UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN

MATEMATICA

CICLO XXXVI

Modal Symbolic Learning: From Theory to Practice

Coordinatore:
Prof. Leonardo BILIOTTI

Tutore:
Prof. Guido SCIAVICCO

Dottorando: Giovanni PAGLIARINI

Anni Accademici 2020/2021 – 2022/2023

Abstract

Symbolic Learning (SL) studies learning algorithms for computational models that rely on formal (or symbolic) logic and, as such, it provides AI models that perform transparent, *explicit* logical reasoning. Inspite of its central role in the Artificial Intelligence (AI) of the 80s/90s, the last two decades of research in AI had been marked by great advances in Deep Learning, which shifted the focus on highly accurate, black-box models. While this caused a slowdown in SL research, over the very last years, the field has been making a comeback as a promising solution to the so-called *black box problem*, due to the interpretable nature of its underlying knowledge representation.

In its current state, SL appears fragmented, underdeveloped, and it is generally not regarded as a serious competitor to Deep Learning. This is likely due to two major factors. First, while deep learning methods specialized to time-series, images, graphs, etc., most of the widespread SL approaches are still based on propositional logic or predicate logic, which are only suitable for static, tabular data, and for purely relational data, respectively. Second, while several established programming frameworks exist for Deep Learning (e.g., tensorflow, Theano, Deeplearning4j, scikit-learn), implementations of SL algorithms (when available) are scattered across different programming languages and libraries, making it difficult to appreciate the true potential of explicit knowledge representations.

This thesis addresses both of these gaps. First, with the aim of providing a modern, unifying view on the subject, we present SOLE, a comprehensive abstract framework for SL methodologies. The framework is based on a solid logical foundation that allow for the use of virtually *any* logical formalism, and covers (logic-specific and -agonistic) methods from Symbolic Data Analysis, Symbolic Learning and Symbolic Modelling. SOLE is also implemented and publicly available in the Julia programming language as Sole.jl, the first programming framework specifically tailored for symbolic methods. Second, we report recent, published experimental results showing how extensions of common SL methods to more expressive, *modal logics* can lead to interpretable and accurate models for tasks involving non-tabular data.

The experimental part of this thesis considers four different classification tasks involving multivariate time-series and images. For each task, we provide an experimental evaluation of a novel decision tree learning algorithm based on temporal/spatial modal logics. The learned models display medium to high accuracies, and qualitative inspections of the extracted knowledge reveal insights into the underlying processes of the classification problems at hand.

The objective of this work is to encourage research in SL, and to, ultimately, enable virtuous interactions between human and computational intelligence, when tackling hard tasks.

Acknowledgements

This thesis concludes a few years of study dedicated to modal logic, machine learning, ACLAI laboratory, and Julia. These main actors played a crucial role in my PhD path, and I would like to express my gratitude towards each of them.

Then, I want to thank my PhD supervisor, whom I was fortunate to cross path with. Prof. Guido Sciavicco, you are a grandmaster, you have nurtured our passions for years, and I am grateful to you for every bit of teaching.

I am thankful to the reviewers for putting real effort in reviewing this work, and helping me in improving the quality of this thesis.

Other valuable academics I want to thank are Prof. Ugo Rizzo, Prof. Dario Della Monica, and Prof. Valeria Ruggiero, all for different reasons.

I also want to express my gratutide towards Prof. Sasha Rubin: thank you for showing interest in our research work; thank you for your valuable feedback; and thank you for welcoming me in the beautiful Sydney for a period that I will always remember with great pleasure.

Thanks to my colleagues: Alberto, Anna Chiara, Arrigo, Elena, Francesca, Ilaria, and Matteo. Thank you Federico for all the fish. And thank you Eduard, because of your devotion to research, and because you paved the way I walked in many ways.

Thanks to my closest ones, Claudia, Marta, Aurora, Marco, Luigi, Fabio, Virginia, and Pranav.

Thanks to my family and, most especially, to my parents, Maurizio, and Manuela; two beautiful persons, I could not have done anything if it was not for your constant support.

Oh, and I want to thank the Julia community for building the best programming language that is out there now in 2024!

Contents

List of Figures

List of Tables

To my Ferrara.

CHAPTER 1

INTRODUCTION

"The Answer to the Great Question [...] Of Life, the Universe and Everything [...] is [...] Forty-two," said Deep Thought, with infinite majesty and calm. It was a long time before anyone spoke.

Douglas Adams, The Hitchhiker's Guide to the Galaxy

§1.1 Recent Trends in Artificial Intelligence

Artificial Intelligence (AI) is the subfield of computer science that studies and develops software or machines that can emulate aspects of human intelligence, such as deduction, creativity, speech recognition, ability to learn from past experience and make reasonable inferences from incomplete information [RN20]. While the field witnessed relatively stable progress for about five decades, the last twenty years saw enormous advancements. Recent revolutionary advancements in hardware engineering and neural network learning have propelled AI into a new era and, as of today, AI is ubiquitous, and it is one of the fastest-growing and most funded scientific disciplines. Noteworthy developments include the emergence of transformative technologies such as Deep Neural Networks, Large Language Models (LLMs) and Generative Pre-trained Transformers (GPTs). These technologies have successfully addressed complex challenges in diverse domains, including text generation, image processing, speech recognition, protein folding, and strategic games (e.g., Go and chess), bringing AI closer to achieving capabilities that were once deemed exclusive to human intelligence. Moreover, the advent of multimodal GPTs, capable of processing various data modalities like images, videos, sound, and text, signifies a pivotal step towards a more comprehensive AI that can handle diverse forms of information. This trajectory paves the way for the exploration of "general intelligence", marking an exciting and open path in the evolution of AI.

The disruptive, exponential growth and spread of AI, now widespread and often concealed in its applications, has been accompained by concerns over the potential risks. Ultimately, with applications to high-stakes contexts, ranging from military to industry and healthcare, and the field being compared to alchemy[1] due to the uncertainties that many of the practitioners feel, major debates over the use of AI have emerged, among both experts and the general public. Arguably this trend culminated, so far, with last year's "Pause Giant AI Experiments: An Open Letter", a call to stop the development powerful AI systems, due to perceived "profound risks to society and

[1] Prof. Ali Rahimi (Google), Test of Time Award speech, NIPS 2017

humanity". Although the purposes of this effort are questionable, this letter was signed by eminent researchers, intellectuals, and technology leaders, which emphasizes the need for a cautious approach and careful consideration of the potential consequences of advancing AI technologies. While the concerns regarding the use of AI are many and of different nature [RN20; Coe20; Bry22], ranging from privacy related ones, to accountability related ones, to more philosophically grounded ones[2], we focus, here, on a specific, pervasive problem affecting most modern AI methods.

§1.1.1 The Black Box Problem

According to reports [Das22], in 2014 Amazon Inc developed and began using an AI model for screening the resumes of job applicants. The goal of the model was to rate the candidates on a scale of 5 stars, depending on how closely they resemble prior successful candidates and, by the end of the year, this tool was heavily used throughout the company. In 2015, however, the tool was shown to systematically discriminate against female candidates. The bias was found to be caused by an under representation of successful female applicants in the training dataset. Eventually, Amazon took effort to limit gender biases, but concluded that such a system could theoretically develop into a system which is, at least to some degree, discriminatory. In the following years, a number of research studies showed similar biases in other contexts, which contributed to an increased interest in *machine fairness* [Zaf+17]; for example, racial biases were found in AI models for healthcare [Obe+19] and criminal law [DF18].

In most of these cases, a simple inspection of the model would not have sufficed to identify the bias: instead, the latter only became evident from a (oftentimes late) post-hoc analysis of the model's on-field behavior. This *lack of transparency* exemplifies one of the today's most debated problems: despite outstanding, often super-human performances, state-of-the-art AI technologies are so complex that even practitioners cannot explain their functioning. In many cases, in fact, modern AI fails to provide transparent and reliable intelligent agents, to the point where its application to sensitive domains is severely restricted over ethical concerns. Because of this, the last years have seen a growing interest in alleviating the so-called *black box problem*.

Black box behavior of this kind manifests prominently in neural models, the main pillars of modern AI, and it lies in the fact that their knowledge about the task is represented via mathematical functions that are too complex to be conveyed and understood by human intelligence. Because of this inherent complexity and unintelligibility, neural networks cannot be engineerd from scratch, and are, instead, conveniently induced via *bottom-up* (or *learning*) approaches, often based on data-driven optimization. While these technologies bring significant generalization power, they require large amounts of data, and yield models that encode the target function via unintuitive representations. This also emerges at a small scale. As an example, consider Iris [Fis88], a benchmark dataset for a rather simple classification task. Neural network training can yield optimal accuracy at the task, even by fixing a simple model structure comprehensive of only a few neurons. However, the model ends up solving the problem by computing complex, highly non-linear relations while, in turn, the task is easily solved by plain, thresholding-based

[2] as a sidenote, consider that an AI was recently named co-author of a scientific publication [OC23], and that a simple Google search for "AI PhD thesis generator" yields more than 67 million results.

rules [Dev+11]. When even a small neural network is overparameterized and encompasses an unnecessarily complex knowledge representation, the effect is even more evident for most of the sensational deep models mentioned above, including recurrent and convolutional neural networks (for tasks involving text, images, etc.), as well as LLMs and GPTs, which are based on much more sophisticated technologies (e.g., word embeddings and multi-head attention) and a number of parameters measured in trillions.

§1.1.2 Interpretability

Of course, the need for models which behavior is explainable by humans may depend on the specific domain of application, but it is such a pervasive topic[3], that it initiated new trends in AI specifically focusing on this kind of models. When entrusting high-stakes decisions to AI models, being able to interrogate the model about its inner workings has now become a fundamental right. In this regard, the European Union General Data Protection Regulation (GDPR), operative from 2018, implements a *right to explanation* by guaranteeing subjects "meaningful information about the logic involved" in any automated decision-making process based on their personal data [Gdp], thus expecting models to be available for scrutiny. Moreover, while AI can be used for simply automating hard tasks, any form of transparent AI can also help us gaining insights into the phenomenon that underlies the task and, ultimately, help us making decisions that are more informed. In this sense, great thinkers argue that AI could be used to *augment* humankind [Bry22], and that new types of interactions between humans and AI models should be put into action (e.g., see Human-in-the-loop Artificial Intelligence [Mos+23]).

Many researchers saw a solution in post-hoc interpretation methods for explaining the decision-making processes of neural networks, which developed into the field of Explainable Artificial Intelligence (XAI) [Gun+19]. Other authors pointed out the weaknesses of these post-hoc methods [Rud19; Bab+21], and advocate, instead, for models that are *inherently interpretable*, namely, models with a knowledge representation that is directly understandable to humans. Models with such an explicit knowledge representation can be debugged, audited, amended, and their knowledge can be mixed with human knowledge. Altogether, they offer many opportunities that functional, neural models do not, such as:

- Verifying that the model satisfies some properties, and is adequate for the given task;
- Learning of new insights by simple inspection of the model;
- Post-hoc manual injection of human knowledge into the model.

As a matter of fact, inspite of a possibly lower initial accuracy, interpretable modelling enables a continuous interaction between human and computational intelligence which, in principle, allows for truly understanding and solving hard tasks. In this regard, black box learning appears as delegating the solution of the task to a soundless, alien technology[4] and, confined by the uninterpretable nature of their knowledge representation, black box models reveal limited operating space.

[3] See the "The Great AI Debate" (NIPS 2017)

[4] Prof. Ali Rahimi (Google), Test of Time Award speech, NIPS 2017

§1.2 Symbolic Learning

One of the most promising approaches to interpretable modelling involves turning to what is now known as *Good Old-Fashioned Artificial Intelligence* (GOFAI). This expression refers to what was the dominant paradigm in AI until the late 1980s, when AI was virtually centered around (good) knowledge representation, and knowledge was predominantly encoded using *symbols*, that is, string tokens representing human-readable concepts. For a long time, Symbolic AI focused on rules engines (or *expert systems*), which could draw conclusions about entities and their relations from manually encoded axioms (e.g., Cyc [Len95]). These systems could, essentially, perform logical reasoning, and were based on formalisms proposed in the largely studied, millennial field of *symbolic logic*. Around the 80s, the focus shifted from top-down to bottom-up systems, and methods for *Symbolic Learning* (SL) flourished. Typical symbolic models are *decision trees*, which resemble nested structures of if-then-else constructs, and *decision lists*, which are more similar to switch-case constructs. In general, all SL methods share two common traits: they have a rather straightforward rule-based functioning, and they encode the extracted knowledge via logical formulas that, regardless of whether they are handcrafted or learned, they can always be translated into natural language.

Depending on the domain of application and the kind of reasoning required, different logical formalisms, with different expressive power and algorithmic complexity, have been adopted. Until the 1990s, Symbolic Learning focused on non-relational (or *tabular*) data and all systems were, more or less consciously, based on propositional logic, where formulas are, essentially, Boolean expressions, such as:

$$(\text{Age} > 15 years \ \wedge \ \text{IsAthletic} = FALSE \ \wedge \ \text{HeartRate} < 60 bpm) \ \rightarrow \ \text{HasArrhythmia}$$

In contrast to this extremely simple formalism, new approaches during the 1990s proposed first-order logic, which is able to express quantified concepts in *purely relational data*, such as:

$$\forall x \forall y \ (\exists z \ (\text{Parent(x,z)} \wedge \text{Parent(z,y)}) \rightarrow \ \text{Grandparent(x,y)}) \ .$$

At the time, however, while neural models were inherently deficient in logical reasoning abilities, symbolic methods were deemed unable to express uncertain and approximate concepts [Min91]. As mentioned, then, the conundrum between symbolic and neural (also called *subsymbolic*) AI eventually took a back seat, and research in symbolic methods experienced a sudden slowdown.

Today, the resumption of symbolic methods faces a few challenges. The World Wide Web, the proliferation of digital devices (e.g., smartphones, smartwatches, and sensors), together with the advances in storage and hardware engineering, launched us in the era of big data [GH15], where non-tabular data of different kinds is available for any virtually any domain, including: datasets of billions of images, billions of hours of speech and video, trillions of words of text, genomic data, and sensory data, among others. Neural network learning technologies evolved and specialized on the different kinds of non-tabular data, which embodies at least 80%–95% of all existing data [GH15], As of today, instead, Symbolic Learning methods are still preferred over deep learning ones in the realm of tabular data [GOV22; SA22]; we regard this fact as double-edged sword which perpetuates the idea that Symbolic Learning must be inherently tied to simple logical

formalisms. In this regard, propositional modal logics, which enable a limited form of relational reasoning that can be tailored for specific purposes, recently lead to interesting results. For example, learning algorithms based on spatial/temporal modal logics have been shown to capture well the complexity of some classification problems involving multimodal, multidimensional data, such as time-series and images [Man+23a; Bec+23; Cas+23; PS23; DM+22].

§1.3 SOLE: A Framework for Symbolic Learning

We claim that, as a research field, SL results fragmented on multiple dimensions. Whereas several established programming frameworks for Deep Learning exist (e.g., tensorflow, Theano, Deeplearning4j, scikit-learn, Flux), no unified codebase exists for learning, representing and manipulating symbolic models. Moreover, implementations of SL algorithms currently only exist scattered across different programming languages (primarily, Python, Java, R, Julia) and Machine Learning programming frameworks (primarily, Scikit-learn [Ped+11a], Weka [WF99; Hal+09], mlr [Bis+16], and MLJ [Bla+19]) that, at best, offer very limited tools for manipulating the extracted knowledge in its logical form. As an example, consider Weka, a suite of ML algorithms that arguably gets the closest to covering the role of *SL programming framework*. Weka provides learning algorithms for decision trees and decision lists, but offers no way of mixing the knowledge enclosed in different symbolic models, nor it offers any tool for handling the extracted patterns as logical formulas. This makes it somehow difficult to appreciate the true essence of explicit knowledge representations and, in a way, it leaves the potential unexploited. The lack of a programming framework for SL approaches is likely correlated with a lack of a unified, abstract view on the topic. Different SL approaches use different terminologies, revealing a general lack of formal, comprehensive foundations. While endeavours to unify the neural network learning literature under the same abstract framework exist (e.g., *Geometric Deep Learning* [Cao+20]), the same is, therefore, needed for Symbolic Learning.

In this thesis, we present SOLE, a novel, abstract framework for (modern) Symbolic Modelling and Symbolic Learning. The framework provides a unifying view on the subject, and revolves around the following observation: most of the existing symbolic models base their functioning on the process of checking the truth a logical formula on an input data instance. In logical terms, the instance can be seen as an *interpretation* of a given logical formalism, and this largely studied process is commonly referred to as *model checking*. As such, SOLE builds on three main pillars:

- Symbolic logic, which studies logical formalisms, formulas and checking algorithms;
- A symbolic interpretation of Machine Learning datasets, which act as sets of logical interpretations onto which formulas can be checked;
- A formalization of symbolic models as computational models with rule-based, *if-then* semantics, that check the truth of formulas on the input data.

As depicted in Fig. 1.1, these three pillars allow for the definition of the symbolic counterparts to standard, functional data analysis, learning, and (quite importantly) post-hoc model analysis.

A first implementation of SOLE is currently available as a set of Julia packages (Sole.jl), which collectively make up the first programming framework specifically designed for Symbolic

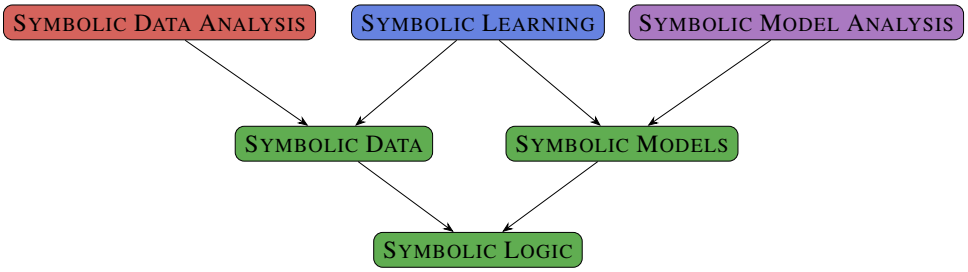


FIGURE 1.1: Building pillars of the SOLE framework.

Learning. The implementation provides tools for representing symbolic models and manipulating their knowledge base, ultimately facilitating fast prototyping of new symbolic approaches for today's hard tasks on possibly (non-)tabular and/or multimodal data. Aside from offering tools for accomodating higher-order and customized logical formalisms, the Julia implementation supports propositional logic, and spatial/temporal modal logics for reasoning on (non-)tabular data, including: time-series, audios, videos, texts, images, graphs, etc.

SOLE is the product of years of research in SL carried out at the Applied Computational Logic and Artificial Intelligence (ACLAI) Laboratory at the University of Ferrara (Ferrara, Italy)[5]. This endeavour has a specific focus on the potential of propositional modal logics, and it is, therefore, defining and uncovering the benefits of *Modal Symbolic Learning*. This thesis can be seen as the natural sequel to Dr. Eduard I. Stan's PhD thesis, titled *Foundations of modal symbolic learning* (2023), which introduced a first formalization of modal decision trees.

§1.4 Thesis Structure

This thesis is organized into two parts. The Part I proposes SOLE as the first comprehensive framework for SL, and presents three (modal logic-based) symbolic algorithms for feature selection, and decision tree learning and analysis. In Part II, the experimental results of four scientific works are presented, where SOLE and its Julia implementation were put into use for learning, manipulating and analysing decision trees on tasks involving non-tabular, multimodal data. A description of each of the upcoming chapters follows:

- Chapter 2 provides an incomplete overview of Symbolic Learning. The overview covers the history of the field and a few of its major research lines, such as decision tree learning, decision list learning and association rule mining;
- Chapter 3 presents SOLE, from its logical foundations, to the data and model representation, the three pillars onto which algorithms can be designed for analysing symbolic data, and learning and analysing symbolic models (see Fig. 1.1). The framework provides a unified view on the different symbolic models reported in the previous chapter.

[5] https://aclai.unife.it/

- Chapter 4 presents three techniques based on propositional modal logics for symbolic feature extraction (ModalFILT), decision tree learning (ModalCART), and post-hoc processing of symbolic models (ModalETEL), respectively;
- Chapter 5 introduces to Part II of the thesis, which reports a few experimental results using ModalCART;
- Chapter 6 reports an application of ModalCART to a land cover classification task involving aereal imagery;
- Chapter 7 reports an application of ModalCART to automated diagnosis of COVID-19 from audio recordings of cough and breath events;
- Chapter 8 reports an application of ModalCART for predicting anomalous behaviours in industrial gas turbines;
- Chapter 9 reports an application of ModalCART for the prediction of the experienced level of beauty from EEG recordings;
- Chapter 10 concludes the thesis by discussing the future of Symbolic Learning and the SOLE framework.

§1.5 Published Results

As of January 2024, I am co-author of about 17 scientific publications on Symbolic Learning, of which 5 are journal articles, and the rest are conference papers. This section surveys my publications, relating them to the content of this thesis.

In Della Monica et al. [DM+22], a formalization of modal decision trees and the ModalCART algorithm for learning them are given. Modal decision trees are an extension of traditional decision trees to propositional modal logics, that allows them to express modal formulas. Since modal logic can be tailored for temporal and spatial reasoning, this novel method allows for the extraction of patterns on temporal, spatial, spatio-temporal and graph data. In Manzella et al. [Man+23b] an efficient implementation for a restricted version of ModalCART is presented. The implementation leverages memoization in order to perform optimized finite model checking. In Pagliarini, Sciavicco, and Stan [PSS21], the multimodal extension of ModalCART is briefly discussed. The algorithm, in its multimodal form, is currently available as a Julia package [Pag+24].

In Pagliarini et al. [Pag+22] neuro-symbolic hybridization opportunities for time-series classification between modal decision trees and neural networks are investigated and evaluated. In Pagliarini et al. [Pag+23], the problem of minimizing class-formulas extracted from modal decision trees is tackled, with a solution based on three heuristic steps. In Cavina et al. [Cav+23], ModalFILT, a set of symbolic methods for feature selection on dimensional data, such as tabular, time-series, image and video are presented and evaluated. In Ghiotti et al. [Ghi+23], ModalETEL, an evolutionary approach to rule extraction from ensembles of modal decision trees is presented and evaluated. In Stan et al. [Sta+22], the APRIORI algorithm for association rule mining is extended to the modal level, laying the foundations for the implementation of this learning

algorithm. In Bonaccorsi et al. [Bon+21], a clinical decision support system is presented, based on rule extraction algorithms for tabular, clinical data. The system is used in everyday operation protocol, and it allows physicians to receive suggestions for prevention and treatment of osteoporosis.

In Pagliarini and Sciavicco [PS21], and its extended version [PS23], ModalCART and its extensions are applied to image datasets for Land Cover Classification, proving to constitute valid alternatives to both neural methods and propositional symbolic methods. A qualitative comparison on the appropriateness of the extracted knowledge is also reported. In Manzella et al. [Man+21] and its extended version [Man+23a], ModalCART and its extensions are applied to cough and breath audio recordings, for automated diagnosis of COVID-19. In Bechini et al. [Bec+23], ModalCART is applied in the the industrial domain to predict trip events (i.e., anomalous behaviours) in gas turbines. In Coccagna et al. [Coc+22], and its extended version [Cas+23], ModalCART and its extensions are applied to EEG recordings, for predicting the experienced level of beauty upon exposition of subjects to art paintings.

With the exceptions of Bonaccorsi et al. [Bon+21], Pagliarini, Sciavicco, and Stan [PSS21] and Pagliarini et al. [Pag+23], all works include experimental results obtained using the Sole.jl programming framework, of which I have, so far, coordinated the development. Part I of this thesis includes the presentation of the ModalFILT feature selection algorithm (Section 4.3), the ModalCART learning algorithm (Section 4.2), and the ModalETEL algorithm for simplifying modal decision tree ensembles (Section 4.1). Part II of this thesis reports experimental evaluations of ModalCART from Pagliarini and Sciavicco [PS23], Manzella et al. [Man+23a], Bechini et al. [Bec+23] and Caselli et al. [Cas+23].

Part I

Theory

CHAPTER 2

OVERVIEW OF SYMBOLIC LEARNING

If you think you're doing something original you're wrong.

IOSONOUNCANE, Il corpo del reato

In this chapter, we summarize research efforts in Symbolic Learning. This term is generally accepted for referring to research fields such as decision tree learning, association rule mining, inductive logic programming, and rule learning/extraction/induction; however, as a field, Symbolic Learning displays a critical fragmentation, which also emerges from the fact that the above areas often overlap each other, and refer to the *learning from example* process using different terms: learning, mining, extracting, and inducing. As a matter of fact, throughout the decades, different communities have attacked symbolic modeling and learning using different terminologies: from engineers [Koh95; AS94], to statisticians [Agr+96], and up to practitioners of Information Retrieval [Che95], Knowledge Representation [Did90], Knowledge Discovery [BDR96], Data Mining [Qui86; Qui93], and, naturally, Machine Learning (ML). To the best of our knowledge, no extensive survey on Symbolic Learning exists, while several surveys and books exist on the individual subfields.

While there is no general consensus on its relation with modern ML, in this context we use this term to precisely devise the subfield of ML techniques that, either knowingly or not, base their functioning on symbolic logic (or formal logic). In fact, while logicians had comparatively little influence on the field, many methods in the literature for the above areas can be seen as relying on checking the truth of formulas of logical formalisms, which is a core concept of formal logic, typically stylized as $\models$. Other than the underlying logical nature of the resulting model, Symbolic Learning has precisely the same objectives as ML, but also offers advantages in terms of transparency of the resulting models. We, hereby, only focus on supervised and unsupervised learning methods, where the goal is to train a statistical model on a dataset of labelled or unlabelled instances, respectively. Following the standard Machine Learning taxonomy, the nature and presence of the labels lead to different types of tasks: classification tasks (with categorical labels), regression tasks (with numerical labels), or unsupervised tasks (with absence of labels).

The roots of Symbolic Learning can be traced back to the dawn of computational AI, as of today, most of the literature is based on propositional and predicate formalisms. Propositional logic, sometimes referred to as *zeroth-order* logic, is able to express concepts via Boolean logical operators such as $\wedge$, $\vee$ and $\neg$. Despite its limits in terms of expressive power, this formalism have found fruitful applications in many domains and covers the vast majority of the Symbolic

Learning literature, with important algorithmic advancements between the 1980s and 1990s [Che95].

During the 1990s, technological advancements and the need for learning methods for relational data shifted the focus to more expressive formalisms and, more specifically, to predicate logic. Predicate logic, also referred to as *first-order* logic, is able to express relational and quantified statements, and while *Propositional Symbolic Learning* can only naturally deal with tabular data, *First-order Symbolic Learning* can learn patterns from relational, graph-structured data. Many efforts were made for extending existing methods to first-order logics, an endeavor that was named *relational learning* (RL) [Rae08; Qui90], *inductive logic programming* (ILP) [BDR96; MR94; RB97; Mug91], or *relational data mining* (RDM) [Dze06; Dze10], depending on the context.

With the turn of the millennium, rapid advancements in Deep Learning enabled the solving of hard tasks on data with relational components, leaving the development of Symbolic Learning behind. Big data shifted the focus on hard problems involving speech, visual data, graphs, time-series and, in general, non-tabular data. As of today, SL still dominates the domain of tabular data over Deep Learning [SA22; GOV22], while symbolic methods for non-tabular data are still outnumbered.

Very recently, attempts were made to extend propositional methods to propositional modal logics, including, among others [DM+22], Signal Temporal Logic [Bom+16; BB21; PDN21; Bar+22; PMN21; Bru+23], (Metric) Interval Temporal Logic [BBS14; Man+23a; Bec+23; Cas+23], and Spatial Logics [PS23; Che+21]. . Propositional modal logics are fragments of first-order logic offering trade-offs between zeroth-order and first-order in terms of expressivity and computational complexity. In terms of computability, these logics are at the verge of decidability, with some logics for which the *satisfiability* problem is decidable (e.g., Pnueli [Pnu77] and Clarke, Emerson, and Sistla [CES86]), and others for which it is undecidable (e.g., Halpern and Shoham [HS91a]), but which can be restricted furthermore to obtain computationally efficient formalisms (e.g., Muñoz-Velasco et al. [Muñoz-V+19]). More importantly, modal logic can be tailored for temporal or spatial reasoning; thus *Modal Symbolic Learning* (as coined in Della Monica et al. [DM+22]), can induce logical patterns from temporal data (e.g., panel, time-series, text data), spatial data (e.g., image data), spatio-temporal data (e.g., climate, video data), as well as relational data.

As a subfield of ML, Symbolic Learning can be broadly divided into Supervised Symbolic Learning (sometimes also called *predictive data mining*), and Unsupervised Symbolic Learning (sometimes also called *descriptive data mining*) Moreover, learning methods can be divided into *tree-based* (e.g., decision tree learning) and *rule-based* (e.g., decision list learning, association rule learning).

§2.1 Supervised Symbolic Learning

The major line of research in supervised SL is decision tree learning, or *tree-based* learning, which is one of the most well-known supervised machine learning method, covering a great share of the SL literature, with some authors using the two as synonyms [Che95]. In spite of their age,

according to a recent Kaggle survey[1], tree-based learning methods and their extensions (forests and boosted trees learning) are still currently among the most commonly used ML algorithms.

The origin of the development of modern decision trees dates back to the 1950s, with tree-like models for producing executable rules, such as Belson [Bel56], and tree-like models were proposed as alternatives to functional regression methods, such as Morgan and Sonquist [MS63]. The proposed learning algorithms typically build the tree in a top-down, heuristic and recursive fashion, by partitioning the input data based on highly discriminating variables [HSM66]. Algorithms for classification came later, with *THeta Automatic Interaction Detection* (THAID) [MM72] and *CHi-squared Automatic Interaction Detection* (CHAID) [Kas80], that introduced more sophisticated loss functions, such as chi-squared test and theta statistics. These approaches were found to be affected by overfitting, which prompted the scientific community to gain interest in investigating further. Around the same years, learning a minimal, optimal decision tree was also shown to be NPTIME-Complete [HR76].

In the 1980s, renowned work by Breiman and Quinlan were iconic in regenerating interest in the subject, with the *Classification And Regression Trees* (CART) [Bre+84], *Iterative Dichotomizer 3* (ID3) [Qui79; Qui86], ASSISTANT [Kon84], M5 [Qui+92; WW97], and C4.5 [Qui93] algorithms. With these algorithms, loss functions were introduced from information theory, such as Shannon entropy [Sha49] or Gini coefficient [Gin21], as well as different splitting criteria, such as information gain and gain ratio. Quinlan, in particular, gave a logical interpretation to decision trees: each decision is easily expressed as a logical atom; since alternative branches can be seen as logical disjunctions, and successive decisions on the same branches as logical conjunctions, a decision tree as a whole can be seen as a set of propositional logical formulas, representing the theory of the dataset on which it is learned. To cope with the overfitting problem, they also introduced some *pruning techniques* to regularise the resulting model [Qui99], such as *reduced error pruning* (REP) [EK01]. More recent contributions include VFDT [HSD01], C5.0 [Qui04], LMT [LHF05] and CUBIST [Kuh+13].

