A_i going from point w_i to point v_i (both include) along each axis i , for $1 \leq i \leq d$. Inspired by the above idea, we have the following definition.

Definition 4.2: Feature extraction functions

Let $\mathbb{X}(\mathcal{D}^d, \mathcal{A})$ be the space of \mathcal{A} -valued signals on \mathcal{D}^d and $\mathcal{X} = \{x_1, \dots, x_m\}$ a dataset. Then, a *feature extraction function* f is defined as:

$$f : \overbrace{2^{\text{dom}(A)} \times \dots \times 2^{\text{dom}(A)}}^{d \text{ times}} \rightarrow \mathbb{R},$$

for some $A \in \mathcal{A}$. Moreover, let \mathbb{F} be the *feature extraction function space*.

Examples of feature extraction functions are the minimum, maximum and average; others, studied ad hoc for time series, are *CANonical Time series CHaracteristics (catch22)* (Lubba et al., 2019) and *Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh)* (Christ et al., 2018); for the more-than-one dimensional cases,

ad-hoc feature extraction functions can be devised for the d -dimensional case, but this discussion goes beyond the scope of this work.

Example 14 (Feature extraction functions). *Consider the signal x in Figure 4.1. Let the \max be the feature extraction function. Then, we have that:*

$$\begin{aligned}\max(x(1 : 5)[1]) &= 40, \\ \max(x(7 : 9)[1]) &= 37, \\ \max(x(2 : 4)[2]) &= 100, \\ \max(x(5 : 8)[2]) &= 120.\end{aligned}$$

Similarly, let \min be the feature extraction function. Then, we have that:

$$\begin{aligned}\min(x(2 : 4)[1]) &= 39, \\ \min(x(6 : 8)[1]) &= 36, \\ \min(x(1 : 5)[2]) &= 90, \\ \min(x(6 : 9)[2]) &= 100.\end{aligned}$$

Since we are considering \mathcal{HS}^d as a representative for the modal symbolic learning framework, the propositional letters are:

$$\mathcal{P} = \{f(A) \bowtie a \mid f \in \mathbb{F}, A \in \mathcal{A}, \bowtie \in \{\leq, <, =, \neq, >, \geq\}, a \in \text{dom}(f(A))\}.$$

However, it is important to stress that, when the worlds are point-based, \mathbb{F} can only be a singleton having the identity function, which is crucial for formalisms such as $\mathcal{LTL}_{\mathbb{F}, \mathbb{P}}$.

We are ready to define how an unstructured object can be seen as a Kripke model.

Definition 4.3: Modal logic transformer for signals on $\mathbb{X}(\Omega, \mathcal{A})$

Let $\mathbb{X}(\Omega, \mathcal{A})$ be the \mathcal{A} -valued signals on Ω , \mathcal{L} a modal logic, \mathbb{P} the space of all sets of propositional letters, and \mathbb{K} the Kripke model space. Then, a \mathcal{L} -transformer $\tau_{\mathcal{L}}$ is defined as:

$$\tau_{\mathcal{L}} : \mathbb{X}(\Omega, \mathcal{A}) \times \mathbb{P} \rightarrow \mathbb{K},$$

which returns a Kripke model $\mathfrak{K} \in \mathbb{K}$ from an input signal $x \in \mathbb{X}$ and a set of proposition letters $\mathcal{P} \in \mathbb{P}$.

Figure 4.11 schematically illustrates the cases captured by the \mathcal{HS}^d transformers and those not directly captured by it. As we have mentioned, NLP processes raw, input text to produce a numerical form of it so that, to some extent, one can imagine that each word in a text is associated with a fixed size, using the standard nomenclature, embedding vector; at this point, \mathcal{HS}^d can be exploited. Moreover, the structure of a graph can be seen as a Kripke frame, and the propositional letters with their truth values can be induced from the original graph to obtain a Kripke model; again, at this point, \mathcal{HS}^d can be exploited.

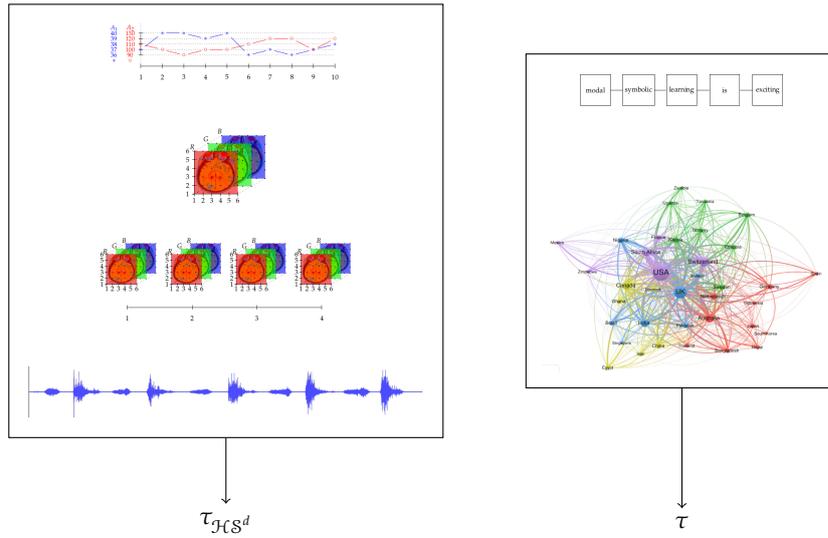


FIGURE 4.11: Cases that are captured naturally by \mathcal{HCS}^d and those that require more assumptions.

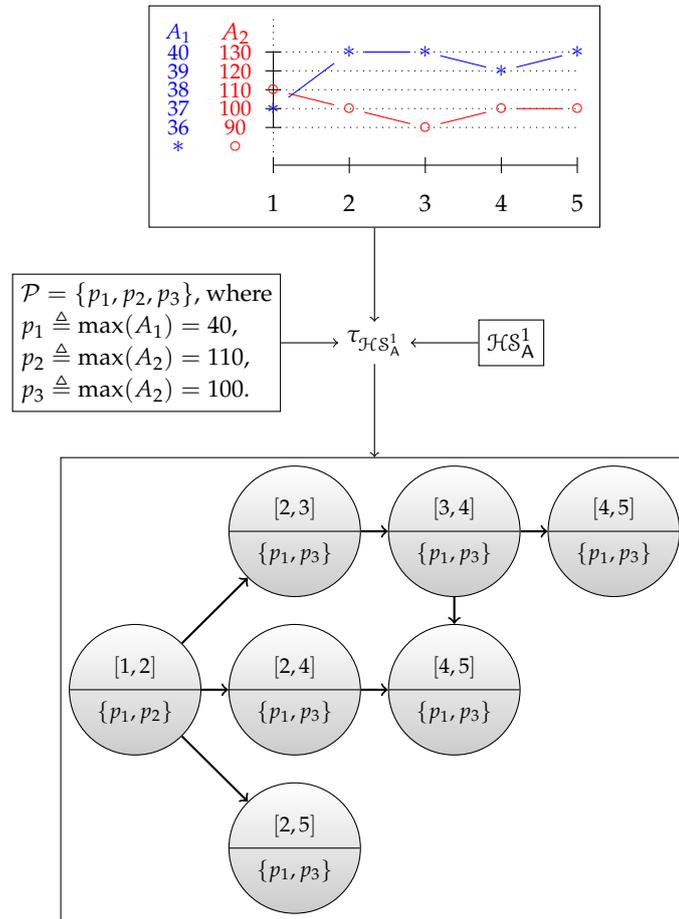


FIGURE 4.12: Kripke model after the application of the \mathcal{HCS}_A^1 -transformer.

We give an example of a concrete transformer for illustrative purposes.

Example 15 ($\mathcal{H}\mathcal{S}^1$ -transformer for temporal data). Consider the signal $x \in \mathbb{X}(\mathcal{D}, \mathcal{A})$ in Figure 4.12, top. In this example, for the sake of simplicity, we fix the following proposition letters:

$$\mathcal{P} = \{\max(A_1) = 40, \max(A_2) = 110, \max(A_2) = 100\},$$

and we consider a fragment of $\mathcal{H}\mathcal{S}^1$ with only one modality $\langle A \rangle$, denoted by $\mathcal{H}\mathcal{S}_A^1$. Then, the $\mathcal{H}\mathcal{S}_A^1$ transformer $\tau_{\mathcal{H}\mathcal{S}_A^1}$ applied on x and \mathcal{P} returns the Kripke model in Figure 4.12, bottom.

We have all the elements to conduct modal symbolic learning from unstructured data.

LEARNING WITH MODAL SYMBOLIC LEARNING

Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful.

—George Edward Pelham Box

This chapter shows how learning is performed with modal symbolic learning. We present several points of view on how an ML practitioner could engage in learning modal theories. At the end of this chapter, we present two real-world applications of modal symbolic learning by instantiating the framework with \mathcal{HS}^1 and \mathcal{HS}^2 for learning from temporal and spatial data, respectively.

§5.1 Regression with Modal Decision Trees

Decision trees have been initially studied for classification. However, decision tree algorithms can also learn regression models (i.e., regression with decision trees) when the target variable is numerical (i.e., $\mathbb{Y} = \mathbb{R}$) by simply adapting the entire theory to handle such a circumstance. In such a case, the *variance reduction* approach is the to-go strategy, which replaces the greedy entropy-based split strategy with the one based on variance. It is immediate, therefore, to inherit the same idea from the propositional case.

Consider a labelled modal dataset $\mathcal{I} = \{(\mathcal{J}_1, y_1), \dots, (\mathcal{J}_m, y_m)\}$. Then, the (*corrected*) *variance of \mathcal{I}* is defined as:

$$\text{Var}(\mathcal{I}) = \frac{1}{m-1} \cdot \sum_{i=1}^m (y_i - \mu_y)^2,$$

where μ_y is the mean of the labels in \mathcal{I} . Now, by simply plugging in the above definition in the entire mechanism discussed in Chapter 3, and in particular in Section 3.4, we obtain regression modal decision trees by letting $\text{Info}(\mathcal{I}) = \text{Var}(\mathcal{I})$.

§5.2 Random Forests with Modal Decision Trees

In the propositional case, the generalization from single trees to forests of trees is relatively simple. The idea that underlies the so-called random forest model is the following one (Breiman, 2001): different independent trees can be learned from different subsets of the training set, using different subsets of attributes. Each tree is precisely a propositional decision tree; a random forest classifier, however, is a classifier whose semantics depends on many trees and is computed via a *voting* function. Introducing modal random forest models can be done in the same way.

ALGORITHM 4: Learning modal random forests.

```

1 function ModalRandomForest( $\mathcal{I}, \Lambda, k, m^{sub}, n^{sub}$ ):
   input : A labelled modal dataset  $\mathcal{I}$ , a set of decisions  $\Lambda$ , the number  $k$  of decision trees in
           the random forest, the number  $m^{sub}$  of instances to give to each single decision
           tree, and the number  $n^{sub}$  of attributes that each single tree learns from.
   output: A modal random forest  $\mathcal{T}$ .
2    $\mathcal{T} \leftarrow \emptyset$ 
3   foreach do
4      $\mathcal{I}' \leftarrow \text{InstancesSubsetSample}(\mathcal{I}, m^{sub})$ 
5      $\Lambda' \leftarrow \text{DecisionsSubsetSample}(\Lambda, n^{sub})$ 
6      $t \leftarrow \text{ModalDecisionTree}(\mathcal{I}', \Lambda')$ 
7      $\mathcal{T} \leftarrow \mathcal{T} \cup \{t\}$ 
8   end
9   return  $\mathcal{T}$ 
10 end

```

Definition 5.1: Modal random forests

Let \mathbb{Y} be the label space. Then, a *modal random forest* is a pair (\mathcal{T}, ζ) , where \mathcal{T} is a collection of k modal decision trees, that is, $\mathcal{T} = \{t_1, \dots, t_k\}$, and $\zeta : \mathbb{Y}^k \rightarrow \mathbb{Y}$ is a *voting aggregation function* of all unit votes of each modal decision tree $t \in \mathcal{T}$.

We can classify an instance with a modal random forest by exploiting the voting aggregation function and the individual runs of each modal decision tree as follows.

Definition 5.2: Run of modal random forests

Let (\mathcal{T}, ζ) be a modal random forest, where $\mathcal{T} = \{t_1, \dots, t_k\}$, and \mathcal{J} an instance of a modal dataset \mathcal{I} . Then, the *run of \mathcal{T} on \mathcal{J}* is defined as:

$$\mathcal{T}(\mathcal{J}) = \zeta(t_1(\mathcal{J}), \dots, t_k(\mathcal{J})).$$

Random forests differ from simple deterministic decision trees in many subtleties, all related to the learning algorithm. Such differences, along with the nature of the model, transform a purely symbolic method, such as decision trees, into a hybrid symbolic and non-symbolic one due to the voting function; however, the voting function can be learned by a (propositional) symbolic learning algorithm to keep the resulting random forest still symbolic. In its simplest form, ζ can be the average if the label is numerical (i.e., regression problem) or can be the maximum class value among the k (base) decision trees if the label is categorical (i.e., classification problem). Algorithm 4 provides a high-level description of how a modal random forest is learned. Each single decision tree t is learned on a subset \mathcal{I}' of \mathcal{I} having m^{sub} instances and on a subset Λ' of Λ having only the decision relative to n^{sub} attributes.

§5.3 Rules Extraction from Modal Decision Trees and Modal Random Forests

Rule-based models, as we have discussed in the introduction of this thesis, have been studied for some time in AI. A *rule* ρ is a symbolic object of the type:

$$\rho : \varphi \Rightarrow y,$$

where φ is called *antecedent* and y is called *consequent*. In general, the antecedent is a formula of some logical formalism, and the consequent can be a label from \mathbb{Y} or another formula. Moreover, as we are in an ML context, to emphasize the difference with respect to \rightarrow , which is a logical implication, in rules we use \Rightarrow .

Suppose that we have to solve a classification problem by means of a *rule-based classifier* Γ , a classification model based on rules. Decision trees provide a natural way to extract rules from them by exploiting the path-, leaf-, and class-formulas. For example, a modal rule-based classifier Γ can be synthesized from a modal decision tree $t = (\mathcal{V}, \mathcal{E}, l, e, b)$ as follows. Let $\mathbb{Y} = \{y_1, y_2\}$. Then, Γ is defined as:

$$\Gamma = \{\varphi_\ell^t \Rightarrow y \mid y \in \mathbb{Y}, \ell \in \text{leaves}^t(y)\}$$

if one wants to make explicit every single formula (i.e., a formula for each branch), or as:

$$\Gamma = \{\varphi_y^t \Rightarrow y \mid y \in \mathbb{Y}\}$$

if one wants to have a more compact representation (i.e., a formula for each class). Since Γ is extracted from a single modal decision tree, Γ can be used without ambiguities because decision trees are known to be *divide-and-conquer* (recall correctness of decision trees), that is, they partition instances at each split level. In other terms, for a testing instance \mathcal{I} , only the antecedent of a single rule $\varphi \Rightarrow y$ will be true for it (i.e., $\mathcal{I} \Vdash \varphi$), and such instance will be classified with the consequent of that rule (i.e., y). Therefore, we have the following definitions.

Definition 5.3: Modal rule-based prediction model

Let \mathbb{Y} be the label space. Then, a *modal rule-based prediction model* $\Gamma = \{\varphi_i \Rightarrow y_i\}_{i=1}^k$ is a set of k rules of the type $\varphi_i \Rightarrow y_i$, where $\varphi_i \in \Phi(\mathcal{ML})$ and $y_i \in \mathbb{Y}$.

Definition 5.4: Run of modal rule-based prediction model

Let $\Gamma = \{\varphi_i \Rightarrow y_i\}_{i=1}^k$ be a modal rule-based prediction model and \mathcal{I} an instance of a modal datasets \mathcal{I} . Then, the *run of Γ on \mathcal{I}* is defined as:

$$\Gamma(\mathcal{I}) = y_i \quad \text{if } \mathcal{I} \Vdash \varphi_i.$$

Imagine, now, that we want to obtain a rule-based classifier Γ from a random forest (\mathcal{T}, ζ) as:

$$\Gamma = \{\varphi_\ell^t \Rightarrow y \mid y \in \mathbb{Y}, \ell \in \text{leaves}^t(y), t \in \mathcal{T}\}$$

or as:

$$\Gamma = \{\varphi_y^t \Rightarrow y \mid y \in \mathbb{Y}, t \in \mathcal{T}\}.$$

In this case, the resulting set of rules is, in general, *ambiguous*: antecedents of many rules may be true for the same testing instance; indeed, a training instance is classified by all single decision trees $t \in \mathcal{T}$. In this case, it is better to *order* the rules in Γ in some arbitrary way. A well-known model is a decision list (Rivest, 1987), which is learned directly from data without extracting rules from decision trees, but the general ideas are still valid. Since the ordering is arbitrary, different metrics have been defined in the literature to break ties between rules, such as support, confidence, lift, and conviction, among many others (e.g., see Tan et al., 2019). Depending on the application-domain, many rules can be discarded from the resulting theory as they may not be *interesting* for the application (e.g., keep the rules that have high support).

§5.4 Multi-Frame Modal Symbolic Learning

There are real-world situations where multiple descriptions describe a single event. In a clinical domain, for example, a patient can be described by a multivariate time series representing the evolution of his/her fever and blood pressure (i.e., temporal data), by an image representing his/her chest X-ray (i.e., spatial data), but also by propositional descriptions such as the number of cigarettes that he/she smokes (i.e., structured data). To address such situations, we propose multi-frame modal symbolic learning to enhance modal symbolic learning methods with the possibility of learning from more-than-one description at the same time and describing the learned knowledge using the correct logic, which, to some extent, can be seen as a product of r modal logics. From this, we need to extend the definition of modal datasets so that multiple descriptions describe each instance.