This solid and renowned research delineates the traditional tree-based learning, which is inherently based on propositional logic, and focuses on top-down learning of binary tree classifiers for tabular, labelled instances. Tree classifiers typically encompass a rooted, ranked, directed tree structure, with internal, split nodes checking a simple atom on the input instance (interpreted as a logical interpretation), and final, leaf nodes holding output values. The property to be tested is often a comparison involving a single variable. The extension to regression is straightforward, with variance measures as a loss function to minimize, and numerical values at the leaves. Decision tree models also encompass cluster hierarchies, hence the extension to the clustering context is straightforward [RB97; LXY00; TF06; BD98]. Many independent extensions to this framework have been proposed, involving the algorithm itself, the loss function, or the shape of the learned tree. For example:

- The atoms to be checked can involve multiple variables (*multivariate splits*). Of particular interest are oblique trees [MKS94], where the conditions are made on linear combinations of the variables;

[1] https://www.kaggle.com/kaggle-survey-2021

- To limit performance-affecting local optima, the learning algorithm can be equipped with lookahead, either in terms of algorithmic recursion, or by extending the formulas to be checked at the internal nodes to be more complex formula than simple atoms (e.g., disjunctions of atoms);
- The splits (and the resulting learned tree) can be more-than-binary;
- The technique can be coupled with bootstrap aggregation methods, known as *bagging* [Bre96], resulting in the commonly known *random forest* models [Bre01], that is, ensembles of trees where the prediction of many trees is averaged;
- The technique can be coupled with *boosting* [KV94] techniques. For example, the commonly known *gradient-boosted trees* approach (e.g., see [Fri01; CG16; Ke+17]) learns many decision trees sequentially, with each subsequent model correcting the errors of the previous model, eventually converging to an optimal metamodel. Note that, in general, both bagging and boosting reliably bring performance improvements [HTF09], but also yield less interpretable models; nevertheless, efforts to recover the simplicity of the model exist (e.g., Deng [Den19]);
- The standard, Boolean logical framework based on two truth values can be replaced with the *fuzzy* (or many-valued) counterpart (from the early work of Łukasiewicz, Post, and Tarski), that allows for more than two truth values [Háj13], producing so-called *fuzzy decision trees* [CP77]. In ML terms, the fuzzy framework can be used to accomodate fuzzy information (e.g., fuzzy inputs, fuzzy targets), and the the literature in this field is vast (see [CC87; Jor94; JJL97; SL99c; WS87; CS92; YS95; Ich+96; Tsu+96; AZZ98; Hay+98; Jan98; BW99; Wan+00; TWY00; OW03]). Fuzzy rules can also be extracted from a propositional tree [TST92; Jan94; CY96]. A recent survey on fuzzy trees and forests can be found in Sosnowski and Gadomer [SG19].

For a more exhaustive and systematic survey on the history of propositional decision trees for classification and regression, refer to one of the many available in the literature [RM05; SL91; de 13; Loh11; Loh14; Kot13; CA21].

More recently, computational logicians proposed SAT-based exact algorithms for learning minimal decision trees [SS21; Nar+18; SCM21; JM20]. While theoretically exponential in time, these algorithms are fixed-parameter tractable (FPT), and they were shown to scale for publicly available datasets of practical interest. Finally, the framework can be extended to underlying logics that are more expressive than the propositional one. The most prominent example is represented by *first-order decision trees*, that initiated the field of inductive logic programming during the 1990s, with the TILDE [BD98], S-CART [KW01] and RRL-TG [DRB01] algorithms. For a survey on the topic, refer to Muggleton and Raedt [MR94]. In recent times, tree-based algorithms were extended to propositional modal logics, with approaches for learning formulas of Signal Temporal Logic [Bom+16; BB21] from continuous signals, and a general, modal extension of CART called ModalCART [DM+22; Man+23b], which showed promising results for classification tasks on time-series and images [Man+23a; Bec+23; Cas+23; PS23].

Hybrid, *neuro-symbolic* approaches mixing decision tree learning with neural network learning have been proposed, combining the strengths of both symbolic and so-called *subsymbolic* methods. In general, the approaches fall into three categories:

- Neural networks are instantiated from learned decision trees (e.g., see [Set90; Bre91; IK95; SL99a; Kub98])
- Decision trees are instantiated from learned neural networks (e.g., see [CS95; KSB99; SAG99; DMB04; ZJ04]).
- Finally, hybrid neuro-symbolic models are designed and learned (e.g., see [LFJ92; GG92a; SL99b; ZC02; Mic+12; SS13; HVD15; Kon+15; Mur+16b; Mur+16a; Ala+21; Wan+21]).

Another relevant family of Supervised SL methods concerns the learning of rule sets [FGL12]. This research area which dates 1960s, with the early AQ [Mic69; Mic+86] separate-and-conquer algorithm, for discovering patterns that completely partition and cover the input data; several variants of the AQ algorithm followed (e.g., INDUCE [Mic80] and AQ11 [Mic83]). Following the rapid growth of interest in decision trees, many algorithms for so-called *rule learning* are devised around the 1990s. Around the same years of ID3 and C4.5, Quinlan proposed C4.5rules [Qui87], an algorithm which functioning is based on tree-based learning, but outputs a set of classification rules. Stemming from REP methods for propositional decision trees, many algorithms for learning *ordered* sets of classification rules followed; the term *decision list* is proposed [Riv87] for referring to such models, although other expressions are commonly used in the literature, such as *decision tables*, or *rule-based models*. In 1989, the CN2 algorithm is devised [CN89], incorporating ideas from ID3 [Qui86] and AQ [Mic+86]. this initiated the well-known approach to decision list learning called *sequential covering*. This approach builds the rule set in a iterative fashion, at each iteration producing a rule that covers a share of the yet-uncovered instances, ultimately obtaining a list that covers the entire instance space. The rules are learned for covering the classes in order of increasing frequency: the algorithm starts with the least common class, learns a rule for it, it eliminates all covered instances, and then moves on to the second least common class, and so on, until the last class, which constitutes the default rule. Depending on the algorithm, sophisticated heuristics can be used throughout the process. Algorithms that follow this approach are PRISM [Cen87], GREEDY3 and GROVE [PH90], GROW [Coh93], OneR [Hol93], IREP [FW94], RIPPER [Coh95a]. Other contributions include the IDTM [Koh95], RIDOR [GC95], PART [FW98], and PRIM [FF99].

Contributions exist based on first-order formalisms: FOIL [Qui90], mFOIL [GL96], BEXA [TC96], PROGOL [Mug95], ALEPH [Sri01] and OPUS [Web95]. More recent contributions include fuzzy extensions (such as FRBS [Riz+15] and FURIA [HH09], the fuzzy counterpart to RIPPER), ensemble methods (e.g., ISLE [FP+03], PREs [FP08], and RuleFit [FP08]), GEDEL [Sön+08] which learns decision lists via a genetic algorithm, CORELS [Ang+17] which is a non-greedy, exact approach, and methods for extracting decision lists from ensembles, such as STEL and InTrees [Den19], SIRUS [Bén+21], RuleCOSI [OKJ19], RuleCOSI+ [OJ23], BELLATREX [Ded+23], and ModalETEL [Ghi+23].

Decision list classifiers can be built from unsupervised rule mining techniques, resulting in the so-called *associative classification* (or *classification by association*) [LHM98; BNZ11].

Algorithms proposed include: CAEP [Don+99], CBA [LMW00], CMAR [LHP01], PRM and CPAR [YH03], ARMC [Zem+08], RCAR [ARB19], and qCBA [KI23].

Similarly to decision trees, neurosymbolic extensions have also been proposed for decision lists, both in the propositional (RuleNet, 1991 [MMS91]) and in the first-order case [Gla+22; CR21; PH11; HP15; Sen+22].

§2.2 Unsupervised Symbolic Learning

In the supervised setting, Symbolic Learning typically involves discovering association between logical patterns in the (unlabelled) data. In this context, rule-based learning methods are widespread, and make up a well-studied subfield of data mining called *association rule mining*.

First defined in Agrawal and Srikant [AS94] and in Agrawal, Imielinski, and Swami [AIS93], the problem of extracting association rules from (large) datasets is often presented in the context of *market basket analysis* [Agr+96; HPY00], and it involves discovering customer purchasing patterns and revealing interesting groupings of products. Because of this initial formulation, the majority of the existing approaches use terminology borrowed from the supermarket analogy, where, for example, the dataset is a set of *transactions*, and the output is composed of frequent *itemsets* (or *patterns*), that is, groupings of products that are correlated. Frequent itemsets generate association rules of the form $X \rightarrowtail Y$, where X and Y are itemsets, and are called *antecedent* and *consequent*, respectively. For example, a rule such as: {Beer} $\rightarrowtail$ {Diapers} indicates that if a customer transaction includes beer, than it likely also includes diapers; note that rules of this kind have a directionality that goes beyond simple correlation.

The widespread approach to extracting (all) association rules involves, first, extracting all frequent patterns, and second, generating all significant association rules from these patterns. The second task is relatively straightforward: to establish the significance of a rule ρ, measures of *support* and *confidence* are often considered. Support indicates how frequent a single pattern is, while confidence measures the probability of Y conditioned on X. More sophisticated measures (e.g., *lift*, *conviction*) can be engineered [GH06]. The first task is #P-hard [Val79; Yan04], and it involves a search in a exponential space (i.e., the power set of the item set). As this search cannot be treated via approximation algorithms, the existing approaches mainly differ by the subprocedure used to extract frequent patterns, and its optimization opportunities (e.g., parallelization).

In 1996, the APRIORI [Agr+96] algorithm (and a few variants) are presented. The approach is based on a breadth-first search, and leverages the *downward closure property*, namely, the monotonicity that frequency of patterns display in the search space, which was first pointed out by Valiant [Val79]. The following years, the Eclat [Zak+97], FP-Growth [HPY00] and LCM [Uno+03; UKA+04] algorithms are presented; Eclat is a depth-first approach, FP-Growth is an optimized method based on repeated projections of the frequent itemsets, and LCM is based on geometric operations (e.g., hypercube decomposition). As of today, heavily optimized implementations of these algorithms make it the state-of-the-art.

More recent approaches include RELim and its fuzzy version SaM [Bor10], PrefRec [MPR22], and extensions on transactions with structural components (e.g., *structured association rule mining*) as well as fuzzy extensions have been presented (e.g., [AR00; MF00; Jul+12; Mat+13;

LHT14; Ge+17; Hu+12; Wan+18]). Of particular interest is the case where the data is temporally distributed, (*sequential (or temporal) association rule mining*), in which APRIORI and FP-Growth are generalized by GSP [SA96] e PrefixSpan [Pei+04], respectively.

As seen in the previous section, techniques for association rule mining can be also used in supervised settings. A deeper overview of this field is outside the scope of this thesis, and can be found in Aggarwal et al. [Agg+15]; more recent surveys also exist [YK16; GT19; TGS20].

§2.3 Available Implementations

Numerous standardization efforts exist, but available implementations of symbolic learning methods are generally made by different scientific sub-communities, and scattered across different programming framework, often with different purposes. Most importantly, to the best of our knowledge, none of the programming frameworks involved leverages nor highlights the logical representations of knowledge. As of January 2024, established implementations of symbolic learning algorithms exist, primarily, in the WEKA [WF99; Hal+09] and Apache Spark [Zah+16] programming frameworks as libraries in the R [R C21] and Python [VRDJ95] programming languages.

As for tree-based learning methods, CART seems to be the most widespread method. WEKA provides implementations for C4.5, M5, LMT, VFDT. Apache Spark and R provides CART, and methods for bagging and boosting trees. In R, CART is available in the `tree` package, and methods for bagging and boosting in the `ranktreeEnsemble` package. R also provides an implementation for C5.0 (`C50` package). Python has implementations for C4.5 (ORANGE [Dem+13] and chefboost [Ser21] libraries), CART (scikit-learn [Ped+11b] and chefboost [Ser21] libraries), and ID3, CHAID, Regression Trees (chefboost [Ser21]). Implementations for CART-like learning algorithms also exists in the Julia programming language [Sad+22; Pag+24].

As for supervised rule learning, several R packages exist for CORELS (`corels` package), OneR (`OneR` package), PREs (`pre` and `rules` packages), PRIM (`prim` and `supervisedPRIM` packages), RuleFit (`xrf` package), FRBS (`frbs` package), inTrees (`inTrees` package), and SIRUS (`sirus` package). WEKA provides IDTM, RIPPER, 1R, RIDOR and PART; implementations exist for IREP and RIPPER are also available in the `wittengstein` Python library, while CN2 is available in ORANGE. An implementation for SIRUS [HBH23] also exists in Julia, and of RuleCOSI and RuleCOSI+ in Python [Obr19].

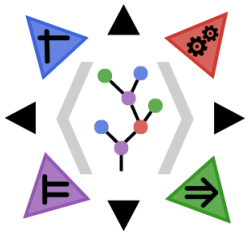
Several implementations are also available for unsupervised rule learning. APRIORI is available in R (`arules` package [HGH05]), WEKA, Python (`lmextend` library [Ras18]) and Julia [Ste+20]. `arules` also offers Eclat, and `arulesNBMiner` [Hah06] `arulesSequences` provides an implementation for cSPADE [Zak00] for sequence mining. FP-Growth is available in Apache Spark, Python (`lmextend` [Ras18] and `efficient-apriori` libraries) and WEKA. Apache Spark also provides PrefixSpan. R offers PrefRec (`RecAssoRules` package) for recommendation-based association rule mining. Implementations for associative classification algorithms are available in R's `arc`, `qCBA`, and `arulesCBA` packages [Hah+19], including: CBA, qCBA, CMAR, PRM, CPAR, and RCAR.

CHAPTER 3

THE SOLE FRAMEWORK

A day without sunshine is like, you know, night.

Steve Martin



As mentioned in the previous chapter, the terminology used in the literature for defining Symbolic Learning methods displays a wide variety. Some early attempts have been made at formalizing symbolic models (e.g., Rivest [Riv87]), but they are typically tailored to specific logical formalisms (e.g., propositional or first-order logics) and, although the concepts of rule, tree, and list are quite intuitive, no generally accepted definition exist.

In this chapter, we present a comprehensive formalization of both symbolic models and symbolic datasets, which provides a common language that makes few assumptions on the underlying logical formalism, and generalizes (at least) well-known definitions decision trees, decision lists, classification/regression rules, and association rules mentioned in Chapter 2. The presented formalization is composed of different layers, with some closer to standard notions of symbolic logic, and others closer to traditional symbolic learning methods. As such, we actually regard this formalization as an abstract framework upon which to base and design data-driven symbolic methods; we refer to this framework as SOLE. Note that the focus of this thesis is limited to propositional logics and propositional modal logics and, as such, the characterization of logical languages (Section 3.2.1) is limited to these kinds of formalisms; nevertheless, symbolic models based on higher-order logics (e.g., first-order models, ILP models), which generalize these formalisms, can be included in the framework by generalizing the logical foundations. Contextually, it must be noted that logical foundation of the framework encompasses well-known concepts in the symbolic logic literature, for which it provides no theoretical contribution.

This chapter begins with a presentation of the overall structure of the SOLE framework (Section 3.1). Subsequently, the main layers of formalization are presented: Section 3.2 introduces a view on well-known concepts of propositional and modal formal logics, formally defining logical formulas and the model checking problem; Section 3.3 lays the formal foundations for

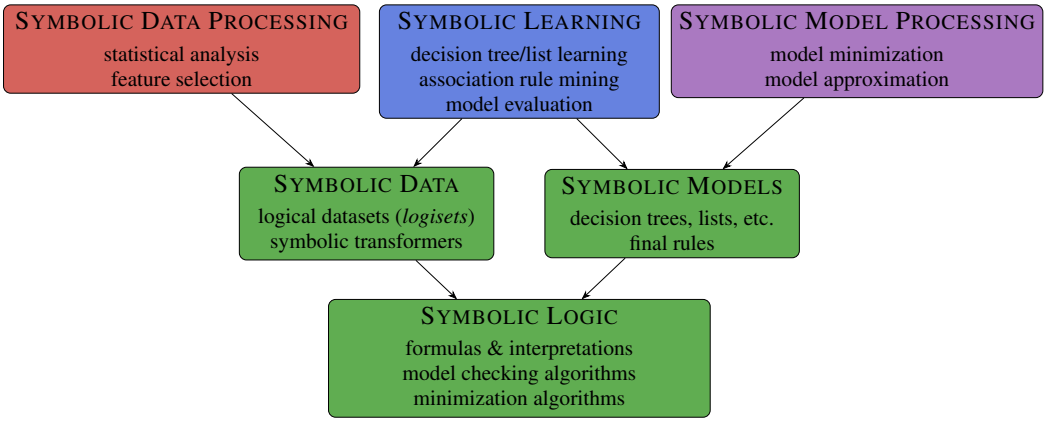


FIGURE 3.1: Layered architecture of the SOLE framework.

representing machine learning datasets in symbolic form; Section 3.4 uses the introduced notions for defining symbolic models as computational AI models based on logical formulas that process symbolic datasets. Section 3.5 contextualises Symbolic Data Processing, Symbolic Learning and Symbolic Model Processing within the framework, and discusses opportunities for iterative workflows. Finally, Section 3.6 comment on the state of the current Julia implementation of the framework.

§3.1 Architecture

Most of the existing (un)supervised symbolic models base their functioning on the process of checking whether an input data instance satisfies a property, encoded as logical formula. In logical terms, thus, the instance is seen as an *interpretation* of a given logical formalism, and this checking process is a largely studied one, often referred to as *model checking*. From this key observation (which, these days, we may fondly phrase as *(model) checking is all you need*), three main pillars for symbolic modelling emerge:

- Symbolic logic, which studies logical formalisms, formulas and checking algorithms;
- A symbolic rendition of Machine Learning datasets, which act as sets of logical interpretations onto which formulas are checked;
- A formalization of symbolic models as computational models with rule-based semantics, that check the truth of formulas on the input data instances.

Symbolic logical formalisms constitute the core engine, onto which symbolic modelling of data and models are, independently, defined. Then, as depicted in Fig. 3.1, these three modules allow for the definition of the symbolic counterpart to standard, functional data processing, learning, and model processing. Symbolic Data Processing (SDP) and Symbolic Model Processing (SMP) solely rely on the symbolic representations of data and models, respectively. Symbolic learning algorithms, instead, require symbolic data as input, and yield symbolic models as output. Note

that analysis and learning algorithms are typically tailored to specific logical formalisms, or classes of logical formalisms. As an example, the CART learning algorithm [Bre+84] assumes an underlying propositional logic; therefore, its input data is a dataset of propositional interpretations, and its output is a propositional decision tree model, that is, a decision tree based on propositional formulas.

§3.2 Symbolic Logic

Formal, symbolic logic has been defined as the *science of reasoning*, and it develops, studies and categorizes systems for representing knowledge and reasoning. Within the field, a wide variety of logical formalisms (or *languages*), was designed for different and many contexts; each language is characterized by its definition of *(well-formed) logical formulas*, expressing statements which truth can be evaluated, and of *logical interpretations*, representing objects which properties are expressed by formulas. Symbolic logic is deeply focused on the categorization of languages according to the their expressive power and computational properties; as a matter of fact, it is intimately related to the fields of complexity and computability [Dav18]. After overviewing some basic concepts in symbolic logic, this section gives a formalization of propositional logic, and well-known propositional modal logics useful for modern learning contexts.

§3.2.1 Basic concepts

A logical formalism, or *language* is a tuple $\mathrm{L} = (\Phi, \mathcal{I}, \models)$, where Φ is a set of *well-defined logical formulas*, $\mathcal{I}$ is a set of *logical interpretations*, and $\models \subseteq \mathcal{I} \times \Phi$ is a *truth relation*. Typically, Φ is inductively defined by a context-free grammar $\mathcal{G}$ on:

- a set of *atoms* $\mathcal{AP}$ (the *alphabet*), representing basic statements,
- a set of *truth values* $\mathcal{T}$, representing different degrees of truth, and
- a finite set of *connectives* C, representing different ways of composing statements,

with production rules of the types:

- $\mathrm{N} ::= p$, with $p \in \mathcal{AP}$,
- $\mathrm{N} ::= t$, with $t \in \mathcal{T}$, or
- $\mathrm{N} ::= c(\overbrace{\mathrm{N}, \ldots, \mathrm{N}}^{\mathfrak{a}(c) \text{ times}})$, with $c \in C$, and $\mathfrak{a}(c) \in \mathbb{N}^+$ being the *arity* of c,

where $\mathrm{N} \in \mathcal{N}$, and $\mathcal{N}$ is a set of nonterminal symbols. Atoms, truth values and connectives are referred to as *syntax tokens*, and the set of connectives can be written as the union of all connectives of increasing arity, up to the highest arity $\overline{\mathfrak{a}}$, as in $C = C_1 \cup \ldots \cup C_{\overline{\mathfrak{a}}}$. We use $p, p', \ldots$ to denote atoms, and $\varphi, \psi, \ldots$ to denote formulas.

In such context, a formula $\varphi \in \Phi$ is exhaustively represented via a *syntax tree*, namely, a ranked, rooted tree where each node v:

- holds a syntax token $\text{tok}(\nu)$,
- is, itself, root of a tree, and can be, thus, thought of as a (sub)formula (i.e., $\nu \in \Phi$), and
- has a (possibly empty) ordered set of children $\text{ch}(\varphi) \subseteq \Phi, |\text{ch}(\varphi)| = \mathfrak{a}(\text{tok}(\nu))$.

Given a formula, the definitions of *length*, *depth*, *connective depth* with respect to a subset of connectives $C' \in C$, *subformulas* and *atoms* follow:

$$
\begin{aligned}
l(\varphi) &= 1 && + \sum_{\psi \in \text{ch}(\varphi)} l(\psi) \\
d(\varphi) &= 1 && + \max_{\psi \in \text{ch}(\varphi)} d(\psi) \\
d_{C'}(\varphi) &= \begin{cases} 1 & \text{if } \text{tok}(\varphi) \in C', \\ 0 & \text{otherwise} \end{cases} && + \max_{\psi \in \text{ch}(\varphi)} d_{C'}(\psi) \\
sub(\varphi) &= \{\varphi\} && \cup \bigcup_{\psi \in \text{ch}(\varphi)} sub(\psi) \\
\text{atoms}(\varphi) &= \begin{cases} \{\varphi\} & \text{if } \text{tok}(\varphi) \in \mathcal{AP}. \\ \emptyset & \text{otherwise} \end{cases} && \cup \bigcup_{\psi \in \text{ch}(\varphi)} \text{atoms}(\psi)
\end{aligned}
$$

Typically, the length of a formula φ is also denoted as $|\varphi|$.
Furthermore, in most contexts, $\mathcal{G}$ is instantiated as a grammar with a single, starting nonterminal symbol φ, as in:

$$\varphi := p \mid t \mid c_1(\varphi) \mid c_2(\varphi, \varphi) \mid \ldots,$$

with $p \in \mathcal{AP}, t \in \mathcal{T}, c_1 \in C_1, c_2 \in C_2$, etc. Parentheses are typically elided for unary connectives, and infix notation is used for binary connectives; for example, $\neg\varphi$ and $\varphi_1 \wedge \varphi_2$ is used instead of $\neg(\varphi)$ and $\wedge(\varphi_1, \varphi_2)$, respectively.[1]

Given an interpretation $I \in \mathcal{I}$ and a formula $\varphi \in \Phi$, each language L defines whether $(I, \varphi) \in \models$ or not by structural induction on φ. Typically, interpretations are structures somehow assigning degrees of truth to the atoms, and $\models$ depends on an *interpret* function i_L such that $i_L(\varphi, I)$ computes a degree of truth of φ on I; in a way, this degree of truth represents the way that a certain property, expressed as φ, holds (or not) on the interpretation. A very common case occurs when $\mathcal{T}$ contains at least a value representing truth and one representing falsehood, typically denoted as $\top$ (*top*) and $\bot$ (*bottom*), respectively. In this case, the truth relation $\models$ is naturally defined as the set of all pairs (I, φ) such that $i_L(\varphi, I) = \top$. We often denote $(I, \varphi) \in \models$ by $I \models \varphi$, and read it as "I satisfies φ", "φ is satisfied by I" or "I is a *model of* φ". We denote the negation of this statement by $I \not\models \varphi$.

Similarly to algebraic mathematical languages, the logical language for expressing this property encompasses constants (the truth values), variables (the atoms) and uses operators for combining them (the connectives); as such, i_L results in an inductive routine which propagates truth values (both the constant ones in φ, and those depending on I) upwards in the syntax tree, with atoms and truth values generating base cases, and connectives generating induction cases. In

[1] Connective associativity and precedence are typically introduced to disambiguate ambiguouos cases.

a rather general setting, however, interpretations only provide degrees of truths for a subset of the atoms $\mathcal{AP}' \subseteq \mathcal{AP}$, which covers scenarios with missing information (e.g., unknown attributes for a given data instance). In this case, some atoms cannot be resolved to a truth value, so the interpret function becomes, more generally, a function which simplifies a formula using the information provided by the interpretation and, when the interpretation provides all the information needed, then the formula simplifies to a truth value: $\text{atoms}(\varphi) \subseteq \mathcal{AP}' \Rightarrow i(\varphi, I) \subseteq \mathcal{T}$. We, thus, refer to $i(\varphi, I)$ as the *simplification* of φ on I. Finally, some non-classical (e.g., modal) languages may need additional information for interpreting a formula; we model this information by introducing a set of *checking conditions* $\mathcal{P}$, and by, ultimately, defining i_L as a function: $\Phi \times \mathcal{I} \times \mathcal{P} \mapsto \Phi$. When a language does not need such additional specifications, we assume a singleton set $\mathcal{P} = \{\emptyset\}$, and we use $i(\varphi, I)$ as an abbreviation for $i(\varphi, I, \emptyset)$. Similarly, for some languages that generally need additional specifications ($|\mathcal{P}| > 1$), there may be a subset of formulas $\Phi_d \subseteq \Phi$ that does not need any. We call a formula *detached*, if $i(\varphi, I, p)$ yields the same result for any condition $p \in \mathcal{P}$. If φ^d is detached, we use $i(\varphi^d, I)$ as an abbreviation for $i(\varphi, I, p)$.

Any language, along with its definition of $\models$, raises interesting decision problems, among which are:

- the *(model) checking problem*, that is, given an interpretation I and a formula φ, decide whether $I \models \varphi$;
- the *satisfiability problem*, that is, given a formula φ, decide whether $\exists I \in \mathcal{I}, I \models \varphi$;
- the *validity problem*, that is, given a formula φ, decide whether $\forall I \in \mathcal{I}, I \models \varphi$;
- the *entailment problem*, that is, given two formulas φ_1 and φ_2, decide whether $\forall I \in \mathcal{I}, I \models \varphi_1 \Rightarrow I \models \varphi_2$;
- the *equivalence problem*, that is, given two formulas φ_1 and φ_2, decide whether $\forall I \in \mathcal{I}, I \models \varphi_1 \Leftrightarrow I \models \varphi_2$ (also stylized as $\varphi_1 \equiv \varphi_2$);
- the *(syntactic) minimization problem*, that is, given two formulas φ_1 and φ_2 such that φ_1 is equivalent to φ_2 decide whether there exists a formula φ_2' such that $|\varphi_2'| < |\varphi_2|$ and φ_2' is equivalent to φ_1.[2]

In AI, where formulas represent real-world processes, and interpretations represent data measurements of realizations of these processes, these problems all represent different tasks involving reasoning and knowledge manipulation. For example, the minimization problem corresponds to the synthesis of a more concise, equivalent statement that describes the same process. As another example, model checking corresponds to verifying that a certan property holds for a given data instance. Depending on the formalism, the time complexity of these problems ranges from polynomial to undecidable. Among these problems, the checking problem is the most important and studied, and, in its *finite* case, it is also the most relevant in data-driven, symbolic AI. The time complexity for checking is typically measured with respect to the size of the interpretation (*data complexity*), and that of the formula (*expression complexity*).

[2] Different formulations of the problem exist, considering different metrics for the complexity of the formula.

In the following, we present a few formalisms that are relevant in learning contexts. For a language L, we denote the set of formulas and interpretations as Φ_{L} and $\mathcal{I}_{\mathrm{L}}$, respectively. Because Boolean truth values (truth and falsehood) alone cover a wide range of applications, the following discussion will limit $\mathcal{T}$ to be exactly $\{\top, \bot\}$. This choice leads to so-called *crisp logics*; extensions to *many-valued logics* based on partial orders between truth values (e.g., fuzzy logics, probabilistic logics) can be investigated at a later time, and are out of the scope of this thesis.

§3.2.2 Propositional Logic

Propositional logic, denoted here as P, is the simplest logical formalism. Although its millenary roots, it was studied more recently, among others, by Gottfried Leibniz, George Boole and Augustus De Morgan, independently and in different forms. In the crisp setting P is the logical counterpart to Boolean algebra. In the following discussion, we use c for denoting any connective in $\mathcal{C}$, t_i for any truth value in $\mathcal{T}$, and φ for any formula in Φ.

Propositional logic encloses two connectives for expressing the negation of a concept and the disjunction between two concepts. The set of connectives is, thus:

$$C = C_1 \cup C_2, \quad C_1 = \{\neg\}, \quad C_2 = \{\vee\},$$

where $\neg$ and $\vee$ are read as *not* and *or*, and formulas are given by:

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

The semantics of a propositional connective c of arity a is defined by a rule for *transfering* truth; that is, a rule that, given truth values $t_1, \ldots, t_a$, computes the truth value semantically equivalent to $c(t_1, \ldots, t_a)$. Let $s_{\mathrm{P}}(c, (t_1, \ldots, t_a))$ compute this truth value (with s abbreviating *simplify*); the semantics of negation and conjunction are defined by:

$$\begin{aligned} s_{\mathrm{P}}(\neg, (t)) &= \begin{cases} \top & \text{if } t = \bot, \\ \bot & \text{otherwise;} \end{cases} \\ s_{\mathrm{P}}(\vee, (t_1, t_2)) &= \begin{cases} \top & \text{if } t_1 = \top \text{ or } t_2 = \top, \\ \bot & \text{otherwise;} \end{cases} \end{aligned}$$

For example, this states that $\top \equiv \neg\bot$, and $\top \vee \bot \equiv \top$. Note that semantic equivalences also hold for generic formulas; for example, for any formula φ, it is the case that $\bot \vee \varphi \equiv \varphi$. In fact, s_{P} is easily extended for locally simplifying arbitrary formulas:

$$\begin{aligned} s_{\mathrm{P}}(\neg, (\varphi)) &= \neg\varphi, \\ s_{\mathrm{P}}(\vee, (\varphi_1, \varphi_2)) &= \varphi_1 \vee \varphi_2, \\ s_{\mathrm{P}}(\vee, (\varphi, \top)) &= \top, \\ s_{\mathrm{P}}(\vee, (\top, \varphi)) &= \top, \\ s_{\mathrm{P}}(\vee, (\varphi, \bot)) &= \varphi, \\ s_{\mathrm{P}}(\vee, (\bot, \varphi)) &= \varphi, \end{aligned}$$

for $\varphi, \varphi_1, \varphi_2 \in \Phi \setminus \mathcal{T}$. Note that $s_{\mathrm{P}}(c, (\varphi_1, \ldots, \varphi_a)) \equiv c(\varphi_1, \ldots, \varphi_a)$.

A *propositional interpretation* (or *propositional mapping*) consists of a bijection $I : \mathcal{AP}' \mapsto \mathcal{T}$ assigning truth values to a subset $\mathcal{AP}'$ of atoms in $\mathcal{AP}$, and i_{P} is defined by:

$$\begin{array}{lcl} i_{\mathrm{P}}(p, I) & = & \begin{cases} I(p) & \text{if } p \in \mathcal{AP}', \\ p & \text{otherwise;} \end{cases} \\ i_{\mathrm{P}}(t, I) & = & t; \\ i_{\mathrm{P}}(c(\varphi_1, \ldots, \varphi_a), I) & = & s_{\mathrm{P}}(c, (i_{\mathrm{P}}(\varphi_1, I), \ldots, i_{\mathrm{P}}(\varphi_a, I))). \end{array}$$

Note how when $\mathcal{AP}' \subseteq \mathrm{atoms}(\varphi)$, then $i_{\mathrm{P}}(I, \varphi) \in \mathcal{T}$. Otherwise, $i_{\mathrm{P}}(I, \varphi)$ represents a (partial) simplification of φ using the information in I. From the definition above of i_{P}, a checking algorithm emerge with linear data and expression time complexities.

Finally, additional connectives can be introduced by extending s_{P}; for example, a binary connective for expressing conjunction ($\wedge$), such that:

$$\psi_1 \wedge \psi_2 \equiv \neg(\neg\psi_1 \vee \neg\psi_2),$$

can be defined by:

$$\begin{array}{lcl} s_{\mathrm{P}}(\wedge, (\varphi_1, \varphi_2)) & = & \varphi_1 \wedge \varphi_2, \\ s_{\mathrm{P}}(\wedge, (\varphi, \top)) & = & \varphi, \\ s_{\mathrm{P}}(\wedge, (\top, \varphi)) & = & \varphi, \\ s_{\mathrm{P}}(\wedge, (\varphi, \bot)) & = & \bot, \\ s_{\mathrm{P}}(\wedge, (\bot, \varphi)) & = & \bot. \end{array}$$

Implication ($\rightarrow$), double implication ($\leftrightarrow$), and exclusive disjunction ($\oplus$) can also be defined as shortcuts:

$$\begin{array}{lcl} \psi_1 \rightarrow \psi_2 & \equiv & \neg\psi_1 \vee \psi_2; \\ \psi_1 \leftrightarrow \psi_2 & \equiv & (\psi_1 \rightarrow \psi_2) \wedge (\psi_2 \rightarrow \psi_1); \\ \psi_1 \oplus \psi_2 & \equiv & (\psi_1 \wedge \neg\psi_2) \vee (\neg\psi_2 \wedge \psi_1). \end{array}$$

In logics with the $\neg$ connective, whenever two connectives c_1 and c_2 are such that $c_1(\varphi_1, \ldots, \varphi_a) \equiv \neg c_1(\neg\varphi_1, \ldots, \neg\varphi_a)$, they are called *duals* to one another; conjunction and disjunction, for example, are duals to one another (De Morgan's laws).

These equivalences can also be leveraged for constructing synctactic simplification rules that are more sophisticated, such as:

$$\begin{array}{lcl} s_{\mathrm{P}}(\neg, (\wedge(\neg\varphi_1, \neg\varphi_2))) & = & s_{\mathrm{P}}(\vee, (\varphi_1, \varphi_2)), \\ s_{\mathrm{P}}(\neg, (\vee(\neg\varphi_1, \neg\varphi_2))) & = & s_{\mathrm{P}}(\wedge, (\varphi_1, \varphi_2)). \end{array}$$

§3.2.3 Modal Logic K

Propositional, modal logics are an extension of classical propositional logic that allows to characterize the validity of arguments with *modal* premises and conclusions. Narrowly speaking, they were initially investigated for expressing the *necessity* and *possibility* of judgments, and their first modern formalization traces back to about a century ago Lewis [Lew18]. With respect to propositional logic it can express, in limited form, relational, quantified statements. Modal

logic K, named after Saul Kripke, is of particular interest because, although very abstract, it is archetypal of many modal logics for temporal and spatial reasoning.

We introduce K in a simple setting, in which it extends propositional logic by adding a single unary connective $\Diamond$ (referred to as *diamond*). The set of connectives is, thus:

$$C = C_1 \cup C_2, \quad C_1 = \{\neg, \Diamond\}, \quad C_2 = \{\vee\},$$

and formulas are given by:

$$\varphi ::= p \mid \neg\varphi \mid \Diamond\varphi \mid \varphi \vee \varphi.$$

In the relational rendition of K [Kri63], an interpretation is a (possibly infinite) directed graph of propositional mappings on the same set of atoms. Such structures, typically referred to as Kripke structures, are conveniently represented by separating the relational and the propositional components. A *Kripke frame* $\mathcal{F} = (\mathcal{W}, R)$, representing the relational component, consists of a non-empty set of *worlds* $\mathcal{W}$, and an binary *accessibility relation* $R \subseteq \mathcal{W} \times \mathcal{W}$ over them. A *Kripke structure* $I = (\mathcal{F}, ev)$, is a Kripke frame $\mathcal{F} = (\mathcal{W}, R)$ enriched with a bijection $ev : \mathcal{W} \to \mathcal{I}_{\mathrm{P}}$, which associates each world w a propositional interpretation. A Kripke $\mathcal{F}$ is navigable via a function enumerating the accessible worlds from a world $w \in \mathcal{W}$:

$$\mathrm{acc}(\mathcal{F}, w) = \{w' \mid (w, w') \in R\}.$$

The truth of formulas is relativized to single worlds. Let a *formula mapping* be a bijection $\mathcal{M} : \mathcal{W} \mapsto \Phi$ assigning a formula to each world of a Kripke frame, and let us use the notation $\{w_1 \mapsto \varphi_1, w_2 \mapsto \varphi_2, \ldots\}$, for denoting the mapping $\mathcal{M}$ such that $\mathcal{M}(w_1) = \varphi_1$ and $\mathcal{M}(w_2) = \varphi_2$. Similarly to the propositional logic, in K the semantics of a connective c is defined by a rule for transfering formula mappings on a frame; that is, a rule that, given mappings $\mathcal{M}_1, \ldots, \mathcal{M}_{\mathfrak{a}(c)}$, computes a mapping associated to $c(\mathcal{M}_1, \ldots, \mathcal{M}_{\mathfrak{a}(c)})$. Let $s_{\mathrm{K}}(\mathcal{F}, c, (\mathcal{M}_1, \ldots, \mathcal{M}_a))$ compute this formula mapping on frame $\mathcal{F}$; the semantics of the three connectives are defined by:

$$\begin{array}{lcl}
s_{\mathrm{K}}(\mathcal{F}, \neg, (\mathcal{M})) & = & \{w \mapsto s_{\mathrm{P}}(\neg, (\mathcal{M}(w))) \mid w \in \mathcal{W})\}, \\
s_{\mathrm{K}}(\mathcal{F}, \vee, (\mathcal{M}_1, \mathcal{M}_2)) & = & \{w \mapsto s_{\mathrm{P}}(\vee, (\mathcal{M}_1(w), \mathcal{M}_2(w))) \mid w \in \mathcal{W})\}, \\
s_{\mathrm{K}}(\mathcal{F}, \Diamond, (\mathcal{M})) & = & \{w \mapsto s'_{\mathrm{P}}(\vee, (\mathcal{M}(w'_1), \ldots, \mathcal{M}(w'_n))) \mid w \in \mathcal{W})\},
\end{array}$$

where $\mathcal{W}$ are the worlds in $\mathcal{F}$, $\mathrm{acc}(\mathcal{F}, w) = \{w'_1, \ldots, w'_n\}$, and s'_{P} extends s_{P} so that an arbitrary number of formulas can be simplified via $\vee$:

$$s'_{\mathrm{P}}(\vee, (\varphi_1, \ldots, \varphi_n)) = \begin{cases} \bot & \text{if } n = 0, \\ \varphi_1 & \text{if } n = 1, \\ s_{\mathrm{P}}(\varphi_1, \varphi_2) & \text{if } n = 2, \\ s_{\mathrm{P}}(\varphi_1, s'_{\mathrm{P}}(\vee, \varphi_2, \ldots, \varphi_n)) & \text{if } n > 2. \end{cases}$$

Similarly to the propositional case, s_{K} is made the base for the checking algorithm. As mentioned, the truth of a formula is relativized to a world ($\mathcal{P} = \mathcal{W}$), thus, the interpret function is:

$$i_{\mathrm{K}}(\varphi, I, w) = M(w),$$

where $M = \tilde{i}_{\mathrm{K}}(\varphi, I)$, and:

$$\begin{array}{lcl} \tilde{i}_{\mathrm{K}}(p, I) & = & \{w \mapsto i_{\mathrm{P}}(p, ev(w)) \mid w \in \mathcal{W}\}, \\ \tilde{i}_{\mathrm{K}}(t, I) & = & \{w \mapsto t \mid w \in \mathcal{W}\}, \\ \tilde{i}_{\mathrm{K}}(c(\varphi_1, \ldots, \varphi_a), I) & = & s_{\mathrm{K}}(c, (\tilde{i}_{\mathrm{K}}(\varphi_1, I), \ldots, \tilde{i}_{\mathrm{K}}(\varphi_a, I))). \end{array}$$

Essentially, $\Diamond\varphi$ holds on a world w (that is, $i(\Diamond\varphi, I, w) = \top$) if and only if that there exists at least an accessible world from w where φ holds; it, therefore, acts as an existential quantifier. $\Diamond$ has a dual connective $\Box$, called *box*, with universal semantics: $\Box\varphi$ states that φ holds on all accessible worlds.

From the definition of i_{K}, a checking algorithm emerge with linear data and expression time complexities [Cla+18]. Note that, similarly to propositional logic, propositional connectives can be introduced: such as implication ($\rightarrow$), double implication ($\leftrightarrow$), and exclusive disjunction ($\oplus$).

This logic can be extended to the case of multiple relations of arbitrary arities, resulting in a modal logic K with many diamond connectives (sometimes also called K^n). In this case, Kripke frames are, essentially, directed hypergraphs over a set of worlds, and different diamond connectives can express different, quantified statements on the accessible worlds over different relations. This allows to represent and express richer relational concepts.

Let a *multirelation Kripke frame* be an object $\mathcal{F} = (\mathcal{W}, \mathcal{R})$, consisting of a non-empty set of *worlds* $\mathcal{W}$, and a set $\mathcal{R} = \{R_{X_1}, \ldots, R_{X_r}\}$ of (more-than-unary) *accessibility relations* over them, with each relation R_X being of arity $\mathfrak{a}(R_X) > 1$: $R_X \subseteq \mathcal{W}^{\mathfrak{a}}(R_X)$. Similarly to the case of connectives, let us define $\mathcal{R}_i$ as the set of relations of arity i; thus, $\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_{\overline{\mathfrak{a}}}$, where $\overline{\mathfrak{a}} \in \mathbb{N}^+$ is the highest relational arity. Any multirelation frame $\mathcal{F}$ is navigable through a function enumerating the accessible world tuples from a world $w \in \mathcal{W}$ via a specific relation $R_X \in \mathcal{R}$ of arity a :

$$\mathrm{acc}(\mathcal{F}, w, X) = \{(w'_1, \ldots, w'_{a-1}) \mid (w, w'_1, \ldots, w'_{a-1}) \in R_X\}.$$

Instead of a single $\Diamond$ connective, an existential quantifier $\langle X \rangle$ (referred to as *diamond* X) exists for each relation $R_X \in \mathcal{R}$, with arity equal to $\mathfrak{a}(R_X) - 1$. The set of connectives is, thus:

$$C = \{\neg, \vee\} \cup \{\langle X \rangle \mid R_X \in \mathcal{R}\},$$

and $\mathfrak{a}(\langle X \rangle) = \mathfrak{a}(R_X) - 1$. Formulas are given by:

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

The definition of i_{K} is the same as the one in the single-relation setting; however, s_{K} now encompasses simplification rules for the new connectives:

$$\begin{array}{lcl} s_{\mathrm{K}}(\mathcal{F}, \neg, (\mathcal{M})) & = & \{w \mapsto s_{\mathrm{P}}(\neg, (\mathcal{M}(w))) \mid w \in \mathcal{W})\}; \\ s_{\mathrm{K}}(\mathcal{F}, \vee, (\mathcal{M}_1, \mathcal{M}_2)) & = & \{w \mapsto s_{\mathrm{P}}(\vee, (\mathcal{M}_1(w), \mathcal{M}_2(w))) \mid w \in \mathcal{W})\}; \\ s_{\mathrm{K}}(\mathcal{F}, \langle X \rangle, (\mathcal{M}_1, \ldots, \mathcal{M}_{a-1})) & = & \{w \mapsto s'_{\mathrm{P}}(\vee, (\hat{\varphi}_i, \ldots, \hat{\varphi}_n)) \mid w \in \mathcal{W})\}, \end{array}$$

$$\begin{array}{rcl} \hat{\varphi}_i & = & s'_{\mathrm{P}}(\wedge, (\mathcal{M}(w^i_1), \ldots, \mathcal{M}(w^i_{a-1}))), \\ \mathrm{acc}(\mathcal{F}, w, X) & = & \{(w^1_1, \ldots, w^1_{a-1}), \ldots, (w^n_1, \ldots, w^n_{a-1})\}, \end{array}$$

where $a = \mathfrak{a}(R_X)$, and s'_{P} extends s_{P} for $\wedge$ and arbitrary numbers of formulas:

$$s'_{\mathrm{P}}(\wedge, (\varphi_1, \ldots, \varphi_n)) = \begin{cases} \bot & \text{if } n = 0, \\ \varphi_1 & \text{if } n = 1, \\ s_{\mathrm{P}}(\varphi_1, \varphi_2) & \text{if } n = 2, \\ s_{\mathrm{P}}(\varphi_1, s'_{\mathrm{P}}(\wedge, \varphi_2, \ldots, \varphi_n)) & \text{if } n > 2. \end{cases}$$

The dual of each $\langle X \rangle$ is a connective $[X]$ (*box X*) with universal semantics.

Relational properties other than standard ones (e.g., transitivity, symmetry) are relevant for the resulting K formalism. A relation R is called *detaching* when:

$$(w_1, w'_1, \ldots, w'_n) \in R \Leftrightarrow \forall w_2 \in \mathcal{W}\ (w_2, w'_1, \ldots, w'_n) \in R.$$

If a relation R_X is detaching, interpreting a formula such as $\langle X \rangle \varphi$ on any world yields the same results; as a matter of fact, the simplification of such a formula becomes a property of the interpretation, as opposed of the world itself. For example, the *global (binary) relation* $R_G = \mathcal{W} \times \mathcal{W}$ is detaching, therefore $\langle X \rangle \varphi$ simplifies to either $\top$ or $\bot$ on all worlds; thus, depending on whether a world where φ holds exists, the interpretations will either satisfy the formula φ or not on any world.

More generally, if $R^d_{X_1}, R^d_{X_2}, \ldots$ are detaching relations, and $R_{X_1}, R_{X_2}, \ldots$ are not detaching relations, then formulas of type φ^d, produced by the grammar:

$$\begin{array}{rcl} \varphi^d & ::= & \neg\varphi^d \mid \varphi^d \vee \varphi^d \mid \langle X^d \rangle \varphi, \\ \varphi & ::= & p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi, \mid \langle X^d \rangle \varphi, \end{array}$$

yields the same results on any world. These formulas are detached, and we can, thus, simply write $i_{\mathrm{K}}(\varphi_d, I)$ instead of $i_{\mathrm{K}}(\varphi_d, I, w)$, for any $w \in \mathcal{W}$.

§3.2.4 HS^d: A Modal Logic for Multidimensional Domains

Modal logic K can be specialized and embedded in geometrical relational structures, resulting in formalisms for geometrical reasoning. Historically, the modal logic literature has been mostly focused on one-dimensional formalisms for temporal reasoning, where the worlds are intervals and/or point on a linear order, and the accessibility relations model relations between these geometrical entities. Interval temporal logic, or HS (named after its authors Halpern and Shoham [HS91b]), based on the 13 binary *interval relations* defined by James F. Allen [All83], is an important, representative example of such formalisms. Allen's interval relations provide a qualitative vocabulary for describing the different arrangements between any two intervals: for example, one is *after* the other, one is *later than* the other, overlapping *overlapping* the other, or *preceding one another*. Higher-dimensional logics have also been studied, typically in the context of *qualitative spatial reasoning* (*QSR*), and most often in algebraic terms [RN07].

Essentially, any geometrical formalism depends on the definitions of $\mathcal{W}$, that is, a set of *entities*, and $\mathcal{R}$, that is, a set of relevant *relations* between the entities. Points are elementary entities that cannot express the extension of objects, thus they are not suitable in contexts

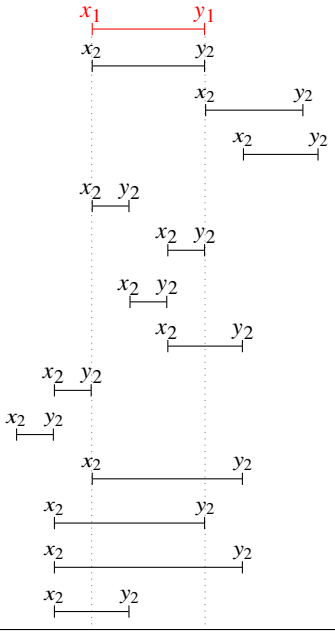
Name	Definition	Example
= (identity)	$R_{=} = \{([x_1, y_1], [x_2, y_2]) \mid x_1 = x_2 \text{ and } y_1 = y_2\}$	
A (after)	$R_{\mathsf{A}} = \{([x_1, y_1], [x_2, y_2]) \mid y_1 = x_2\}$	
L (later than)	$R_{\mathsf{L}} = \{([x_1, y_1], [x_2, y_2]) \mid y_1 < x_2\}$	
B (begins)	$R_{\mathsf{B}} = \{([x_1, y_1], [x_2, y_2]) \mid x_1 = x_2 \text{ and } y_2 < y_1\}$	
E (ends)	$R_{\mathsf{E}} = \{([x_1, y_1], [x_2, y_2]) \mid y_1 = y_2 \text{ and } x_1 < x_2\}$	
D (during)	$R_{\mathsf{D}} = \{([x_1, y_1], [x_2, y_2]) \mid x_1 < x_2 \text{ and } y_2 < y_1\}$	
O (overlaps)	$R_{\mathsf{O}} = \{([x_1, y_1], [x_2, y_2]) \mid x_1 < x_2 < y_1 < y_2\}$	
$\overline{\mathsf{A}}$ (after inverse)	$R_{\overline{\mathsf{A}}} = \{([x_1, y_1], [x_2, y_2]) \mid [x_2, y_2] R_{\mathsf{A}} [x_1, y_1]\}$	
$\overline{\mathsf{L}}$ (later than inverse)	$R_{\overline{\mathsf{L}}} = \{([x_1, y_1], [x_2, y_2]) \mid [x_2, y_2] R_{\mathsf{L}} [x_1, y_1]\}$	
$\overline{\mathsf{B}}$ (begins inverse)	$R_{\overline{\mathsf{B}}} = \{([x_1, y_1], [x_2, y_2]) \mid [x_2, y_2] R_{\mathsf{B}} [x_1, y_1]\}$	
$\overline{\mathsf{E}}$ (ends inverse)	$R_{\overline{\mathsf{E}}} = \{([x_1, y_1], [x_2, y_2]) \mid [x_2, y_2] R_{\mathsf{E}} [x_1, y_1]\}$	
$\overline{\mathsf{D}}$ (during inverse)	$R_{\overline{\mathsf{D}}} = \{([x_1, y_1], [x_2, y_2]) \mid [x_2, y_2] R_{\mathsf{D}} [x_1, y_1]\}$	
$\overline{\mathsf{O}}$ (overlaps inverse)	$R_{\overline{\mathsf{O}}} = \{([x_1, y_1], [x_2, y_2]) \mid [x_2, y_2] R_{\mathsf{O}} [x_1, y_1]\}$	

TABLE 3.1: 13 Allen's interval relations.

where objects have diverse shapes, or when mereological aspects of space are relevant (e.g., grade of overlap between regions). Regions, on the other hand, can express extension, but are more complex to deal with. When regions are considered, they are usually assumed to be *connected*, which is in agreement with an implicit locality hypothesis on the objects of the reasoning. Although limited in terms of expressive power, convex regions are typically preferred as extended entities, or they are used to approximate complex shapes (e.g., by means of *convex hulls*, [Coh95b]). Regions can be further constrained in order to achieve higher specificity or lower complexity; for example, forcing all entities to be of the same size can be beneficial for specific domains such as *optical character recognition*. Another sensible constraint is *orthogonality*: in the case of polygons, forcing the edges to be parallel to a given reference set of axes limits the set of possible relations, and sometimes lowers computational costs; in particular, rectangles with edges parallel to the axes (such as in Rectangle Algebra, presented in [BCF98]) have been deeply used in computer vision and QSR due to their computational benefits.

When defining relations, different different aspects can be captured (e.g., distance, size, shape of the entities), and two of the most studied relational aspects are *directionality* and *topology*. Directional relations account for the relative spatial arrangement of the two objects (e.g., *next to*, *to the right of*); they are not invariant under rotation/reflection, and are not as easily definable for generic types of regions. On the algebraic side, many binary calculi have been proposed, both with punctual entities [Fra96] and regions of various types [BCF98; Nav+13; SK04; Ski+07]. Directional algebras could, in principle, give rise to their modal logic counterparts, but only a few attempts have been made in this sense [Bre+10; MR99; MPS09; MNNS07]. While in this

context an absolute frame of reference is generally adopted, there are examples of spatial logics and calculi that account for the *orientation* of objects, and are able to describe relations such as *in front of*, *behind* or *facing each other* [Fre92; WZ19]. As for topological relations, they generally address the different modalities of intersection between regions and their boundaries, which makes them invariant under rotation and reflection; being purely qualitative and easily definable for generic regions, topological approaches are often preferred to directional ones. The most popular characterization of topological binary relations is the set of Region Connection Calculus relations, often referred to as RCC8 [ESM11; RCC92]. In its extended form, the set includes the 8 relations: *disconnected* (*DC*), *externally connected* (*EC*), *partially overlapping* (*PO*), *tangential proper part* (*TPP*), *non-tangential proper part* (*NTPP*), *tangential proper part inverse* (*TPPi*), *non-tangential proper part inverse* (*NTPPi*), and *identity*. Coarser sets are derived by considering unions of RCC8 relations, deriving, for example, RCC5. In [LW06] the modal logics entailed by these sets of relations are studied.

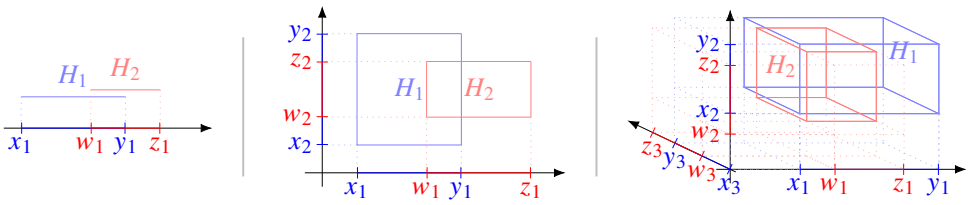
Notably, compared to topological algebras, *jointly exhaustive* region-based directional algebras (namely, algebras in which each pair of entities are in at least a relation) tend to have a higher number of relations, as well as a higher granularity; moreover, their relations can generally be partitioned and joined to capture topological aspects, and, ultimately, to form coarser topological systems. With this in mind, and also considering the convenient properties of orthogonal rectangles, in this context, we present HS^d, a general-purpose logic where the interpretations are (multirelation) Kripke structures on *hyperrectangles* in a multidimensional space, their worlds are orthogonal hyperrectangles, and the binary relations capture qualitative, but fine geometric arrangements between them. The logic elegantly generalizes at least propositional logic (P), interval temporal logic (HS [HS91b]), and rectangle logic (presented in algebraic form in [BCF98]), and its fragments include important topological formalisms such as RCC8 and RCC5. Altogether, the logic inherits the importance of all these formal systems, and provides for useful applications in many data-oriented fields, including Time-Series Analysis and Computer Vision. It is, in fact, virtually representative of any modal logic for temporal, spatial and spatio-temporal reasoning. With concrete objectives in mind, the focus is, therefore, on interpretations represented by finite, discrete spaces, so that data represented by multidimensional arrays (e.g., time-series, images, videos, text). For simplicity, the presentation is made on structures that have the same length N along each dimension (i.e., hypercubes).

Consider a d-dimensional grid, $\mathbb{D}^d$, where $\mathbb{D} = \langle\{1, \ldots, N\}, \leq\rangle$ is an initial segment of the positive natural numbers up to some cardinality N. Within this context, we define an *(orthogonal) hyperrectangle* as an ordered collection of d pairs of natural numbers, outlining a localized subregion of interest in the grid, that is:

$$H = ([x_1, y_1], \ldots, [x_d, y_d]) \subseteq \mathbb{D}^d,$$

where each pair $[x_i, y_i]$ defines the bounds of the interval along the i-th dimension, with $1 \leq x_i \leq y_i \leq N$ for each dimension i such that $1 \leq i \leq d$. When $x_i = 1$ and $y_i = N$ across all dimensions, the hyperrectangle H covers the full extent of $\mathbb{D}^d$. Let $\mathbb{H}(\mathbb{D}^d)$ be the set of all $(N(N+1)/2)^d$ hyperrectangles in $\mathbb{D}^d$.

We now devise a set of binary relations over $\mathbb{H}(\mathbb{D}^d)$, building on the 13 Allen's binary



$(H_1, H_2) \in R_O$ $(H_1, H_2) \in R_{OD}$ $(H_1, H_2) \in R_{D\overline{O}O}$

FIGURE 3.2: Examples of hyperrectangles and HS^d relations for $d \in \{1, 2, 3\}$.

interval relations [All83], which qualitatively describe different arrangements between any pair of intervals on a linear order $\mathbb{H}(\mathbb{D})$. The relation set includes the following binary relations, which are defined and depicted in Tab. 3.1:

$$\mathcal{R}^{\text{HS}} = \{R_=, R_\mathsf{A}, R_\mathsf{L}, R_\mathsf{B}, R_\mathsf{E}, R_\mathsf{D}, R_\mathsf{O}, R_{\overline{\mathsf{A}}}, R_{\overline{\mathsf{L}}}, R_{\overline{\mathsf{B}}}, R_{\overline{\mathsf{E}}}, R_{\overline{\mathsf{D}}}, R_{\overline{\mathsf{O}}}\}.$$

As suggested by the notation, each relation R has a corresponding converse relation, denoted by $\overline{R}$. We define relations in HS^d by lifting Allen's interval relations to d-dimensional hyperrectangles:

$$\begin{aligned} \mathcal{R}^{\text{HS}^d} &= \{R_{X_1 \dots X_d} \mid R_{X_i} \in \mathcal{R}^{\text{HS}}\}, \\ R_{X_1 \dots X_d} &= \{([x_1, y_1], \dots, [x_d, y_d]) \mid [x_1, y_1] \in R_{X_1} \dots [x_d, y_d] \in R_{X_d}\}. \end{aligned}$$

Naturally, $\mathcal{R}^{\text{HS}^d}$ has 13^d relations; see Fig. 3.2 for a graphical representation of three relations in 1, 2, and 3 dimensions, respectively.

Ultimately, for a given $d \in \mathbb{N}^+$, formulas of HS^d are given by:

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

with $X_i \in \{=, \mathsf{A}, \mathsf{L}, \mathsf{B}, \mathsf{E}, \mathsf{D}, \mathsf{O}, \overline{\mathsf{A}}, \overline{\mathsf{L}}, \overline{\mathsf{B}}, \overline{\mathsf{E}}, \overline{\mathsf{D}}, \overline{\mathsf{O}}\}$. Let a *$d$-dimensional Kripke frame* on a grid $\mathbb{D}$ be an object $\mathcal{F} = (\mathbb{H}(\mathbb{D}^d), \mathcal{R}^{HS^d})$. Interpretations in HS^d are Kripke structures on d-dimensional Kripke frames (on grids of possibly different sizes). The definition for the interpret function is inherited from K.

While encompassing a large number of relations and connectives, HS^d is intuitive, and fine enough for capturing different aspects of relative pair-wise, spatial arrangements. Its relations are mutually exclusive and jointly exhaustive with respect to $\mathbb{H}(\mathbb{D}^d)$; as such, exactly one HS^d relation holds between each pair of hyperrectangles in $\mathbb{H}(\mathbb{D}^d)$, and the global relation, that allows to express global patterns (e.g., "there exists a rectangular region *anywhere* ...") is defined by union of all relations:

$$R_G = \bigcup_{R \in \mathcal{R}^{\text{HS}^d}} R.$$

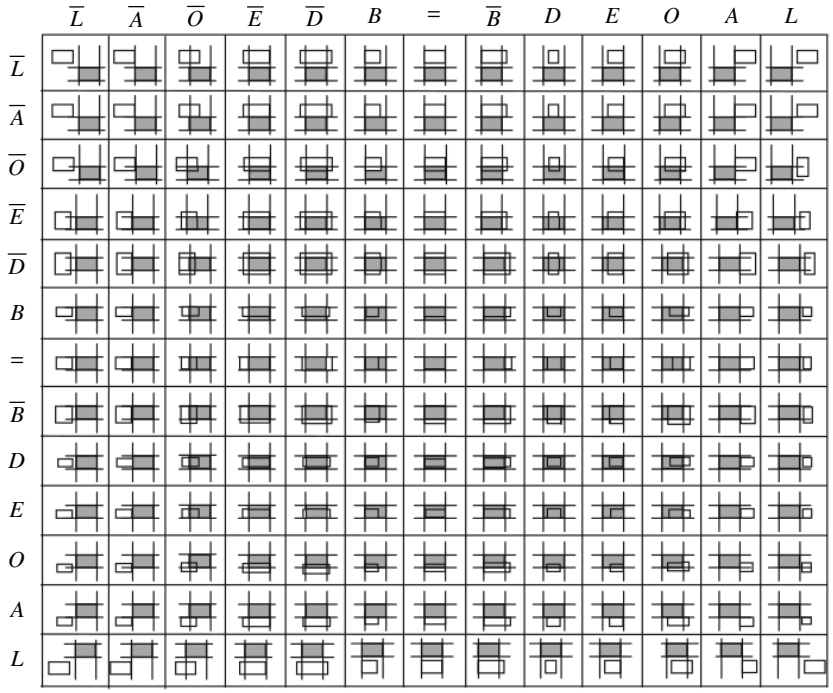


FIGURE 3.3: Pictorial examples for the 169 relations of HS^2, arranged in a matricial structure. For each relation X, a reference entity r_1 (gray) and a secondary entity r_2 (transparent), with $(r_1, r_2) \in R_X$.