Definition 5.5: Multi-frame modal datasets

Let \mathbb{K} be the Kripke model space and \mathbb{Y} the label space. Then, a *multi-frame modal dataset* $\mathcal{I} = \{\mathcal{J}_1, \dots, \mathcal{J}_m\}$ is a set of m multi-frame instances each of which is associated with r Kripke models, that is, $\mathcal{J}_i \in \mathbb{K}^r$, for all $1 \leq i \leq m$. The dataset \mathcal{I} is called *labelled* if each instance is labelled with an element from \mathbb{Y} , that is, $\mathcal{I} = \{(\mathcal{J}_1, y_1), \dots, (\mathcal{J}_m, y_m)\}$, where $y_i \in \mathbb{Y}$, for all $1 \leq i \leq m$. A *label function* $Y : \mathbb{K}^r \rightarrow \mathbb{Y}$ is a function that associates each labelled instance to its true label.

Multi-frame modal datasets capture d -dimensional situations quite naturally, but just as it happens in modal symbolic learning, in the multi-frame case, too, we need to *concretize* the learning models and the associated languages to specific modal logics to adapt them to real-world cases. The above definition also captures challenging real-world scenarios such as different alignments in temporal data (e.g., the patient has two audio recordings sampled on different days) and different scales in spatial data (e.g., the patient has two medical images that have different resolutions). Therefore, we can plug in the mechanism of $\mathcal{H}\mathcal{S}^d$ to learn from multiple descriptions at the same time (i.e., $\mathcal{H}\mathcal{S}^0$ for structured data, $\mathcal{H}\mathcal{S}^1$ for temporal data, $\mathcal{H}\mathcal{S}^2$ for spatial data, etc.).

§5.5 Blueprint of Modal Symbolic Learning

Figure 5.1 schematically illustrates the blueprint of modal symbolic learning. We start with an input dataset $\mathcal{X} = \{x_1, \dots, x_m\}$ from which we would like to learn a

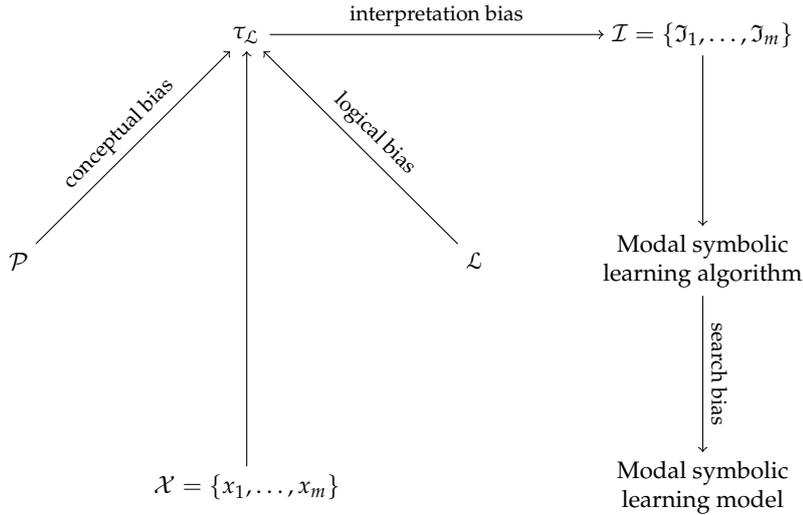


FIGURE 5.1: Blueprint of modal symbolic learning.

modal symbolic learning model. We must make some initial assumptions in terms of conceptual bias (\mathcal{P}) and logical bias (\mathcal{L}) so that a \mathcal{L} -transformer $\tau_{\mathcal{L}}$ can be defined to obtain a modal dataset $\mathcal{I} = \{\mathcal{I}_1, \dots, \mathcal{I}_m\}$ by leveraging the interpretation bias (e.g., what is the truth value of a propositional letter). At this point, a modal symbolic learning algorithm can be applied on \mathcal{I} to build a modal symbolic learning model by exploiting the search bias for learning \mathcal{L} patterns.

§5.6 Real-world Data Experiments: Temporal Data

In this section, we show how to learn with temporal (modal) symbolic learning. We consider COVID-19 cough and breath audio recordings data to diagnose positive from negative subjects. Since such data are represented as time series and, following our general framework, instantiate such a problem with \mathcal{HS} . In this practical context, we choose for this task temporal decision trees and temporal random forests; moreover, we model the task as a multi-frame setting since breath and cough recordings can be described, each, by a Kripke model. These results have been published by Manzella et al. (2021).

Dataset. The dataset used in this work, presented by Brown et al. (2020), was originally crowdsourced and compiled by researchers at the University of Cambridge, and it is available upon request. It has the following structure. The dataset in its entirety is composed of 9986 audio samples recorded by 6613 volunteers. Each audio recording is encoded in the *Waveform Audio File (WAV)* format and consists of a discrete sampling of the perceived sound pressure caused by (continuous) sound waves. Out of all volunteers, 235 declared to be *positive* to COVID-19. The subjects are quasi-normally distributed by age, with an average between 30 and 39 and a frequency curve slightly left-skewed towards younger ones; the data is not gender-balanced, with more than double as many male subjects than female ones. Besides recording sound samples, subjects were asked to fill in a very brief clinical history, plus information about their geographical location. Brown et al. (2020) used this data to derive smaller datasets, each posing a different form of the same task of COVID-19 diagnosis. In particular, the location of the subject was used to distinguish among

those that, at the moment of the recording, were living in almost-COVID-free countries; by combining this information with the subjects' declaration concerning a diagnostic test for COVID-19, only a subset of the subjects who declared to be negative could indeed be reliably considered as negative. With this approach, three tasks were designed:

1. to distinguish between declared positive subjects from non-positive ones that have a *clean medical history*, have *never smoked*, have *no symptoms*, and live in countries in which the virus spread at that moment was very low (so they can be reliably considered negative subjects);
2. to distinguish between declared positive subjects with *cough as symptom* from non-positive ones that have a *clean medical history*, have *never smoked*, have *cough as a symptom*, and live in countries in which the virus spread at that moment was very low (so they can be reliably considered negative subjects with a cough);
3. to distinguish between declared positive subjects with *cough as symptom* from non-positive ones that have *asthma*, that have *never smoked*, have *cough as a symptom*, and live in countries in which the virus spread at that moment was very low (so they can be reliably considered negative subjects with cough and asthma).

We refer to these tasks and datasets as *TA1*, *TA2*, and *TA3*, respectively. To counteract the small amount of control data, the authors of the original dataset also produced and released two *augmented* versions for *TA2* and *TA3* (referred to, here, as *TA2+* and *TA3+*, respectively), obtained with standard audio augmentation methods. Each task was declined into three versions by Brown et al. (2020), which differ by how subjects are represented, that is, using only their cough sample, only their breath sample, or both. In their original release temporal $\mathcal{H}\mathcal{S}$ -based decision trees and forests were not designed for multi-frame representation of the data. Nevertheless, thanks to the multi-frame formalization, these models are also able to treat subjects represented as the union of a cough and a breath sample. With respect to the original work, we have eliminated 14 instances that presented cough/breath labeling mistakes, empty audio recording, and/or a too-noisy background; barring such differences, it is possible to directly compare our results with the ones from the original paper and from other models trained on the same data. Moreover, because of the nature of interval temporal trees, it also makes sense to explore the possibility of learning from *single* coughs/breath cycles, instead of whole recordings (which present, each, several episodes); for each version of the dataset, we produced a *segmented* variant containing single episodes from the original ones.

Preprocessing techniques. In audio signal processing, it is customary to extract *spectral* representations of sounds, facilitating their interpretation in terms of audio frequencies. To this end, we adopt a variation of a widespread representation technique, which goes under the name of *Mel-Frequency Cepstral Coefficients (MFCC)*. MFCC, first proposed by Davis and Mermelstein (1980), is still the preferred technique for extracting sensible spectral representations of audio data, and its use in ML has been fruitful for tackling hard AI tasks, such as speech recognition, music genre recognition, noise reduction, and audio similarity estimation. Computing the MFCC representation involves the following steps:

1. the raw audio is divided into (often overlapping) chunks of small size (e.g. 25ms), and a *Discrete Fourier Transform (DFT)* is applied to each of the chunks, to produce a spectrogram of the sound at each chunk, that is, a continuous distribution of sound density across the frequency spectrum;
2. the frequency spectrum is then converted and represented in the so-called *Mel scale*, a logarithmic representation which causes the frequency space to better reflect human ear perception of frequencies;
3. a set of n triangular band-pass filters is convolved across the frequency spectrum, discretizing it into a finite number of frequencies; finally,
4. a *Discrete Cosine Transform (DCT)* is applied to the logarithm of the discretized spectrogram along the frequency axis, which compresses the spectral information at each point in time into a fixed number of coefficients.

This transformation does not modify the temporal ordering of the audio events; nevertheless, the classical approach at this point is to feed data to off-the-shelf classification methods which do not make use of such ordering (e.g., SVMs, neural networks). Moreover, step 4 does not preserve the spectral component, which makes this description not directly interpretable in terms of sound frequencies; as such, we applied MFCC up to step 3, ultimately obtaining a multivariate time series representation where the n attributes describe the power of the different sound frequencies. Furthermore, different techniques were used to clean and normalize the data prior to the MFCC step:

1. a *noise gate* filter to attenuate signals that register below a threshold to remove background noises;
2. a *peak normalization* filter where the amplitude is scaled on the highest signal level present in the recording granting consistent amplitude between audio tracks;
3. silence removal filter to remove unwanted long silences.

Additionally, to make the model invariant to different *tones*, a *pitch normalization* step was applied, where instead of the Mel scale, the frequency spectrum was represented via the *semitone scale*, which is still logarithmic, but relative to a fundamental frequency. Such a fundamental frequency for each sample was found using a *Fast Fourier Transform (FFT)* as the most prevalent frequency between 200Hz and 700Hz, which is generally accepted as appropriate for human cough in normal conditions (e.g., see Korpáš, Sadloňová, and Vrabec, 1996; Singh, Rohith, and Mittal, 2015).

Experimental settings. For each of the 30 problems described in a previous paragraph, a number of $\mathcal{H}\mathcal{S}$ -based temporal decision trees and temporal random forests were trained and evaluated via standard performance metrics for binary classification: overall accuracy (*acc*), precision (*prec*), precision (*rec*) and F1-score (*F1*). To minimise the bias, datasets were balanced by downsampling the majority class and randomly split into two (balanced) sets for training (80%) and test (20%). This process was repeated 10 times (randomized 10-folds cross-validation), and the average and standard deviation of the performance metrics were considered. Furthermore, for any training/test split, random forests were run 5 times with different initial random conditions, and their average performance was considered. As for the parametrization of random forests, after a pre-screening phase, we set $m^{sub} = 70\%$

of the cardinality of each training set (m), $k = 100$, and $n^{sub} = 50\%$ of the number of attributes (n). While experiments for single decision trees were run with a standard pre-pruning setting, that is, minimum entropy gain of 0.01 for a split to be meaningful and maximum entropy at each leaf of 0.6, random forests grow full trees without being (pre)pruned. In all cases we let $\bowtie \in \{\leq, \geq\}$. As for f , as we have already mentioned, there are many possible choices; in order to maximize the interpretability of our models, however, we used only *minimum* and *maximum*, in their *softened* version, which corresponds to the 20th and the 80th percentile, respectively; thus, $f \in \{\min_{80}, \max_{80}\}$. As for the preprocessing, the chunk size and overlap for the DFT were fixed to the standard values of 25ms and 10ms, respectively. Pre-screening was also applied to the parameters that partially drive the form of the data, that is, number of frequencies (i.e., the number of attributes n), the size of the moving average filter (w) used to lower the number of resulting points, and step of the moving window (s); as a result of such a pre-screening, we fixed $n = 30$, $w = 30$, and $s = 20$. In any case, for further data dimensionality reduction, the resulting series were capped at a maximum of 50 time points each. The pre-screening also found that noise gate, peak normalization, and silence removal were effective for cough samples, while peak normalization and silence removal were effective for breath samples. Pitch normalization proved to be effective both with cough and breath samples. Furthermore, all audio with a sample rate lower than 16000Hz were discarded.

Results. As much as the suitability of our method is concerned, let us focus on Table 5.1 and Table 5.2. As already mentioned, each row is the average of 10 executions of a specific combination of dataset settings; each performance is associated with its experimental standard deviation, for a better assessment of the solidity of the results. It is immediately clear that the datasets with segmented coughs and breath cycles perform better than the original ones; this is probably due to two aspects: first, temporal decision trees and random forests can focus on the relevant acoustic aspects of positive versus negative samples with a single episode at the time, and, second, segmented datasets are, in general, much bigger than non-segmented ones, which allowed us to train better models. We have, therefore, followed two different rules to highlight the results in Table 5.1 and Table 5.2: for the non-segmented datasets, rows with accuracies better than 85% have been highlighted, while for the segmented ones, we have focused our attention on rows with better than 95% of accuracy. A second, immediate, observation is that multi-frame learning performs decidedly better than single-frame one; this means that positive samples are more easily recognized from negative ones from a combination of (a single) breath and cough episode than they are from breath and cough separately; the performances of the models on the tasks $TA1$ and $TA2$, in particular, benefit from this approach. This is consistent with the results obtained by Brown et al. (2020). As a third consideration, we notice, as expected, that temporal random forests perform consistently better than their single tree counterpart; yet, very high accuracies are obtained with single trees in some cases. In accordance with Brown et al. (2020), augmented datasets give rise to better models, in virtually all cases. The worst results emerge from $TA3$ (arguably, the most challenging among the three problems), partly because of the intrinsic difficulty of the problem, but, most importantly, because of the small size of datasets, especially after downsampling; $TA3+$, its augmented counterpart, however, allowed us to train much better models. The best result with non-segmented datasets and single trees has been obtained precisely with $TA3+$, with an average accuracy of 90.6%; in this case, temporal random forests do not improve the accuracy

Dataset		<i>acc</i>	<i>prec</i>	<i>rec</i>	<i>F1</i>	training time	
decision tree	cough	TA1	67.8 ± 7.9	68.9 ± 9.1	66.0 ± 8.7	67.2 ± 7.9	15.89
		TA2	79.0 ± 13.7	81.4 ± 17.9	80.0 ± 13.3	79.6 ± 12.4	1.05
		TA3	60.0 ± 21.1	65.0 ± 25.4	70.0 ± 25.8	63.7 ± 17.5	0.09
		TA2+	83.3 ± 6.4	82.1 ± 7.9	86.7 ± 11.5	83.7 ± 6.5	2.89
		TA3+	90.6 ± 6.4	96.4 ± 5.9	84.5 ± 10.7	89.7 ± 7.5	1.07
	breath	TA1	68.8 ± 6.4	70.7 ± 7.5	65.2 ± 10.2	67.4 ± 7.1	110.30
		TA2	82.0 ± 11.4	87.8 ± 14.2	76.0 ± 15.8	80.5 ± 12.0	4.28
		TA3	55.0 ± 15.8	53.7 ± 26.1	60.0 ± 39.4	51.0 ± 29.1	0.10
		TA2+	82.8 ± 8.9	86.1 ± 15.6	83.3 ± 12.0	83.2 ± 7.6	10.91
		TA3+	85.0 ± 9.4	93.5 ± 8.8	75.0 ± 15.6	82.6 ± 12.2	15.51
	cough+breath	TA1	66.4 ± 6.6	67.4 ± 7.7	64.4 ± 5.8	65.8 ± 6.2	56.07
		TA2	77.0 ± 14.9	81.5 ± 18.5	74.0 ± 16.5	76.4 ± 14.1	1.09
		TA3	65.0 ± 12.9	70.0 ± 21.9	75.0 ± 26.4	67.3 ± 11.0	0.08
		TA2+	85.0 ± 8.7	86.1 ± 10.5	84.5 ± 9.4	85.0 ± 8.6	3.58
		TA3+	84.4 ± 4.4	91.7 ± 7.2	76.3 ± 9.2	82.8 ± 5.5	6.11
random forest	cough	TA1	76.6 ± 7.5	79.4 ± 9.2	72.4 ± 7.9	75.6 ± 7.7	1,654.57
		TA2	83.4 ± 12.0	85.3 ± 13.9	83.6 ± 18.0	83.2 ± 12.6	53.34
		TA3	70.5 ± 18.8	79.3 ± 23.0	70.0 ± 35.0	66.5 ± 28.2	6.28
		TA2+	88.7 ± 6.3	91.9 ± 6.9	85.3 ± 10.5	88.1 ± 6.9	247.03
		TA3+	89.2 ± 6.8	96.3 ± 5.8	81.6 ± 11.3	87.9 ± 8.3	96.57
	breath	TA1	74.5 ± 6.5	76.3 ± 8.7	72.3 ± 6.6	74.0 ± 6.0	5,184.86
		TA2	84.0 ± 9.7	88.9 ± 12.2	80.0 ± 18.9	82.7 ± 11.3	238.98
		TA3	62.0 ± 21.1	63.0 ± 32.0	63.0 ± 33.4	59.7 ± 26.7	80.99
		TA2+	87.9 ± 5.7	91.9 ± 7.7	84.0 ± 10.6	87.2 ± 6.3	1,079.55
		TA3+	94.5 ± 4.1	98.9 ± 3.5	90.3 ± 8.9	94.0 ± 4.7	598.27
	cough+breath	TA1	74.8 ± 6.4	76.1 ± 7.8	73.0 ± 7.0	74.3 ± 6.2	5,355.72
		TA2	84.2 ± 9.9	87.3 ± 10.1	81.2 ± 17.7	83.0 ± 11.4	170.95
		TA3	69.5 ± 19.1	76.0 ± 26.1	69.0 ± 25.1	68.6 ± 18.5	22.40
		TA2+	89.8 ± 5.5	93.5 ± 5.4	85.8 ± 10.5	89.1 ± 6.3	1,048.96
		TA3+	89.5 ± 7.1	95.4 ± 6.4	83.3 ± 11.3	88.5 ± 8.1	667.19

TABLE 5.1: Cross-validated results on five non-segmented datasets for COVID-19 diagnosis, using different approaches based on temporal decision trees and temporal random forests. For each approach, the average and the standard deviation across 10 repetitions of the following performance metrics are reported: overall accuracy, mean precision, mean recall, and F1 score. Results are reported in percentage points and, for each dataset, the average performance of the best decision tree and the best random forest approach is highlighted. Average training time in seconds is also reported.