Note that adding R_G to the the set would increase the logical expressive power, as the *diamond G* can be defined by disjunction of all modal connectives:

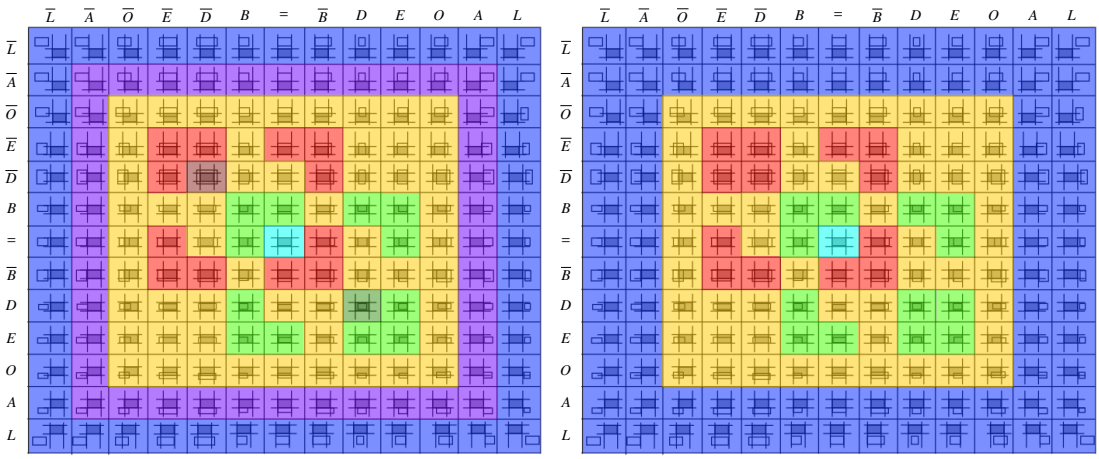
$$\langle G \rangle \varphi = \bigvee_{R_X \in \mathcal{R}^{\mathrm{HS}^d}} \langle X \rangle \varphi,$$

In a similar way, unions of HS^d relations give rise to coarser directional and topologicalrelations (and thus, coarser modal connectives). For example, consider the bidimensional specialization, HS^2, which 169 relations are depicted in Fig. 3.3; the RCC8 relation *disconnected* on $\mathbb{H}(\mathbb{D}^2)$ is defined as:

$$R_{DC} = \bigcup_{X \in \{A,L,B,E,D,O,=\}} (R_{\overline{L},X} \cup R_{L,X} \cup R_{X,\overline{L}} \cup R_{X,L}).$$

As a matter of fact, among the many fragments of HS^d, we find two notable topological ones (originally presented in [LW06]) which we call, here, HS^d_{RCC8} (8 relations, shown in Fig. 3.4a) and HS^d_{RCC5} (5 relations, shown in Fig. 3.4b).

As we shall see, this logic has important applications in learning contexts on multidimensional data (e.g., images, videos, etc.).



(A) (B)

FIGURE 3.4: Partitions of HS^2 relations into HS^2_{RCC8} (a) and HS^2_{RCC5} (b) relations. HS^2_{RCC8} relations are: *disconnected* (blue), *externally connected* (violet), *partially overlapping* (yellow), *tangential proper part* (light green), *non-tangential proper part* (dark green), *tangential proper part inverse* (red), *non-tangential proper part inverse* (maroon), and *identity* (turquoise). HS^2_{RCC5} relations are: *discrete from* (blue), *partially overlapping* (yellow), *proper part* (green), *proper part inverse* (red), *identity* (turquoise).

§3.3 Symbolic Data

Much like Machine Learning, Symbolic Learning studies algorithms for extracting knowledge from data; in SL, however, the data must be represented in symbolic form. In this section, we discuss how to transform common, ML datasets into sets of logical interpretations (*logisets*).

As mentioned, while for a long time Symbolic Learning focused on non-relational (or *tabular*) data, modern AI is mostly focused on relational and possibly multimodal data, which embodies at least 80%–95% of all existing data [GH15]. We, therefore, start by defining tabular and non-tabular datasets, which include the wide variety of multidimensional datasets (i.e., datasets of time-series, images, videos, and graphs). After showing the generality of relational representations of non-tabular datasets, we introduce the general concept of *symbolic transformer*, and discuss the transformation of multidimensional instances to interpretations of HS^d.

Ultimately, we extend the provided framework to the multimodal setting, where instances are simultaeously described by many *modalities*.

In the following discussion we consider datasets of numeric, real attributes, and no missing value. Although this setting is not the most general, it is representative, and covers most of the real-world needs. Nonetheless, strategies can be devised for handling non-numerical attributes (e.g., one-hot encodings), and missing information (e.g., imputation methods, fuzzy extensions).

§3.3.1 Tabular and Non-tabular Datasets

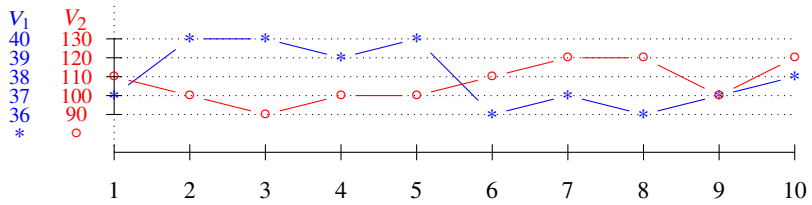
In the simplest ML setting, a dataset is, essentially, a set of *(data) instances*, each being an entity with some numerical attributes associated (or *attributes*). Let $\mathcal{A}$ an (unordered) set of attributes $\{A_1, \ldots, A_n\}$, a *tabular instance* on $\mathcal{A}$ is a map $I : \mathcal{A} \mapsto \mathbb{R}$. A *tabular dataset* on a set of attributes $\mathcal{A}$ is a set $\mathcal{I} = \{I_1, \ldots, I_m\}$ of tabular instances on $\mathcal{A}$. The term *tabular* comes from the fact that each dataset in this form is naturally represented as table where each instance is a row, and each attribute is a column; for example, consider the following tabular dataset on $\mathcal{A} = \{\text{Age}, \text{Weight}, \text{HeartRate}\}$:

Instance	Age	Weight	HeartRate
I_1	27	70	60
I_2	49	55	70
I_3	20	81	75
...	...	...	...

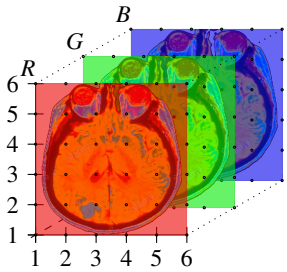
While any tabular dataset is completely described by a table, which (picturially) forces an order on the attributes, no relation at all exists between the attributes. In most modern, real-world settings, however, difficult ML tasks lies in datasets with relational components. For example, speech recognition, image classification, action recognition and community detection involve time-series, images, video and graph data, respectively. Throughout time, these datasets have been named *non-tabular* (or, sometimes, *unstructured*); we give, here, a graph-based characterization, highlighting their relational components.

A *non-tabular instance* on $\mathcal{A}$ is a bijection $I : \mathcal{A} \mapsto \mathbb{R}$, with $A \in \mathcal{A}$ plus a finite set of relations $\mathcal{R} = \{R_1, \ldots, R_r\}$ over $\mathcal{A}$, where each relation R_X has arity $\mathfrak{a}(R_X)$. Similarly to before, a *non-tabular dataset* on a set of attributes $\mathcal{A}$ is a set $\mathcal{I} = \{I_1, \ldots, I_m\}$ of non-tabular instances on $\mathcal{A}$. Depending on the nature of the data, the relations over the attributes can encode aspects of very different kind; from the fact that two features are measured using the same sensor (with, perhaps, a different sensitivity), to structural aspects (e.g., temporal, spatial, or spatio-temporal arrangements), to more abstract relationships that depend on domain-specific, expert knowledge. Of course, this representation generalizes its non-relational counterpart, and decays to it when $\mathcal{R} = \emptyset$.

A quite common of non-tabular instances includes those naturally represented via multidimensional arrays of real numbers, covering the great realm of time-series, images, videos, and common ML renditions of textual data (e.g., word embeddings). Time-series, for example, are series of temporally ordered measurements, typically registered at a constant rate. Observations are either *univariate* or *multivariate*, depending on the number of different aspects that are measured at any point in time. Fig. 3.5a illustrates an example of time-series of 10 points representing the evolution of the temperature and blood pressure of a medical subject. Note that time-series also include audio recordings. In a similar way, a digital image is a 2D structure of *pixels*, each containing the measurement of different electromagnetic frequencies, commonly called *color channels*. Fig. 3.5b illustrates an image representing on the canonical RGB color channels, which measures the electromagnetic response for red (R), green (G), and blue (B). Digital videos are usually encoded as a set of temporally ordered, equally spaced images. For all these kinds of data, a set of variables vary across a multidimensional domain.



(A) A multivariate time-series with two variables.



(B) An RGB image.

FIGURE 3.5: Two examples of a (non-tabular) multidimensional data instance.

Let $\mathcal{V}$ be a collection of *variables* $\{V_1, \ldots, V_n\}$. A *d-dimensional instance* on variables $\mathcal{V}$ and of size $(S_1, \ldots, S_d)$, is a bijection $I : \mathcal{V} \mapsto \mathcal{M}_{S_1,\ldots,S_d}$, where elements of $\mathcal{M}_{S_1,\ldots,S_d}$ are d-dimensional matrices of real numbers $M_{s_1,\ldots,s_d}$, with $s_i \in [1, S_i]$. Essentially, a d-dimensional instance I assigns a real value to each variable $V \in \mathcal{V}$ at any point $(s_1, \ldots, s_d)$ in a discrete, d-dimensional space; we denote this value as $I(V, (x, y))$.

Multidimensional instances (resp., datasets) are specializations of non-tabular instances (resp., datasets). To exemplify an encoding, consider a single-channel (or *grayscale*) image I of size X, Y on a variable V. Such an object is represented in raw relational form I^R by fixing attributes $\mathcal{A} = \{A_{1,1}, A_{1,2}, \ldots, A_{1,Y}, \ldots, A_{X,1}, A_{X,2}, \ldots, A_{X,Y}\}$, a set of binary, cardinal relations $\mathcal{R} = \{R_{NORTH}, R_{SOUTH}, R_{EAST}, R_{WEST}\}$, cardinal for expressing the matricial arrangement (see Fig. 3.6), and a bijection such that $I^R(A_{x,y}) = I(V, (x, y))$. More specifically, the relations capture the adjacency and relative spatial arrangement of the pixels; for example, R_{NORTH} contains all tuples of type $(A_{x,y}, A_{x,y-1})$, thus encoding the fact that two attributes are spatially adjacent and one is above the other. With more than a variable, the encoding only requires more relations. For example, a digital image of size (X, Y) with RGB channels is represented via a set of $X \times Y \times 3$ attributes of type $A_{x,y,c}$, with $c \in \{R, G, B\}$ representing one of the three channels, and a set of relations that includes a ternary relation for encoding the pixels grouping of attributes $\mathcal{R} = \{R_{NORTH}, R_{SOUTH}, R_{EAST}, R_{WEST}, R_{RGB}\}$:

$$
\begin{aligned}
R_{NORTH} &= \{(A_{x,y,c_1}, A_{x,y-1,c_2}) \mid x \in [1, X], y \in [2, Y], c_1, c_2 \in \{R, G, B\}\}, \\
R_{SOUTH} &= \{(A_{x,y,c_1}, A_{x,y+1,c_2}) \mid x \in [1, X], y \in [1, Y-1], c_1, c_2 \in \{R, G, B\}\}, \\
R_{EAST} &= \{(A_{x,y,c_1}, A_{x-1,y,c_2}) \mid x \in [2, X], y \in [1, Y], c_1, c_2 \in \{R, G, B\}\}, \\
R_{WEST} &= \{(A_{x,y,c_1}, A_{x+1,y,c_2}) \mid x \in [1, X-1], y \in [1, Y], c_1, c_2 \in \{R, G, B\}\}, \\
R_{RGB} &= \{(A_{x,y,R}, A_{x,y,G}, A_{x,y,B}) \mid x \in [1, X], y \in [1, Y]\}.
\end{aligned}
$$

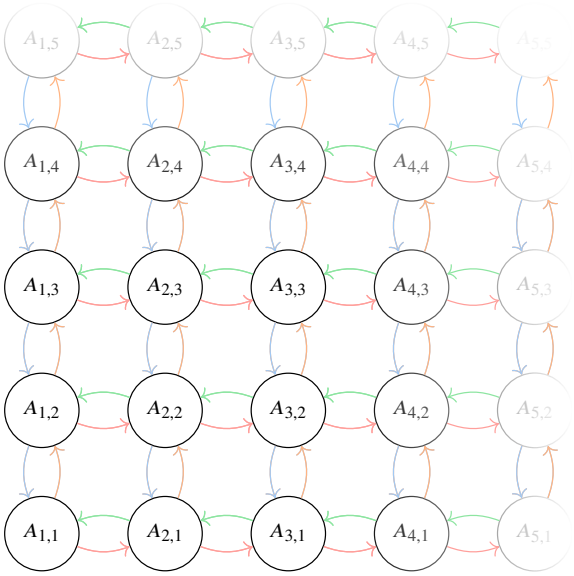


FIGURE 3.6: Relational representation of a grayscale image as a non-tabular instance. The attributes (for which the image has scalar values) are related to each other via binary, cardinal relations R_{NORTH} (brown), R_{SOUTH} (blue), R_{EAST} (red), R_{WEST} (green).

As we shall see, the HS^d formalism suited for logical renditions of this kind of data.

§3.3.2 Symbolic Transformers

A noticeable symmetry emerges between tabular, non-tabular instances, and interpretations for the languages P and K, respectively, with the major difference being that ML instances enclose real numbers, while logical interpretations enclose truth values. Therefore, a symbolic rendition of ML instances involves designing an alphabet of conditions on attributes and real numbers. For example, a rather simple logical rendition of tabular instances in P may encompass an infinite number of atoms that are thresholding on the value of single attributes, such as:

$$\mathcal{AP} = \{A \bowtie v \mid A \in \mathcal{A}, \bowtie \in \{<, \leq, =, \neq, \geq, >\}, v \in \mathbb{R}\},$$

where $\bowtie$ is referred to as the *test operator*. In this case, a tabular instance I can act as a propositional mapping by defining it such that $I(A \bowtie v)$ is equal to $\top$ if $I(A) \bowtie v$, and to $\bot$ otherwise. This intuitive atom design covers most of the propositional decision tree learning, but it also results limited. In principle, atoms can be as complex as needed, from being simple thresholding to single attributes, up to relying on deep neural networks (e.g., Guo and Gelfand [GG92b] and Murthy et al. [Mur+16b]). As an example, oblique decision trees (e.g., Setiono and Liu [SL99b]) complicate this framework by thresholding on affine combinations of the attributes:

$$\mathcal{AP} = \{([A_1, \ldots, A_n] \cdot \mathbf{w} + b) \bowtie v \mid A \in \mathcal{A}, \bowtie \in \{<, \leq, =, \neq, \geq, >\}, v \in \mathbb{R}\},$$

where $\mathbf{w} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. More broadly, atoms involve thresholdings on functional transformations.

Let a *tabular feature extraction function* on attributes $A_1, \ldots, A_n$ be a bijection:

$$f : \mathbb{R}^n \to \mathbb{R}.$$

Such a function, essentially, extracts real numbers from tabular instances on attributes $A_1, \ldots, A_n$, ideally summarizing the information within the instances. Let $\mathcal{F}$ be a set of tabular feature extraction functions, atoms are generally defined by:

$$\mathcal{AP} = \{f \bowtie v \mid f \in \mathcal{F}, A \in \mathcal{A}, \bowtie \in \{<, \leq, =, \neq, \geq, >\}, v \in \mathbb{R}\},$$

and, using a rather loose notation, instances are such that $I(f \bowtie v)$ is equal to $\top$ if:

$$f(I(A_1), \ldots, I(A_n)) \bowtie v,$$

and to $\bot$ otherwise. Note that the thresholding step could, in principle, be incorporated into the very functions in $\mathcal{F}$, which could, more straightforwardly, produce truth values instead of real values. However, it is convenient to explicit the thresholding step on $\mathbb{R}$ as the final operation that produces the truth value; althought out of the scope of this thesis, this atom formulation extends more naturally to the many-valued generalization, that are based on a (generally partial) order between truth values (i.e., a *truer than* partial relation). Note that this design choice enforces a structure in the atoms in $\mathcal{AP}$; for example, it is the case that:

$$f(A_1, ..., A_n) > v_1 \to f(A_1, ..., A_n) > v_2 \Leftrightarrow v_1 \leq v_2$$

Similar expedients could be used for transforming non-tabular instances into modal interpretations, it would not leverage the higher expressive power in a sensible way.

Let us, first, abstract the transformation process of ML data into logical interpretations. A *symbolic transformer* for a language L is an object that transforms ML instances into logical interpretations of L:

$$\tau_{\mathrm{L}} : \mathbb{I} \mapsto \mathcal{I}_{\mathrm{L}},$$

where $\mathbb{I}$ is the space of all instances. Naturally, any transformer can also be applied to datasets, by a straightforward extension:

$$\tau_{\mathrm{L}}(\mathcal{I}) = \{\tau_{\mathrm{L}}(I) \mid I \in \mathcal{I}\}.$$

A transformer can, thus, transform a ML dataset into a a set of logical interpretations, which we refer to as *logiset*. As non-tabular instances and modal interpretations generalize propositional interpretations and propositional interpretations, respectively, we focus here on relevant non-tabular transformers for K.

Non-tabular transformers for K transform a non-tabular instance I^{ML} on attributes $\mathcal{A} = \{A_1, \ldots, A_n\}$ and relations $\mathcal{R}^{\mathrm{ML}} = \{R_1^{\mathrm{ML}}, \ldots, R_r^{\mathrm{ML}}\}$ onto a Kripke structure I on worlds $\mathcal{W}$, on accessibility relations $\mathcal{R} = \{R_{X_1}, \ldots, R_{X_r}\}$, and modal assigment ev. Once again, the main

difference between non-tabular ML instances and K-interpretations lies in the the fact that non-tabular instances are hypergraphs over attributes, while modal interpretations are hypergraphs over propositional mappings. One could, therefore, be tented to mantain the relational structure by fixing $\mathcal{W} = \mathcal{A}$ and $\mathcal{R} = \mathcal{R}^{\text{ML}}$. Then, scalar atoms can be designed so that at a given world A_i, scalar conditions on the value of A_i are tested:

$$\begin{aligned} \mathcal{AP} &= \{(\bowtie, v) \mid \bowtie \in \{<, \leq, =, \neq, \geq, >\}, v \in \mathbb{R}\} \\ ev(A_i) &= \left\{ (\bowtie, v) \mapsto \begin{cases} \top & \text{if } I(A_i) \bowtie v, \\ \bot & \text{otherwise.} \end{cases} \mid (\bowtie, v) \in \mathcal{AP} \right\} \end{aligned}$$

As an example, consider, again, the case of a grayscale image, as represented in Fig. 3.6; this logical rendition entails a punctual modal formalism that allows to express concepts such as *for all pixels, if the color is less than 10, then the pixel above has color higher than 20* via a formula:

$$[G]((<, 10) \rightarrow \langle N \rangle((>, 20))).$$

However, when designing the transformer, one should take into account aspects of pragmatics and qualitativeness, as the resulting language should be appropriate for describing sensible patterns in the data in a possibly succinct way. In fact, while the above formalism may be suited for tasks involving shift-invariant, pixel-based reasoning, it would result inappropriate for more complex tasks. For example, consider the task of recognizing whether the image is a picture of the eye of a human or a cat. In this case, the formalism may be able to express a logical statement that holds on images of one kind, and do not hold on images of the other kind, effectively telling the two classes apart; however, such description would result unnatural, unneccessary complex, hard to interpret, and, perhaps most importantly, hard to *learn*. While the raw relational representation may not be the desired one, it must be clarified that the transformation should also be simple enough to allow human practitioners to interpret the patterns, and trace them back to the data in its original form. As such, single worlds, relations, and atoms holding on worlds should all represent meaningful, interpretable concepts.

For the case of multidimensional data, modal logic HS^d on hyperrectangles offers many advantages. Worlds represent localized geometric areas that exist in the instance, so that logical patterns are interpreted on confined areas. Furthermore, the granularity of the relation set allows to express with elasticity relational aspects of different kind. For example, HS^2 on an image can describe patterns in both directional and topological terms simultaneously, such as *if there exists an orthogonal rectangular area where φ_1, partially overlapping with another area where φ_2, then all rectangular areas with a left edge as the right edge of an area where φ_1, have φ_3*:

$$\langle G \rangle(\varphi_1 \wedge \langle PO \rangle(\varphi_2)) \rightarrow [G](\varphi_1 \rightarrow \langle L, = \rangle(\varphi_3)).$$

Note that the relation R_L,= includes all pairs of rectangles . As for the atoms, they should, then, capture some properties of the hyperrectangular areas in the instance. With respect to the punctual rendition presented above, where a single world has a single value per variable, a world in HS^d would, instead, correspond to more than a value for an attribute. Furthermore, given a d-dimensional instance I, any hyperrectangle H of size $(s_1, \ldots, s_d)$ delineates a portion of the

instance $I[H]$ that is a d-dimensional instance I of size $(s_1, \ldots, s_d)$. For example, any rectangle of pixels in an RGB image is, itself, an RGB image. Thus, atoms are designed to involve some (multidimensional) feature extraction process, localized to single hyperrectangles.

Let a *d-dimensional feature extraction function* on variables $\mathcal{V}$ be a bijection:

$$f : \mathbb{I}^d_{\mathcal{V}} \to \mathbb{R},$$

where $\mathbb{I}^d_{\mathcal{V}}$ is the space of all d-dimensional instances on variables $\mathcal{V}$. A d-dimensional feature extraction function, essentially, extracts a real number from a d-dimensional instance. Ideally, the real number captures relevant (and interpretable) aspects in the image. For example, feature functions for any instances of any dimensions are the minimum, maximum, and average functions; on an image, such functions compute the minimum, maximum and average value across all color channels, respectively.

Let $\mathcal{F}$ be set of d-dimensional feature extraction functions $\{f_1, \ldots, f_n\}$ on variable in $\mathcal{V}$. Scalar atoms for the d-dimensional data are thresholdings on extracted features:

$$\mathcal{AP} = \{f \bowtie v \mid V \in \mathcal{V}, \bowtie \in \{<, \leq, =, \neq, \geq, >\}, v \in \mathbb{R}\}.$$

Finally, we transform any d-dimensional instance I of size $(S_1, \ldots, S_d)$ into an interpretation $I_{\mathrm{HS}^d} = (\mathcal{F}, ev)$, where $\mathcal{F}$ is a d-dimensional Kripke frame:

$$(\mathbb{H}(\langle\{1, \ldots, S_1\}, \leq\rangle, \ldots, \langle\{1, \ldots, S_d\}, \leq\rangle), \mathcal{R}^{HS^d}),$$

and $I_{\mathrm{HS}^d,H} = ev(H)$ is a propositional mapping such that:

$$I_{\mathrm{HS}^d,H}(f \bowtie v) = \begin{cases} \top & \text{if } f(I[H]) \bowtie v \\ \bot & \text{otherwise.} \end{cases}$$

General-purpose (e.g., *Catch22* [Lub+19]) and context-specific (e.g., *STFT* and *MFCC* [DM80]) feature functions exists, and the literature for the case multidimensional data is especially vast, reaching out to different fields including Time-Series Analysis, Signal Processing, and Computer Vision. As mentioned, feature functions extract relevant feature representations of the instances/hyperrectangles; as such, they can be designed to be as complex as required, and their choice mainly depends on the context and goals of the AI exercise. With this regard, note that simple functions favor interpretability, and that some associations of feature functions and test operators have particularly intuitive semantics. For example, the *minimum*, and *maximum* functions are peculiarly simple, and pairing them with operators such as $>$ and $<$, respectively, leads to straightforward concepts: $\min > 10$ holds on (a hyperrectangle of) an instance if and only if all values within (the hyperrectangular area of) the instance are greater than 10; conversely, $\max < 10$ if and only if all values are smaller than 10. It, thus, make sense to parameterize the alphabet on the set $\mathcal{FT}$ of feature-operator pairs. While the minimum and maximum value across all variables may result limited, *univariate* versions of these functions applied to single variables may represent a good trade-off in terms of granularity. Within the atoms, we denote the univariate version of a feature function f, that is, the feature function that applies f to variable V as $f(V)$;

atoms of this kind gives rise to univariate alphabets of the kind:

$$\mathcal{AP} = \{f(V) \bowtie v \mid (f, \bowtie) \in \mathcal{FT}, V \in \mathcal{V}, v \in \mathbb{R}\}.$$

Finally, let us note that, since tabular data is a special case of multidimensional data (with $\mathcal{V} = \mathcal{A}$), these same atoms can be used in propositional formalisms, where it is arguably more common to see feature extraction as a data preprocessing step, and the *identity* function is used, then, as feature function in the atoms; in this case, $identity(V_i) \bowtie v$ can be simply abbreviated into $V_i \bowtie v$.

§3.3.3 Multimodal Datasets

In modern Machine Learning it is common to deal with *multimodal* data. In the multimodal setting, instances are simultaneously described by more than a single data kind (or *modality*). For example, a classification task may require to classify an object given a photo of it, and an audio recording of its sound.[3]

A *multimodal instance* of *rank* q (or *q-instance*) is a tuple of q non-tabular instances (or *modalities*) $(I_1, \ldots, I_n)$ on sets of attributes $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and sets of relations $\mathcal{R}_1, \ldots, \mathcal{R}_n$, respectively. Similarly, *multimodal dataset* of rank q (or *q-dataset*) on sets of attributes $\mathcal{A}_1, \ldots, \mathcal{A}_n$ and sets of relations $\mathcal{R}_1, \ldots, \mathcal{R}_n$ is a set of $\mathcal{I} = \{I_1, \ldots, I_m\}$ of tabular instances on $\mathcal{A}$. In the classification example above, an instance is easily modeled by a modality I_1 for describing the images, and a modality I_2 for describing the audio recording.

The logical framework presented above is easily extended to the multimodal setting. A q-instance can be transformed to its logical rendition via a *q-transformer* τ, which is a tuple of q transformers $(\tau_1, \ldots, \tau_q)$ on a collection of q formalisms $\mathcal{L} = (\mathrm{L}_1, \ldots, \mathrm{L}_q)$. Each transformer is applied to the q modalities independently, and the output of the transformation a *q-interpretation* $I^{\mathcal{L}} = (I_1^{\mathcal{L}}, \ldots, I_q^{\mathcal{L}})$ (is a tuple of q interpretations), which can be thought of as an interpretation of a propositional meta-formalism $\mathrm{P}^{\mathcal{L}}$. Meta-formulas of $\mathrm{P}^{\mathcal{L}}$ are, essentially, Boolean combinations of formulas of $\mathrm{L}_1, \ldots, \mathrm{L}_m$, that are to be interpreted on single modalities:

$$\mathcal{AP}^{\mathcal{L}} = \{\{\mu\}\varphi \mid \mu = [1, \ldots, q], \varphi \in \Phi_{\mathrm{L}_\mu}\},$$

$$\varphi^{\mathcal{L}} ::= p \mid \neg\varphi^{\mathcal{L}} \mid \varphi^{\mathcal{L}} \wedge \varphi^{\mathcal{L}},$$

where $p \in \mathcal{AP}^{\mathcal{L}}$. Checking a $\mathrm{P}^{\mathcal{L}}$ formula on a q-interpretation is similar to checking a formula of a propositional logic: the interpret function $i_{\mathrm{P}^{\mathcal{L}}}$ follows the same definition of i_{P}, and any q-interpretation $I^{\mathcal{L}} \in \mathcal{I}_{\mathrm{P}^{\mathcal{L}}}$ is such that:

$$I^{\mathcal{L}}(\{\mu\}\varphi) = i_{\mathrm{L}_\mu}(I_\mu^{\mathcal{L}}, \varphi),$$

where $I_\mu^{\mathcal{L}}$ is the μ-th modality of $I^{\mathcal{L}}$.

[3] Note that while, incidentally, the words *multimodal* and *modality* are also used in the logic literature for denoting multirelation modal logics and their modal connectives, in this context we only use it in its ML interpretation.

§3.4 Symbolic Models

In Machine Learning, a model is an abstraction of an algorithm that, typically, produces an output from an input interpretation. A model can be as simple as linear regressions, or as complex as (deep) neural networks. As the name of the field suggests, any model is, often, the result of a optimization, *learning* meta-algorithm; however, its definition is independent from the learning context, and only concerns its structure and how it computes the output from the input.

Broadly speaking, all symbolic models (such as decision trees, lists, and sets) share a rule-based functioning and encode knowledge via logical formulas. More specifically, any symbolic model must be such that some conditions involving logical satisfaction (i.e., $\models$) are tested on the input, and exactly one condition eventually applies, ultimately costituting an explanation of the output. This desideratus ensures that only part of the model is responsible for the output, making the size of the explanation typically much smaller than the model. Note that this is not the case with neural networks, where the information flows throughout the whole structure of the model, making it difficult to justify the output.

Similarly to mathematical function defined by cases, then, any symbolic model must be equivalent to a set of non-overlapping IF-THEN rules; where the condition to be checked is based on the logical satisfaction. In general, logical satisfaction requires checking conditions to be specified $\mathcal{P}$, which means that each symbolic model should also encompass the interpret conditions for checking $I \models \varphi$. However, the following discussion assumes that models only encompass detached formulas, so no interpret (checking) condition is required. Besides simplifying the discussion, this choice is also justified by the fact that symbolic models typically check properties of the interpretations while they do not, for example, check properties of specific worlds. In the following, we fix a logical formalism L, and we base the discussion on logical satisfaction $\models$, rather than the interpret operation i.

As for the outcome of the model, let us call $\mathbb{O}$ the output space of an ML task. Definitions of symbolic models must cover the supervised and unsupervised settings. Supervised settings include classification, where $\mathbb{O}$ is finite, unordered and discrete, and regression, where, for simplicity, we assume $\mathbb{O} = \mathbb{R}$. As mentioned, Unsupervised Symbolic Learning concides with association rule mining, where the outputs are formula, themselves: $\mathbb{O} = \Phi_{\mathrm{L}}$.

As symbolic models are rule-based, they should be able to be nested and composed to form more complex reasoning structures. We build towards the definition bottom-up by, first, defining simple models that determine the final output. A *leaf (or final) model* is an object $\mathfrak{f}$ from the grammar:

$$\begin{array}{lcl} \mathfrak{f} & ::= & \mathfrak{f}_c \mid \mathfrak{f}_s, \\ \mathfrak{f}_c & ::= & o, \\ \mathfrak{f}_s & ::= & \mathrm{sub}, \end{array}$$

with $o \in \mathbb{O}$, and sub being any (possibly subsymbolic) model that outputs a value in $\mathbb{O}$, given an input logical interpretations. The operational semantics are given recursively by defining how these models compute outputs from input interpretations, that is, by defining a function:

$$\mathrm{apply} : \mathfrak{M} \times \mathcal{I}_{\mathrm{L}} \mapsto \mathbb{O}.$$

The simplest model is the *constant model* $o \in \mathbb{O}$, which outputs a constant value:

$$\text{apply}(o, I) = o.$$

Subsymbolic computation (e.g., neural networks) can be wrapped into a *subsymbolic model* sub, which is simply a function $\Phi_{\text{L}} \mapsto \mathbb{O}$ to be applied on the input interpretation:

$$\text{apply}(\text{sub}, I) = \text{sub}(I).$$

A symbol model is constructed on these basic blocks by wrapping them into IF-THEN, IF-THEN-ELSE, IF-THEN-ELSEIF-ELSE, and ensembling constructs.

A symbolic model encodes nestings of conditional constructs, and has a recursive structure involving *decision nodes* that are symbolic models, themselves. A *(generalized) symbolic model* on final models $\mathfrak{F}$ and output space $\mathbb{O}$ is an object $\mathfrak{m}$ from the following grammar:

$$\begin{array}{lcl}
\mathfrak{m} & ::= & \mathfrak{f} \mid \mathfrak{r} \mid \mathfrak{b} \mid \mathfrak{l} \mid \mathfrak{e}, \\
\mathfrak{r} & ::= & \varphi \rightarrowtail \mathfrak{m}, \\
\mathfrak{b} & ::= & \varphi \rightarrowtail (\mathfrak{m}, \mathfrak{m}), \\
\mathfrak{l}' & ::= & \mathfrak{m} \mid \mathfrak{l}' \cdot \mathfrak{m}, \\
\mathfrak{l} & ::= & [\mathfrak{l}'], \\
\mathfrak{e}' & ::= & \mathfrak{m} \mid \mathfrak{e}', \mathfrak{m}, \\
\mathfrak{e} & ::= & (\mathfrak{e}', \text{ag}),
\end{array}$$

where $\mathfrak{f} \in \mathfrak{F}, o \in \mathbb{O}, \varphi \in \Phi_{\text{L}}$, and $\text{ag} \in \text{AG}(\mathbb{O})$, where $\text{AG}(\mathbb{O})$ is a set of *output aggregation functions* of type $2^{\mathbb{O}} \mapsto \mathbb{O}$. A symbolic model is called:

- a *(decision) rule* if it is of type $\varphi \rightarrowtail \mathfrak{m}$;
- a *(decision) branch* if it is of type $\varphi \rightarrowtail (\mathfrak{m}^+, \mathfrak{m}^-)$;
- a *(decision) list* if it is of type $[\mathfrak{m}_1 \cdot \ldots \cdot \mathfrak{m}_n, \mathfrak{m}_{\text{default}}]$;
- a *(decision) ensemble* if it is of type $(\mathfrak{m}_1, \ldots, \mathfrak{m}_n, \text{ag})$.

The operational semantics of decision rules, branches, lists and ensembles are given recursively:

$$\begin{array}{lcl}
\text{apply}(\varphi \rightarrowtail \mathfrak{m}, I) & = & \begin{cases} \text{apply}(\mathfrak{m}, I) & \text{if } I \models \varphi, \\ \text{nothing} & \text{if } I \not\models \varphi, \end{cases} \\
\text{apply}(\varphi \rightarrowtail (\mathfrak{m}^+, \mathfrak{m}^-), I) & = & \begin{cases} \text{apply}(\mathfrak{m}^+, I) & \text{if } I \models \varphi, \\ \text{apply}(\mathfrak{m}^-, I) & \text{if } I \not\models \varphi, \end{cases} \\
\text{apply}([\mathfrak{m}_1 \cdot \ldots \cdot \mathfrak{m}_n, \mathfrak{m}_{\text{default}}], I) & = & \begin{cases} \text{apply}(\mathfrak{m}_i, I) & \text{if } |cov(\{\mathfrak{m}_1, \ldots, \mathfrak{m}_n\})| > 0 \text{ and} \\ & i = \min\limits_{\mathfrak{m}_j \in cov(\{\mathfrak{m}_1, \ldots, \mathfrak{m}_n\})} j, \\ \text{apply}(\mathfrak{m}_{\text{default}}, I) & \text{otherwise}, \end{cases} \\
\text{apply}((\mathfrak{m}_1, \ldots, \mathfrak{m}_n, \text{ag}), I) & = & \text{ag}(\text{apply}(\mathfrak{m}_1, I), \ldots, \text{apply}(\mathfrak{m}_n, I)),
\end{array}$$

where nothing is a special value in $\mathbb{O}$ representing an empty output, and

$$cov(\{\mathfrak{m}_1, \ldots, \mathfrak{m}_n\}) = \{\mathfrak{m}_i \mid \mathfrak{m}_i(I) \neq \text{nothing}\}.$$

Each model has set of immediate submodels (or *consequents*), that is, the set of models produced by recursive expansions of $\mathfrak{m}$ in the derivation tree:

- for a rule $\varphi \rightarrowtail \mathfrak{m}$, the formula φ is the *antecedent* and $\mathfrak{m}$ is the *consequent*;
- for a branch $\varphi \rightarrowtail (\mathfrak{m}^+, \mathfrak{m}^-)$, φ is the *antecedent*, and $\mathfrak{m}^+$ and $\mathfrak{m}^-$ are the *positive* and *negative consequents*;
- a list $[\mathfrak{m}_1 \cdot \ldots \cdot \mathfrak{m}_n, \mathfrak{m}_{\text{default}}]$ encompasses an ordered collection of *n non-default consequents* $\mathfrak{m}_1, \ldots, \mathfrak{m}_n$, and a *default consequent* $\mathfrak{m}_{\text{default}}$;
- an ensemble $(\mathfrak{m}_1, \ldots, \mathfrak{m}_n, \text{ag})$ encompasses a set of *contributing consequents* $\mathfrak{m}_1, \ldots, \mathfrak{m}_n$, and an aggregation function ag.

Note that, in terms of expressivity, the decision list construct generalizes the branch construct, since any branch $\varphi \rightarrowtail (\mathfrak{m}^+, \mathfrak{m}^-)$ is equivalent to a list $[\varphi \rightarrowtail \mathfrak{m}^+, \mathfrak{m}^-]$. Furthermore, decision ensembles generalize decision lists, since the aggregation function can encode a *firing order* of the immediate submodels.

With constant models as final models ($\mathfrak{F} = \mathbb{O}$), this definition generalizes classical definitions of decision trees, rules, stumps, branches, lists, forests and gradient-boosted tree. For example, traditional decision list and decision tree models emerge from the subgrammars:

$$\mathfrak{tree} \quad ::= \quad \mathfrak{f} \mid \varphi \rightarrowtail (\mathfrak{tree}, \mathfrak{tree}),$$

and

$$\begin{aligned} \mathfrak{list}' \quad &::= \quad \varphi \rightarrowtail \mathfrak{f} \mid \mathfrak{list}' \cdot \varphi \rightarrowtail \mathfrak{f}, \\ \mathfrak{list} \quad &::= \quad [\mathfrak{list}'], \end{aligned}$$

where $\mathfrak{f} \in \mathbb{O}$. Similarly, forest models are tree ensembles of the kind:

$$\mathfrak{forest} \quad ::= \quad (\mathfrak{tree}, \ldots, \mathfrak{tree}, f),$$

where f is a measure of central tendency; typically, the mean function when an order sussists on $\mathbb{O}$ (e.g., classification setting), and the mode function otherwise (e.g., regression setting). Boosted trees can be encoded via a more sophisticated (e.g., weighted) aggregation function. Note that more complex aggregation functions (e.g., deep) are undesired in this context. When $\mathfrak{F}$ is the set of neural models, then the definition also generalizes neuro-symbolic models where neural network is used after some rule-based reasoning (e.g., Zhou and Chen [ZC02]).

As mentioned, a symbolic model is equivalent to a set of non-overlapping IF-THEN rules. Let a *final rule* be a symbolic model with a final consequent; the set of final rules with non-overlapping antecedents, representing the knowledge about the task, is computed by a function r

defined as:

$$
\begin{aligned}
r(\mathfrak{f}) &= \{\top \rightarrowtail \mathfrak{f}\} \\
r(\varphi \rightarrowtail \mathfrak{m}) &= \{\varphi \wedge \psi \rightarrowtail \mathfrak{f} \mid \psi \rightarrowtail \mathfrak{f} \in r(\mathfrak{m})\} \\
r(\varphi \rightarrowtail (\mathfrak{m}^+, \mathfrak{m}^-)) &= r(\varphi \rightarrowtail \mathfrak{m}^+) \cup r(\neg\varphi \rightarrowtail \mathfrak{m}^-) \\
r([\mathfrak{m}]) &= r(\mathfrak{m}) \\
r([\mathfrak{m}_1 \cdot \ldots \cdot \mathfrak{m}_n]) &= r(\mathfrak{m}_1) \cup \{\neg\psi \wedge \psi' \rightarrowtail \mathfrak{f}' \mid \psi \rightarrowtail \mathfrak{f} \in r(\mathfrak{m}_1), \psi' \rightarrowtail \mathfrak{f}' \in r([\mathfrak{m}_2 \cdot \ldots \cdot \mathfrak{m}_n])\} \\
r((\mathfrak{m}_1, \ldots, \mathfrak{m}_n, \mathrm{ag})) &= \bigcup_{\psi_1 \rightarrowtail \mathfrak{f}_1 \in r(\mathfrak{m}_1), \ldots, \psi_n \rightarrowtail \mathfrak{f}_n \in r(\mathfrak{m}_n),} \{\psi_1 \wedge \ldots \wedge \psi_n \rightarrowtail \mathcal{AG}(\mathfrak{f}_1, \ldots, \mathfrak{f}_n)\},
\end{aligned}
$$

where $\mathfrak{f} \in \mathfrak{F}$, and $\mathcal{AG} : 2^{\mathfrak{F}} \mapsto \mathfrak{F}$ returns a final model that outputs the aggregation via ag of a collection of final models. That is, $\mathfrak{f}_{\mathrm{ag}} = \mathcal{AG}(\mathfrak{f}_1, \ldots, \mathfrak{f}_n)$ if and only if:

$$\forall I \in \mathcal{I}_{\mathrm{L}} \ \mathrm{apply}(\mathfrak{f}_{\mathrm{ag}}, I) = \mathrm{ag}(\mathrm{apply}(\mathfrak{f}_1, I), \ldots, \mathrm{apply}(\mathfrak{f}_n, I)).$$

Note that, when $\mathfrak{F}$ is a set of constant models (i.e., $\mathfrak{F} \subseteq \mathbb{O}$,), then:

$$\mathcal{AG}(\mathfrak{f}_1, \ldots, \mathfrak{f}_n) = \mathrm{ag}(\mathfrak{f}_1, \ldots, \mathfrak{f}_n).$$

Note that final ruleset extracted from a model can, itself, be interpreted as a symbolic model; for example, as a decision list by fixing a random order between the rules. Also note that symbolic models involving rule nodes may produce nothing values; in such case we call the model *open*. On the contrary, if no rule construct is involved in the model subtree (and the aggregation function do not produce nothing values), then symbolic models always produce non-empty outputs. In this case, the model is *closed*, and the antecedents of the extracted rules induce a partition of the instance space $\mathcal{I}_{\mathrm{L}}$. It is important to point out that the very essence of the transparency of symbolic models lies in the possibility of deriving such rules, translating them to natural language, and manipulating them using the tools of symbolic logic.

Similarly to any other computational model, symbolic models and their associated rules can be large and complex, depending on the task. A measure of complexity for symbolic models emerges by considering, for each of the four constructs, how the the size of the extracted ruleset compares with respect to the size of the immediate submodels. In these terms, rule and branch constructs have linear *symbolic complexity*, while decision list and ensemble constructs have quadratic and exponential complexity, respectively. The exponential blowup for ensemble models is of particular interest, exemplified by the fact that getting rid of the subsymbolic aggregation step from forests of decision trees is a well-known problem (e.g., see Section 2.1).

Besides standard ML evaluation methods, specific methods for evaluating the knowledge enclosed symbolic models can be applied. The individually extracted rules are evaluated in terms of complexity of both the antecedent and of the final consequent. Antecedents complexity can be measured, for example, in terms of synctactic depth, synctactic length, number of atoms, and modal depth (that is, the connective depth with respect to the set of modal connectives $d_{\{\Diamond,\Box\}}$). Consequent complexity is particularly relevant for subsymbolic final models, and can include aspects of algorithmic and/or structural complexity. Moreover, given a logiset

$\mathcal{I} = \{I_1, \ldots, I_n\} \subset \mathcal{I}_\mathrm{L}$, the goodness of a final rule $\varphi \rightarrowtail \mathfrak{f}$ can be evaluated. The performance evaluation, then, is done in terms of confusion metrics in the classification setting, distance metrics in the regression setting, and comparisons between truth values in the unsupervised case (i.e., $\mathbb{O} = \Phi_\mathrm{L}$). Note that the supervised setting, $\mathcal{I}$ is labelled, that is, each interpretation I_i in it is associated to a label $o_i \in \mathbb{O}$. Rule evaluation metrics are, first, defined for the single interpretation and, then, extended to logisets by aggregation via central tendency; metrics are, thus, functions of the type: $\mathfrak{R}_f \times \mathcal{I}_\mathrm{L} \times \mathbb{O} \to \mathbb{R}$, where $\mathfrak{R}_f$ is the space of final rules. In the unsupervised case, interpretations are unlabelled, thus metric functions are of the type: $\mathfrak{R}_f \times \mathcal{I}_\mathrm{L} \to \mathbb{R}$.

Common performance metrics are straightforwardly adapted from the rule extraction literature. For the classification and association cases, for example, notable metrics depend on whether the antecedent and the consequent of the rule apply to the interpretation. A rule $\varphi \rightarrowtail \mathfrak{f}$ is said to *cover* an interpretation I when $I \models \varphi$. Furthermore, a rule is said to *fulfil* an interpretation when its consequent also applies to the interpretation; the formal definition differs in the supervised and unsupervised cases. In the supervised, classification setting, where any interpretation I_i is associated to a label $o_i \in \mathbb{O}$, fulfilment compares the consequent with the label:

$$\mathrm{fulfils}(\varphi \rightarrowtail \mathfrak{f}, I_i, o_i) \quad = \quad \begin{cases} \text{true} & \text{if } o = \mathrm{apply}(\mathfrak{f}, I_i), \\ \text{false} & \text{otherwise.} \end{cases}$$

In the unsupervised case, where the consequent is a formula, fulfilment checks that the consequent is satisfied by the instance:

$$\mathrm{fulfils}(\varphi \rightarrowtail \psi, I_i) \quad = \quad \begin{cases} \text{true} & \text{if } I_i \models \psi, \\ \text{false} & \text{otherwise.} \end{cases}$$

Let n be the number of interpretations in a logiset $\mathcal{I}$, Let n_a be the number of interpretations in $\mathcal{I}$ that the rule covers, n_c be the number of interpretations in $\mathcal{I}$ that the rule fulfils, and n_{ac} the number of interpretations in $\mathcal{I}$ that the rule covers and fulfils. Typical performance metrics for rule extraction are *support*, and *confidence*, defined as:

$$\begin{aligned} \mathit{support} &= \tfrac{n_a}{n}, \\ \mathit{confidence} &= \tfrac{n_{ac}}{n_a}. \end{aligned}$$

Let I be any instance from the space $\mathcal{I}_\mathrm{L}$; the support of a rule measures the probability that the rule applies to I, while the confidence of a rule measures the probability that the rule correctly applies to I (e.g., it correctly classifies I). More sophisticated evaluation metrics include *lift* and *conviction* [Bri+97], defined as:

$$\begin{aligned} \mathit{lift} &= \tfrac{n_{ac}}{n_a \cdot n_c} \\ \mathit{conviction} &= \tfrac{1 - n_c}{1 - \frac{n_a}{n}}. \end{aligned}$$

Note that fulfilment can be adapted to the regression case by using distance metrics between the label and the model's output, in place of plain equality.

§3.5 AI Workflows in SOLE

Machine Learning focuses on inducing algorithms from datasets representing past experience [Mit97]. An ML task typically coincides with an ML dataset (labelled or unlabelled), and the goal is to build a model on the dataset by means of optimization meta-algorithms, called *learning algorithms*. The process typically involves a data processing step, a learning step, and finally, a deployment step. Knowledge represented in symbolic form, however, facilitates the interpretability of the learned models offering more opportunities with respect to standard ML models, and data-driven AI workflows can be designed accordingly. As pictured in Fig. 3.7, approaching a data-driven task broadly results in a iterative process, which involves a *symbolic transformation* step where the dataset is transformed into a logiset, followed by cycles of data processing, model learning, and post-hoc processing of the learned models. At any step, new knowledge about the data can be acquired, augmenting the human, expert knowledge about the task, which can be fed back into the workflow. While in principle, the ciclicity is also possible with subsymbolic ML models, it is particularly facilitated, here, by the transformation of the dataset, which ensures that the extracted knowledge in symbolic form is reconducible to intelligible patterns and concepts.

Symbolic workflows encompass three kinds of algorithms. Symbolic Data Processing (SDP) algorithms process (unlabelled or labelled) logiset, and their output depends on the specific goal. For example, methods for symbolic feature selection, engineering and extraction can be designed to output a list of most relevant attributes, variables, features and/or atoms. Symbolic Learning (SL) algorithms also take logisets as inputs, and extract one or more symbolic models (e.g., a decision tree, or a set of classification rules). Similarly to SDP algorithms, supervised SL algorithms (e.g., CART and RIPPER) require the logiset to be accompanied by labels, while unsupervised SL algorithms do not. Learned models can either be inspected manually, or they can be input to (post-hoc) Symbolic Model Processing (SMP) algorithms, which goal can range from simple automated knowledge inspection, to simplification and/or approximations of the models (see Section 2.1). SMP algorithms can also require an additional (labelled or unlabelled) logisets, and involve some learning steps (e.g., data-driven pruning). For example, branches of decision trees can be removed, and replaced with decision lists learned with a different learning algorithm. Symbolic algorithms can range from being general-purpose and logic-agnostic, up to tailored to a very specific formalisms. For example, feature selection methods can also be tailored to modal logics and/or to multidimensional logisets for returning the most relevant worlds, variables, and/or features. In all cases, of course, each algorithm may encompass a set of (hyper)parameters.

§3.6 Sole.jl: The Julia implementation

At the current state, the framework is implemented in the Julia programming language as *Sole.jl*, which is, to the best of our knowledge, the first programming framework specifically tailored for symbolic modelling and learning. Sole.jl includes Julia packages for different purposes, and provides well-known and novel procedures for Symbolic Data Processing (e.g., feature extraction), Symbolic Learning (e.g., decision trees), and Symbolic Model Processing (e.g., post-hoc analysis

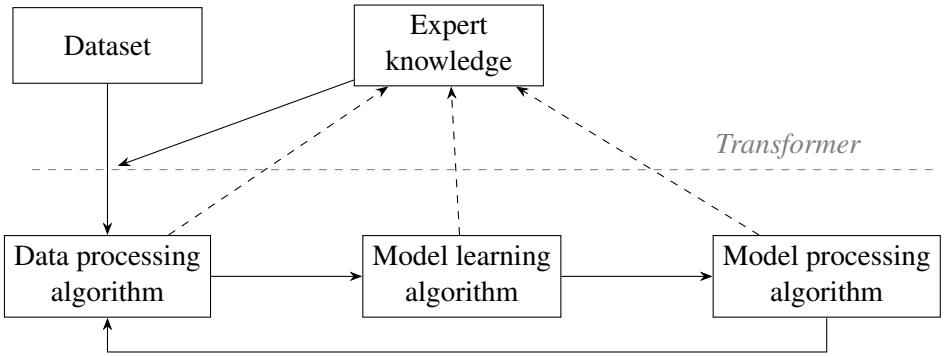


FIGURE 3.7: Iterative symbolic approach at a data-driven task.

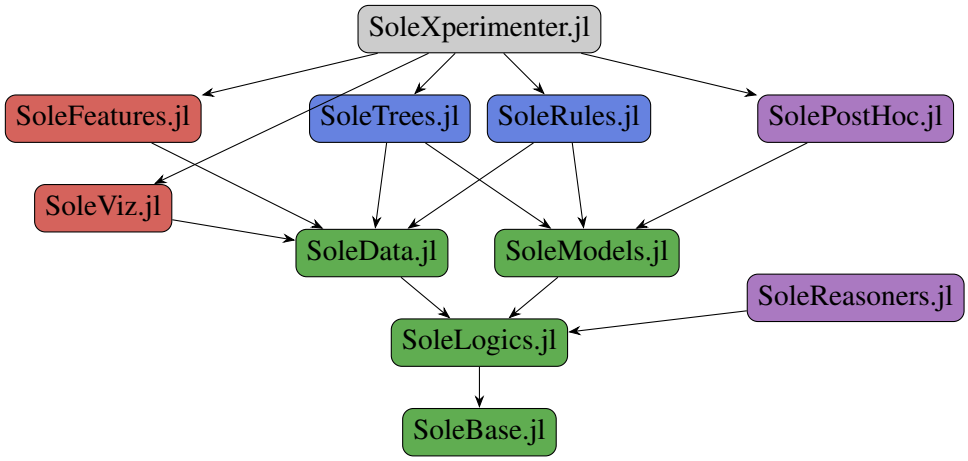


FIGURE 3.8: Dependency graph for packages in Sole.jl, the Julia implementation of SOLE.

of the learned models). The packages are described as follows, with their dependency graph depicted in Fig. 3.8:

- *SoleLogics.jl* provides the infrastructure for representing logical formulas, and algorithms for general-purpose, and logic-specific model checking;
- *SoleData.jl* provides implementations of logisets that are optimized for checking many (similar) formulas. Note that this is crucial for efficient symbolic learning;
- *SoleModels.jl* provides implementations of symbolic models, and tools for extracting their knowledge in logical form;
- *SoleFeatures.jl* provides algorithms for feature engineering on symbolic data;
- *SoleViz.jl* provides routines for producing visualizations of symbolic data;
- *SoleTrees.jl* provides algorithms for learning tree-based models;

- *SoleRules.jl* provides algorithms for learning rules (e.g., decision list learning, association rule mining);
- *SoleReasoners.jl* provides algorithms for deciding satisfiability, validity, entailment of logical formulas;
- *SolePostHoc.jl* provides algorithms for inspecting and simplifying the learned models.

Online documentation for some packages is available online[4].

The code makes extensive use of Julia's functional programming capabilities, flexible type system and multiple dispatch capabilities [Kwo20], which are earning this language much attention from the scientific computing community. Sole.jl follows principles of *modularity*, *abstraction* and *extensibility*. Objects are mainly defined by their abstract interface, with extensive use of programming traits, which allow optimized compilation of the code. At all scales, the defined type trees cover many abstraction levels, in which the currently implemented composite types meet the needs for specific (but, often, elegantly complete) cases. As an example, a single data structure defined in SoleData.jl, with an internal memoization policy described in [Man+23b], implements all HS^d-logisets on scalar atoms; this structure is useful for learning from the great realm of tabular, time-series, images, video data. This design pattern allows for elasticity in the implementation of algorithms for data analysis and learning, which can be either generic, agnostic of the shape of the data, or tailored to special kinds of logisets.

Altogether, SOLE is an ongoing effort for unifying symbolic methods for machine learning and data mining. The Julia implementation have been presented at JuliaCon2022 and JuliaCon2023, and received appreciation within the community. Online lectures and material (including code and workflow examples) for a 10-hour PhD tutorial on Modal Decision Tree learning in Sole.jl are available online[5].

[4] For example, the documentation for SoleLogics.jl is available at: https://aclai-lab.github.io/SoleLogics.jl/

[5] https://github.com/aclai-lab/modal-symbolic-learning-course/

CHAPTER 4

A FEW SYMBOLIC ALGORITHMS

A system where all parts react the same way is a system with a fatal flaw.

Mamoru Oshii, Ghost in the Shell

In this chapter we build upon the foundations of data-driven symbolic methods based on modal logics, and present three approaches based on K formalisms for Symbolic Data Processing (SDP), Symbolic Learning (SL), and Symbolic Model Processing (SMP), respectively. Section 4.1 presents a filter-based method for supervised and unsupervised symbolic feature selection on HS^d logisets, previously presented in [Cav+23; PC24]. Section 4.2 presents a supervised algorithm for tree-based learning on multimodal K-logisets, previously presented in [DM+22; Man+23b]. Section 4.3 presents a post-hoc processing method for extracting a decision list from a learned ensemble of modal decision trees, previously presented in [Ghi+23]. In these sections we do not report experimental results; however, these methods are all implemented and publicly avaliable in the Sole.jl programming framework, and they have all shown to provide interesting results. For an experimental evaluation of the first and third method, please refer to the original publications; on the other hand, the second part of this thesis presents experimental results obtained with the tree-based learning method presented in Section 4.2.

§4.1 ModalFILT: Feature Extraction and Selection for Modal Data

As mentioned in Chapter 3, and as we will see in Part II of this thesis, when symbolic models are used to probe d-dimensional datasets, they can unearth patterns expressed in HS^d. Examples of such patterns, translated to natural language, range over all possible qualitative expressions concerning values of features on hyperrectangles. In an hypothetical case in which instances are patients under observation and are described by multivariate time-series that include the variable *temperature* and the variable *blood pressure*, for example, we may be looking at sentences such as:

when the number of temperature peaks in a patient is greater than 3 in an interval of time, his/her blood pressure decreases quicker than 10mmHG/hour in some interval contained in the reference one,

which would correspond to a formula such as:

$$[G](peaks(temp) > 3 \rightarrow \langle D \rangle (deriv(blood) > 10)).$$

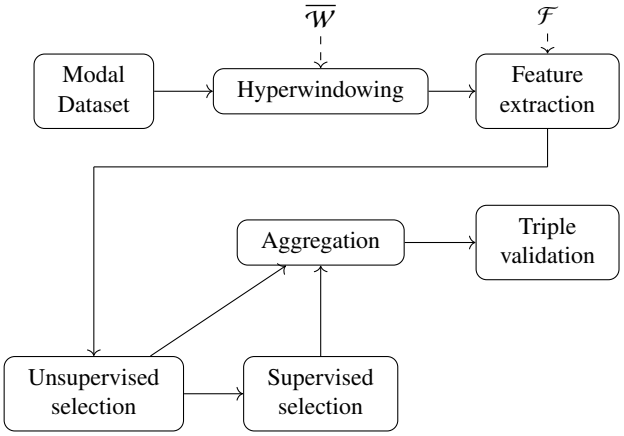


FIGURE 4.1: Pictorial representation of the feature extraction and selection approach presented.

As another example, in a different hypothetical case in which instances are aereal images whose pixels are described by hyperspectral color channels, we may express that:

when the maximum value of the red color in an image if greater that 125 in some rectangle, there exist some adjacent rectangle with a minimum value of green that is less than 64,

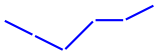
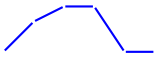
which would give rise to a formula such as:

$$[G](max(red) > 125 \rightarrow \langle EC \rangle (min(green) < 64)).$$

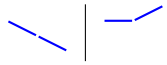
Modal formulas are discovered from a dataset in the same way in which propositional ones are: by systematic model checking on all instances. Such a computational process is expensive by nature, and the number of variables $\mathcal{V}$, feature extraction functions $\mathcal{F}$, and worlds $\mathcal{W}$ to be examined has a great influence on the computational load.

As we consider the scope of typical high-dimensional datasets, teeming with hundreds or even thousands of variables across multiple dimensions and subject to numerous candidate feature extraction functions, the resulting set of potential features easily spirals into the tens of thousands. Hence, a pivotal aspect of data processing is the thoughtful selection of a substantially smaller and more tractable subset of these features for in-depth analysis. This selection must be informed by the symbolic examination of hyperrectangles and their underlying relations within the multidimensional analytical space. A systematic method for feature extraction and selection for dimensional data can be pictorially represented as in Fig. 4.1 and outlined as follows.

Recall that the number of hyperrectangles in a d-dimensional space $\mathbb{D}^d$ of size $(N, \ldots, N)$ is $(N(N+1)/2)^d$. Within these hyperrectangles, evaluating variables under distinct functions intensifies the computational load. Directly addressing every potential combination of variables and functions over all hyperrectangles is computationally impractical, if not outright infeasible. As a practicable alternative, we introduce an approach which partitions a finite and discrete d-dimensional space into a manageable collection $\overline{\mathcal{W}} \subset \mathbb{H}(\mathbb{D}^d)$ of hyperrectangles, designated

$\mathcal{V}$	V_1	V_2
I_1		
I_2		

(A) Two (raw) input multivariate time-series.

$\mathcal{V}$	V_1		V_2	
$\overline{\mathcal{W}}$	w_1	w_2	w_1	w_2
I_1				
I_2				

(B) Hyperwindowing of the time-series using two windows $\overline{\mathcal{W}} = \{w_1, w_2\}$.

$\mathcal{V}$	V_1						V_2					
$\overline{\mathcal{W}}$	w_1			w_2			w_1			w_2		
$\mathcal{F}$	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3
I_1	$v^{(1)}_{111}$	$v^{(1)}_{112}$	$v^{(1)}_{113}$	$v^{(1)}_{121}$	$v^{(1)}_{122}$	$v^{(1)}_{123}$	$v^{(1)}_{211}$	$v^{(1)}_{212}$	$v^{(1)}_{213}$	$v^{(1)}_{221}$	$v^{(1)}_{222}$	$v^{(1)}_{223}$
I_2	$v^{(2)}_{111}$	$v^{(2)}_{112}$	$v^{(2)}_{113}$	$v^{(2)}_{121}$	$v^{(2)}_{122}$	$v^{(2)}_{123}$	$v^{(2)}_{211}$	$v^{(2)}_{212}$	$v^{(2)}_{213}$	$v^{(2)}_{221}$	$v^{(2)}_{222}$	$v^{(2)}_{223}$

(C) Feature extraction using a set of functions $\mathcal{F} = \{f_1, f_2, f_3\}$.