(best accuracy in this case is 89.2%). As for segmented datasets, the best result with single trees is a very notable 98.8%, obtained in *TA2*, with only 2% of standard deviation over 10 executions. With temporal random forests, *TA2+* allowed us to obtain an astonishing 99.4% of averaged accuracy, 1% of standard deviation, which is the best-known performance of a COVID-19 acoustic diagnostic system, across all existing datasets and learning methods in the segmented, multi-frame setting; however, temporal random forest models have accuracies better than 96% for all tasks, except for *TA3*, which is an indication of the reliability of this method. The multi-frame approach, as it can be seen, has also the effect of lowering the standard deviation across the different sets of experiments, at least in the segmented case.

Comparison with state-of-the-art competitors. Because COVID-19 is a humankind threat there has been an enormous effort by researchers to tackle any problem related

Dataset		<i>acc</i>	<i>prec</i>	<i>rec</i>	<i>F1</i>	training time	
decision tree	cough	TA1	72.8 ± 7.6	75.0 ± 8.0	68.4 ± 9.4	71.4 ± 8.3	4.22
		TA2	91.0 ± 9.9	96.7 ± 10.5	86.0 ± 13.5	90.3 ± 10.2	0.18
		TA3	72.5 ± 24.9	79.6 ± 26.1	70.0 ± 35.0	68.3 ± 31.8	0.01
		TA2+	93.2 ± 8.4	92.3 ± 8.0	94.6 ± 11.5	93.1 ± 9.0	0.40
		TA3+	90.0 ± 5.3	92.4 ± 6.6	87.5 ± 8.3	89.7 ± 5.7	0.22
	breath	TA1	74.3 ± 2.9	74.6 ± 4.1	74.1 ± 3.4	74.3 ± 2.5	122.47
		TA2	84.0 ± 5.4	84.0 ± 6.2	84.7 ± 10.9	83.9 ± 5.9	3.54
		TA3	61.7 ± 19.3	61.0 ± 29.5	66.7 ± 35.1	59.9 ± 26.6	0.23
		TA2+	88.2 ± 4.3	91.3 ± 3.8	84.6 ± 8.8	87.5 ± 5.3	13.50
		TA3+	91.9 ± 6.7	98.4 ± 3.4	85.4 ± 13.3	90.8 ± 8.5	2.78
	cough+breath	TA1	95.9 ± 1.5	96.5 ± 1.7	95.3 ± 2.1	95.9 ± 1.5	82.65
		TA2	98.8 ± 1.9	100.0 ± 0.0	97.7 ± 3.7	98.8 ± 1.9	1.10
		TA3	90.0 ± 17.5	92.6 ± 14.7	90.0 ± 31.6	86.0 ± 31.3	0.01
		TA2+	97.8 ± 1.7	98.5 ± 1.1	97.0 ± 3.9	97.7 ± 1.8	14.08
		TA3+	84.4 ± 18.0	87.1 ± 21.3	86.3 ± 15.0	85.6 ± 16.0	0.14
random forest	cough	TA1	80.4 ± 5.7	84.0 ± 5.6	75.2 ± 8.0	79.2 ± 6.3	136.81
		TA2	92.4 ± 6.9	99.3 ± 2.1	85.6 ± 13.1	91.4 ± 8.1	2.06
		TA3	73.5 ± 23.2	78.0 ± 25.2	77.0 ± 24.1	74.6 ± 20.2	0.15
		TA2+	95.5 ± 4.4	98.5 ± 2.8	92.6 ± 9.5	95.1 ± 5.2	5.94
		TA3+	92.9 ± 7.5	100.0 ± 0.0	85.8 ± 14.9	91.5 ± 9.7	2.94
	breath	TA1	81.9 ± 2.2	84.0 ± 3.1	79.0 ± 2.7	81.4 ± 2.2	8,271.27
		TA2	86.7 ± 6.7	91.5 ± 7.2	82.3 ± 15.6	85.4 ± 9.3	124.35
		TA3	66.3 ± 12.4	68.1 ± 19.3	67.3 ± 30.9	63.7 ± 18.7	5.03
		TA2+	90.5 ± 2.5	95.7 ± 2.8	84.9 ± 5.6	89.9 ± 3.0	947.30
		TA3+	92.0 ± 8.2	99.9 ± 0.5	84.2 ± 16.5	90.5 ± 10.6	101.68
	cough+breath	TA1	98.0 ± 0.5	99.4 ± 0.6	96.7 ± 1.1	98.0 ± 0.5	8,717.38
		TA2	97.2 ± 3.1	100.0 ± 0.0	94.5 ± 6.3	97.0 ± 3.4	28.56
		TA3	86.5 ± 15.6	93.7 ± 8.6	81.0 ± 32.5	81.5 ± 27.6	0.67
		TA2+	99.4 ± 0.5	99.0 ± 0.9	99.9 ± 0.4	99.4 ± 0.5	905.60
		TA3+	96.1 ± 5.4	100.0 ± 0.0	92.3 ± 10.8	95.6 ± 6.2	21.10

TABLE 5.2: Cross-validated results on five segmented datasets for COVID-19 diagnosis, using different approaches based on temporal decision trees and temporal random forests. For each approach, the average and the standard deviation across 10 repetitions of the following performance metrics are reported: overall accuracy, mean precision, mean recall, and F1 score. Results are reported in percentage points and, for each dataset, the average performance of the best decision tree and the best random forest approach is highlighted. Average training time in seconds is also reported.

to it. We, however, considered the problem of COVID-19 diagnosis and many others have done the same; thus, we briefly discuss the main aspects of such proposals (see Manzella et al., 2022 for an in-depth analysis and discussion). Many datasets related to the COVID-19 diagnosis have been created and collected. Such datasets are either public or private; the latter may be due to law-compliance issues (e.g., data anonymity). The majority of the results are based on neural networks because there are many ready-to-use deep learning frameworks. Non-ML experts need to be made aware that symbolic approaches are easier to train and are often preferable over the state-of-the-art deep learning ones which, on the other hand, are harder to train and require an intensive hyper-parametrization. Cough recordings are preferred over breath ones, while others use also speech recordings; nevertheless, just few address the problem by jointly learning to diagnose from both cough and breath recordings. With respect to dataset the that we used for our experiments, few researches addressed the tasks TA2 and TA3 preferring their augmented versions, namely, TA2+

and TA3+. In any case, our methodology is robust in terms of performance metrics and provides interpretable formulas that may be interesting for medical personnel and physicians in their work. Indeed, we found out that our (innovative) methodology is superior to traditional ones, both symbolic and non, applied to the same data while allowing the interpretation of the results and enabling visualization (and transformation into audible sounds) of models that enclose the distinguishing characteristics of cough/breath samples of positive (and negative) subjects.

§5.7 Real-world Data Experiments: Spatial Data

As a second example of real-world data application, we consider the spatial case where we instantiate our framework with \mathcal{HS}^2 , and, in particular, where the relations are the eight Egenhofer and Franzosa’s topological relations, and their coarse version having only five relations, denoted by \mathcal{HS}_{RCC8}^2 and \mathcal{HS}_{RCC5}^2 , respectively. Here, we choose spatial modal decision trees and propositional decision trees at different degrees of experimental settings. Pagliarini and Sciavico (2021) have published these results.

Datasets. We consider five datasets for the problem of *land cover classification (LCC)*, that is, the problem of classifying pixels in remotely sensed images, associating each pixel with a label that captures the use of the land it depicts (e.g., forest, urban area, crop field), typically known as *Indian Pines*, *Pavia University*, *Pavia Centre*, *Salinas*, and *Salinas-A*, respectively. Each dataset consists of a hyperspectral image, or *scene*, of a piece of land, coupled with a *ground-truth label mask*, providing the correct class for some of the pixels in the scene. In all cases, the scene is captured by a dedicated sensor during a flight campaign: specifically, Pavia University and Pavia Centre were collected using a ROSIS sensor (Reflective Optics System Imaging Spectrometer), and the remaining ones using an AVIRIS sensor (Airborne Visible/Infrared Imaging Spectrometer). The ROSIS and AVIRIS yield spectral detections covering a range of frequencies from $0.43\mu\text{m}$ to $0.86\mu\text{m}$ and from $0.4\mu\text{m}$ to $2.45\mu\text{m}$, with a number of channels of 103 and 200, respectively. The size of the scenes varies from 86×83 pixels for Salinas-A to 1096×1096 pixels for Pavia Centre; note, however, that not all pixels are labelled.

Preprocessing techniques. The typical approach to LCC involves collecting either all labelled pixels, or a randomly sampled subset, and applying a learning algorithm for multiclass classification. In some cases, authors have used the spatial structure in a simple way, and considered, for the classification of every single pixel, a set of neighbouring pixels; the neighbourhood is generally in the form of a $N \times N$ window centred in the pixel, for a natural odd number N . We adopted the same solution, by extracting, for each dataset, a collection of $N \times N$ labelled images (one for each labelled pixel). All five datasets present a class imbalance, with some classes appearing thousands of times, while others only appear a few dozen times. To level out this imbalance, which can easily cause biases towards the most occurring classes, a fixed number P of pixels is randomly sampled for each class, discarding classes with less than P samples. The experiments were carried out in a randomized cross-validation setting, where, this sampling step is performed 10 times, and each time the set is partitioned into two balanced sets, one for training the model, and the other for testing and evaluating it. In this round of experiments, we set $N = 3$ and $P = 100$, for each dataset, 10 balanced (sub)datasets of 3×3 images, that is, 3×3 images are extracted

from the original image; 4 classes with less than 100 samples were discarded from Indian Pines so that the resulting subdatasets only encompassed 12 of the 16 original classes. Furthermore, an 80%–20% balanced split was performed when splitting each subdataset into training and test, and a policy for keeping them *strongly disjointed* was implemented. Such a policy prevents any two images with non-empty overlap on the original scene to end up on different sides of the training/test border, to avoid the so-called *data leakage* phenomenon (i.e., when data leakage occurs, part of the information in the test set appears in the training set as well, biasing the algorithm and, ultimately, affecting the performance estimation).

Experimental settings. For each training-test split, different approaches to decision tree modelling are deployed and compared. We separate the approaches to symbolic learning for LCC into *pure* approaches, where the tree is applied on the 3×3 image itself, and *derived* ones, where the input image is processed beforehand using functional filters. The pure approaches included in this experiment are referred to as *single-pixel* propositional approach, in which standard, propositional decision trees are applied to the numerical description of the pixel to be classified (i.e., the central pixel), disregarding the neighboring pixels, *flattened* propositional approach, in which standard decision trees are applied to the numerical descriptions of all pixels in the 3×3 image, disregarding the spatial structure, and *RCC8* (resp., *RCC5*) modal approaches, in which spatial (modal) decision trees with \mathcal{HS}_{RCC8}^2 (resp., \mathcal{HS}_{RCC5}^2) is applied to the 3×3 image, and the image channels are regarded as spatial variables; each of the above settings has a corresponding derived one, obtained by applying, before training, a 3×3 average convolutional filter. In derived settings we first used the outer 5×5 box around each 3×3 image to compute a 3×3 *average image* which is, then, fed *as is* to the learning algorithm in pure case; in this way, each pair of approaches is immediately comparable. To fix the ideas, consider the case of *Indian Pines*, which has 200 channels. In the pure single-pixel approach we trained decision trees that take decisions on the 200 featured values of the central pixel; on the other hand, in the corresponding derived setting (referred to as *avg*), the trees learn from instances described by 200 values that are, each, the average of a channel with the 3×3 image. Instead, trees trained with the pure flattened approach take decisions on the $3^2 \times 200 = 1800$ featured values within the 3×3 image, while the corresponding derived setting (*avg+flattened*) uses $3^2 \times 200 = 1800$ values, that are the result of the 3×3 average convolution performed within the 5×5 outer box. As for the spatial approaches, they are always applied on 3×3 images, except that in the derived cases (*avg+RCC8* and *avg+RCC5*) the image is the result of a convolution. Note that, because LCC is invariant under rotation and reflection, spatial approaches are more likely to grasp complex patterns when using topological relations, as opposed to directional ones. At the code level, each approach differs from the others by how data are preprocessed before the learning phase, and by the algorithm parametrization: when applied to datasets of 1×1 images, spatial decision trees behave exactly as traditional ones, so that the implementation used in this experiment is the same in the traditional and the spatial case. With regards to pre-pruning, different parametrizations were tested using the single-pixel baseline approach, and the one achieving the highest cross-validation performance was, finally, fixed. The fixed pre-pruning conditions are the minimum number of samples per leaf of 4, minimum information gain of 0.01 for a split to be meaningful, and maximum entropy at each leaf of 0.3. Each model was evaluated using standard performance metrics for multiclass classification, namely, *overall accuracy*, κ *coefficient* (which relativizes the accuracy to the probability of a random

		Dataset	κ	<i>acc</i>	<i>prec</i>	training time
derived	avg	Indian Pines	71.2 ± 2.9	73.6 ± 2.7	74.5 ± 2.6	43.75
		Pavia University	75.9 ± 2.5	78.6 ± 2.2	79.1 ± 2.5	14.06
		Pavia Centre	89.5 ± 1.0	90.7 ± 0.9	91.1 ± 1.1	11.34
		Salinas	92.2 ± 1.7	92.7 ± 1.6	93.0 ± 1.5	61.66
		Salinas-A	94.9 ± 2.0	95.8 ± 1.6	95.9 ± 1.6	8.49
	avg+flattened	Indian Pines	66.1 ± 3.0	68.9 ± 2.7	69.9 ± 2.7	239.04
		Pavia University	61.2 ± 5.4	65.5 ± 4.8	66.8 ± 4.6	137.82
		Pavia Centre	64.1 ± 5.3	68.1 ± 4.7	69.6 ± 4.5	248.32
		Salinas	95.5 ± 1.3	95.8 ± 1.2	96.0 ± 1.1	379.55
		Salinas-A	67.8 ± 4.7	73.2 ± 3.9	74.4 ± 4.0	446.97
	avg+RCC8	Indian Pines	77.0 ± 2.0	78.9 ± 1.9	79.7 ± 1.8	7,644.26
		Pavia University	80.0 ± 2.4	82.2 ± 2.1	82.8 ± 2.2	3,004.28
		Pavia Centre	89.6 ± 2.2	90.7 ± 1.9	91.3 ± 1.8	4,147.41
		Salinas	92.8 ± 1.7	93.2 ± 1.6	93.5 ± 1.5	6,381.60
		Salinas-A	98.3 ± 1.0	98.6 ± 0.8	98.7 ± 0.7	1,790.53
avg+RCC5	Indian Pines	77.7 ± 2.3	79.6 ± 2.1	80.1 ± 2.2	6,154.47	
	Pavia University	80.1 ± 2.8	82.3 ± 2.5	82.9 ± 2.4	3,837.40	
	Pavia Centre	89.4 ± 2.3	90.6 ± 2.0	91.1 ± 1.9	2,788.61	
	Salinas	92.3 ± 1.5	92.8 ± 1.4	93.0 ± 1.4	7,621.10	
	Salinas-A	98.3 ± 1.0	98.6 ± 0.8	98.7 ± 0.7	4,310.79	
pure	single-pixel	Indian Pines	62.7 ± 2.6	65.8 ± 2.3	66.3 ± 2.8	29.25
		Pavia University	70.9 ± 3.2	74.1 ± 2.9	74.6 ± 3.1	12.09
		Pavia Centre	86.4 ± 4.1	87.9 ± 3.6	88.5 ± 3.4	9.39
		Salinas	89.1 ± 1.9	89.8 ± 1.8	90.3 ± 1.8	31.20
		Salinas-A	94.9 ± 2.0	95.8 ± 1.7	95.9 ± 1.7	4.55
	flattened	Indian Pines	59.0 ± 4.2	62.4 ± 3.8	63.6 ± 3.5	173.49
		Pavia University	42.5 ± 6.5	48.9 ± 5.8	49.8 ± 6.0	115.18
		Pavia Centre	40.4 ± 4.2	47.0 ± 3.7	48.2 ± 3.4	112.28
		Salinas	94.8 ± 1.4	95.1 ± 1.4	95.4 ± 1.2	149.81
		Salinas-A	64.8 ± 6.3	70.7 ± 5.2	71.8 ± 5.0	39.67
	RCC8	Indian Pines	73.2 ± 4.1	75.4 ± 3.8	76.2 ± 3.8	1,034.12
		Pavia University	77.9 ± 1.5	80.4 ± 1.3	81.1 ± 1.2	599.83
		Pavia Centre	89.1 ± 2.8	90.3 ± 2.5	90.8 ± 2.4	518.45
		Salinas	90.5 ± 1.5	91.1 ± 1.4	91.5 ± 1.4	1,811.41
		Salinas-A	96.0 ± 1.5	96.7 ± 1.2	96.9 ± 1.1	428.22
RCC5	Indian Pines	74.5 ± 4.5	76.6 ± 4.1	77.4 ± 4.0	1,795.79	
	Pavia University	77.7 ± 1.5	80.2 ± 1.3	81.0 ± 1.1	729.35	
	Pavia Centre	88.9 ± 3.0	90.2 ± 2.7	90.8 ± 2.6	665.75	
	Salinas	90.4 ± 1.5	91.0 ± 1.4	91.4 ± 1.5	2,291.02	
	Salinas-A	96.0 ± 1.5	96.7 ± 1.2	96.9 ± 1.1	562.50	

TABLE 5.3: Cross-validation results on five datasets for land cover classification, using different approaches based on propositional and spatial decision trees. For each approach, the average and the standard deviation across 10 repetitions of the following performance metrics are reported: κ coefficient, overall accuracy, and mean precision. Results are reported in percentage points and, for each dataset, the average performance of the best pure approach and the best derived approach is highlighted. Average training time in seconds is also reported.

answer being correct (Cohen, 1960), and *mean precision*. Note that with balanced test sets the overall accuracy also corresponds to the *mean recall*.

Results. Table 5.3 shows the results of pure and derived approaches applied to the five datasets; for each approach and dataset, the average and standard deviation of κ coefficient, overall accuracy and mean precision are reported. The discussion that

follows is mainly based on the κ coefficient, but the other two metrics reveal similar insights. The results show that pure approaches always attain lower performances than their derived counterpart, with the only exception of Salinas-A, where the pure single-pixel approach and the avg approach yield the same average performance. For each dataset, the best performances in the pure and derived cases are generally attained by spatial approaches. On the contrary, the worst performances are always attained by either single-pixel approaches or flattened ones. Except for Salinas, which does not seem to follow this general schema, flattened approaches appear to cause a degradation of performance: when compared to single-pixel approaches, the degradation ranges from about 4 percentage points (Indian Pines) up to 30 percentage points (Salinas-A). Instead, Salinas is the only dataset showing a clear preference for the flattened approaches which, concerning the spatial approaches, improve κ from 90.5% to 94.8% in the pure case, and from 92.8% to 95.5% in the derived case. Altogether, the results show that across four out of five datasets, spatial approaches consistently yield better results than propositional ones. The improvements seem to be proportional to the intrinsic hardness of each classification dataset; compared with the propositional approach, the average improvement of spatial decision trees ranges from about 1 percentage point (Salinas-A) to 11 percentage points (Indian Pines). From a qualitative perspective, there does not seem to be a clear winner between the two topological logics: spatial decision trees with RCC8 and RCC5 behave quite similarly, with the greatest difference in terms of κ being as low as 1.3 percentage points (Indian Pines, pure spatial approaches).

Comparison with state-of-the-art competitors. As we shall see in Chapter 7, learning from spatial data is generally preferred with neural networks; in particular, with convolutional neural networks. Indeed, the datasets that we considered are commonly used to benchmark neural network-based methods for LCC (e.g., see Hu et al., 2015; Mou, Ghamisi, and Zhu, 2017; Roy et al., 2020; Li, Zhang, and Shen, 2017; Lee and Kwon, 2017; Hong et al., 2022; Audebert, Le Saux, and Lefèvre, 2019; Cao et al., 2018; Jiang et al., 2019; Santara et al., 2017). A proper literature review reveals how, since the beginning, due to a declared need for explicit classification rules, the task has been frequently addressed using propositional symbolic learning (e.g., see Goel et al., 2003; Zhang and Wang, 2003). This suggests how, despite of their benefits in terms of transparency, the inability of known symbolic learning algorithms to deal with complex data caused researchers and practitioners to favour higher statistical performances. But, in a way, while the life of black box models ends with their statistical performances, that of symbolic ones starts with it, and symbolic models enable a continuous interaction between artificial and human intelligence; in this sense, symbolic learning is still a relatively poorly understood field. Note that although these datasets are commonly used in literature to evaluate non-symbolic methods, none of these methods addresses interpretability matters, and thus, a comparison against these methods makes little sense. In a similar way, the existing symbolic methods used for the task are generally not spatial in nature, except for the work by Jiang et al. (2012b) for which, however, data are not available. Therefore, canonical symbolic approaches are less adopted because, for example, they are less resilient to symmetries in spatial data for which convolutional neural networks, and the like, are outstanding. Nevertheless, our methodology is mature enough to learn robust classifiers from which interpretable formulas can be extracted for further investigation.

EXTENSIONS

There is nothing permanent except change.

—Heraclitus

This chapter briefly discusses some possible extensions of the modal symbolic learning framework. Such extensions are not meant to be complete but should guide the eager symbolic ML practitioners in such an exciting field. The following ideas, except for the last one which is an inspiration from the field of deep learning, are known at the propositional level, and the objective is to lift them at the modal level.

§6.1 Neural-Symbolic Modal Decision Trees

Decision trees and neural networks are well-known alternatives for pattern recognition, and their strengths and weaknesses have been studied for over three decades (e.g., see Atlas et al., 1989; Shavlik, Mooney, and Towell, 1991). As suggested in the literature (e.g., see d’Avila Garcez et al., 2019; d’Avila Garcez, Lamb, and Gabbay, 2009; Minsky, 1991, among others), to solve the symbolic versus non-symbolic ML duality, one can think of an *hybrid* approach: hybrid systems combine the strengths of both symbolic and non-symbolic methods, to guarantee high degrees of interpretability of the learned models, while retaining high enough statistical performances. Notoriously, decision trees favour the interpretability of their decisions, which, due to their symbolic nature, represent coarse concepts in numeric domains, whereas neural networks are difficult to interpret, but have a better generalization capability.

Let us focus on the literature concerning the hybridization of these two models. Neural networks can be initialized from decision trees (e.g., see Sethi, 1990; Brent, 1991; Ivanova and Kubat, 1995; Setiono and Leow, 1999; Kubat, 1998). Decision trees can be initialized by neural networks (e.g., see Craven and Shavlik, 1995; Krishnan, Sivakumar, and Bhattacharya, 1999; Schmitz, Aldrich, and Gouws, 1999; Dancey, McLean, and Bandar, 2004; Zhou and Jiang, 2004). Finally, hybrid neural-symbolic decision trees can be learned (e.g., see Li, Fang, and Jennings, 1992; Guo and Gelfand, 1992; Setiono and Liu, 1999; Zhou and Chen, 2002; Micheloni et al., 2012; Srivastava and Salakhutdinov, 2013; Hinton, Vinyals, and Dean, 2015; Kotschieder et al., 2015; Murthy et al., 2016; Murdock et al., 2016; Alaniz et al., 2021; Wan et al., 2021).

In an attempt at taxonomizing the existing neural-symbolic work on decision trees, one could argue that there are at least three independent parameters that can be combined. First, the possibility of using a network for initial screening of the dataset, to be later dealt with in a more precise way by one of several potentially different trees (*root hybridization*). Second, the possibility of querying an external network as a feature extractor to take split decisions in a single tree (*split hybridization*).

Third, the possibility of consulting one of several different networks at the leaves of a decision tree before deciding the class (*leaf hybridization*). When the underlying neural networks' architectures are alike, different hybridization types become comparable as well. Therefore, *neural-symbolic modal decision trees* are possible since we can exploit any of the above hybridization proposals.

§6.2 Fuzzy Modal Decision Trees

Propositional fuzzy (or many-valued) logics (from the early work of Łukasiewicz, Post, and Tarski) extend Boolean \mathcal{PL} by allowing more than two truth values (Hájek, 2013). Fuzzy modal logics were introduced by Fitting (1991) and have enjoyed sustained attention in recent years (e.g., see Bou et al., 2011; Caicedo and Rodríguez, 2010; Hájek, 2005; Vidal, Esteva, and Godo, 2017). Fitting, in particular, gives a very general approach to fuzzy modal logic in which propositions and accessibility relations are not just true or false but may take different truth values.

Fuzzy logics ameliorate some of the shortcomings of crisp, Boolean logics. In ML terms, the fuzzy framework can be used in the decision tree technique to manage fuzzy information (e.g., fuzzy inputs, fuzzy classes, fuzzy rules) and improve the resulting models' predictive capability. The term fuzzy decision tree was coined by Chang and Pavlidis (1977), and since then, many other contributions have been made to the literature. The constants a in propositional decisions (recall Definition 3.2) and the final decisions (in the leaves) can be softened in a probabilistic way without the usage of so-called fuzzy sets (e.g., see Carter and Catlett, 1987; Jordan, 1994). The crisp rules extracted from a crisp propositional decision tree can be fuzzified to obtain fuzzy rules (e.g., see Tani, Sakoda, and Tanaka, 1992; Jang, 1994; Chi and Yan, 1996). Other proposals start by learning a crisp propositional decision tree structure and then search the degree of softness in every node to fuzzify the original tree (e.g., see Jeng, Jeng, and Liang, 1997; Suárez and Lutsko, 1999). Moreover, a fuzzy decision tree can be learned directly by integrating fuzzy techniques in the learning process (e.g., see Wang and Suen, 1987; Cios and Sztandera, 1992; Yuan and Shaw, 1995; Ichihashi et al., 1996; Tsuchiya et al., 1996; Apolloni, Zamponi, and Zanaboni, 1998; Hayashi et al., 1998; Janikow, 1998; Boyen and Wehenkel, 1999; Wang et al., 2000; Tsang, Wang, and Yeung, 2000; Olaru and Wehenkel, 2003).

Bringing together the two schools of thought, namely, fuzzy modal logics and fuzzy propositional decision trees, we can elegantly obtain, thanks to our framework, *fuzzy modal decision trees* to learn more realistic modal patterns from complex real-world scenarios.

§6.3 Gradient-boosted Modal Decision Trees

Boosting is a sequential process of improving weak learners, where each subsequent model corrects the errors of the previous model, trying to converge to an optimal metamodel. Gradient boosting is the extension of boosting that sequentially fits the negative gradients (Friedman, 2000). Typical boosting involves a regression task, although it can also be used for classification with minor adjustments.

For any differentiable loss function a different gradient boosting algorithm can be defined, such as *AdaBoost* (Freund and Schapire, 1996), *stochastic gradient boosting* (Friedman, 2002), *eXtreme Gradient Boosting (XGBoost)* (Chen and Guestrin, 2016) and *Light Gradient-Boosted Machines (LightGBM)* (Ke et al., 2017). Each weak learner

can be any model, but propositional decision trees are preferred in the literature, and this, again, justifies our choice of decision trees as representatives of symbolic learning. We can, thus, exploit the learning mechanisms of such contributions, and their relatively efficient implementations, by simply replacing propositional with modal decision trees to obtain *gradient-boosted modal decision trees*. In general, moving from simple decision trees to random forests and then to boosted trees, reduces the interpretability of the final model, but increases its performances (Hastie, Tibshirani, and Friedman, 2009); nevertheless, there are ways to recover the interpretability of the model (e.g., see Deng, 2019 for the case of random forests).

§6.4 Incremental Modal Decision Trees Learning

In ML terms, *concept drift* means that the statistical properties of the target value that the trained model tries to predict change over time as new observations arrive, and the performances of the model deteriorate over time as chances of misprediction increase. For example, imagine a situation where a validated model in production works well, then a sudden unforeseen situation arises (e.g., a pandemic like COVID-19), and the model starts to be unreliable. Deployed ML models face such situations periodically and must be revalidated to assess their performances over time for critical tasks. There are at least two ways to address such an issue. On the one hand, the model can be *retrained* periodically (e.g., during each weekend) or if triggered by some user-defined condition (e.g., the accuracy goes below a certain threshold), but training is costly, depending on the volume of the data. *Incremental learning* (or, alternatively, *online learning*), on the other hand, integrates into the original model the information enclosed in the new observations in the original model as they arrive without retraining on past instances; therefore, training, in this case, is more feasible.

Typically, ML models are *batch* trained, where the learning set is fixed beforehand, and models are learned from it; this kind of setting is also known as *non-incremental learning* (or *offline learning*). In incremental learning, a sequence of instances is observed, one at a time, which might not be equally spaced in a time interval, and a trained model is incrementally updated with the information contained in such instances. *Online modal decision tree* learning can be achieved by exploiting the known results for the same problem at the propositional level (e.g., see Schlimmer and Fisher, 1986; Utgoff, 1988; Crawford, 1989; Utgoff, 1989; Utgoff, 1994; Utgoff, Berkman, and Clouse, 1997; Domingos and Hulten, 2000; Hulten, Spencer, and Domingos, 2001; Gama, Fernandes, and Rocha, 2006; Manapragada, Webb, and Salehi, 2018, among others).

§6.5 Geometric Modal Symbolic Learning

Geometric deep learning is a recent exciting research line in the deep learning realm that tries to give a unifying view to the zoo of deep learning approaches in the literature (Bronstein et al., 2017). The general idea of geometric deep learning is the study of symmetries in data and injecting of such symmetries in the learning pipeline as inductive biases. As we have recalled in our introduction, deep learning architectures excel at learning from unstructured data as they exploit inductive biases. From a philosophical point of view, deep learning could have been called *connectionist learning*, to be opposed to symbolic learning, as the subset of ML that designs connectionist algorithms (i.e., neural networks) to build models from data. In the same

spirit, we could have called modal symbolic learning as *modal learning* since we also address the problem of learning from unstructured data, but it is just a matter of taste.

Geometric modal symbolic learning is a research effort similar to geometric deep learning: symmetries can be studied also in the case of Kripke models. In the field of modal logics, symmetries have been studied related to the model checking problem, called *symmetry-based model checking* (e.g., see Ip and Dill, 1996; Clarke et al., 1996; Emerson and Sistla, 1996; Sistla, Gyuris, and Emerson, 2000), which, recall, is at the core of learning any modal logic theory. Regarding modal symbolic learning, we must discuss some important aspects of symmetry-based model checking. First, learning is asymptotically more efficient since model checking requires less time in smaller, but bisimilar to the original, models, and this ameliorates the polynomial blow-up when transforming an input signal to a Kripke model. Second, state-of-the-art deep learning architectures exploit the inductive biases by exploiting symmetries (e.g., shift-invariant and rotation-invariant patterns in images), and symmetry-based model checking for modal symbolic learning goes in the same direction; we can, thus, obtain better generalized modal symbolic models. Finally, logicians need to study symmetry-based model checking for other logics, such as \mathcal{HS}^d , to advance the discipline of geometric modal symbolic learning.

RELATED WORK

Sometimes you put walls up not to keep people out, but to see who cares enough to break them down.

—Socrates

ML is a thrilling field. The literature is widespread on the argument, and we must find a degree of comparability with our work. We must, therefore, restrict our horizon in the *mare magnum* of contributions that are present today in the literature. Decision trees have been used as emblematic for the entire symbolic learning framework, and we must dedicate some time to discussing their history. Then we must review the contributions of temporal and spatial learning for completeness. We do so by reviewing the proposals for temporal and spatial data that are not directly related to our framework, and finally, we discuss symbolic approaches for similar tasks.

§7.1 Brief History on Propositional Decision Trees

Figure 7.1 illustrates the reviewed literature on the history of propositional decision trees. The origin of the development of modern decision trees dates back to Belson (1956) with his seminal work, which serves as a precursor to a new line of decision tree development that employs ML algorithms to produce *executable* rules. Based on Belson's work, Morgan and Sonquist (1963) proposed *Automatic Interaction Detection (AID)* as an alternative to functional regression (i.e., regression with decision trees). Like the earlier approaches, *Concept Learning System (CLS)* (Hunt, Stone, and Marin, 1966) works progressively with recursion partitioning the input data based on highly discriminating variables. Whereas AID is used for regression tasks, *THeta Automatic Interaction Detection (THAID)* (Messenger and Mandell, 1972), which is the first implementation of a decision tree for classification, and *CHI-squared Automatic Interaction Detection (CHAID)* (Kass, 1980) extend AID for classification tasks by introducing new impurity information-based functions as entropy or Gini index needed to partition the dataset in a node. The main issue with AID-based approaches is overfitting, which prompted the scientific community to gain interest in investigating further.