$\mathcal{V}$	V_1						V_2					
$\overline{\mathcal{W}}$	w_1			w_2			w_1			w_2		
$\mathcal{F}$	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3	f_1	f_2	f_3
I_1	$v^{(1)}_{111}$	$v^{(1)}_{112}$	$v^{(1)}_{113}$	$v^{(1)}_{121}$	$v^{(1)}_{122}$	$v^{(1)}_{123}$	$v^{(1)}_{211}$	$v^{(1)}_{212}$	$v^{(1)}_{213}$	$v^{(1)}_{221}$	$v^{(1)}_{222}$	$v^{(1)}_{223}$
I_2	$v^{(2)}_{111}$	$v^{(2)}_{112}$	$v^{(2)}_{113}$	$v^{(2)}_{121}$	$v^{(2)}_{122}$	$v^{(2)}_{123}$	$v^{(2)}_{211}$	$v^{(2)}_{212}$	$v^{(2)}_{213}$	$v^{(2)}_{221}$	$v^{(2)}_{222}$	$v^{(2)}_{223}$

(D) Unsupervised selection step, followed by a supervised selection step. Red cells represent tuples (variable-window pairs and variable-window-feature triples) that did not pass the selection, while green cells represent tuples that passed the selection.

FIGURE 4.2: Layered pipeline for feature extraction and selection for high-dimensional datasets. The case of time-series data is shown.

as *hyperwindows*. This set must cover every point within the space, ensuring comprehensive coverage without exhaustive enumeration, but they need not to be mutually exclusive. The selection of hyperwindow parameters determines the granularity and coverage of the dataset's features. By evaluating features within these hyperrectangles, one can approximate the behaviors of the features across the broader sets of all hyperrectangles with a controlled computational effort.

Upon establishing the hyperwindowing of the dataset, feature extraction begins. Extracting features across all hyperwindows in $\overline{\mathcal{W}}$ results in a transformed dataset with dimensions influenced by the number of feature extraction functions applied. Normalization of extracted features is essential for maintaining comparability across different scales. This process results in a tabular dataset, offering a structured compilation of features encapsulating the intrinsic variations within hyperwindows. The tabular dataset (comprised of the application of each feature extraction function within each hyperwindow) serves as a foundation for subsequent selection steps.

Two feature selection steps follow. First, an unsupervised feature selection is applied, leveraging normalization to ascertain an individual score for each variable-hyperwindow-function combination. A predefined fraction of the highest-scoring combinations is retained, condensing the dataset further. This step operates independently and, through its application, *prunes* the dataset, eliminating features least likely to carry pertinent information. This dimensionality reduction facilitates a more focused investigation in the later stages of analysis, particularly as supervised methods tend to be computationally more intensive. Thes, a supervised feature selection step applies a similar scoring and pruning process but within a target-oriented context. By considering the relationship between features and outcomes, supervised selection assigns relevance scores and filters the dataset to isolate the most predictive feature combinations.

Applying multidimensional feature extraction and selection returns a set of triples of the type *(variable, hyperwindow, function)*. These, in turn, can be combined with *aggregation functions*. A very simple aggregation process consists of evaluating a specific variable with the number of times in which it has been selected in the selected triples; the same can be done with windows and feature extraction function. Other, more elaborate aggregations can be designed, that take into account the specific scores. In general, aggregations may meld scores across multiple dimensions, allowing us to concentrate on the most influential variables, functions, or windows. As a final consideration, we observe that in the supervised case there exist popular feature selection methods based on hypothesis testing. Such a strategy in presence of several thousands of potential selectees is cautioned against, given the risk of cumulative probabilistic errors leading to unreliable outcomes. However, the selected triples *can* be tested for their ability to distinguish classes (which is different from using the result of the test as part of the selection process). In the proposed framework, we include a *a posteriori* set of tests on a random sub-selection of triples, that allows us to evaluate the quality of the results. Fig. 4.2 exemplifies the process for time-series data.

The outlined methodology balances computational feasibility and exhaustive feature inspection within d-dimensional datasets. Hyperwindowing represents a compromise, reducing an otherwise insupportable analytical load into a tractable strategy. However, the size and arrangement of hyperwindows are pivotal and may necessitate empirical tuning to adequately capture the nuances of the dataset. Feature extraction and the two-layer selection process exemplify

an approach focusing computational resources effectively from unspecific (unsupervised) to targeted (supervised) reduction. Refer to Cavina et al. [Cav+23] and Patrik Cavina [PC24] for an experimental evaluationo.

§4.2 ModalCART: Tree-Based Learning for Modal Data

While classical, propositional decision tree learning is well-defined for the case of tabular data, propositional trees cannot deal natively with non-tabular data; in fact, it is often the case that, when dealing with non-tabular datasets, machine learning practitioners apply feature engineering methods that preventively reduce the data to a non-relational description of itself, prior to learning a decision tree. This process denatures the data and it can easily challenge the transparency and performance of resulting the model. Recently, efforts have made to extend known learning algorithms to modal logics. In this section, we present ModalCART [DM+22; Man+23b], the extension of the well-known CART algorithm [Bre+84] to modal logic K. Of course, while the extension is made to K, it also straightforwardly applies to other modal logics, including HS^d and its fragments.

Learning optimal decision trees is known to be NP-hard [HR76], and requires heuristic strategies for exploring the space of formulas for a given formalism. The CART algorithm, for example, follows a divide-and-conquer approach, and is based on a simple strategy where at each recursive, greedy step, only atoms and their negations are explored; the most informative one is, then, adopted as antecedent of a decision branch and, ultimately, the resulting propositional decision tree represents a formula in *disjunctive normal form* (DNF), that is, a disjunction of conjunctions of possibly negated atoms, as in:

$$(\overline{p}_1^1 \wedge \ldots \wedge \overline{p}_{n_1}^1) \vee \ldots \vee (\overline{p}_1^m \wedge \ldots \wedge \overline{p}_{n_m}^m),$$

where $\overline{p} \in \mathcal{AP} \cup \{\neg p \mid p \in \mathcal{AP}\}$. A well-known result in propositional logic states that any formula has an equivalent one in DNF; thus, more having so-simple antecedents does not affect the expressive completess of this model with respect to P.

At the modal level, however, there is no generally accepted normal form [Bie09] and, in order to mantain expressive completeness, this strategy must be extended, so that complex modal formulas involving nestings can be built. To explore the space of modal formulas in an expressively complete way we, first, define an ad-hoc computational model called a *K-tree*. K-trees can encode any formula in K, and can be translated into (generalized) decision tree models, as defined in Section 3.4. Then, the extension of the CART learning algorithm for K-trees, called ModalCART, is presented. While generalizing the approach to multiple classes is immediate, for the sake of simplicity, we assume a binary classification setting, that is, a setting with only two possible output values C_0 and C_1, referred to as the *classes*. The input is, thus, a labelled modal logiset $\mathcal{I} = \{I_1, \ldots, I_n\}$ with labels $o_1, \ldots, o_n$, such that $o_i \in \mathbb{O} = \{C_0, C_1\}$.

§4.2.1 K-trees: An Alternative Representation of K-formulas

K-trees are (enriched) binary trees, with leaves and edges are equipped with label values. Leaf labels identify the different classes; edge labels are atomic logical elements which are, then,

composed to obtain complex formulas. A K-tree associates a formula to every class it features (i.e., every different leaf label) and it classifies an instance (i.e., an interpretation of K) into a class if and only if the instance satisfies the formula corresponding to that class.

In the following, let $t = (\mathcal{V}, \mathcal{E})$ denote a *full directed binary tree* with *nodes* in $\mathcal{V}$ and *edges* in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We denote by $\mathcal{V}^{\ell,t}$ (or simply $\mathcal{V}^{\ell}$, if t is clear from the context) the set of its *leaf nodes* (or, simply, *leaves*), by $\mathcal{V}^{\iota,t}$ (or simply $\mathcal{V}^{\iota}$, if t is clear from the context) the set of its *internal nodes* (i.e., non-root and non-leaf nodes), and by $\rho(t)$ its *root*. Moreover, we usually use $v, v', \ldots, v_1, v_2, \ldots$ to refer to generic nodes, and $\ell, \ell', \ldots, \ell_1, \ell_2, \ldots$ to refer to leaves. Each non-leaf node v of a tree t has a *left* (resp. *right*) *child* $\swarrow(v)$ (resp., $\searrow(v)$), and each non-root node v has a *parent* $\uparrow(v)$. For a node v, the set of its *ancestors* (v included) is denoted by $\uparrow^*(v)$, where $\uparrow^*$ is the transitive and reflexive closure of $\uparrow$; we also define $\uparrow^+(v) = \uparrow^*(v) \setminus \{v\}$, and we say that if $v' \in \uparrow^*(v)$, then v is a *descendant* of v'.

Moreover, given a tree t, a *path* $\pi^t = v_0 \rightsquigarrow v_h$ *in* t (or, simply, π if t is clear from the context) of *length* $h \geq 0$ between two nodes v_0 and v_h is a finite sequence of $h+1$ nodes such that $v_i = \uparrow(v_{i+1})$, for each $i = 0, \ldots, h-1$. We denote by $\pi_1 \cdot \pi_2$ the operation of *appending* the path π_2 to path π_1. We also say that a path $v_0 \cdot v_1 \rightsquigarrow v_h$ is *left* (resp., *right*) if $v_1 = \swarrow(v_0)$ (resp., $v_1 = \searrow(v_0)$). For a node v in t, π^t_v (or, simply, π_v if t is clear from the context) denotes the unique path $\rho(t) \rightsquigarrow v$. A *branch of* t is a path π^t_ℓ, for some $\ell \in V^\ell$. Finally, given two paths π^t_1, π^t_2, we denote by $\pi^t_1 \sqsubseteq \pi^t_2$ the fact that π^t_1 is a, not necessarily proper, prefix of π^t_2.

Let $\mathcal{AP}$ a finite set of atoms, and define the set of *modal decisions*:

$$\Lambda = \{p, \neg p \mid p \in \mathcal{AP}\} \cup \{\top, \bot, \Diamond\top, \Box\bot\}.$$

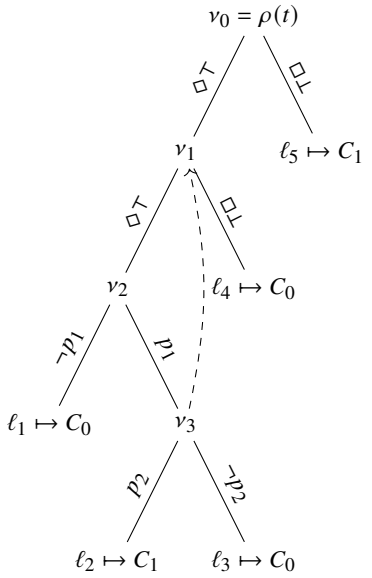
Then, an *K-tree* (*over* $\mathbb{O}$) is an object of the type:

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

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

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

An example of K-tree can be seen in Fig. 4.3 (left-hand side). Intuitively, K-tree is a rooted binary tree, with backward- and forward-edges enriching the underlying tree structure; as we will see, these additional edges ensure the expressive completeness of the model, that is, the ability of a K-tree to express any formula of K. Condition 1 states that the edge labels of each pair of outgoing edges (v, v') and (v, v'') from a node v must be one the (logical) negation of the other. Condition 2 states that if the backward-edge links of two nodes v and v' are not self-loops, then such links are not equal. Condition 3 states that if v' is the backward-edge link of v, v'' is an



$$
\begin{aligned}
\varphi_{\nu\rightsquigarrow\nu} &= \top, \text{ for all } \nu \in \mathcal{V},\\
\varphi_{\uparrow(\nu)\rightsquigarrow\nu} &= e(\uparrow(\nu),\nu), \text{ for all } \nu \in \mathcal{V}\setminus\{\rho(t)\},\\
\varphi_{\nu_1\rightsquigarrow\ell_1} &= \Diamond(\top \wedge \neg p_1),\\
\varphi_{\rho(t)\rightsquigarrow\nu_2} &= \Diamond(\top \wedge \Diamond\top),\\
\varphi_{\rho(t)\rightsquigarrow\ell_1} &= \Diamond(\top \wedge \Diamond(\top \wedge \neg p_1)),\\
\varphi_{\nu_1\rightsquigarrow\nu_3} &= \Box(\top \rightarrow p_1),\\
\varphi_{\rho(t)\rightsquigarrow\nu_3} &= \Box(\top \rightarrow \Box(\top \rightarrow p_1)),\\
\varphi_{\rho(t)\rightsquigarrow\ell_2} &= \Diamond(\Box(\top \rightarrow p_1) \wedge p_2),\\
\varphi_{\rho(t)\rightsquigarrow\ell_3} &= \Box(\Box(\top \rightarrow p_1) \rightarrow \neg p_2),\\
\varphi_{\rho(t)\rightsquigarrow\ell_4} &= \Box(\top \rightarrow \Box\bot),\\
\varphi_{\ell} &= \bigwedge_{\pi\sqsubseteq\pi_\ell} \varphi_\pi, \text{ for all } \ell \in \mathcal{V}^\ell,\\
\varphi_{C_0} &= \varphi_{\ell_1} \vee \varphi_{\ell_3} \vee \varphi_{\ell_4},\\
\varphi_{C_1} &= \varphi_{\ell_2} \vee \varphi_{\ell_5}.
\end{aligned}
$$

FIGURE 4.3: On the left-hand side, a K-tree t; on the right-hand side, all its path-, leaf-, and class-formulas. For each node, non-displayed backward-edges and forward-edges are assumed to be self-loops. Moreover, in this example, $\nu_3 = \zeta(\nu_0 \rightsquigarrow \ell_2) = \zeta(\nu_0 \rightsquigarrow \ell_3)$, and, within $\nu_0 \rightsquigarrow \ell_2$, it holds that $\mathcal{A}(\nu_1, \nu_3)$, while we have $\mathcal{D}(\nu_i, \nu_j)$ for all nodes ν_i, ν_j within path $\nu_0 \rightsquigarrow \ell_3$. Finally, $\varphi_{\rho(t)\rightsquigarrow\ell_1}$ is an example of non-implicative formula, and $\varphi_{\rho(t)\rightsquigarrow\ell_3}$ is an example of an implicative one.

ancestor of ν (different from ν) and ν' is an ancestor of ν'' (different from ν''), then ν' must be an ancestor of $b(\nu'')$ (different from $b(\nu'')$), that is, $b(\nu'')$ must be a node on the path $\nu' \rightsquigarrow \nu''$; in other words, crossing backward-edges are not allowed. Condition 4 states that f is idempotent and that forward edges are separated from each other, and, finally, Condition 5 states that, for any edge (ν, ν'), where ν' is not a leaf, if the edge label is either $\bot$ or $\Box\bot$, then the backward-edge link of ν' cannot be ν' itself.

We now show how a K-tree gives rise to a modal formula for each of the classes. As mentioned, no normal form for K exist, for allowing one to bound the nesting of operators, and this makes the construction of formulas more complicated. The following concepts pave the way towards such a construction. Let $t = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a K-tree, and $\pi = \nu_0 \rightsquigarrow \nu_h$ be a path in t of length greater than 1. Note that, by condition 3 above, two backward-edges within a path cannot lead to the same node. Then, the *contributor of* π, denoted by $\zeta(\pi)$, is defined as the only node $\nu_i \in \pi$ such that $\nu_i \neq \nu_1$, with $0 < i < h$, and $b(\nu_i) = \nu_1$, if it exists, and ν_1, otherwise. Let $t = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a K-tree and $\pi = \nu_0 \rightsquigarrow \nu_h$ be a path in t of length greater than 1. Then, given two nodes $\nu_i, \nu_j \in \pi$, with $i, j < h$, we say that they *agree*, denoted by $\mathcal{A}(\nu_i, \nu_j)$, if $\nu_{i+1} = \swarrow(\nu_i)$ (resp., $\nu_{i+1} = \searrow(\nu_i)$) and $\nu_{j+1} = \swarrow(\nu_j)$ (resp., $\nu_{j+1} = \searrow(\nu_j)$); otherwise, we say that they *disagree*, denoted by $\mathcal{D}(\nu_i, \nu_j)$. 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*. Let $t = (\mathcal{V}, \mathcal{E}, l, e, b,$

f) be a K-tree. Then, for each path $\pi = \nu_0 \rightsquigarrow \nu_h$ in t, the *path-formula* φ_π is defined inductively as:

- if $h = 0$, then $\varphi_\pi = \top$;
- if $h = 1$, then $\varphi_\pi = e(\nu_0, \nu_1)$;
- if $h > 1$, $\lambda = e(\nu_0, \nu_1)$, $\pi_1 = \nu_1 \rightsquigarrow \zeta(\pi)$, and $\pi_2 = \zeta(\pi) \rightsquigarrow \nu_h$, then:

$$\varphi_\pi = \begin{cases} \lambda \wedge (\varphi_{\pi_1} \wedge \varphi_{\pi_2}) & \text{if} \quad \lambda \neq \Diamond\top, \mathcal{A}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \notin Im, \\ & \text{or} \quad \lambda \neq \Diamond\top, \mathcal{D}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \in Im; \\ \lambda \rightarrow (\varphi_{\pi_1} \rightarrow \varphi_{\pi_2}) & \text{if} \quad \lambda \neq \Diamond\top, \mathcal{D}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \notin Im, \\ & \text{or} \quad \lambda \neq \Diamond\top, \mathcal{A}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \in Im; \\ \Diamond(\varphi_{\pi_1} \wedge \varphi_{\pi_2}) & \text{if} \quad \lambda = \Diamond\top, \mathcal{A}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \notin Im, \\ & \text{or} \quad \lambda = \Diamond\top, \mathcal{D}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \in Im; \\ \Box(\varphi_{\pi_1} \rightarrow \varphi_{\pi_2}) & \text{if} \quad \lambda = \Diamond\top, \mathcal{D}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \notin Im, \\ & \text{or} \quad \lambda = \Diamond\top, \mathcal{A}(\nu_0, \zeta(\pi)), \text{ and } \varphi_{\pi_2} \in Im. \end{cases}$$

Moreover, for each leaf $\ell \in \mathcal{V}^\ell$, the *leaf-formula* φ_ℓ (or, simply, φ_ℓ if t is clear from the context) is defined as:

$$\varphi^t_\ell = \bigwedge_{\pi \sqsubseteq \pi_\ell} \varphi_\pi.$$

Finally, for each class C, the *class-formula* φ_L (or, simply, φ_L if t is clear from the context) is defined as:

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

In Fig. 4.3 (right-hand side) we show, by way of example, all path-, leaf-, and class-formulas of the K-tree depicted on the left-hand side.

Path-formulas are at the core of the operational semantics of K-trees. In the following definition, we define how K-trees classify modal interpretations. Let $t = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a K-tree, ν a node in t, and I an interpretation of K. Then, the *run of t on I from ν*, denoted by $t(I, \nu)$, is defined as:

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

The *run of t on I*, denoted by $t(I)$, is defined as $t(I, \rho(t))$. Moreover, I is *classified into $C \in \mathbb{O}$* by t if and only if $t(I) = C$. Following the above definition, a K-tree classifies an interpretation using its class-formulas, and it does so by checking, progressively, the path-formulas that contribute to build a leaf-formula, which, in turn, is one of the disjuncts that take part in a class-formula. Considering, again, by way of example, the tree in Fig. 4.3; ultimately, it classifies any interpretation as C_0, if and only if either one of the following three statements holds on the interpretation:

1. *there exists an accessible world from an accessible world from w_0 where $\neg p_1$ holds* (φ_{ℓ_1}), or

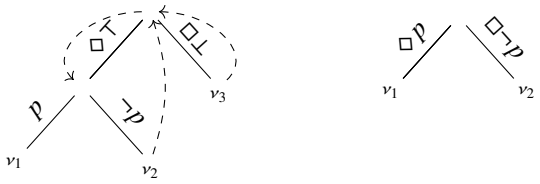


FIGURE 4.4: Compact visualiaziton of a constrained K-tree. On the left, a K-tree. On the right, its compact visualization.

2. *there exists an accessible world w'' from an accessible world w' from w_0, p_1 holds on all such worlds w'', and for each accessible world w' from w_0, it is the case that if an accessible world w'' exists where p_1 holds, then $\neg p_2$ holds on w'* (φ_{ℓ_3}), or
3. *there exists an accessible world w' from w_0, but no accessible world exists from any such world w'.* (φ_{ℓ_4}).

Since each internal node of the tree, essentially, checks a path-formula, a recursive procedure can be easily devised for translating any K-tree into a decision tree model, as presented in Section 3.4; the procedure translates any internal node of the tree into a decision branch with the local path-formula as antecedent, and any leaf node with label $C \in \mathbb{O}$ as a (final) constant model C. It is also immediate to see that K-trees can be constrained to represent (albeit with some redundancy between antecedents) traditional propositional trees with atomic antecedents, by fixing decisions $\Lambda = \{p, \neg p \mid p \in \mathcal{AP}\}$ and, for all internal node ν, $b(\nu) = f(\nu) = \nu$. Other constrained versions of K-trees have been explored. For example, by replacing Conditions 1– 5 with the following conditions:

1. if $\nu \notin \mathcal{V}^{\ell}$, then $e(\nu, \swarrow(\nu)) \equiv \neg e(\nu, \searrow(\nu))$;
2. if $e(\nu, \nu') = \Diamond\top$, then: $\nu' = \swarrow(\nu)$, $f(\nu) = \nu'$, $e(\nu', \swarrow(\nu')) \in \mathcal{AP}$;
3. if $\searrow(\nu) = \nu'$ and $e(\uparrow(\nu), \nu) = \Diamond\top$, then $b(\nu') = \uparrow(\nu)$;
4. if $\searrow(\nu) = \nu'$ and $e(\uparrow(\nu), \nu) \neq \Diamond\top$, then $b(\nu') = \nu$;
5. if $e(\uparrow(\nu), \nu) \neq \Diamond\top$, then $f(\nu) = \nu$,

we end up with a K-tree with simpler semantics and additional properties. Such a model produces constrained path-formulas from the grammar:

$$\varphi_\pi ::= p \mid (p \wedge \varphi_\pi) \mid \Diamond p \mid \Diamond(p \wedge \varphi_\pi),$$

with $p \in \mathcal{AP}$; since this kind of formulas are free of alternation of quantifiers, they allow for the design and usage of more efficient, ad-hoc model checking algorithms. As a result, these trees are easier to learn, and we can represent them in a more compact and intuitive form (e.g., Fig. 4.4).

§4.2.2 CART-like Learning of K-trees

The most typical approach to sub-optimal decision tree learning schema is simple. At the beginning, the root is associated with the entire labelled dataset $\mathcal{I}$; then, we recursively partition (or, in decision tree terms, *split*) $\mathcal{I}$ into subsets $\mathcal{I}_1, \mathcal{I}_2, \ldots$, that contain progressively more similar intra-node classes and less similar inter-node classes at any given level of the tree, in such a

ALGORITHM 1: High-level description of ModalCART, the modal extension of CART.

input : $\mathcal{I}, t$ – labelled modal logiset, amount of lookahaed
output : t – K-tree

function *ModalCART*($\mathcal{I}, t$)**:**
  t.root $\leftarrow$ *Learn*($\mathcal{I}, \emptyset, t$)
  return t
end

function *Learn*($\mathcal{I}, \mathcal{N}, t$)**:**
  if *no stopping condition applies* **then**
    $v \leftarrow$ *FindBestSubTree*($\mathcal{I}, \mathcal{N}, t$)
    v.forth.left $\leftarrow$ *Learn*($\mathcal{I}_{\swarrow(f(v))}, \mathcal{N} \cup \{v\}, t$)
    v.forth.right $\leftarrow$ *Learn*($\mathcal{I}_{\searrow(f(v))}, \mathcal{N} \cup \{v\}, t$)
    return v
  else
    $v \leftarrow$ *CreateLeafNode*($\mathcal{I}$)
    return v
end

function *FindBestSubTree*($\mathcal{I}, \mathcal{N}, t$)**:**
  $\Lambda \leftarrow$ *SplitDecisions*($\mathcal{I}$)
  $\epsilon \leftarrow -\infty$
  $v_\epsilon \leftarrow nil$
  foreach $i \in 1, \ldots, t$ **do**
    foreach $v \in$ *SubTrees*($\Lambda, \mathcal{N}, i$) **do**
      if $\epsilon <$ *InfoSplit*($\mathcal{I}, \mathcal{I}_{\swarrow(f(v))}, \mathcal{I}_{\searrow(f(v))}$) **then**
        $\epsilon \leftarrow$ *InfoSplit*($\mathcal{I}, \mathcal{I}_{\swarrow(f(v))}, \mathcal{I}_{\searrow(f(v))}$)
        $v_\epsilon \leftarrow v$
  return v_ϵ
end

way that the quantity of *information* (e.g., the *entropy*) carried by each (dataset associated to a) node decreases. The process continues until a stopping criteria applies, in which case a leaf is created. Stopping conditions are typically based on thresholding, and may require, for example, a minimum information for a split to be informative, a minimum number of samples per leaf, and/or a maximum information at each leaf. For classification tasks, stopping conditions can be *precise* (i.e., all the instances in the node having the same class) or *imprecise* (e.g., accepting non-pure nodes under a given cardinality of the associated dataset). Imprecise stopping conditions produce decision trees that tend to behave better from a practical point of view and to generalize well from training to test data. A precise stopping condition, however, trivially produces an optimal decision tree, which shows that the problem is PTIME. Algorithms based on this idea are generally called *information-based* algorithms, the most influential being CART [Bre+84]. Despite having been introduced in the seventies, recent results show that its performances are still superior to those of more recent proposals [Zha+21]. Their polynomial complexity (irrespective of the stopping criteria) is also based on assuming that the number n of attributes/variables is fixed a priori, irrespective from the number m of instances. Towards lifting the same greedy approach to the modal case, the corresponding (reasonable) assumption is that the number W of worlds of each

Kripke structure of a modal logiset is fixed a priori.

Given a K-tree $t = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ and a path π_ν from the root to a node ν, let $\hat{\pi}^t_\nu = \nu_0, \nu_1, \ldots, \nu$ (or, simply, $\hat{\pi}_\nu$, when t is clear from the context) be the *shortened path* from $\rho(t)$ to ν defined as the unique sub-sequence, if any, of the path π_ν in which $\nu_0 = \rho(t)$ and, for each i, ν_{i+1} is the successor of $f(\nu_i)$ in π_ν. If such a sub-sequence does not exist, then the shortened path is undefined. Observe that shortened paths are undefined for all nodes ν'' for which there exist nodes ν, ν' such that $\nu' = f(\nu)$ and ν'' is a descendant of ν but not of ν' (see Condition 4).

Let $t = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a K-tree, ν a node of t for which the shortened path $\hat{\pi}_\nu$ is defined, and $\mathcal{I}$ a labelled modal logiset. Then, the *ν-dataset* is defined as:

$$\mathcal{I}_\nu = \{I \in \mathcal{I} \mid I \models \bigwedge_{\nu' \in \hat{\pi}_\nu} \varphi_{\pi_{\nu'}}\}.$$

Observe that the ν-dataset only depends on the shortened path, and thus it is undefined if the shortened path $\hat{\pi}_\nu$ is; moreover, if ν is the root of t then its corresponding ν-dataset is the original dataset $\mathcal{I}$.

Splitting, which is the main greedy operation in CART, is, here, performed over a modal formula. Similarly to a propositional decision tree, then, a K-tree t splits a ν-datasets $\mathcal{I}_\nu$ based on the formulas $\varphi_{\swarrow(f(\nu))}$ and $\varphi_{\searrow(f(\nu))}$. Let $\mathcal{I}$ be a labelled modal logiset, ν a non-leaf node of t whose ν-dataset is defined. Then, the *(binary) split* of $\mathcal{I}_\nu$ is the pair defined as:

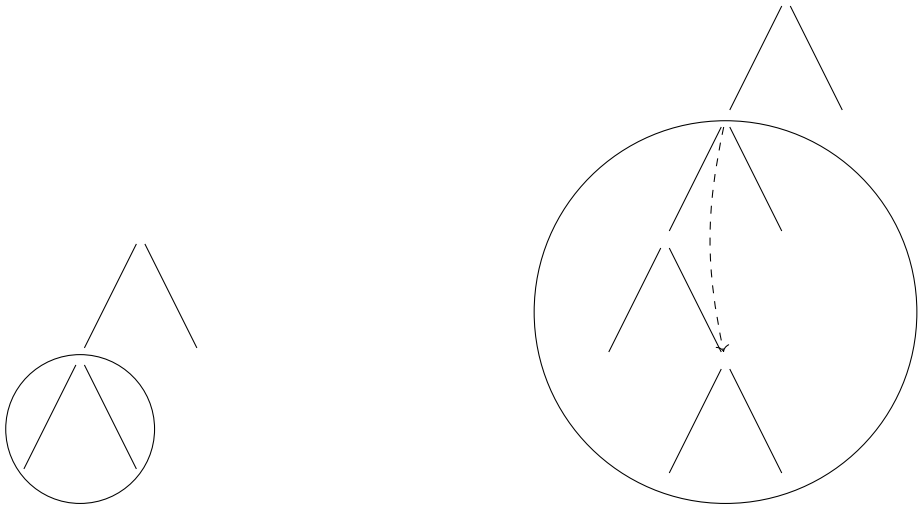
$$\left(\mathcal{I}_{\swarrow(f(\nu))}, \mathcal{I}_{\searrow(f(\nu))}\right).$$

Now, let $\mathrm{Info}(\mathcal{I})$ be any scalar measure of the information with respect to the classes contained in a labelled dataset; typical examples include *Shannon's entropy* and *Gini index*. Let $(\mathcal{I}_{\swarrow(f(\nu))}, \mathcal{I}_{\searrow(f(\nu))})$ be the split of $\mathcal{I}_\nu$, representing the two subdatasets corresponding to the children of ν. The *split information* of $\mathcal{I}_\nu$ is defined as the average information of the subdatasets, weighted by their number of instances:

$$\mathrm{InfoSplit}(\mathcal{I}_\nu) = \frac{\left|\mathcal{I}_{\swarrow(f(\nu))}\right|}{|\mathcal{I}_\nu|} \cdot \mathrm{Info}\left(\mathcal{I}_{\swarrow(f(\nu))}\right) + \frac{\left|\mathcal{I}_{\searrow(f(\nu))}\right|}{|\mathcal{I}_\nu|} \cdot \mathrm{Info}\left(\mathcal{I}_{\searrow(f(\nu))}\right).$$

Algorithm 1 shows ModalCART, the adaptation to the modal setting of CART. Let us, first, focus on the *SplitDecisions* routine, which we do not formalize, but qualitatively describe as the function that produces the smallest finite number of decisions that are useful for learning a symbolic classifier. Note that for any logiset $\mathcal{I}$ defined on a possibily infinite alphabet $\overline{\mathcal{AP}}$, there exists a finite alphabet $\mathcal{AP}$ of cardinality polynomial in the size of $\mathcal{I}$ which includes all atoms relevant for learning algorithms. We call the decisions on $\mathcal{AP}$ the *decisions induced by* $\mathcal{I}$, and we let *SplitDecisions* return these decisions.

The *SubTrees* routine, given the set of decisions Λ, a set of ancestors $\mathcal{N}$ and a positive integer i, returns all trees of height i with exactly two nodes at any given level greater than 0, with edge labels chosen from Λ, where all outgoing backward-edges lead to a node in $\mathcal{N}$, and whose root is linked via forward-edge to the (only) non-leaf node at level $i - 1$. This generalizes the splitting operation to be able to use forward edges: as schematized in Fig. 4.5a, propositional decision tree splitting consists of simply exploring all possible trees of height 1 that can be attached to the



(A) Classic, propositional split (B) Modal split with forward edge.