Classification And Regression Trees (CART) (Breiman et al., 1984) were iconic in regenerating interest in the subject. In particular, the method follows the same greedy approach as the AID-based methods but adds several features; for example, *pruning techniques* to regularise the resulting model, such as *Reduced Error Pruning (REP)* (Elomaa and Kaariainen, 2001), to cope with the overfitting problem. Quinlan (1986) entered this field with an ML perspective formalizing the development of an inductive process for *knowledge acquisition*, which resulted in the so-called *Iterative Dichotomizer 3 (ID3)* algorithm, that has been extended with pruning techniques

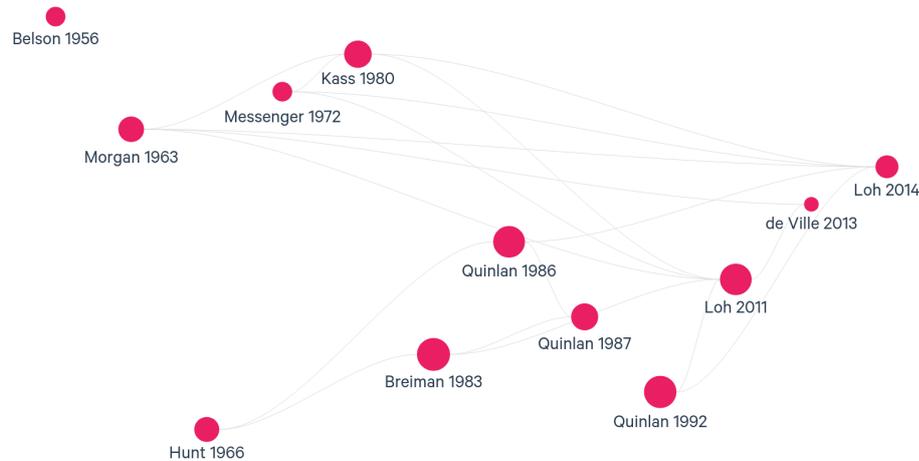


FIGURE 7.1: Literature map on the history on propositional decision trees (generated with [litmaps](#)).

some years later by the same author (Quinlan, 1999). ID3 was a breakthrough for the *rule induction* knowledge acquisition process: each decision can be expressed as a propositional letter; since alternative branches can be seen as logical disjunctions, and successive decisions on the same branches can be seen as logical conjunctions, a decision tree as a whole can be seen as a set of \mathcal{PL} formulas. As we have seen, a decision tree is a propositional description of the dataset on which it is learned, representing the theory underlying the given task. Moreover, Quinlan compared different entropy-based splitting criteria, namely, *information gain* and *gain ratio*. Subsequently, the same author developed the *C4.5* algorithm (Quinlan, 1993) to cope with the main limitation of ID3 of handling only categorical data.

With the dawn of the era of big data, other, more sophisticated learning approaches have been proposed to handle unstructured data, leaving, in a way, decision tree development behind. As such, there are few decision tree-like proposals in the literature for the need for more expressive decision trees. Finally, there are more exhaustive, systematic surveys on the history of the development of decision trees for either classification or regression (e.g., see de Ville, 2013; Loh, 2011; Loh, 2014).

§7.2 Approaches for Learning from Temporal Data

In the temporal case, the most representative objects are time series since, to some extent, every other temporal object can be seen as one. Classification, on the other hand, as we have discussed, is the cornerstone of every learning task because it provides the principles of ML. Therefore, we discuss the literature on time series classification summarized in Figure 7.2.

Classification of time series can be *distance-based* (i.e., based on the notion of distance/similarity between series) or not, its underlying ontology can be *point-based* or *interval-based*, and the method itself can be *feature-based* (i.e., based on the notion of

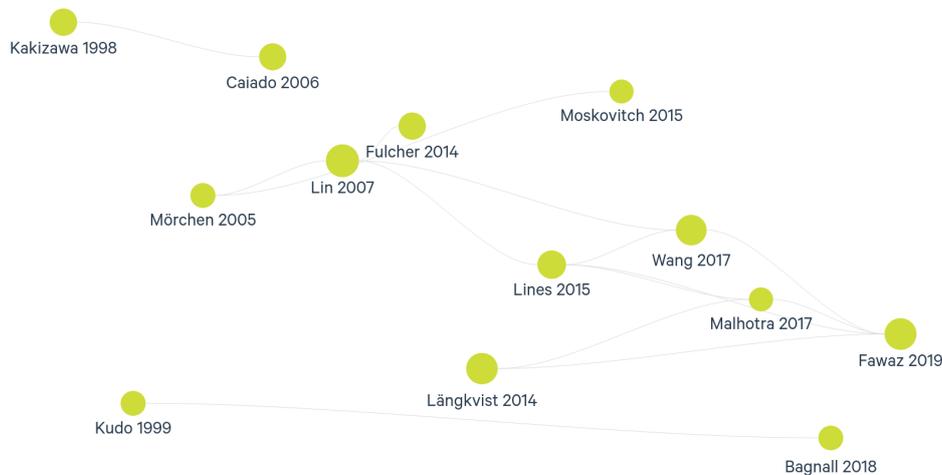


FIGURE 7.2: Literature map for approaches for time series classification (generated with [litmaps](#)).

extracting features from the series and then using some off-the-shelf, non-temporal classifier) or not. Finally, a time series classification method may or may not require a *transformation* of the raw data. The plethora of existing methods cannot be immediately partitioned into a taxonomy because, first, many proposals present different combinations of these characteristics, and, second, there may be other dimensions that are not adequately captured by our list.

Transforming a time series is modifying how the information in the series is presented; following a transformation, one applies a learning method. In some cases, the transformation phase is presented as an essential step of a classification method; nonetheless, at a conceptual level, it is not a classification method but part of the preprocessing. Feature extraction from a time series can be regarded as a form of transformation; however, it is often preferred to reserve the term transformation for *order-preserving* processes, and so do we. Order-preserving transformations can be further separated into *numerical* and *categorical*; the former includes methods that return numerical time series (usually for dimensionality reduction, smoothing, and noise elimination), while the latter produces time sequences that represent the same coarse information. Categorical transformation can be point-based (i.e., each value is substituted by its discrete category), but the interval-based representation often makes more sense (i.e., each interval of values is replaced by a category).

Simple numerical transformations include *Equal Width Discretization (EWD)* and *Equal Frequency Discretization (EFD)*; the former divides the domain of each attribute into equal-width *bins* and then identifies how many values fall inside each bin, while the latter forms a uniform distribution of the values inside each bin. More sophisticated transformations are spectral decomposition based on *Discrete Fourier Transform (DFT)* (Agrawal, Faloutsos, and Swami, 1993), wavelet decomposition based on *Discrete Wavelet Transform (DWT)* (Chan and Fu, 1999) (a process similar to DFT), *Singular Value Decomposition (SVD)* (Chan and Fu, 1999), and *Piecewise Aggregate Approximation (PAA)* (Keogh et al., 2001). At the high level, PAA divides the original

raw time series into equal-width segments and then computes the mean of the values that belong to each segment. *Symbolic Aggregate Approximation (SAX)* (Lin et al., 2007) uses PAA to reduce the dimensionality of a N -points time series to a K -points mean values time series, with $K < N$, and then assigns K symbolic labels to each segment.

Persist (Mörchen and Ultsch, 2005) maximizes the duration of the resulting time intervals which explicitly considers the temporal ordering. In contrast, the transformation proposed by Sciavicco, Stan, and Vaccari (2019) for multivariate time series that produces a categorical interval-based representation (or abstraction) of a series, which inspired the possibility of categorical temporal decision tree extraction from time series, and, more in general, the notion of *timeline*, that represents a time series. Timelines allow interpreting fuzzy interval temporal logics on time series (Conradie et al., 2020; Conradie et al., 2022).

Other categorical transformations include the work by Moskovitch and Shahar (2015) who defined a discretization of time series into a set of symbolic, state-based time intervals, which are used as features for classification tasks. In their work, three versions of their temporal discretization procedure, called *Temporal Discretization For Classification (TD4C)*, were compared with EWD and SAX, resulting in a better approach to classifying time series data. Once the raw time series are abstracted into an interval-based representation, employing TD4C, a set of temporal interval relation patterns are mined as features, a process very similar to *frequent patterns mining* (Agrawal and Srikant, 1994), which are subsequently used for classification via any (standard) classification schema.

Several non-symbolic approaches have been proposed in the literature for time series classification. Kakizawa, Shumway, and Taniguchi (1998) developed optimal bivariate discriminants using multivariate time-invariant forms of discriminant functions. Kudo, Toyama, and Shimbo (1999) proposed a methodology for classifying sets of data points in a multidimensional space based on the common regions through which only time series of one class pass. Their method transforms multivariate signals into binary vectors, where each element of this vector corresponds to one rectangular region of the space value-time and tells if the signal passes through this region, and then, a procedure, called subclass method, is applied to build rules from these binary vectors. Caiado, Crato, and Peña (2006) presented a new measure of distance between time series based on the normalized periodogram, which estimates the spectral density of a signal. Fulcher and Jones (2014) presented a highly-comparative method for learning feature-based classifiers for univariate time series. Their method automatically computes more than 9000 features which are further automatically selected for classification tasks; the trained model is a linear discriminant classifier that fits a multivariate normal density to each class using a pooled estimate of covariance. Functional methods for time series classification in which the notion of distance plays a central role have been developed and tested by Lines and Bagnall (2015), in which the classifier is a *nearest neighbour* (Tan, Steinbach, and Kumar, 2005; Han, Kamber, and Pei, 2011) equipped with *Dynamic Time Warping (DTW)* (Shokoohi-Yekta, Wang, and Keogh, 2015) as dissimilarity measure. The latter methods have also been tested on several univariate time series by Bagnall et al. (2018) and in the multivariate setting by Pasos Ruiz et al. (2021).

A *generative* deep learning model (i.e., an unsupervised model that finds a good representation of the raw time series prior to training a classifier) for time series classification has been proposed by Malhotra et al. (2017), where a *Sequence Auto-Encoder*

(SAE) based on a sequence-to-sequence model (Sutskever, Vinyals, and Le, 2014) is trained. In particular, the model consists of two multi-layered *Recurrent Neural Networks* (RNNs), an *encoder* and a *decoder*, with gated recurrent units in the hidden layers. Once the SAE is learned, the encoder RNN is used as a pre-trained model to obtain embeddings for time series, and, as already pointed out, such embeddings can be used as input features to other off-the-shelf classifiers. Moreover, the authors evaluated their method by training two non-linear *Support Vector Machines* (SVMs). Finally, unlike *Multi-Layer Perceptrons* (MLPs), where the temporal information is lost and the features learned are no longer time-invariant, *Convolutional Neural Networks* (CNNs) are probably the most used architectures for time series classification due to their robustness to learning space-invariant filters and the relatively small amount of training time because, in general, they need to learn fewer parameters (i.e., the weights of the network) with respect to, for example, RNNs or MLPs (Fawaz et al., 2019). Wang, Yan, and Oates (2017) proposed a simple, but robust, CNN-based baseline with three *discriminative* deep learning models (i.e., models that directly learn the mapping between the raw input time series and the output): MLPs, *Fully-Connected Networks* (FCNs), and *Residual Networks* (ResNets). The spectrum of deep learning approaches for classifying time series is vast, and it is currently a very hot topic in the time series mining research community. A systematic treatment of deep learning methods is beyond the scope of this work, but the reader can refer to the work by Fawaz et al. (2019), an up-to-date and very clear review on this topic, where, for example, they extend the taxonomy by Långkvist, Karlsson, and Loutfi (2014) for neural network-based methods.

§7.3 Approaches for Learning from Spatial Data

The most emblematic problem for learning from spatial data is image classification. Deep learning architectures are preferred for such a problem. Arguably, the medical context is where image classification is most used. Thus, we discuss the literature on image classification from a deep learning point of view. There are many reviews on deep learning in healthcare (e.g., see Shen, Wu, and Suk, 2017; Topol, 2019; Esteva et al., 2019), and many more on CNNs related to the medical field (e.g., see Akkus et al., 2017; Litjens et al., 2017; Ker et al., 2018; Raghunath et al., 2019; Khan et al., 2020; Alzubaidi et al., 2021). The take-home message is that CNN architectures, and the like, are the preferred approaches for learning to classify images in the medical context. Figure 7.3 illustrates the literature map of the reviewed literature on this topic.

CNNs have been formalized by LeCun et al. (1989) and have been a major breakthrough in the field of computer vision. There has been a significant interest in the literature on CNN-like architectures, and we briefly discuss some of the important contributions. Szegedy et al. (2015) formalized the *Inception* neural network, a sparsely-connected architecture, instead of a fully-connected one; subsequent improvements have followed (e.g., see Szegedy et al., 2016; Szegedy et al., 2017; Chollet, 2017). ResNets learn *residual* functions with reference to the layer inputs instead of learning unreferenced functions (He et al., 2016); other improvements have been followed (e.g., Wang et al., 2017). *Capsule Networks* (CapsNets) are neural network architectures that learn hierarchical relationships (Sabour, Frosst, and Hinton, 2017). Finally, *Dense Networks* (DenseNets) connect each layer to every other layer in a feed-forward way (Huang et al., 2017).

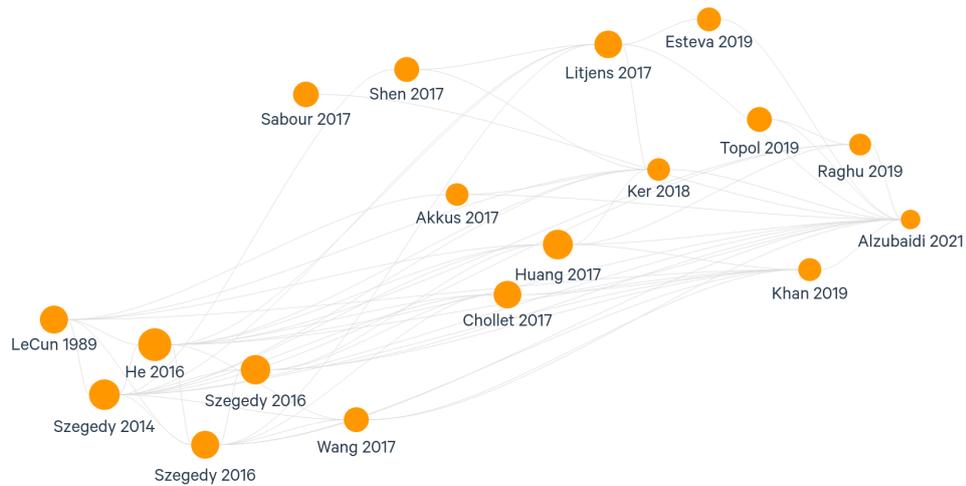


FIGURE 7.3: Literature map for approaches for image classification (generated with *litmaps*).

§7.4 Symbolic Approaches for Learning from Unstructured Data

Learning from unstructured data has also been approached symbolically over the years. Figure 7.4 illustrates some of the contributions to this problem, which we briefly review in this section.

Symbolic methods have been developed for learning from temporal data. Rodríguez Diez, González, and Boström (2001) developed a method for building an ensemble of (base) classifiers with boosting. The method extracts a set of rules having only one antecedent. Moreover, point-based and interval-based predicates are defined to cope with the temporal component. In particular, point-based predicates are introduced to test the results obtained with boosting without using interval-based predicates. To some extent, predicates can be seen as features, and this method falls into the realm of *inductive logic programming*. Geurts (2001) proposed a feature-based approach that integrates extracted temporal patterns into decision trees. Yamada et al. (2003) presented a decision tree-based procedure to classify time series data where the splitting step is done by exhaustively searching a time sequence that is present in data based on class and shape information using DTW as a distance measure. A similar approach is the one proposed by Balakrishnan and Madigan (2006) extending regression trees to deal with functional variables (e.g., multivariate time series) and standard variables (i.e., non-functional). Representative curves are learned to split the dataset using clustering techniques with similarity measures (i.e., Euclidean distance and DTW), where the cluster representative is set to be the instance that is closest (i.e., has a smaller combined distance) to all other examples in the cluster, and, then, reassign instances to the groups based on their distance to the representatives (i.e., complete-link hierarchical clustering). In the work by Baydogan and Runger (2015), each (multivariate) time series includes also the first differences (representing trends) for each numerical variable. *Shapelets* (Ye and Keogh, 2009) have

(from which the decision tree is learned). Ding, Ding, and Perrizo (2002) developed a decision tree learning algorithm that incorporates a data structure, called Peano Count Tree, to capture the semantics of the input images. Li and Claramunt (2006) proposed a propositional decision tree mechanism that models the spatial distribution, which is embedded in the *spatial* entropy-based splits to learn from a structured dataset that resembles a spatial situation. Properties in *linear superposition (spatial) logic*, a spatial logic based on spatial superposition, can be mined from networks of cardiac myocytes with decision trees (Grosu et al., 2009). In the work by Jiang et al. (2012a), inter-pixel correlations are considered by integrating the *local* feature vector with additional scalar features capturing the information of neighbouring pixels. A survey on spatial prediction models is provided by Jiang (2019).

CONCLUSIONS

Before I had studied Chan for thirty years, I saw mountains as mountains, and rivers as rivers. When I arrived at a more intimate knowledge, I came to the point where I saw that mountains are not mountains, and rivers are not rivers. But now that I have got its very substance I am at rest. For it's just that I see mountains once again as mountains, and rivers once again as rivers.

—Qingyuan Weixin

This thesis presented the foundations of modal symbolic learning which brings together two mathematical disciplines: modal logics and machine learning. Modal logics have been studied for more than a century, and their study have been focused mainly on deductive reasoning. In the era of big data, where unstructured data are flourishing daily, machine learning is performed generally in a non-symbolic way, such as with the aid of (deep) neural network architectures, while symbolic learning is still under-represented. Modal symbolic learning is a research effort that combines both symbolic learning and modal logics by exploiting the known results in the literature to enhance symbolic learning algorithms with the ability to learn modal logic theories from the plethora of unstructured data, benefiting from the inductive biases at the symbolic level.

We took propositional decision trees as representatives for the propositional symbolic learning paradigm and motivated such a choice. After presenting propositional decision trees, the quintessential expression of traditional symbolic learning, we gave their modal generalization, called modal decision trees. We then showed classification efficiency, correctness, and completeness for modal decision trees, and we studied the complexity of learning modal decision trees from modal datasets. Modal datasets, which are needed to conduct modal symbolic learning, emerge from real-world applications described by various unstructured data. We showed how to transform a dataset into a modal one by discussing more in detail tailored modal logics. Motivated by the need for a unique formalism, we made an effort by bringing all such logics under the same umbrella. Learning with modal symbolic learning can be done by investigating the properties of modal decision trees, and we discussed two real-world applications where modal theories are preferred. We also discussed possible extensions of our framework. First, we argued how to learn (hybrid) neural-symbolic (modal) decision tree structures leveraging neural networks. Second, we discussed how fuzzy modal decision trees could be achieved by exploiting the known results on fuzzy modal logics and fuzzy decision trees. Third, we presented how gradient-boosted modal decision trees could be obtained, taking motivation from the propositional level. Fourth, we discussed how to learn incrementally with modal decision trees inspired by the known results of incremental learning with propositional decision trees. Finally, motivated by the recent advances in

the field of geometric deep learning, we presented geometric modal symbolic learning by pointing to the relevant work on symmetry-based model checking for modal logics. Furthermore, we reviewed the relevant related work in the literature. We reviewed the history of the development of propositional decision trees since such structures have been used throughout this thesis to construct the foundations of our framework. Then, we discussed the non-symbolic related work for learning from temporal and spatial data. Finally, we examined the symbolic proposals for learning from unstructured data.

Symbolic learning needs to be more represented. Many researchers prefer to approach the learning problem without exploiting the symbolic side (e.g., using neural networks). Distinctively, our framework provides the foundations for embarking on such an exciting field of symbolic learning, exploring the different expressivity power and applicability of (propositional) modal logics. The take-home message is that many mathematical logicians, focusing on deductive reasoning for years, and computer scientists, moved by the hype around neural networks, could find their position in this field by studying new mathematical properties that emerge from researching other symbolic-related inductive approaches. To steer them in such an effort, we pointed out many ambitious extensions of our framework, which can be considered future directions.

Bibliography

- Aceto, L., D. Della Monica, V. Goranko, A. Ingólfssdóttir, A. Montanari, and Guido Sciavicco (2016). "A complete classification of the expressiveness of interval logics of Allen's relations: the general and the dense cases". In: *Acta Informatica* 53.3, pp. 207–246.
- Agrawal, R., C. Faloutsos, and A. N. Swami (1993). "Efficient Similarity Search In Sequence Databases". In: *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms (FODO)*. Vol. 730. Lecture Notes in Computer Science. Springer, pp. 69–84.
- Agrawal, R. and R. Srikant (1994). "Fast Algorithms for Mining Association Rules in Large Databases". In: *Proceedings of 20th International Conference on Very Large Data Bases (VLDB)*, pp. 487–499.
- Akkus, Z., A. Galimzianova, A. Hoogi, D. L. Rubin, and B. J. Erickson (2017). "Deep Learning for Brain MRI Segmentation: State of the Art and Future Directions". In: *Journal of Digital Imaging* 30.4, pp. 449–459.
- Alaniz, S., D. Marcos, B. Schiele, and Z. Akata (2021). "Learning Decision Trees Recurrently Through Communication". In: *Proceedings of the 34th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13518–13527.
- Allen, J. F. (1983). "Maintaining Knowledge about Temporal Intervals". In: *Communication of the ACM* 26.11, pp. 832–843.
- Alur, R., T. Feder, and T. A. Henzinger (1996). "The Benefits of Relaxing Punctuality". In: *Journal of the ACM* 43.1, pp. 116–146.
- Alzubaidi, L., J. Zhang, A. J. Humaidi, A. Q. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan (2021). "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions". In: *Journal of Big Data* 8.1, p. 53.
- Apolloni, B., G. Zamponi, and A. M. Zanaboni (1998). "Learning fuzzy decision trees". In: *Neural Networks* 11.5, pp. 885–895.
- Atlas, L. E., J. T. Connor, D.-C. Park, M. A. El-Sharkawi, R. J. Marks II, A. F. Lippman, R. A. Cole, and Y. K. Muthusamy (1989). "A performance comparison of trained multilayer perceptrons and trained classification trees". In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 915–920.
- Audebert, N., B. Le Saux, and S. Lefèvre (2019). "Deep Learning for Classification of Hyperspectral Data: A Comparative Review". In: *IEEE Geoscience and Remote Sensing Magazine* 7.2, pp. 159–173.
- Aydin, S. K. and E. A. Gol (2020). "Synthesis of Monitoring Rules with STL". In: *Journal of Circuits Systems and Computers* 29.11, 2050177:1–2050177:26.
- Bagnall, A. J., H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh (2018). *The UEA multivariate time series classification archive*. arXiv: [1811.00075](https://arxiv.org/abs/1811.00075) [cs].

- Balakrishnan, S. and D. Madigan (2006). "Decision Trees for Functional Variables". In: *Proceedings of the 6th International Conference on Data Mining (ICDM)*, pp. 798–802.
- Bartocci, E., L. Bortolussi, L. Nenzi, and G. Sanguinetti (2015). "System design of stochastic models using robustness of temporal properties". In: *Theoretical Computer Science* 587, pp. 3–25.
- Bartocci, E., L. Bortolussi, and G. Sanguinetti (2014). "Data-Driven Statistical Learning of Temporal Logic Properties". In: *Proceedings of the 12th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*. Vol. 8711. Lecture Notes in Computer Science. Springer, pp. 23–37.
- Baydogan, M. G. and G. C. Runger (2015). "Learning a symbolic representation for multivariate time series classification". In: *Data Mining and Knowledge Discovery* 29.2, pp. 400–422.
- Bechini, G., E. Losi, L. Manservigi, G. Pagliarini, G. Sciavicco, I. E. Stan, and M. Venturini (2023). "Statistical Rule Extraction for Gas Turbine Trip Prediction". In: *Journal of Engineering for Gas Turbines and Power* 145.5.
- Belson, W. A. (1956). "A Technique for Studying the Effects of Television Broadcast". In: *Journal of the Royal Statistical Society* 5.3, pp. 195–202.
- Blackburn, P., M. de Rijke, and Y. Venema (2001). *Modal Logic*. Cambridge University Press.
- Blockeel, H. and L. De Raedt (1998). "Top-Down Induction of First-Order Logical Decision Trees". In: *Artificial Intelligence* 101.1-2, pp. 285–297.
- Bombara, G. and C. Belta (2018). "Online Learning of Temporal Logic Formulae for Signal Classification". In: *Proceedings of the 16th European Control Conference (ECC)*, pp. 2057–2062.
- (2021). "Offline and Online Learning of Signal Temporal Logic Formulae Using Decision Trees". In: *ACM Transactions on Cyber-Physical Systems* 5.3, 22:1–22:23.
- Bombara, G., C. I. Vasile, F. Penedo, H. Yasuoka, and C. Belta (2016). "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.
- Bou, F., F. Esteva, L. Godo, and R. O. Rodríguez (2011). "On the Minimum Many-Valued Modal Logic over a Finite Residuated Lattice". In: *Journal of Logic and Computation* 21.5, pp. 739–790.
- Boyen, X. and L. Wehenkel (1999). "Automatic induction of fuzzy decision trees and its application to power system security assessment". In: *Fuzzy Sets and Systems* 102.1, pp. 3–19.
- Breiman, L. (1996). "Bagging Predictors". In: *Machine Learning* 24.2, pp. 123–140.
- (2001). "Random Forests". In: *Machine Learning* 45.1, pp. 5–32.
- Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and regression trees*. Wadsworth Publishing Company.
- Brent, R. P. (1991). "Fast training algorithms for multilayer neural nets". In: *IEEE Transactions on Neural Networks* 2.3, pp. 346–354.
- Bresolin, D., D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco (2014). "Interval temporal logics over strongly discrete linear orders: Expressiveness and complexity". In: *Theoretical Computers Science* 560, pp. 269–291.
- Bresolin, D., A. Kurucz, E. Muñoz-Velasco, V. Ryzhikov, G. Sciavicco, and M. Zakharyashev (2017). "Horn Fragments of the Halpern-Shoham Interval Temporal Logic". In: *ACM Transactions on Computational Logic* 18.3, 22:1–22:39.

- Bresolin, D., D. Della Monica, A. Montanari, P. Sala, and G. Sciavicco (2019). "Decidability and complexity of the fragments of the modal logic of Allen's relations over the rationals". In: *Information and Computation* 266, pp. 97–125.
- Bresolin, D., A. Montanari, P. Sala, and G. Sciavicco (2009). "A Tableau-Based System for Spatial Reasoning about Directional Relations". In: *Proceedings of the 18th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX)*. Vol. 5607. Lecture Notes in Computer Science. Springer, pp. 123–137.
- Bronstein, M. M., J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst (2017). "Geometric Deep Learning: Going beyond Euclidean data". In: *IEEE Signal Processing Magazine* 34.4, pp. 18–42.
- Brown, C., J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo (2020). "Exploring Automatic Diagnosis of COVID-19 from Crowdsourced Respiratory Sound Data". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 3474–3484.
- Brunello, A., E. Marzano, A. Montanari, and G. Sciavicco (2019). "J48SS: A Novel Decision Tree Approach for the Handling of Sequential and Time Series Data". In: *Computers* 8.1, p. 21.
- Brunello, A., G. Sciavicco, and I. E. Stan (2019). "Interval Temporal Logic Decision Tree Learning". In: *Proceedings of the 16th European Conference on Logics in Artificial Intelligence (JELIA)*. Vol. 11468. Lecture Notes in Computer Science. Springer, pp. 778–793.
- Brynjolfsson, E. (2022). "The Turing Trap: The Promise & Peril of Human-Like Artificial Intelligence". In: *Daedalus* 151.2, pp. 272–287.
- Caiado, J., N. Crato, and D. Peña (2006). "A periodogram-based metric for time series classification". In: *Computational Statistics and Data Analysis* 50.10, pp. 2668–2684.
- Caicedo, X. and R. O. Rodríguez (2010). "Standard Gödel modal logics". In: *Studia Logica* 94.2, pp. 189–214.
- Cao, X., F. Zhou, L. Xu, D. Meng, Z. Xu, and J. W. Paisley (2018). "Hyperspectral Image Classification With Markov Random Fields and a Convolutional Neural Network". In: *IEEE Transactions on Image Processing* 27.5, pp. 2354–2367.
- Carter, C. and J. Catlett (1987). "Assessing Credit Card Applications Using Machine Learning". In: *IEEE Expert* 2.3, pp. 71–79.
- Chan, K. and A. W. Fu (1999). "Efficient Time Series Matching by Wavelets". In: *Proceedings of the 15th International Conference on Data Engineering (ICDE)*, pp. 126–133.
- Chang, R. L. P. and T. Pavlidis (1977). "Fuzzy Decision Tree Algorithms". In: *IEEE Transactions on Systems, Man, and Cybernetics* 7.1, pp. 28–35.
- Chaochen, Z., C. A. R. Hoare, and A. P. Ravn (1991). "A Calculus of Durations". In: *Information Processing Letters* 40.5, pp. 269–276.
- Chen, T. and C. Guestrin (2016). "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.
- Chen, Y., J. Tumova, and C. Belta (2012). "LTL robot motion control based on automata learning of environmental dynamics". In: *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5177–5182.
- Chen, Y., J. Tumova, A. Ulusoy, and C. Belta (2013). "Temporal logic robot control based on automata learning of environmental dynamics". In: *International Journal of Robotics Research* 32.5, pp. 547–565.

- Chi, Z. and H. Yan (1996). "ID3-derived fuzzy rules and optimized defuzzification for handwritten numeral recognition". In: *IEEE Transactions on Fuzzy Systems* 4.1, pp. 24–31.
- Chollet, F. (2017). "Xception: Deep Learning with Depthwise Separable Convolutions". In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807.
- Christ, M., N. Braun, J. Neuffer, and A. W. Kempa-Liehr (2018). "Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh - A Python package)". In: *Neurocomputing* 307, pp. 72–77.
- Cios, K. J. and L. M. Sztandera (1992). "Continuous ID3 algorithm with fuzzy entropy measures". In: *Proceedings of the 1992 IEEE International Conference on Fuzzy Systems*, pp. 469–476.
- Clark, P. and T. Niblett (1989). "The CN2 Induction Algorithm". In: *Machine Learning* 3, pp. 261–283.
- Clarke, E. M., E. A. Emerson, and A. P. Sistla (1986). "Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications". In: *ACM Transactions on Programming Languages and Systems* 8.2, pp. 244–263.
- Clarke, E. M., O. Grumberg, D. Kroening, D. A. Peled, and H. Veith (2018). *Model Checking*. 2nd. MIT Press.
- Clarke, E. M., S. Jha, R. Enders, and T. Filkorn (1996). "Exploiting Symmetry in Temporal Logic Model Checking". In: *Formal Methods in Systems Design* 9.1/2, pp. 77–104.
- Coeckelbergh, M. (2020). *AI Ethics*. MIT Press.
- Cohen, J. (1960). "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* 20.1, pp. 37–46.
- Conradie, W., D. Della Monica, E. Muñoz-Velasco, G. Sciavicco, and I. E. Stan (2020). "Time Series Checking with Fuzzy Interval Temporal Logics". In: *Proceedings of the 21st Italian Conference on Theoretical Computer Science (ICTCS)*. Vol. 2756. CEUR Workshop Proceedings. CEUR-WS.org, pp. 250–262.
- (2022). "Fuzzy Halpern and Shoham's interval temporal logics". In press in *Fuzzy Sets and Systems*.
- Craven, M. W. and J. W. Shavlik (1995). "Extracting Tree-Structured Representations of Trained Networks". In: *Proceedings of the 8th Advances in Neural Information Processing Systems (NIPS)*, pp. 24–30.
- Crawford, S. L. (1989). "Extensions to the CART Algorithm". In: *International Journal of Man-Machine Studies* 31.2, pp. 197–217.
- Dancey, D., D. McLean, and Z. Bandar (2004). "Decision Tree Extraction from Trained Neural Networks". In: *Proceedings of the 7th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pp. 515–519.
- d'Avila Garcez, A. S., M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran (2019). "Neural-symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning". In: *Journal of Applied Logics* 6.4, pp. 611–632.
- d'Avila Garcez, A. S., L. C. Lamb, and D. M. Gabbay (2009). *Neural-Symbolic Cognitive Reasoning*. Cognitive Technologies. Springer.
- Davis, M. (2018). *The Universal Computer: The Road from Leibniz to Turing*. 3rd. CRC Press, Inc.
- Davis, S. B. and P. Mermelstein (1980). "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 28.4, pp. 357–366.

- de Ville, B. (2013). "Decision trees". In: *WIREs Computational Statistics* 5.6, pp. 448–455.
- Della Monica, D., D. de Frutos-Escrig, A. Montanari, A. Murano, and G. Sciavicco (2017). "Evaluation of Temporal Datasets via Interval Temporal Logic Model Checking". In: *Proceedings of the 24th International Symposium on Temporal Representation and Reasoning (TIME)*. Vol. 90. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 11:1–11:18.
- Della Monica, D., A. Montanari, G. Sciavicco, and I. E. Stan (2020). "A Note on Ultimately-Periodic Finite Interval Temporal Logic Model Checking". In: *Proceedings of the 2nd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata (OVERLAY)*. Vol. 2785. CEUR Workshop Proceedings. CEUR-WS.org, pp. 11–15.
- Della Monica, D., G. Pagliarini, G. Sciavicco, and I. E. Stan (2022). "Decision Trees with a Modal Flavor". In press in the Proceedings of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA).
- Dempster, A., F. Petitjean, and G. I. Webb (2020). "ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels". In: *Data Mining and Knowledge Discovery* 34.5, pp. 1454–1495.
- Deng, H. (2019). "Interpreting tree ensembles with inTrees". In: *International Journal of Data Science and Analytics* 7.4, pp. 277–287.
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei (2009). "ImageNet: A large-scale hierarchical image database". In: *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255.
- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186.
- Ding, Q., Q. Ding, and W. Perrizo (2002). "Decision tree classification of spatial data streams using Peano Count Trees". In: *Proceedings of the 2002 ACM Symposium on Applied Computing (SAC)*, pp. 413–417.
- Domingos, P. M. and G. Hulten (2000). "Mining high-speed data streams". In: *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 71–80.
- Dreyfus, H. L. (1972). *What Computers Can't Do*. MIT Press.
- Egenhofer, M. J. and R. D. Franzosa (1991). "Point-set topological spatial relations". In: *International Journal of Geographical Information Systems* 5.2, pp. 161–174.
- Elomaa, T. and M. Kaariainen (2001). "An Analysis of Reduced Error Pruning". In: *Journal of Artificial Intelligence Research* 15, pp. 163–187.
- Emerson, E. A. and A. P. Sistla (1996). "Symmetry and Model Checking". In: *Formal Methods in System Design* 9.1/2, pp. 105–131.
- Esteva, A., A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean (2019). "A guide to deep learning in health-care". In: *Nature Medicine* 25.1, pp. 24–29.
- Fawaz, H. I., G. Forestier, J. Weber, L. Idoumghar, and P. Muller (2019). "Deep learning for time series classification: a review". In: *Data Mining and Knowledge Discovery* 33.4, pp. 917–963.
- Fawaz, H. I., B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean (2020). "InceptionTime: Finding AlexNet for time series classification". In: *Data Mining and Knowledge Discovery* 34.6, pp. 1936–1962.