FIGURE 4.5: Split: propositional versus modal case.

current node, while, as in Fig. 4.5b, when using forward edges, the single split explores taller trees. Not all trees of a given height must be examined: forward edges implement a form of lookahead, by means of which we explore pairs of (mutually opposite) potential formulas.

FindBestSubTree implements the modal splitting with some lookahead t: in this way, if a class is determined by a modal formula of length less than or equal to t, such a formula will be certainly found and expressed a branch of the tree, which implies that Algorithm 1 correctly finds an optimal tree provided that t is the maximum length of a characteristic formula in the considered dataset and that the stopping condition is instantiated to checking that a node is pure. As we have observed, this is the same requirement that one should impose to an information-based greedy algorithm designed to learn a propositional decision tree in order to guarantee completeness; however, practical applications of such algorithms in both the propositional and the modal case relax such a requirement and explore a series of heuristic choices in order to extract non-overfitting models.

Let us, now, discuss on the computational complexity by, first, considering the function *FindBestSubTree*, shown in Algorithm 1. For a dataset $\mathcal{I}$ with m instances each with W distinct worlds, a set of nodes $\mathcal{N}$, and for a lookahead amount t, we show that the running time of *FindBestSubTree*$(\mathcal{I}, \mathcal{N}, t)$ is:

$$\mathcal{O}(tm^t|\Lambda|^t m^2 N^2),$$

where Λ is the set of decisions induced by $\mathcal{I}$. A single split in a K-tree is the result of model-checking one path-formula (and its negation) on every instance I of the current dataset; given a generic modal formula φ, one can simply apply a finite model-checking algorithm that labels every world with the truth value of all sub-formulas (in non-decreasing order of length). Using a

ALGORITHM 2: High-level description of the function SubTrees.

```
function SubTrees(Λ, N, i):
    Trees ← ∅
    foreach λ ∈ Λ do
        v̂ ← CreateNonLeafNode(λ)
        foreach v_b ∈ N ∪ {v̂} do
            v ← copy(v̂)
            v.back ← v_b
            if i = 1 then
                v.forth ← v
                Trees ← Trees ∪ {v}
            else
                foreach v_sub ∈ SubTrees(Λ, N ∪ {v}, i − 1) do
                    (v′, v″) ← (copy(v), copy(v))
                    (v′.left, v″.right) ← (v_sub, v_sub)
                    (v′.forth, v″.forth) ← (v_sub.forth, v_sub.forth)
                    Trees ← Trees ∪ {v′v″}
    return Trees
end
```

standard finite model checking algorithm [CES86], the cost of such a check, for all instances, is:

$$O(|\varphi|N^2 m).$$

Executing *FindBestSubTree* encompasses repeating such a process for all possible formulas that could be generated at a given step; with lookahead t, this single step requires trying all sub-trees of height at most t. As we have observed, completeness guarantees that one may limit the exploration of incomplete sub-trees with exactly two nodes at each given height. Given a set of ancestors $\mathcal{N}$ to which backward-edges may lead, the number of structurally different such sub-trees, of height $1 \leq i \leq t$, is bounded by $2^{i-1}(|\mathcal{N}| + i - 1)^i$; for each of such sub-trees, we must check $|\Lambda|^i$ different formulas which are obtained by choosing a different decision on each edge, where Λ is the the set of decisions induced by $\mathcal{I}$. Summarizing, *FindBestSubTree* runs in time:

$$O(\Sigma_{i=1}^{t} 2^{i-1}(|\mathcal{N}| + i - 1)^i |\Lambda|^i i m N^2).$$

We can bound the above expression by taking into consideration that both the number of any set of ancestors and the length of any formula to be checked during the process are bounded by m. Thus, if we bound every element of the summation with $2^{t-1}(m + t - 1)^t |\Lambda|^t m^2 N^2$, we obtain:

$$\begin{array}{lll}
& O(\Sigma_{i=1}^{t} 2^{i-1}(|\mathcal{N}| + i - 1)^i |\Lambda|^i i m N^2) & \\
= & O(t^2 2^{t-1}(m + t - 1)^t |\Lambda|^t m^2 N^2) & \text{bounding} \\
= & O(t^2 2^t (m + t)^t |\Lambda|^t m^2 N^2) & t - 1 < t \\
= & O(t^2 2^t (2m)^t |\Lambda|^t m^2 N^2) & t \leq m \\
= & O(t^2 2^{2t} m^t |\Lambda|^t m^2 N^2) & \\
= & O(t^2 m^t (4|\Lambda|)^t m^2 N^2) & \\
= & O(t^2 m^t |\Lambda|^t m^2 N^2). &
\end{array}$$

Now, we discuss the computational complexity of the ModalCART. For a dataset $\mathcal{I}$ with m instances, each W distinct worlds, and for a lookahead amount t, we show that the running time of ModalCART($\mathcal{I}, t$) is:

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

in the worst case, and:

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

in the average case, assuming constant W. *ModalCART* is a recursive procedure whose complexity can be approached via a recurrence; even with lookahead $t > 1$, each step consists, at most, of 2 recursive calls. Assuming a constant W corresponds to studying the complexity of ModalCART as the number of instances grows without changing in nature, and entails assuming that $|\Lambda| = O(m)$. Thus, the running time of *FindBestSubTree* becomes:

$$O(m^{2(t+1)}).$$

In the worst case, every split of a dataset of cardinality m ends up assigning exactly one instance to a branch and exactly $m-1$ instances to the other one, so that the recurrence is:

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

which ends up being bounded by:

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

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

$$\begin{array}{rcll} \mathcal{T}(m) & = & \frac{1}{m-1}\sum_{i=1}^{m-1}(\mathcal{T}(i)+\mathcal{T}(m-i)) + O(m^{2(t+1)}) & \\ & = & \frac{2}{m-1}\sum_{i=1}^{m-1}\mathcal{T}(i) + O(m^{2(t+1)}). & \text{by symmetry} \end{array}$$

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

$$\begin{array}{rcll} \mathcal{T}(m) & = & \frac{2}{m-1}\sum_{i=1}^{m-1}\mathcal{T}(i) + O(m^k) & \\ & \leq & \frac{2}{m-1}\sum_{i=1}^{m-1}\alpha i^k \lg(i) + O(m^k) & \text{substituting} \\ & \leq & \frac{2}{m-1}\alpha \lg(m)\sum_{i=1}^{m-1} i^k + O(m^k). & \end{array}$$

The above summations of the first $m-1$ powers can be rephrased using the Faulhaber formula, and, then, further bounded [GPW21] to obtain:

$$\begin{array}{rcl} \mathcal{T}(m) & \leq & \frac{2\alpha\lg(m)}{m-1}\left(\frac{(m-1)^{k+1}}{k+1} + \frac{(m-1)^k}{2} + \frac{k(m-1)^{k-1}}{12}\right) + O(m^k), \\ & = & \frac{2\alpha\lg(m)}{(k+1)(m-1)}(m-1)^{k+1} + \frac{(k+1)(m-1)^k}{2} + \frac{(k+1)k(m-1)^{k-1}}{12} + O(m^k). \end{array}$$

Since we assumed constant t, we have that $k \leq m-2$, and therefore, $k < k+1 \leq m-1$. Thus:

$$\begin{aligned}\mathcal{T}(m) &\leq \frac{2\alpha \lg(m)}{(k+1)(m-1)}(m-1)^{k+1} + \frac{(m-1)^{k+1}}{2} + \frac{(m-1)^{k+1}}{12}) + O(m^k),\\ &= \frac{2\alpha \lg(m)(m-1)^k}{(k+1)}(1+\frac{1}{2}+\frac{1}{12}) + O(m^k).\end{aligned}$$

For large enough values of m, all of the above amounts to proving that:

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

which is implied by:

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

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

Note how, given a modal logiset $\mathcal{I}$ with m instances it is easy to see that the length of any characteristic formula that distinguishes a class in $\mathcal{I}$ is constant as m grows if the number W of distinct worlds of the instances in $\mathcal{I}$ is. Therefore, the required amount of lookahead in a run of ModalCART can be bounded by a constant and, as a consequence, in this setting the running time of ModalCART is polynomial in m. Thus, there is a polynomial algorithm that, given a modal logiset $\mathcal{I}$ with m instances, computes an optimal decision tree for it.

Given the conjunctive nature of path-formulas, the extension to the multimodal setting also emerge naturally. In the multimodal setting, each node of the K-tree is adorned with the modality μ onto which the decision is to be interpreted, and it checks the path-formula constructed only considering the nodes adorned with the same modality. The resulting tree, then, encompasses meta-formulas, instead of K-formulas. In a similar way, ModalCART extends to the multimodal case simply by iterating, by producing SubTrees where the nodes are adorned with any modality μ. The algorithmic time complexity remains unchainged with respect to the size of the logiset.

Note that the above formulations entails that path-formulas emerging from K-trees must be detached. However, there may be some cases where we may want to fix a different condition for decisions satisfaction. For example, we can envision an *initial world* w_0 to be associated to any interpretation, onto which the pattern can be learned. w_0, then, represents the initial condition for K-tree runs, and it can become a parameter of both the model and the learning procedure. As an example, on an image the initial world could be interpreted as a privileged observation point, and could consist in the unit rectangle at the center of the image, or the largest rectangle, consisting of the image frame. Alternatively, with the suitable language, global patterns can be captured by allowing initial decisions of the kind $\langle G \rangle\, f(V) \bowtie v$.

Similarly to CART, ModalCART can be coupled with bagging techniques, resulting in the ModalCART-RF algorithm for learning modal random forests. ModalCART-RF trains k different trees, each overfitting to a random subspace of n^{sub} variables and a random subspace of m^{sub} instances. ModalCART and ModalCART-RF are available in the Julia package ModalDecisionTrees.jl, and they have already shown to be capable of extracting meaningful, non-trivial patterns; as mentioned, Part II of this thesis presents experimental results obtained with these two algorithms.

§4.3 ModalETEL: Evolutionary Rule Extraction from Modal Forests

To address the natural tendency of CART-like learning methods to overfit to the training data, ensemble learning methods are used, resulting in the well-known *random forest* approach [Bre01]. The approach trains several trees on different subsets of the data, and uses them in conjunction, producing *forest of trees* (denoted here as $\mathfrak{f}$), which work by aggregating the results of individual trees. The approach yields improvements in terms of generalization power; however, such ensembles become equivalent to black-box models, as single trees of the same forest may have very different behavior, sometimes making their interpretation very difficult. The development of methods for interpreting and explaining random forests is paramount, and several solutions have been proposed in the past. In this section, we address the problem of extracting small rulelists (that is, rulelists with final rules as immediate submodels, denoted here as $\mathfrak{I}$) from forests, both at the propositional and modal level. To this end, we propose to generalize the STEL algorithm [Den19] along two directions: from propositional to modal logic and from a cascade of multiple optimization problems to a single multi-objective optimization one. To solve the resulting optimization problem, we propose some genetic operators, resulting in the *Modal Evolutionary Tree Ensemble Learner* (or *ModalETEL*) algorithm, which fully generalizes the original one, and produces simple (modal) rulelists from (modal) forests optimizing performance and interpretability. Similarly to Section 4.2, for the sake of simplicity, we consider a binary classification setting, that is, we fix $\mathbb{O} = \{C_0, C_1\}$.

§4.3.1 Multi-objective Optimization Problems and Evolutionary Approaches

Let $\vec{x} \in \mathbb{R}^n$ be a vector of decision variables, and $o, u, s \in \mathbb{N}$. As in Collette and Siarry [CS13] and Deb [Deb01], a *multi-objective optimization problem (MOOP)* is a problem defined as follows:

$$\begin{array}{rll} \min \ / \ \max & \vec{f}(\vec{x}) & (o \text{ objective functions}) \\ \text{s.t.} & \vec{g}(\vec{x}) \leq 0 & (u \text{ inequality constraints}) \\ \text{and} & \vec{h}(\vec{x}) = 0 & (s \text{ equality constraints}), \end{array}$$

where $\vec{f}(\vec{x}) \in \mathbb{R}^o, \vec{g}(\vec{x}) \in \mathbb{R}^u$, and $\vec{h}(\vec{x}) \in \mathbb{R}^s$. A MOOP is *constrained* if $u > 0$ or $s > 0$; otherwise, it is *unconstrained*. A MOOP can be *continuous*, in which we look for real values, or *combinatorial*, in which we look for objects from a countably (in)finite set, typically integers, permutations, or graphs. For a solution to be interesting, there must be a *dominance* relation between solutions. A point $\vec{x}$ *dominates* $\vec{x}'$ if and only if $f(\vec{x}) \leq f(\vec{x}')$ for all $f \in \vec{f}$ and $f(\vec{x}) < f(\vec{x}')$ for some $f \in \vec{f}$. A solution is *non dominated* (or *Pareto optimal*) if and only if there is no other solution that dominates it, and the set of non dominated solutions is called *Pareto front*.

Solving MOOPs can be approached in several ways, such as scalar, interactive, fuzzy, and metaheuristic-based methods [CS13], which include evolutionary strategies. In evolutionary strategies for MOOPs, the solving procedure begins by generating an initial population of candidate solutions, either through random generation or guided by background knowledge. Each solution is then evaluated by calculating its objective functions and constraints. Next, the population is modified iteratively using genetic operators: selection/reproduction, crossover,

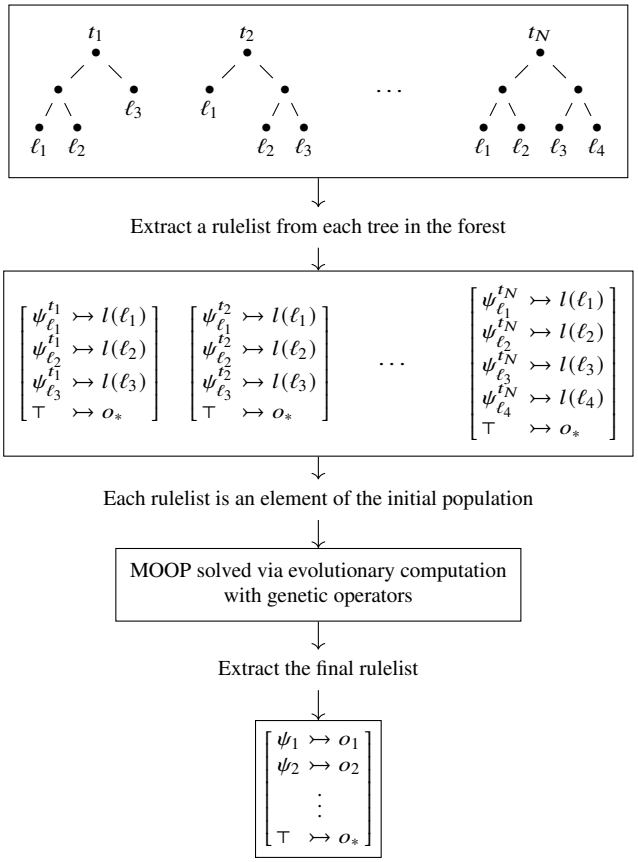


FIGURE 4.6: The proposed evolutionary approach.

and mutation. The fittest solutions are chosen for the next generation, while the unfit ones are discarded. The remaining solutions undergo a crossover operator, producing novel solutions that are further mutated to maintain population diversity. Finally, the new population is evaluated, and the process is repeated until the termination criterion is satisfied. Evolutionary algorithms have been applied to solve MOOPs in several cases (e.g., see [Deb01]).

Among the many evolutionary algorithms for solving MOOPs, NSGA-II [Deb+02] uses a fast non-dominated sorting algorithm, sharing, elitism, and crowded comparison. Elitism implies that the best solutions of the previous iteration are kept unchanged in the current one, which significantly increases the convergence speed of the algorithm. Additionally, its use of a fast non-dominated sorting algorithm contributes to a significant reduction of its computational complexity.

§4.3.2 Forest to Rulelist: An Evolutionary Approach

Extracting a rulelist $\mathfrak{l}$ from a forest $\mathfrak{f}$ model can be formalized by solving the following MOOP:

$$\begin{aligned} \min \quad & \mathit{error}(\mathfrak{l}) \\ \min \quad & \mathit{complexity}(\mathfrak{l}) \\ \text{s.t.} \quad & |\mathfrak{l}| \geq |\mathbb{O}| \\ & |\mathfrak{l}| \leq K, \end{aligned}$$

where *error* and *complexity* are measures of the statistical error and the complexity of a rulelist $\mathfrak{l}$, and $K \geq |\mathbb{O}|$ is the maximum number of rules in $\mathfrak{l}$ (including the default rule). Observe that a model must have at least a rule for each label; solutions to this MOOP are rulelist objects with high performance and low complexity. An approach is, then, designed based on multi-objective evolutionary algorithm, and is graphically represented in Fig. 4.6. In particular, given a forest $\mathfrak{f}$ of N trees, the initial population of candidate solutions is computed as the set of N rulelists derived from the trees in the forest. Then, a multi-objective evolutionary algorithm is applied on the initial population, having chosen a specific measure of error and a specific measure of complexity as objective functions. Finally, from the last evolved population by the algorithm, single rulelists are extracted and analyzed.

In a classification scenario, the statistical performance of a rulelist can be measured via global metrics such as the *overall accuracy*, *average accuracy*, *average precision*, *F1-score*, or the *error* (ε). As for complexity measures for rulelists, we identify structural ones such as the *number of rules* (ρ) and the *total length of the antecedents* (σ), defined as the total number of syntax tokens within the formulas; specifically, they only depend on the structure of the rulelist. Furthermore, inspired by Otero and Freitas [OF16], we also consider the *average delay* (δ), computed as the average index of the fired rule within the rulelist, or, in a similar way, the number of rules that are tested before the input is classified given a set of inputs to be classified by the rulelist. While performance and complexity will be objects of optimization, the delay will only be used to evaluate the results.

Similarly to Deng [Den19], the underlying idea involve listing the rules within the trees of the forest (see Section 3.4), and exploring the search space in their vicinity. With an evolutionary approach, this translates into genetic crossover/mutation operators designed to perform minimal modifications to the individuals (i.e., rulelists) in the current population. Considering that one of the objectives is to lower the complexity (in terms of interpretability) of the solutions, such operators propose rulelists with the same or fewer rules, having the rules' antecedents with the same or fewer conjuncts, which, in turn, are equally long or smaller. The proposed genetic operators can be elegantly ordered by their level of abstraction (see Fig. 4.7). From the most abstract one, *crossover*, which affects pairs of rulelists in the population, to mutation at the *rulelist level*, which affects a single rulelist, at the *rule level*, which affects a single rule of a single rulelist, down to the *conjunct level*, which affects a single conjunct, and to the *leaf level*, which affects a single atom of a single conjunct.

The *crossover* operator considers two individuals $\mathfrak{l}_1, \mathfrak{l}_2$ such that at least one of them has at least one non-default rule (if it exists) and two random indexes $1 \leq i \leq r_1$ and $1 \leq j \leq r_2$ (r_1, r_2 being, respectively, $\mathfrak{l}_1$'s and $\mathfrak{l}_2$'s number of non-default rules). Then, it returns two new

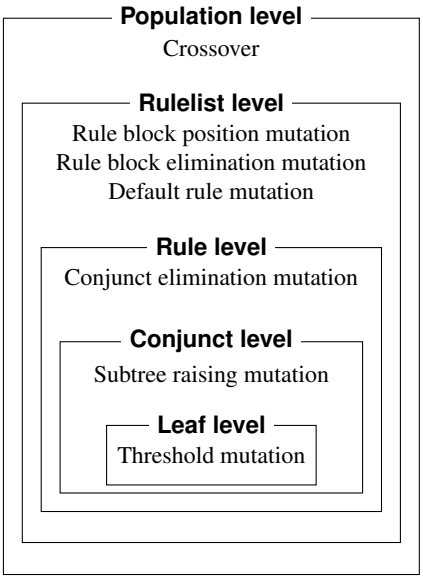


FIGURE 4.7: Hierarchical representation of the proposed genetic operators.

individuals $\mathfrak{I}'_1, \mathfrak{I}'_2$, as follows:

$$\mathfrak{I}_1 = \begin{bmatrix} & \cdots & \\ \psi_i^1 & \rightarrowtail & o_i^1 \\ & \cdots & \end{bmatrix} \qquad \mathfrak{I}_2 = \begin{bmatrix} & \cdots & \\ \psi_j^2 & \rightarrowtail & o_j^2 \\ & \cdots & \end{bmatrix}$$

$$\Downarrow$$

$$\mathfrak{I}'_1 = \begin{bmatrix} & \cdots & \\ \psi_j^2 & \rightarrowtail & o_j^2 \\ & \cdots & \end{bmatrix} \qquad \mathfrak{I}'_2 = \begin{bmatrix} & \cdots & \\ \psi_i^1 & \rightarrowtail & o_i^1 \\ & \cdots & \end{bmatrix}.$$

At the rulelist level, three mutation operators are proposed. The first one, *rule block position mutation*, considers a rulelist $\mathfrak{I}$ with at least two rules (if it exists) and chooses two indexes $1 \leq i \leq j \leq r$. Then, it returns $\mathfrak{I}'$ obtained by moving all rules from the i-th to the j-th to a different position, as in the following example:

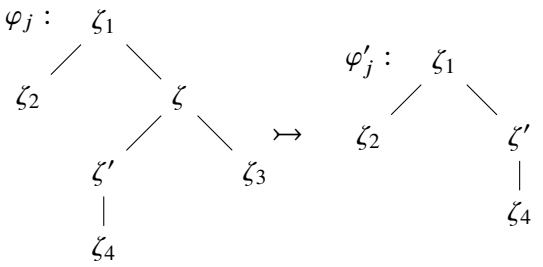
$$\mathfrak{I} = \begin{bmatrix} \psi_1 & \rightarrowtail & o_1 \\ & \cdots & \\ \psi_{i-1} & \rightarrowtail & o_{i-1} \\ \psi_i & \rightarrowtail & o_i \\ & \cdots & \\ \psi_j & \rightarrowtail & o_j \\ \psi_{j+1} & \rightarrowtail & o_{j+1} \\ & \cdots & \\ \psi_r & \rightarrowtail & o_r \\ \top & \rightarrowtail & o_* \end{bmatrix} \quad \Rightarrow \quad \mathfrak{I}' = \begin{bmatrix} \psi_i & \rightarrowtail & o_i \\ & \cdots & \\ \psi_j & \rightarrowtail & o_j \\ \psi_1 & \rightarrowtail & o_1 \\ & \cdots & \\ \psi_{i-1} & \rightarrowtail & o_{i-1} \\ \psi_{j+1} & \rightarrowtail & o_{j+1} \\ & \cdots & \\ \psi_r & \rightarrowtail & o_r \\ \top & \rightarrowtail & o_* \end{bmatrix}.$$

The second operator, *rule block elimination mutation*, considers a rulelist I with at least one non-default rule (if it exists) and chooses two indexes $1 \leq i \leq j \leq r$. Then, it returns I' obtained by removing all rules from the i-th to the j-th, as follows:

$$\mathrm{I} = \begin{bmatrix} \psi_1 & \rightarrowtail & o_1 \\ & \cdots & \\ \psi_{i-1} & \rightarrowtail & o_{i-1} \\ \psi_i & \rightarrowtail & o_i \\ & \cdots & \\ \psi_j & \rightarrowtail & o_j \\ \psi_{j+1} & \rightarrowtail & o_{j+1} \\ & \cdots & \\ \psi_r & \rightarrowtail & o_r \\ \top & \rightarrowtail & o_* \end{bmatrix} \quad \Rightarrow \quad \mathrm{I}' = \begin{bmatrix} \psi_1 & \rightarrowtail & o_1 \\ & \cdots & \\ \psi_{i-1} & \rightarrowtail & o_{i-1} \\ \psi_{j+1} & \rightarrowtail & o_{j+1} \\ & \cdots & \\ \psi_r & \rightarrowtail & o_r \\ \top & \rightarrowtail & o_* \end{bmatrix}.$$

The third operator, *default rule mutation*, considers a rulelist I and returns I' obtained by substituting the default rule's consequent with a different, randomly chosen one.

At the rule level, we propose the *conjunct elimination mutation* which, given a a rulelist I, it considers one of the non-default rules with an antecedent consisting of more than one conjunct, and returns a new individual I', removing a randomly selected conjunct of the chosen (non-default) rule. As for mutation at the conjunct level, recall that every single conjunct in a rule's antecedent may be, in general, a complex formula. Considering that any formula is naturally represented by a syntax tree (see Section 3.2), and inspired by a decision tree pruning technique called *subtree raising*, a *subtree raising mutation* operator is proposed. The operator considers a rulelist I with at least one non-default rule (if it exists) and an index $1 \leq i \leq r$ of a non-default rule, chooses an index $1 \leq j \leq k$ (k being the number of conjuncts in the antecedent in the chosen rule) of a rule $\mathfrak{r}_j$ with a non-propositional antecedent (if it exists), and two nodes ζ, ζ' in the syntax tree that represents the antecedent $\mathfrak{r}_i$ such that ζ' is a descendant of ζ. Then, it returns the rulelist I' obtained by substituting $\mathfrak{r}_i$ with $\mathfrak{r}'_i$, whose j-th conjunct φ_j, in turn, is obtained by transplanting the subtree rooted in ζ' over the one rooted in ζ, as follows:



$$\mathfrak{I} = \left[\begin{array}{lll} & & \dots \\ \mathfrak{r}_i : & \varphi_1 \wedge \ldots \wedge \varphi_j \wedge \ldots \wedge \varphi_k & \rightarrowtail \quad o_i \\ & & \dots \end{array}\right]$$
$$\Downarrow$$
$$\mathfrak{I}' = \left[\begin{array}{lll} & & \dots \\ \mathfrak{r}'_i : & \varphi_1 \wedge \ldots \wedge \varphi'_j \wedge \ldots \wedge \varphi_k & \rightarrowtail \quad o_i \\ & & \dots \end{array}\right].$$

In the above example, $\zeta, \zeta', \ldots$ are syntax tokens of logic formulas; in the modal case, for instance, ζ could be $\wedge$ and ζ' could be $\Diamond$. Observe that, as expected, subtree rising too changes a rule, and therefore an individual, in the general direction of searching for simpler ones; shorter antecedents are, in fact, more general ones (being less restrictive).

When applied to (non-tabular and) d-dimensional data, the resulting logiset encompasses atoms that are of the type $f \bowtie v$, for some feature extraction function f and test operator $\bowtie \in \{<, \leq, \geq, >\}$; for these cases, we propose a genetic operator which mutates the threshold value v. Let $std(f)$ be the standard deviation of the values that the instances in the logiset have for the extracted feature f. The *threshold mutation* operator takes as input an individual $\mathfrak{I}$ with at least one non-default rule (if it exists) and an index $1 \leq i \leq r$ of a non-default rule $\mathfrak{r}_i$. It, then, randomly chooses an index $1 \leq j \leq k$ (k being the number of conjuncts in the antecedent in the chosen rule) of $\mathfrak{r}_i$, and an atom p of the type $f \bowtie v$ in the conjunct φ_j of $\mathfrak{r}_i$. Then, it returns the individual $\mathfrak{I}'$ obtained by substituting $\mathfrak{r}_i$ with $\mathfrak{r}'_i$, whose j-th conjunct φ_j is substituted by φ'_j, in turn, obtained by replacing p with $f \bowtie v'$, where is perturbed as in $v' \in \pm 0.1 \cdot std(f) + v$, as follows:

$$p : f \bowtie v \quad \Rightarrow \quad p' : f \bowtie v',$$

$$\mathfrak{I} = \left[\begin{array}{lll} & & \dots \\ \mathfrak{r}_i : & \varphi_1 \wedge \ldots \wedge \varphi_j \wedge \ldots \wedge \varphi_k & \rightarrowtail \quad o_i \\ & & \dots \end{array}\right]$$
$$\Downarrow$$
$$\mathfrak{I}' = \left[\begin{array}{lll} & & \dots \\ \mathfrak{r}'_i : & \varphi_1 \wedge \ldots \wedge \varphi'_j \wedge \ldots \wedge \varphi_k & \rightarrowtail \quad o_i \\ & & \dots \end{array}\right].$$

Overall, the methodology allows for gaining insights from the knowledge enclosed in random forest models based on propositional and modal logics, providing a range of smaller models, that explore approximation trade-offs between performance and complexity. For an experimental evaluation of the method, the reader is referred to Ghiotti et al. [Ghi+23].

Part II

Practice

CHAPTER 5

INTRODUCTION TO PART II

No one should be scorned in physical disputes for not holding to the opinions which happen to please other people best [...]. One of these is the stability of the sun and mobility of the Earth, a doctrine believed by Pythagoras and all his followers [...].

Galileo Galilei

Although workflows consisting of symbolic methods presented in Chapter 4 produced interesting results, the most consistent results were found on classification tasks. This part of the thesis reports the results of four scientific works where the SOLE framework was used for learning and analysing modal decision trees and forests via ModalCART and ModalCART-RF. The works apply variations of the same methodology, involving factorial experimental design where different data and algorithmic parameters are explored. For each problem, the motivation for the symbolic approach is discussed, and the dataset and the data processing steps are detailed. Then, a performance analysis in cross-validation of trained K-trees (or HS-tree) and/or forests is performed, followed by an evaluation of model complexity and/or training times. Finally, a post-hoc interpretation of selected trained trees is performed.

The nature of the tasks is quite different, but they all share the aspect of dimensionality; one task is an image classification task, and the remaining three tasks are multivariate time-series classification tasks. In all cases, for random forest models (i.e., ModalCART-RF), we set an instance subspace $m^{sub} = 70\%$ of the cardinality of each training set (m), a number of trees $k = 100$, and a variable subspace $n^{sub} = 50\%$ of the number of variables (n). All the experiments were executed on a dual-socket Dell PowerEdge server with Intel Xeon Gold 6238R processors using multithreading and varying number of threads.

CHAPTER 6

LAND COVER CLASSIFICATION FROM AERIAL DATA

Keep your face always toward the sunshine,
and shadows will fall behind you.

Unknown

This section considers the task of Land Cover Classification, which involves classifying pixels in remotely sensed images. The imagery involved displays short-distance inter-pixel dependencies, which makes it a suitable case study for the classification of small image patches. In this context, we evaluate ModalCART and ModalCART-RF based on the topological, bidimensional logic HS^2_{RCC8}, achieving accuracies of 88%–99% on five multilabel classification datasets (with 6–16 classes). An extensive, quantitive comparison with several baseline models based on P is performed, as well as a qualitative comparison on the appropriateness of the extracted knowledge. Ultimately, classification rules in the two cases are extracted, and their antecedents read as simply as: *within* 3×3 *window surrounding the pixel, the minimum value of variable* 43 *is less than* 1978*, and there exists at least one rectangle of neighboring pixels in which all of the pixels have the value for variable* 83 *greater than or equal to* 638 *and the value for variable* 6 *less than or equal to* 1625. The presented results were published in Pagliarini and Sciavicco [PS23].