- Fedus, W., B. Zoph, and N. Shazeer (2022). "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity". In: *Journal of Machine Learning Research* 23.120, pp. 1–39.
- Fitting, M. (1991). "Many-valued modal logics". In: *Fundamenta Informaticae* 15.3-4, pp. 235–254.
- Fitting, M. and R. L. Mendelsohn (1998). *First-Order Modal Logic*. Kluwer Academic Publishers.
- Freund, Y. and R. E. Schapire (1996). "Experiments with a New Boosting Algorithm". In: *Machine Learning, Proceedings of the 13th International Conference on Machine Learning (ICML)*, pp. 148–156.
- Friedman, J. H. (2000). "Greedy Function Approximation: A Gradient Boosting Machine". In: *Annals of Statistics* 29, pp. 1189–1232.
- (2002). "Stochastic gradient boosting". In: *Computational Statistics and Data Analysis* 38.4, pp. 367–378.
- Fulcher, B. D. and N. S. Jones (2014). "Highly Comparative Feature-Based Time-Series Classification". In: *IEEE Transactions on Knowledge and Data Engineering* 26.12, pp. 3026–3037.
- Fürnkranz, J. (1999). "Separate-and-Conquer Rule Learning". In: *Artificial Intelligence Review* 13.1, pp. 3–54.
- Gama, J., R. Fernandes, and R. Rocha (2006). "Decision trees for mining data streams". In: *Intelligent Data Analysis* 10.1, pp. 23–45.
- Gambella, C., B. Ghaddar, and J. Naoum-Sawaya (2021). "Optimization problems for machine learning: A survey". In: *European Journal of Operational Research* 290.3, pp. 807–828.
- Gandomi, A. and M. Haider (2015). "Beyond the hype: Big data concepts, methods, and analytics". In: *International Journal of Information Management* 35.2, pp. 137–144.
- Genesereth, M. R. and N. J. Nilsson (1988). *Logical foundations of artificial intelligence*. Morgan Kaufmann.
- Geurts, P. (2001). "Pattern Extraction for Time Series Classification". In: *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*. Vol. 2168. Lecture Notes in Computer Science. Springer, pp. 115–127.
- Gnewuch, M., H. Pasing, and C. Weiß (2021). "A generalized Faulhaber inequality, improved bracketing covers, and applications to discrepancy". In: *Mathematics of Computation* 90.332, pp. 2873–2898.
- Goel, P. K., S. O Prasher, R. M Patel, J. A Landry, R. B Bonnell, and A. A Viau (2003). "Classification of hyperspectral data by decision trees and artificial neural networks to identify weed stress and nitrogen status of corn". In: *Computers and Electronics in Agriculture* 39.2, pp. 67–93.
- Goldblatt, R. (2003). "Mathematical modal logic: A view of its evolution". In: *Journal of Applied Logic* 1.5-6, pp. 309–392.
- Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. MIT Press.
- GoRanko, V., A. Montanari, and G. Sciavicco (2004). "A Road Map of Interval Temporal Logics and Duration Calculi". In: *Journal of Applied Non-Classical Logics* 14.1-2, pp. 9–54.
- Grosu, R., S. A. Smolka, F. Corradini, A. Wasilewska, E. Entcheva, and E. Bartocci (2009). "Learning and detecting emergent behavior in networks of cardiac myocytes". In: *Communications of the ACM* 52.3, pp. 97–105.
- Gunning, D. and D. W. Aha (2019). "DARPA's Explainable Artificial Intelligence (XAI) Program". In: *AI Magazine* 40.2, pp. 44–58.

- Gunning, D., M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang (2019). "XAI - Explainable artificial intelligence". In: *Science Robotics* 4.37.
- Guo, H. and S. B. Gelfand (1992). "Classification trees with neural network feature extraction". In: *IEEE Transactions on Neural Networks* 3.6, pp. 923–933.
- Hájek, P. (2005). "Making fuzzy description logic more general". In: *Fuzzy Sets and Systems* 154.1, pp. 1–15.
- (2013). *Metamathematics of fuzzy logic*. Springer.
- Halpern, J. Y., R. Harper, N. Immerman, P. G. Kolaitis, M. Y. Vardi, and V. Vianu (2001). "On the Unusual Effectiveness of Logic in Computer Science". In: *Bulletin of Symbolic Logic* 7.2, pp. 213–236.
- Halpern, J. Y. and Y. Shoham (1991). "A Propositional Modal Logic of Time Intervals". In: *Journal of the ACM* 38.4, pp. 935–962.
- Han, J., M. Kamber, and J. Pei (2011). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
- Hastie, T., R. Tibshirani, and J. H. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. Springer Series in Statistics. Springer.
- Hayashi, I., T. Maeda, A. Bastian, and L. C. Jain (1998). "Generation of fuzzy decision trees by fuzzy ID3 with adjusting mechanism of AND/OR operators". In: *Proceedings of the 1998 IEEE International Conference on Fuzzy Systems*, pp. 681–685.
- He, K., X. Zhang, S. Ren, and J. Sun (2016). "Deep Residual Learning for Image Recognition". In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.
- Hinton, G., O. Vinyals, and J. Dean (2015). *Distilling the Knowledge in a Neural Network*. DOI: [10.48550/arXiv.1503.02531](https://doi.org/10.48550/arXiv.1503.02531).
- Hong, D., Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot (2022). "SpectralFormer: Rethinking Hyperspectral Image Classification With Transformers". In: *IEEE Transactions on Geoscience and Remote Sensing* 60, pp. 1–15.
- Hornik, K. (1991). "Approximation capabilities of multilayer feedforward networks". In: *Neural Networks* 4.2, pp. 251–257.
- Hornik, K., M. B. Stinchcombe, and H. White (1989). "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5, pp. 359–366.
- Hoxha, B., A. Dokhanchi, and G. Fainekos (2018). "Mining parametric temporal logic properties in model-based design for cyber-physical systems". In: *International Journal on Software Tools for Technology Transfer* 20.1, pp. 79–93.
- Hu, W., Y. Huang, W. Li, F. Zhang, and H.-C. Li (2015). "Deep Convolutional Neural Networks for Hyperspectral Image Classification". In: *Journal of Sensors* 2015, 258619:1–258619:12.
- Huang, G., Z. Liu, L. van der Maaten, and K. Q. Weinberger (2017). "Densely Connected Convolutional Networks". In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269.
- Hulten, G., L. Spencer, and P. M. Domingos (2001). "Mining time-changing data streams". In: *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 97–106.
- Hunt, E. B., P. J. Stone, and J. Marin (1966). *Experiments in Induction*. Academic Press New York.
- Hurley, P. J. (2014). *A Concise Introduction to Logic*. 12th. Cengage Learning.
- Huth, M. and M. D. Ryan (2004). *Logic in Computer Science: Modelling and Reasoning about Systems*. 2nd. Cambridge University Press.
- Hyafil, L. and R. L. Rivest (1976). "Constructing Optimal Binary Decision Trees is NP-Complete". In: *Information Processing Letters* 5.1, pp. 15–17.

- Ichihashi, Hidetomo, T. Shirai, Kazunori Nagasaka, and Tetsuya Miyoshi (1996). "Neuro-fuzzy ID3: a method of inducing fuzzy decision trees with linear programming for maximizing entropy and an algebraic method for incremental learning". In: *Fuzzy Sets and Systems* 81.1, pp. 157–167.
- Ip, C. N. and D. L. Dill (1996). "Better Verification Through Symmetry". In: *Formal Methods in System Design* 9.1/2, pp. 41–75.
- Ivanova, I. and M. Kubat (1995). "Initialization of neural networks by means of decision trees". In: *Knowledge-Based Systems* 8.6, pp. 333–344.
- James, G., D. Witten, T. Hastie, and R. Tibshirani (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.
- Jang, J.-S. R. (1994). "Structure determination in fuzzy modeling: a fuzzy CART approach". In: *Proceedings of the 3rd IEEE International Fuzzy Systems Conference*, pp. 480–485.
- Janikow, C. Z. (1998). "Fuzzy decision trees: issues and methods". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 28.1, pp. 1–14.
- Jeng, B., Y.-M. Jeng, and T.-P. Liang (1997). "FILM: a fuzzy inductive learning method for automated knowledge acquisition". In: *Decision Support Systems* 21.2, pp. 61–73.
- Jha, S., A. Tiwari, S. A. Seshia, T. Sahai, and N. Shankar (2019). "TeLEx: learning signal temporal logic from positive examples using tightness metric". In: *Formal Methods in System Design* 54.3, pp. 364–387.
- Jiang, J., J. Ma, Z. Wang, C. Chen, and X. Liu (2019). "Hyperspectral Image Classification in the Presence of Noisy Labels". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.2, pp. 851–865.
- Jiang, Z. (2019). "A Survey on Spatial Prediction Methods". In: *IEEE Transactions on Knowledge and Data Engineering* 31.9, pp. 1645–1664.
- Jiang, Z., S. Shekhar, P. Mohan, J. Knight, and J. Corcoran (2012a). "Learning Spatial Decision Tree for Geographical Classification: A Summary of Results". In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*, pp. 390–393.
- Jiang, Z., S. Shekhar, P. Mohan, J. F. Knight, and J. Corcoran (2012b). "Learning spatial decision tree for geographical classification: a summary of results". In: *Proceedings of the 20th 2012 International Conference on Advances in Geographic (SIGSPATIAL)*. ACM, pp. 390–393.
- Jin, X., A. Donzé, J. V. Deshmukh, and S. A. Seshia (2015). "Mining Requirements From Closed-Loop Control Models". In: *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems* 34.11, pp. 1704–1717.
- Jones, A., Z. Kong, and C. Belta (2014). "Anomaly detection in cyber-physical systems: A formal methods approach". In: *Proceedings of the 53rd IEEE Conference on Decision and Control (CDC)*, pp. 848–853.
- Jordan, Michael I. (1994). "A Statistical Approach to Decision Tree Modeling". In: *Proceedings of the 11th International Conference on Machine Learning*, pp. 363–370.
- Kakizawa, Y., R. H. Shumway, and M. Taniguchi (1998). "Discrimination and Clustering for Multivariate Time Series". In: *Journal of the American Statistical Association* 93.441, pp. 328–340.
- Kaminska, J., E. Lucena-Sánchez, G. Sciavicco, and I. E. Stan (2020). "Rule Extraction via Dynamic Discretization with an Application to Air Quality Modelling". In: *Proceedings of the 14th International Rule Challenge, 4th Doctoral Consortium, and 6th Industry Track (RULE+RR)*. Vol. 2644. CEUR Workshop Proceedings. CEUR-WS.org, pp. 42–57.

- Kass, G. V. (1980). "An Exploratory Technique for Investigating Large Quantities of Categorical Data". In: *Journal of the Royal Statistical Society* 29.2, pp. 119–127.
- Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Proceedings of the 31st Advances in Neural Information Processing Systems (NIPS)*, pp. 3146–3154.
- Kearns, M. J. and L. G. Valiant (1994). "Cryptographic Limitations on Learning Boolean Formulae and Finite Automata". In: *Journal of the ACM* 41.1, pp. 67–95.
- Keogh, E. J., K. Chakrabarti, M. J. Pazzani, and S. Mehrotra (2001). "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases". In: *Knowledge and Information Systems* 3.3, pp. 263–286.
- Ker, J., L. Wang, J. Rao, and T. C. C. Lim (2018). "Deep Learning Applications in Medical Image Analysis". In: *IEEE Access* 6, pp. 9375–9389.
- Khan, A., A. Sohail, U. Zahoor, and A. S. Qureshi (2020). "A survey of the recent architectures of deep convolutional neural networks". In: *Artificial Intelligence Review* 53.8, pp. 5455–5516.
- Kong, Z., A. Jones, and C. Belta (2017). "Temporal Logics for Learning and Detection of Anomalous Behavior". In: *IEEE Transactions on Automatic Control* 62.3, pp. 1210–1222.
- Kong, Z., A. Jones, A. Medina Ayala, E. Aydin Gol, and C. Belta (2014). "Temporal logic inference for classification and prediction from data". In: *Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control (HSCC)*, pp. 273–282.
- Kontschieder, P., M. Fiterau, A. Criminisi, and S. Rota Bulò (2015). "Deep Neural Decision Forests". In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1467–1475.
- Korpáš, J., J. Sadloňová, and M. Vrabec (1996). "Analysis of the cough sound: an overview". In: *Pulmonary pharmacology* 9.5-6, pp. 261–268.
- Kripke, S. A. (1963). "Semantical Analysis of Modal Logic I. Normal Propositional Calculi". In: *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 9.5-6, pp. 67–96.
- Krishnan, R., G. Sivakumar, and P. Bhattacharya (1999). "Extracting decision trees from trained neural networks". In: *Pattern Recognition* 32.12, pp. 1999–2009.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 26th Advances in Neural Information Processing Systems (NIPS)*, pp. 1106–1114.
- Kubat, M. (1998). "Decision Trees Can Initialize Radial-Basis Function Networks". In: *IEEE Transactions on Neural Networks* 9.5, pp. 813–821.
- Kudo, M., J. Toyama, and M. Shimbo (1999). "Multidimensional Curve Classification Using Passing—through Regions". In: *Pattern Recognition Letters* 20.11, pp. 1103–1111.
- Långkvist, M., L. Karlsson, and A. Loutfi (2014). "A review of unsupervised feature learning and deep learning for time-series modeling". In: *Pattern Recognition Letters* 42, pp. 11–24.
- LeCun, Y., B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4, pp. 541–551.
- Lee, H. and H. Kwon (2017). "Going Deeper With Contextual CNN for Hyperspectral Image Classification". In: *IEEE Transactions on Image Processing* 26.10, pp. 4843–4855.
- Lewis, C. I. (1918). *A Survey of Symbolic Logic*. University of California Press.

- Li, T., L. Fang, and A. Jennings (1992). "Structurally Adaptive Self-Organizing Neural Trees". In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 329–334.
- Li, X. and C. Claramunt (2006). "A Spatial Entropy-Based Decision Tree for Classification of Geographical Information". In: *Transactions in GIS* 10.3, pp. 451–467.
- Li, Y., H. Zhang, and Q. Shen (2017). "Spectral-Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network". In: *Remote Sensing* 9.1, p. 67.
- Lin, J., E. J. Keogh, L. Wei, and S. Lonardi (2007). "Experiencing SAX: a novel symbolic representation of time series". In: *Data Mining and Knowledge Discovery* 15.2, pp. 107–144.
- Lines, J. and A. J. Bagnall (2015). "Time series classification with ensembles of elastic distance measures". In: *Data Mining and Knowledge Discovery* 29.3, pp. 565–592.
- Litjens, G., T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. van der Laak, B. van Ginneken, and C. I. Sánchez (2017). "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42, pp. 60–88.
- Loh, W. Y. (2011). "Classification and regression trees". In: *WIREs Data Mining and Knowledge Discovery* 1.1, pp. 14–23.
- (2014). "Fifty Years of Classification and Regression Trees". In: *International Statistical Review* 82.3, pp. 329–348.
- Lubba, C. H., S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones (2019). "catch22: CAnonical Time-series CHaracteristics - Selected through highly comparative time-series analysis". In: *Data Mining and Knowledge Discovery* 33.6, pp. 1821–1852.
- Lucena-Sánchez, E., E. Muñoz-Velasco, G. Sciavicco, I. E. Stan, and A. Vaccari (2019). "Towards Interval Temporal Logic Rule-Based Classification". In: *Proceedings of the 1st Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY)*. Vol. 2509. CEUR Workshop Proceedings. CEUR-WS.org, pp. 65–70.
- Lucena-Sánchez, E., G. Sciavicco, and I. E. Stan (2020). "Symbolic Learning with Interval Temporal Logic: the Case of Regression". In: *Proceedings of the 2nd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata (OVERLAY)*. Vol. 2785. CEUR Workshop Proceedings. CEUR-WS.org, pp. 5–9.
- (2021). "Feature and Language Selection in Temporal Symbolic Regression for Interpretable Air Quality Modelling". In: *Algorithms* 14.3, p. 76.
- Lutz, C. and F. Wolter (2006). "Modal Logics of Topological Relations". In: *Logical Methods in Computer Science* 2.2.
- Maler, O. and D. Nickovic (2004). "Monitoring Temporal Properties of Continuous Signals". In: *Proceedings of the 2004 Joint Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS) and Formal Techniques in Real-Time and Fault-Tolerant Systems (FTRTFT)*. Vol. 3253. Lecture Notes in Computer Science. Springer, pp. 152–166.
- Malhotra, P., V. TV, L. Vig, P. Agarwal, and G. M. Shroff (2017). "TimeNet: Pre-trained deep recurrent neural network for time series classification". In: *Proceedings of the 25th European Symposium on Artificial Neural Networks (ESANN)*.
- Manapragada, C., G. I. Webb, and M. Salehi (2018). "Extremely Fast Decision Tree". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 1953–1962.
- Manzella, F., G. Pagliarini, G. Sciavicco, and I. E. Stan (2021). "Interval Temporal Random Forests with an Application to COVID-19 Diagnosis". In: *Proceedings of the 28th International Symposium on Temporal Representation and Reasoning (TIME)*. Vol. 206. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 7:1–7:18.

- (2022). “The Voice of COVID-19: Breath and Cough Recording Classification with Temporal Decision Trees and Random Forests”. In press in *Artificial Intelligence in Medicine*.
- McCarthy, J., M. Minsky, N. Rochester, and C. E. Shannon (2006). “A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955”. In: *AI Magazine* 27.4, pp. 12–14.
- McCarthy, John (1974). “Professor Sir James Lighthill, FRS. Artificial Intelligence: A General Survey”. In: *Artificial Intelligence* 5.3, pp. 317–322.
- McCulloch, W. and W. Pitts (1943). “A Logical Calculus of Ideas Immanent in Nervous Activity”. In: *Bulletin of Mathematical Biophysics* 5, pp. 127–147.
- McKinsey, J. C. C. and A. Tarski (1944). “The Algebra of Topology”. In: *Annals of Mathematics* 45.1, pp. 141–191.
- Messenger, R. and L. Mandell (1972). “A Modal Search Technique for Predictive Nominal Scale Multivariate Analysis”. In: *Journal of the American Statistical Association* 67.340, pp. 768–772.
- Michelsoni, C., A. Rani, S. Kumar, and G. L. Foresti (2012). “A balanced neural tree for pattern classification”. In: *Neural Networks* 27, pp. 81–90.
- Minsky, M. (1991). “Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy”. In: *AI Magazine* 12.2, pp. 34–51.
- Minsky, M. and S. Papert (1969). *Perceptrons*. MIT Press.
- Mitchell, T. M. (1997). *Machine learning*. McGraw-Hill.
- Montanari, A., G. Sciavicco, and N. Vitacolonna (2002). “Decidability of Interval Temporal Logics over Split-Frames via Granularity”. In: *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA)*. Vol. 2424. Lecture Notes in Computer Science. Springer, pp. 259–270.
- Morales, A., I. Navarrete, and G. Sciavicco (2007). “A new modal logic for reasoning about space: spatial propositional neighborhood logic”. In: *Annals of Mathematics and Artificial Intelligence* 51.1, pp. 1–25.
- Mörchen, F. and A. Ultsch (2005). “Optimizing time series discretization for knowledge discovery”. In: *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 660–665.
- Morgan, J. N. and J. A. Sonquist (1963). “Problems in the Analysis of Survey Data, and a Proposal”. In: *Journal of the American Statistical Association* 58.302, pp. 415–434.
- Moskovitch, R. and Y. Shahar (2015). “Classification-driven temporal discretization of multivariate time series”. In: *Data Mining and Knowledge Discovery* 29.4, pp. 871–913.
- Mou, L., P. Ghamisi, and X. X. Zhu (2017). “Deep Recurrent Neural Networks for Hyperspectral Image Classification”. In: *IEEE Transactions on Geoscience and Remote Sensing* 55.7, pp. 3639–3655.
- Muñoz-Velasco, E., M. Pelegrín-García, P. Sala, G. Sciavicco, and I. E. Stan (2019). “On coarser interval temporal logics”. In: *Artificial Intelligence* 266, pp. 1–26.
- Muggleton, S. H. (1991). “Inductive Logic Programming”. In: *New Generation Computing* 8.4, pp. 295–318.
- Muñoz-Velasco, E., G. Sciavicco, and I. E. Stan (2017). “Implementation of a Tableau-based Satisfiability Checker for HS3”. In: *Joint Proceedings of the 18th Italian Conference on Theoretical Computer Science (ICTCS) and the 32nd Italian Conference on Computational Logic (CILC)*. Vol. 1949. CEUR Workshop Proceedings. CEUR-WS.org, pp. 326–340.

- Murdock, C., Z. Li, H. Zhou, and T. Duerig (2016). “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.
- Murthy, V. N., V. Singh, T. Chen, R. Manmatha, and D. Comaniciu (2016). “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.
- Neider, D. and I. Gavran (2018). “Learning Linear Temporal Properties”. In: *Proceedings of the 2018 Formal Methods in Computer Aided Design (FMCAD)*, pp. 1–10.
- Newell, A. and H. A. Simon (1976). “Computer Science as Empirical Inquiry: Symbols and Search”. In: *Communications of the ACM* 19.3, pp. 113–126.
- Nguyen, L. V., J. Kapinski, X. Jin, J. V. Deshmukh, K. Butts, and T. T. Johnson (2017). “Abnormal Data Classification Using Time-Frequency Temporal Logic”. In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control (HSCC)*, pp. 237–242.
- Obermeyer, Z., B. Powers, C. Vogeli, and S. Mullainathan (2019). “Dissecting racial bias in an algorithm used to manage the health of populations”. In: *Science* 366.6464, pp. 447–453.
- Olaru, C. and L. Wehenkel (2003). “A complete fuzzy decision tree technique”. In: *Fuzzy Sets Syst.* 138.2, pp. 221–254.
- Pagliarini, G., S. Scabro, G. Serra, G. Sciavicco, and I. E. Stan (2022). “Neural-Symbolic Temporal Decision Trees for Multivariate Time Series Classification”. In: *Proceedings of the 29th International Symposium on Temporal Representation and Reasoning (TIME)*. Vol. 247. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 13:1–13:15.
- Pagliarini, G. and G. Sciavicco (2021). “Decision Tree Learning with Spatial Modal Logics”. In: *Proceedings of the 12th International Symposium on Games, Automata, Logics, and Formal Verification (GANDALF)*. Vol. 346, pp. 273–290.
- Pagliarini, G., G. Sciavicco, and I. E. Stan (2021). “Multi-Frame Modal Symbolic Learning”. In: *Proceedings of the 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY)*. Vol. 2987. CEUR Workshop Proceedings. CEUR-WS.org, pp. 37–41.
- Papadimitriou, C. H. (1994). *Computational Complexity*. Addison-Wesley.
- Pasos Ruiz, A., M. Flynn, J. Large, M. Middlehurst, and A. J. Bagnall (2021). “The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances”. In: *Data Mining and Knowledge Discovery* 35.2, pp. 401–449.
- Pnueli, A. (1977). “The Temporal Logic of Programs”. In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 46–57.
- Quinlan, J. R. (1986). “Induction of Decision Trees”. In: *Machine Learning* 1, pp. 81–106.
- (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- (1999). “Simplifying decision trees”. In: *International Journal of Human-Computer Studies* 51.2, pp. 497–510.
- Raghu, M., C. Zhang, J. M. Kleinberg, and S. Bengio (2019). “Transfusion: Understanding Transfer Learning for Medical Imaging”. In: *Proceedings of the 32nd Advances in Neural Information Processing Systems (NIPS)*, pp. 3342–3352.
- Rivest, R. L. (1987). “Learning Decision Lists”. In: *Machine Learning* 2.3, pp. 229–246.
- Rodríguez Diez, J. J., C. A. González, and H. Boström (2001). “Boosting interval based literals”. In: *Intelligent Data Analysis* 5.3, pp. 245–262.

- Rosenblatt, F. (1958). "The perceptron: A probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65.6, pp. 386–408.
- Roy, S. K., G. Krishna, S. R. Dubey, and B. B. Chaudhuri (2020). "HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification". In: *IEEE Geoscience and Remote Sensing Letters* 17.2, pp. 277–281.
- Rudin, C. (2019). "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5, pp. 206–215.
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). "Learning Representations by Back-propagating Errors". In: *Nature* 323.6088, pp. 533–536.
- Russell, S. and P. Norvig (2020). *Artificial Intelligence: A Modern Approach*. 4th. Pearson.
- Sabour, S., N. Frosst, and G. E. Hinton (2017). "Dynamic Routing Between Capsules". In: *Proceedings of the 30th Advances in Neural Information Processing Systems (NIPS)*, pp. 3856–3866.
- Saglam, I. and E. A. Gol (2019). "Cause Mining and Controller Synthesis with STL". In: *Proceedings of the 58th IEEE Conference on Decision and Control, (CDC)*, pp. 4589–4594.
- Santara, A., K. Mani, P. Hatwar, A. Singh, A. Garg, K. Padia, and P. Mitra (2017). "BASS Net: Band-Adaptive Spectral-Spatial Feature Learning Neural Network for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.9, pp. 5293–5301.
- Scarselli, F., M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini (2009). "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1, pp. 61–80.
- Schlimmer, J. C. and D. H. Fisher (1986). "A Case Study of Incremental Concept Induction". In: *Proceedings of the 5th National Conference on Artificial Intelligence (AAAI)*, pp. 496–501.
- Schmitz, G. P. J., C. Aldrich, and F. S. Gouws (1999). "ANN-DT: An Algorithm for Extraction of Decision Trees from Artificial Neural Networks". In: *IEEE Transactions on Neural Networks* 10.6, pp. 1392–1401.
- Sciavicco, G. and I. E. Stan (2020). "Knowledge Extraction with Interval Temporal Logic Decision Trees". In: *Proceedings of the 27th International Symposium on Temporal Representation and Reasoning (TIME)*. Vol. 178. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 9:1–9:16.
- Sciavicco, G., I. E. Stan, and A. Vaccari (2019). "Towards a General Method for Logical Rule Extraction from Time Series". In: *Proceedings of the 8th International Work-Conference on the Interplay Between Natural and Artificial Computation (IWINAC)*. Vol. 11487. Lecture Notes in Computer Science. Springer, pp. 3–12.
- Sethi, I. K. (1990). "Entropy nets: from decision trees to neural networks". In: *Proceedings of the IEEE* 78.10, pp. 1605–1613.
- Setiono, R. and W. K. Leow (1999). "On mapping decision trees and neural networks". In: *Knowledge-Based Systems* 12.3, pp. 95–99.
- Setiono, R. and H. Liu (1999). "A Connectionist Approach to Generating Oblique Decision Trees". In: *IEEE Transactions on Systems, Man and Cybernetics – Part B* 29.3, pp. 440–444.
- Shavlik, J. W., R. J. Mooney, and G. G. Towell (1991). "Symbolic and Neural Learning Algorithms: An Experimental Comparison". In: *Machine Learning* 6, pp. 111–143.
- Shen, D., G. Wu, and H. I. Suk (2017). "Deep Learning in Medical Image Analysis". In: *Annual Review of Biomedical Engineering* 19, pp. 221–248.

- Shokoohi-Yekta, M., J. Wang, and E. Keogh (2015). "On the Non-Trivial Generalization of Dynamic Time Warping to the Multi-Dimensional Case". In: *Proceedings of the 15th SIAM International Conference on Data Mining (SDM)*, pp. 289–297.
- Shwartz-Ziv, R. and A. Armon (2022). "Tabular data: Deep learning is not all you need". In: *Information Fusion* 81, pp. 84–90.
- Silver, D., A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489.
- Singh, V. P., J. M. S. Rohith, and V. K. Mittal (2015). "Preliminary analysis of cough sounds". In: *Proceedings of the 2015 Annual IEEE India Conference (INDICON)*, pp. 1–6.
- Sipser, M. (2013). *Introduction to the Theory of Computation*. 3rd. Course Technology.
- Sistla, A. P. and E. M. Clarke (1985). "The Complexity of Propositional Linear Temporal Logics". In: *Journal of the ACM* 32.3, pp. 733–749.
- Sistla, A. P., V. Gyuris, and E. A. Emerson (2000). "SMC: a symmetry-based model checker for verification of safety and liveness properties". In: *ACM Transactions on Software Engineering and Methodology* 9.2, pp. 133–166.
- Srivastava, N. and R. Salakhutdinov (2013). "Discriminative Transfer Learning with Tree-based Priors". In: *Proceedings of the 26th Advances In Neural Information Processing Systems (NIPS)*, pp. 2094–2102.
- Stan, I. E., G. Sciacavico, E. Muñoz-Velasco, Giovanni Pagliarini, Mauro Milella, and Andrea Paradiso (2022). "On Modal Logic Association Rule Mining". In: *Proceedings of the 23rd Italian Conference on Theoretical Computer Science (ICTCS)*. Vol. 3284. CEUR Workshop Proceedings. CEUR-WS.org, pp. 53–65.
- Suárez, A. and J. F. Lutsko (1999). "Globally Optimal Fuzzy Decision Trees for Classification and Regression". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.12, pp. 1297–1311.
- Sutskever, I., O. Vinyals, and Q. V. Le (2014). "Sequence to Sequence Learning with Neural Networks". In: *Proceedings of the 28th Advances on Neural Information Processing Systems (NIPS)*, pp. 3104–3112.
- Sweileh, W. M. (2020). "Bibliometric analysis of scientific publications on "sustainable development goals" with emphasis on "good health and well-being" goal (2015–2019)". In: *Global Health* 68.16.
- Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich (2015). "Going deeper with convolutions". In: *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.
- Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). "Rethinking the Inception Architecture for Computer Vision". In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826.
- Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi (2017). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, pp. 4278–4284.
- Tan, P., M. S. Steinbach, and V. Kumar (2005). *Introduction to Data Mining*. Addison-Wesley.
- Tan, P.-N., M. S. Steinbach, A. Karpatne, and V. Kumar (2019). *Introduction to Data Mining*. 2nd. Pearson.

- Tani, T., M. Sakoda, and K. Tanaka (1992). "Fuzzy modeling by ID3 algorithm and its application to prediction of heater outlet temperature". In: *Proceedings of the 1992 IEEE International Conference on Fuzzy Systems*, pp. 923–930.
- Topol, E. J. (2019). "High-performance medicine: the convergence of human and artificial intelligence". In: *Nature Medicine* 25.1, pp. 44–56.
- Tsang, E. C. C., X. Wang, and D. S. Yeung (2000). "Improving learning accuracy of fuzzy decision trees by hybrid neural networks". In: *IEEE Transactions on Fuzzy Systems* 8.5, pp. 601–614.
- Tsuchiya, T., T. Maeda, Y. Matsubara, and M. Nagamachi (1996). "A fuzzy rule induction method using genetic algorithm". In: *International Journal of Industrial Ergonomics* 18.2, pp. 135–145.
- Turing, A. M. (1950). "Computing Machinery and Intelligence". In: *Mind* LIX.236, pp. 433–460.
- Utgoff, P. E. (1988). "ID5: An Incremental ID3". In: *Proceedings of the 5th International Conference on Machine Learning (ICML)*, pp. 107–120.
- (1989). "Incremental Induction of Decision Trees". In: *Machine Learning* 4, pp. 161–186.
- (1994). "An Improved Algorithm for Incremental Induction of Decision Trees". In: *Proceedings of the 11th International Conference on Machine Learning (ICML)*, pp. 318–325.
- Utgoff, P. E., N. C. Berkman, and J. A. Clouse (1997). "Decision Tree Induction Based on Efficient Tree Restructuring". In: *Machine Learning* 29.1, pp. 5–44.
- van den Oord, A., S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu (2016). "WaveNet: A Generative Model for Raw Audio". In: *The 9th ISCA Speech Synthesis Workshop (SSW)*, p. 125.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). "Attention is All you Need". In: *Proceedings of the 30th Advances in Neural Information Processing Systems (NIPS)*, pp. 5998–6008.
- Venema, Y. (1990). "Expressiveness and Completeness of an Interval Tense Logic". In: *Notre Dame Journal of Formal Logic* 31.4, pp. 529–547.
- (1991). "A Modal Logic for Chopping Intervals". In: *Journal of Logic and Computation* 1.4, pp. 453–476.
- Vidal, A., F. Esteva, and L. Godo (2017). "On modal extensions of product fuzzy logic". In: *Journal of Logic and Computation* 27.1, pp. 299–336.
- Wan, A., L. Dunlap, D. Ho, J. Yin, S. Lee, S. Petryk, S. Adel Bargal, and J. E. Gonzalez (2021). "NBDT: Neural-Backed Decision Tree". In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.
- Wang, F., M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang (2017). "Residual Attention Network for Image Classification". In: *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6450–6458.
- Wang, Q. R. and C. Y. Suen (1987). "Large Tree Classifier with Heuristic Search and Global Training". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9.1, pp. 91–102.
- Wang, X., B. Chen, G. Qian, and F. Ye (2000). "On the optimization of fuzzy decision trees". In: *Fuzzy Sets and Systems* 112.1, pp. 117–125.
- Wang, Z., W. Yan, and T. Oates (2017). "Time series classification from scratch with deep neural networks: A strong baseline". In: *Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1578–1585.

- Yamada, Y., E. Suzuki, H. Yokoi, and K. Takabayashi (2003). "Decision-Tree Induction from Time-Series Data Based on a Standard-Example Split Test". In: *Proceedings of the 12th International Conference on Machine Learning (ICML)*, 840–847.
- Ye, L. and E. J. Keogh (2009). "Time series shapelets: a new primitive for data mining". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 947–956.
- Yuan, Y. and M. J. Shaw (1995). "Induction of fuzzy decision trees". In: *Fuzzy Sets and Systems* 69.2, pp. 125–139.
- Zeitouni, K. and N. Chelghoum (2001). "Spatial Decision Tree-Application to Traffic Risk Analysis". In: *Proceedings of the 2001 ACS / IEEE International Conference on Computer Systems and Applications (AICCSA)*, p. 203.
- Zhang, D., N. Maslej, E. Brynjolfsson, J. Etchemendy, T. Lyons, J. Manyika, H. Ngo, J. C. Niebles, M. Sellitto, E. Sakhaee, Y. Shoham, J. Clark, and R. Perrault (2022). *The AI Index 2022 Annual Report*. URL: <https://arxiv.org/abs/2205.03468>.
- Zhang, Q. and J. Wang (2003). "A rule-based urban land use inferring method for fine-resolution multispectral imagery". In: *Canadian Journal of Remote Sensing* 29.1, pp. 1–13.
- Zhou, J., G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun (2020). "Graph neural networks: A review of methods and applications". In: *AI Open* 1, pp. 57–81.
- Zhou, Z.-H. (2021). *Machine Learning*. Springer.
- Zhou, Z.-H. and Z. Chen (2002). "Hybrid Decision Tree". In: *Knowledge-Based Systems* 15.8, pp. 515–528.
- Zhou, Z.-H. and Y. Jiang (2004). "NeC4.5: Neural Ensemble Based C4.5". In: *IEEE Transactions on Knowledge and Data Engineering* 16.6, pp. 770–773.