§6.1 Problem

Land Cover Classification (LCC) typically refers to the task 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*, or *crop field*). Despite being an instance of standard *image segmentation*, LCC has a few peculiarities that allow it to be dealt with as an *image classification* task. Images of this kind are usually captured from satellite or aerial sensors, and due to the altitude from which the images are captured, a single pixel carries the average electromagnetic response of a wide surface area (e.g., one to hundreds of square meters); as such, the classification of a pixel in the image typically depends only on a close neighborhood of pixels around it. Additionally, the imagery involved is often *hyperspectral*, that is, composed of many spectral channels, describing the strength of signals at different electromagnetic wavelengths; thus, each pixel holds a large number of values (e.g., usually in the hundreds) that, collectively, describe in a satisfactory way the nature of its content. Because of these key features, the task can be dealt with a *per-pixel*

approach: first, a dataset of pixel samples, typically represented with feature vectors, is extracted from the image; then, a machine learning method is trained on the pixel samples; finally, in the operational phase, each pixel is classified independently. Although not related with this work, it should be noted that a plethora of *object-based* approaches exists as well; refer to Kucharczyk et al. [Kuc+20] for a recent review on the topic.

LCC has a long history of being tackled with per-pixel Symbolic Learning, starting from the most influential papers, dating back to 1995–2003 [KMD95; FB97; PM03; Goe+03], and continuing to more recent works [Ber+18; BDS14]. All the work in the symbolic realm is based on propositional logics; most of it focuses on decision trees [FB97; PM03; Goe+03; Ber+18; MM11; Xu+05; KS17; Phi+20], with only a few examples of rule-based models [BDS14; ZW03]. Although some authors made remarks on how functional models can be more statistically accurate [Goe+03; Phi+20], all the literature on this side agrees on the importance of having clear classification rules for the task, as they can be: *a)* interpreted in order to identify spectral similarities and differences between different classes; *b)* validated and used for defining *standards* for class membership conditions. Nonetheless, the per-pixel LCC literature since 2015 has been flooded with highly-cited works using neural network-based methods, effectively raising the bar in terms of statistical accuracies [Hu+15; MGZ17; Roy+20; LZS17; LK17; Hon+22; ALL19; Cao+18; Jia+19; San+17]. One of the reasons why these approaches (mainly based on convolutional networks, but also recurrent networks [MGZ17; ALL19] and transformer models Hong et al. [Hon+22]) are more effective than symbolic methods is that they leverage intrinsic spatial localities that occur in the data. In fact, while the vast majority of symbolic work relies on propositional logic, and a simple feature vector consisting only of the pixel's own spectral footprint (thus disregarding the spatial structure of the image), these deep neural networks can factor in a close neighborhood of pixels around the pixel to be classified, ultimately capturing inter-pixel dependencies and correlations.

The body of symbolic work accounting for inter-pixel correlations is likely confined by the inability of propositional methods to handle spatial data and, to our knowledge, is limited to [Jia+12], which proved the efficacy of integrating the (local) feature vector with additional scalar features capturing information about the neighboring pixels. This approach is, again, based on propositional trees, and relies on a functional feature extraction step, which may undermine the interpretability of the model (on which no comments are made throughout the paper).

§6.2 Data

This case study considers five datasets that are commonly used to benchmark neural network-based methods for Land Cover Classification (LCC); see [Hu+15; MGZ17; Roy+20; LZS17; LK17; Hon+22; ALL19; Cao+18; Jia+19; San+17; Mak+15; AKH17]. The datasets are known as *Indian Pines*, *Pavia University*, *Pavia Centre*, *Salinas*, and *Salinas-A*, respectively; a summary of their characteristics is reported in Tab. 6.1. 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

TABLE 6.1: Specifications for the five public datasets for Land Cover Classification (LCC) in use.

Name	Abbr.	# channels	Spectral coverage	Spatial resolution	Image size (px)	# labels	# classes
Indian Pines	IP	200	0.4–2.45 μm	20 m/px	145×145	10249	16
Pavia University	PU	103	0.43–0.86 μm	1.3 m/px	610×340	42776	9
Pavia Centre	PC	103	0.43–0.86 μm	1.3 m/px	1096×715	148152	9
Salinas	S	200	0.4–2.45 μm	3.7 m/px	512×217	54129	16
Salinas-A	S-A	200	0.4–2.45 μm	3.7 m/px	83×86	5348	6

AVIRIS sensor (Airborne Visible/Infrared Imaging Spectrometer). The ROSIS and AVIRIS yield spectral detections covering a range of frequencies from $0.43\mu m$ to $0.86\mu m$ and from $0.4\mu m$ to $2.45\mu m$, with a number of channels of 103 and 200, respectively. Note that, in the case of hyperspectral images, there exists an implicit order between the variables and, in fact, it is often the case that hyperspectral channels with close wavelengths are correlated. 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.

The typical approach to LCC involves collecting either all labelled pixels, or a randomly sampled subset, and applying a learning algorithm for binary or multiclass classification. In some cases, authors have leveraged the spatial structure in a limited way, by considering, for the classification of each single pixel, a set of neighboring pixels; the neighborhood is generally in the form of a $d \times d$ window centered in the pixel, for a natural odd number d. In this context, we adopt the same solution, and extract, for each dataset, a collection of $d \times d$ labelled images (one for each labelled pixel). All five datasets originally presented a class imbalance, with some classes appearing thousands of times, while others only appearing a few dozen times. For the case of entropy-based decision tree learning, such imbalances generally cause biases towards the most occurring classes, and, since to extract logical descriptions that are equally accurate for all classes is a reasonable objective, this imbalance is leveled out. Thus, a fixed number P of pixels are randomly sampled for each class, using upsampling for classes with less than P samples.

A quick overview of the literature displays a wide variety of experimental practices and evalution settings when dealing with these datasets. In all cases, neural models achieved nearly-optimal performances: for Indian Pines and Pavia University, the average accuracy is in the range ~79–99% [Hu+15; MGZ17; Roy+20; LZS17; LK17; ALL19; Cao+18; San+17; Mak+15]; for Pavia Centre, the average accuracy is at ~99% [Mak+15]; for Salinas, the average accuracy is in ~92–99% [Hu+15; Roy+20; LK17; Jia+19; San+17; Mak+15]; for Salinas-A, the average accuracy is at ~97% [AKH17]. Also, propositional random forest models attained average accuracies in ~69–71% [MGZ17] on Indian Pines and Pavia University, and around ~93% on Salinas [Jia+19].

§6.3 Experimental Setting

The experiments are conducted in a randomized cross-validation setting, in which this sampling step is performed n 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. Within this context, $d = 3$, $P = 100$ and $n = 10$ were set to produce, for each dataset, 10 balanced subdatasets of 3×3 images. Furthermore, a 80%–20% balanced split was performed when splitting each subdataset into training and test, using a policy for ensuring that they were *strongly disjointed*. 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 frontier, in order to avoid *data leakage*, that occurs when part of the information in the test set appears in the training set as well, biasing the algorithm and, ultimately, affecting the performance estimation. Note that only a few works in the literature claim to prevent data leakage [ALL19] and perform a pre-balancing step [Cao+18], which may or may not affect the evaluation of the learned models. From each training-test split, different approaches to decision tree learning for LCC are deployed and compared. We separate the approaches into *pure* approaches, where the 3×3 image itself is considered, and *derived* ones, where the input image is processed beforehand using (simple) functional filters. The pure approaches included in this experiment are referred to as:

- *single-pixel* propositional approach, in which standard CART is applied to the numerical description of the pixel to be classified (that is, the central pixel), disregarding the neighboring pixels;
- *flattened* propositional approach, in which standard CART is applied to the numerical descriptions of all pixels in the 3×3 image, disregarding the spatial structure;
- *RCC8* modal approach, in which ModalCART with HS^2_{RCC8} is applied to the 3×3 image;
- *RCC5* modal approach, in which ModalCART with HS^2_{RCC5} is applied to the 3×3 image;

Each of the above settings has a corresponding derived one, obtained by applying, before training, a 3×3 average convolutional filter. In derived settings, the outer 5×5 box around each 3×3 image is first used to compute a 3×3 *average image* which is, then, fed *as is* to the learning algorithm of the corresponding pure setting; in this way, each pair of approaches is immediately comparable. To fix the ideas, consider the case of *Indian Pines*, which has 200 channels. The pure single-pixel approach trains trees that make decisions on the 200 featured values of the central pixel; on the other hand, in the corresponding derived setting (referred to as *avg*), trees are trained from descriptions of 200 values that are, each, the average of a channel within the 3×3 neighborhood. Instead, trees trained with in the pure flattened approach make 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 resulting from the 3×3 average convolution performed on the 5×5 outer box. As for the modal approaches, they are always applied on 3×3 images, except in the derived cases (*avg+RCC8* and *avg+RCC5*) the image is the result of the convolution. It is of note that, because LCC is invariant under rotation and reflection, modal approaches are more likely to grasp complex patterns when using topological relations, as opposed to directional ones.

From an operational perspective, each approach differs from the others by how data are preprocessed prior to the learning phase, and by the algorithm parametrization. As a matter of fact, when applied to datasets of 1×1 images, ModalCART behaves exactly as traditional

CART, thus the same implementation of ModalCART presented in the previous section can be used for implementing traditional and modal approaches. With regard 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: 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. As for the parametrization of the modal approaches, we fixed w_0 to be the minimum rectangle that contains the central pixel (that is, the pixel to be classified). Moreover, the set of decisions was induced by setting:

$$\mathcal{FT} = \{(max, \leq), (max, \geq), (min, \leq), (min, \geq)\},$$

and global decisions were disallowed throughout the learning.

Ultimately, considering that each approach is tested in both the single-tree and the random forest version, this setting involves comparing $5 \times (4 \times 2) \times 2 = 80$ different approaches. As mentioned above, cross-validation with 10 repetitions was deployed, therefore a total of 800 models were trained. Each model was evaluated via standard performance metrics for multiclass classification, namely, *overall accuracy*, κ *coefficient* (which relativizes the accuracy to the probability of a random answer being correct [Coh60]), and *mean precision*. With balanced test sets the overall accuracy also corresponds to the *mean recall*.

§6.4 Statistical Performance

Overall, the total computation time needed for the experiments was around 523 hours (about 21 days), using multi-threading with a number of threads varying from 26 to 32. Tab. 6.2 shows the statistical cross-validation 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 is reported. The discussion that follows is mainly based on the κ coefficient, but the other two metrics reveal similar insights. It can, first, be observed that the pure and derived flattened approaches lead to a severe performance degradation; the average κ coefficient hovers around 0% in all cases, and it is below 0% in most cases. This is likely due to the large number of decisions, which has a negative effect on the greedy learning strategy; as such, these two approaches can be excluded from a relevant method comparison. Pure approaches tend to attain lower performances than their derived counterpart; this is always the case for the single-pixel approach, while it holds for the modal approaches in fourteen out of twenty cases. Across all datasets, the best performances in the pure and derived cases are always attained with modal approaches, which, therefore, appear to yield consistently better results than propositional ones. The improvements seem to be proportional to the intrinsic hardness of the classification dataset; compared with the single-pixel approach, the average improvement of modal approaches ranges from about 1 percentage point (Salinas-A, decision tree) to 8.5 percentage points (Indian Pines, random forest). From a qualitative perspective, there does not seem to be a clear winner between the two topological logics: RCC8 and RCC5 behave quite similarly, with the greatest difference in terms of κ being as low as 2 percentage points (Indian Pines, derived modal approaches). However, across the 4 tree types, RCC8 yields the best results in eight out of twenty cases, while

TABLE 6.2: Cross-validation results on five datasets using different approaches based on CART and ModalCART (in bold). For each approach, the average and the standard deviation across 10 repetitions of the κ coefficient, the overall accuracy, and the mean precision are reported in percentage points. For each result line, the average performance of the best pure approach and the best derived approach is highlighted.

			Pure approaches												Derived approaches											
Dataset			single-pixel			flattened			**RCC8**			**RCC5**			avg			avg+flattened			**avg+RCC8**			**avg+RCC5**		
			κ	acc	prec	κ	acc	prec	κ	acc	prec	κ	acc	prec	κ	acc	prec	κ	acc	prec	κ	acc	prec	κ	acc	prec
Random Forest	IP	avg	82.9	83.9	84.6	0.5	6.7	6.4	**91.5**	**92.0**	**92.6**	91.2	91.8	92.4	86.8	87.6	88.2	−1.9	4.4	4.7	92.0	92.5	93.1	**94.0**	**94.4**	**94.8**
		std	2.2	2.0	1.9	1.5	1.4	1.0	2.3	2.2	2.2	1.9	1.8	1.7	3.1	2.9	2.9	2.0	1.9	3.6	2.2	2.1	2.0	2.8	2.7	2.5
	PU	avg	81.2	83.3	84.4	−3.5	8.0	11.4	87.2	88.7	89.8	**87.9**	**89.2**	**90.0**	83.4	85.2	86.6	−4.5	7.1	9.9	87.8	89.1	90.0	**87.9**	**89.2**	**90.3**
		std	4.0	3.5	4.1	5.3	4.7	6.7	4.1	3.7	3.0	5.5	4.9	4.4	2.8	2.5	2.3	4.3	3.8	5.9	3.7	3.3	3.3	3.9	3.5	3.4
	PC	avg	88.1	89.4	90.6	−5.5	6.2	12.8	92.0	92.9	93.5	**92.2**	**93.1**	**93.6**	91.8	92.7	93.3	−6.8	5.1	9.0	**92.6**	**93.4**	**94.0**	**92.6**	**93.4**	**94.0**
		std	4.5	4.0	3.5	4.4	3.9	7.1	3.2	2.9	2.7	3.2	2.9	2.6	3.2	2.8	2.6	5.1	4.5	10.5	2.8	2.5	2.3	3.1	2.7	2.6
	S	avg	92.9	93.3	93.6	−4.1	2.4	3.2	**94.7**	**95.1**	**95.4**	94.1	94.5	94.8	94.5	94.9	95.3	−3.9	2.6	4.8	95.8	96.1	96.3	**95.9**	**96.2**	**96.4**
		std	1.4	1.3	1.3	1.5	1.4	2.8	1.7	1.6	1.5	1.4	1.3	1.4	1.3	1.2	1.1	1.7	1.6	2.1	1.3	1.2	1.2	1.5	1.4	1.4
	S-A	avg	97.0	97.5	97.8	−4.2	13.2	11.3	**98.8**	**99.0**	**99.1**	98.6	98.8	98.9	97.6	98.0	98.2	−5.4	12.2	11.8	**98.4**	**98.7**	**98.8**	98.0	98.3	98.5
		std	2.7	2.3	2.0	8.5	7.1	5.9	1.7	1.4	1.3	2.1	1.8	1.6	2.8	2.3	2.2	8.7	7.2	6.1	2.8	2.3	2.1	3.0	2.5	2.3
Decision Tree	IP	avg	70.9	72.7	72.8	0.2	6.4	6.5	78.3	79.6	80.5	**79.4**	**80.7**	**81.6**	76.7	78.2	79.5	0.6	6.8	7.6	80.5	81.7	82.9	**81.3**	**82.5**	**83.6**
		std	4.1	3.8	4.3	2.5	2.3	1.8	3.4	3.1	3.1	3.2	3.0	3.1	3.7	3.4	3.3	1.8	1.7	2.0	4.1	3.9	4.0	2.7	2.6	3.4
	PU	avg	71.6	74.8	75.9	−0.9	10.3	10.8	77.5	80.0	80.6	**77.6**	**80.1**	**80.6**	73.9	76.8	78.2	−1.8	9.6	10.2	76.9	79.4	80.5	**77.5**	**80.0**	**81.0**
		std	4.3	3.9	3.6	5.7	5.0	4.8	7.5	6.7	7.3	6.6	5.8	6.6	5.5	4.9	5.5	5.8	5.2	6.1	4.6	4.1	4.8	4.6	4.1	4.6
	PC	avg	84.9	86.6	88.3	−1.8	9.6	9.8	89.1	90.3	91.5	**89.4**	**90.6**	**91.6**	**89.8**	**90.9**	**91.7**	−2.6	8.8	11.4	88.9	90.1	91.0	88.8	90.0	90.9
		std	4.1	3.7	2.6	5.8	5.2	4.1	3.1	2.8	2.5	2.9	2.6	2.3	3.5	3.1	2.6	5.4	4.8	7.5	4.2	3.8	3.3	4.1	3.6	3.2
	S	avg	88.5	89.2	89.9	0.2	6.4	6.2	**91.2**	**91.8**	**92.2**	90.8	91.4	91.9	92.5	93.0	93.6	−1.4	4.9	4.4	93.2	93.6	94.1	**93.3**	**93.8**	**94.3**
		std	2.4	2.3	2.2	1.5	1.4	1.7	1.9	1.8	1.7	2.2	2.1	2.0	1.4	1.3	1.4	2.4	2.3	2.6	2.6	2.4	2.2	2.2	2.1	1.9
	S-A	avg	95.4	96.2	96.6	−3.6	13.7	12.4	**96.4**	**97.0**	**97.4**	**96.4**	**97.0**	**97.4**	96.6	97.2	97.5	−5.2	12.3	13.0	**98.6**	**98.8**	**98.9**	**98.6**	**98.8**	**98.9**
		std	3.4	2.8	2.5	9.7	8.1	7.1	3.0	2.5	2.2	3.0	2.5	2.2	3.0	2.5	2.2	7.2	6.0	9.0	2.7	2.2	2.0	2.7	2.2	2.0

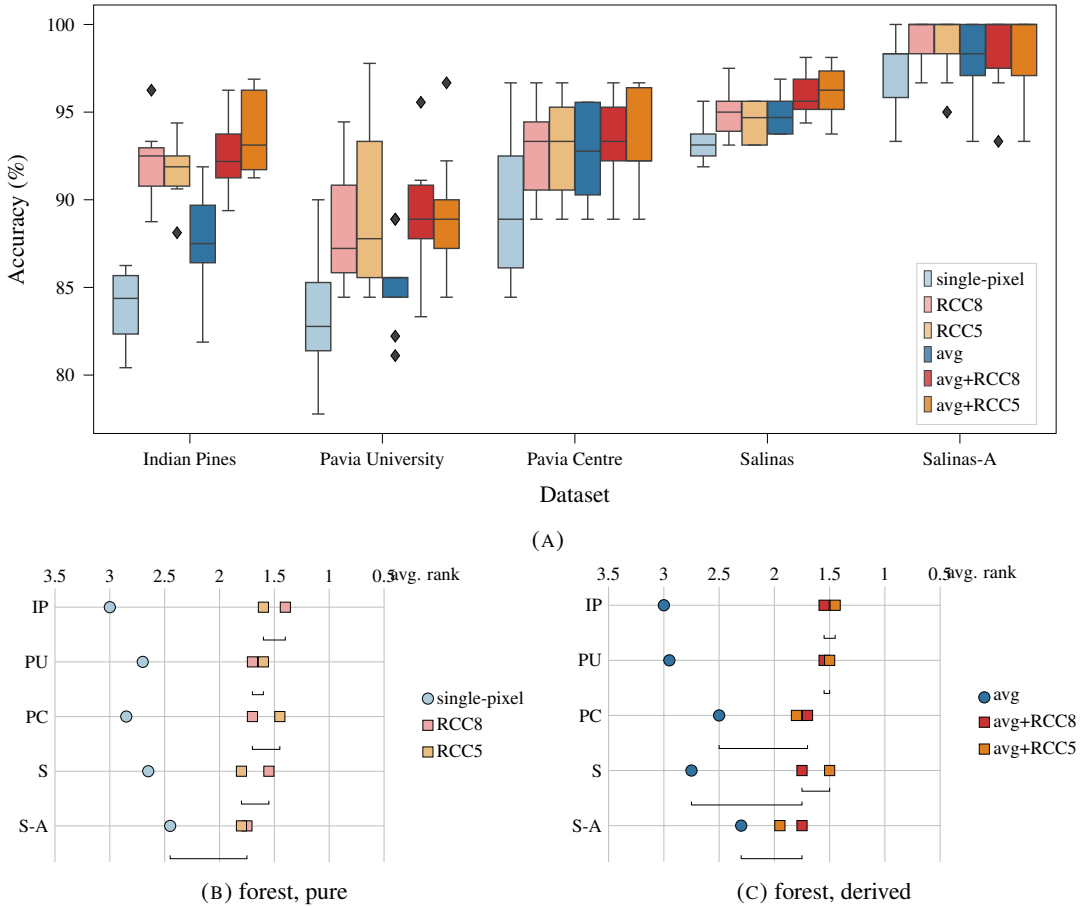


FIGURE 6.1: Hypothesis test results on the average accuracies of the eight approaches and five datasets. At the top (a), the distributions of the average accuracies for the random forest models are shown via a boxplot, reporting a five-number summary (minimum, first quartile, median, third quartile, maximum) and the outliers (i.e., observations farther away than 1.5 the interquartile range). At the bottom, Critical Difference diagrams comparing the pure (b) and derived random forest (c) approaches.

RCC5 in fourteen out of twenty cases. As expected, random forest models always achieve higher performances when compared with their single-tree counterpart, with the only two exceptions represented by the derived modal approach; apart from these isolated cases, which are likely due to fluctuations, the improvement ranges from 12 percentage points (Indian Pines, pure single-pixel) to 1 percentage point (Salinas-A, derived single-pixel). Fig. 6.1a graphically shows the distribution of the average accuracies for the random forest models. As it appears, the modal approaches tend to achieve accuracies of ~90–96% for Indian Pines and Pavia Centre, ~85–95% for Pavia University, ~93–98% for Salinas, and ~96–100% for Salinas-A. Although a proper comparison with the existing literature on the same datasets is hampered by the differences in the experimental setting, we point out that the results for Indian Pines, Pavia University, and Salinas

TABLE 6.3: Cross-validation results on five datasets using different approaches based on CART and ModalCART (highlighted in bold). For each approach, the average across 10 repetitions of the following metrics is reported: κ coefficient (percentage points), number of leaves/rules per tree r_τ, and training time t (measured in seconds and divided by the number of threads). The average value for each performance measure is also shown, referred to random forests, decision trees, and both models (ov. avg). For each result line, the average performance of the pure approach and the derived approach with the smallest number of rules is highlighted.

	Dataset	Pure approaches												Derived approaches											
		single-pixel			flattened			**RCC8**			**RCC5**			avg			avg+flattened			**avg+RCC8**			**avg+RCC5**		
		κ	r_τ	t	κ	r_τ	t	κ	r_τ	t	κ	r_τ	t	κ	r_τ	t	κ	r_τ	t	κ	r_τ	t	κ	r_τ	t
Random Forest	IP	82.9	56.6	39	0.5	153.5	353	**91.5**	**40.2**	**874**	91.2	40.4	351	**86.8**	**50.6**	**16**	−1.9	153.3	140	92.0	73.3	869	94.0	78.2	580
	PU	81.2	54.4	7	−3.5	149.1	59	**87.2**	**37.6**	**215**	87.9	38.1	144	83.4	52.9	7	−4.5	159.3	64	**87.8**	**40.2**	**207**	87.9	40.6	128
	PC	88.1	35.9	10	−5.5	147.5	124	**92.0**	**23.8**	**312**	**92.2**	**23.8**	**188**	91.8	29.4	9	−6.8	155.7	111	**92.6**	**24.8**	**188**	92.6	25.1	114
	S	92.9	46.4	28	−4.1	334.8	385	94.7	33.7	2547	**94.1**	**33.6**	**1342**	94.5	38.9	29	−3.9	338.5	333	95.8	31.5	855	**95.9**	**31.4**	**507**
	S-A	97.0	9.0	3	−4.2	82.8	46	**98.8**	**7.5**	**128**	**98.6**	**7.5**	**79**	97.6	8.6	4	−5.4	81.7	59	98.4	6.6	145	**98.0**	**6.5**	**90**
	avg	88.4	40.5	17	-3.4	173.5	193	**92.8**	**28.6**	**815**	92.8	28.7	421	90.8	36.1	13	-4.5	177.7	141	**93.3**	**35.3**	**453**	93.7	36.4	284
Decision Tree	IP	70.9	66.0	3	0.2	255.0	34	**78.3**	**50.1**	**123**	79.4	51.0	82	76.7	56.9	4	0.6	247.7	47	**80.5**	**46.1**	**99**	81.3	47.0	99
	PU	71.6	23.1	1	−0.9	109.2	10	**77.5**	**22.0**	**24**	77.6	22.1	19	73.9	34.0	1	−1.8	117.3	12	**76.9**	**23.8**	**19**	77.5	24.6	15
	PC	84.9	18.0	2	−1.8	117.4	26	**89.1**	**13.5**	**46**	**89.4**	**13.5**	**29**	89.8	15.4	2	−2.6	129.5	21	88.9	13.8	15	**88.8**	**13.7**	**11**
	S	88.5	22.1	4	0.2	264.1	47	91.2	23.2	296	**90.8**	**21.7**	**241**	**92.5**	**19.4**	**5**	−1.4	248.4	66	93.2	22.7	70	93.3	23.3	59
	S-A	95.4	7.9	0	−3.6	85.5	7	**96.4**	**6.8**	**22**	**96.4**	**6.8**	**20**	96.6	6.9	1	−5.2	78.7	13	**98.6**	**6.0**	**22**	**98.6**	**6.0**	**19**
	avg	82.3	27.4	2	-1.2	166.2	25	86.5	23.1	102	86.7	23.0	78	85.9	26.5	3	-2.1	164.3	32	**87.6**	**22.5**	**45**	87.9	22.9	41
	ov. avg	85.3	33.9	10	-2.3	169.9	109	**89.7**	**25.8**	**459**	**89.8**	**25.8**	**250**	88.4	31.3	8	-3.3	171.0	87	**90.5**	**28.9**	**249**	90.8	29.6	162

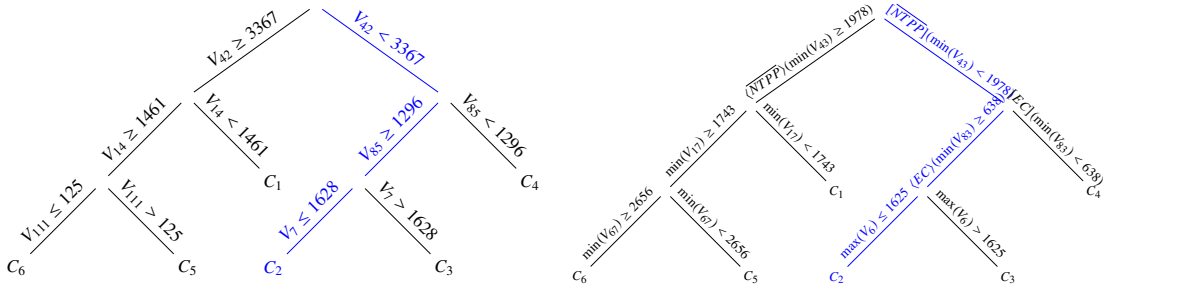
are consistent with those from the neural literature, and are also consistently better than those attained by other works using random forest models (refer to the *Data* section above).

In order to assess whether the performances of modal approaches are significantly better than propositional ones, the validation accuracies throughout the 10 repetitions can be interpreted as statistical populations and subjected to statistical hypothesis tests. Each accuracy population, identified by dataset and approach, is represented via boxplot in Fig. 6.1 (top), which provides an immediate graphical overview of the performances. In this context, we rely on the Holm-adjusted Wilcoxon signed-rank test, as explained in [BCM16], and report the results in form of *Critical Difference diagrams* [Dem06], which is a standard methodology for such comparisons. Fig. 6.1 (bottom) shows two sets of Critical Difference diagrams comparing approaches based on pure random forests, and derived random forests, respectively Each diagram refers to a single dataset, and shows the average rank attained with each approach throughout the 10 repetitions (smaller is better); underneath each diagram, bars are shown, that connect and group approaches that are statistically indistinguishable. The two sets of diagrams reveal virtually the same trends, and a few observations can be made. In all cases, modal approaches have better ranks than single-pixel ones, and are always in the group of significance with the lowest rank. Furthermore, for Indian Pines and Pavia University, both modal approaches show a significant performance improvement over the single-pixel approach, both in the pure and the derived case; across all datasets, the improvement is considered significant in seven out of ten cases.

§6.5 Model Complexity

With the aim of analyzing the structural and computational complexities of the models, and confronting them with the statistical performance, we report in Tab. 6.3 some additional metrics for each case, including the number of rules per decision tree trained and the average training time of the model. The five datasets can be ordered by complexity, by considering the number of rules required for classifying each one of them. Ignoring the flattened approaches, which are less competitive in terms of performance, Indian Pines encloses the task which causes the trained trees to be the largest in eleven out of twelve cases (that is, except for forest avg); in nine out of the remaining eleven cases (that is, except for single-tree RCC8 and forest avg+RCC8), Pavia University, Salinas, Pavia Centre and Salinas-A follow, in this order. With the exceptions of derived single-tree approaches on Salinas and derived random forests on Indian Pines, HS-trees are, on average, smaller than their (non-flattened) propositional counterparts. As for the two exceptions, the single-tree avg approach on Salinas yields an average number of leaves equal to 19.4, which is not too smaller than that of avg+RCC8 (22.7) and avg+RCC5 (23.3), while forest avg on Indian Pines yields substantially smaller trees (50.6 leaves) compared with the modal counterparts (73.3 and 78.2 leaves), but also has a lower κ (86.8%, compared with 92.0% and 94.0%); a possible cause could be the pruning condition causing the algorithm to stop the learning prematurely. When comparing pure and derived approaches, the avg approach always yields smaller trees than the propositional approach, with a gap ranging from 0.4 to 10.9 (single tree on S-A and single tree on PU, respectively); as for the modal approaches, the same pattern only holds nine out of twelve cases. When comparing random forests and decision trees, only in four out of fourty cases (single-pixel, RCC8, RCC5, and avg, all on Indian Pines) the average size of the trees in forest models is less of that of the respective single decision tree model; this is expected, given that forest models do not involve pruning conditions. Fixed a dateset, in general, with larger values for κ, the number of rules seems to decrease; this rule of thumb does not hold in all cases in the table, but in those cases where it does not, the differences are of a few decimals (e.g., pure RCC5 vs. pure RCC8 forests on Salinas). A similar trend also emerges in the fact that, with only three exceptions out of twenty (derived forest on Indian Pines, derived single tree on Pavia Centre, derived single tree on Salinas), for any propositional (pure or derived) approach, there always exists a corresponding modal approach (pure or derived) that yields trees that are both smaller and more accurate.

With respect to the training times, it can be observed how the modal approaches require much more time than propositional ones. With respect to propositional approaches, on average across all datasets, learning a tree with RCC8 requires 31 (derived case) to 46 (pure case) times more time, and RCC5 requires 20 (derived case) to 25 (pure case) times more time. Considering that the performances and structural complexities of RCC8 and RCC5 trees are similar, it can be argued that RCC5 provides an optimal trade-off between expressiveness and computational complexity. It can also be observed that training forests of 100 trees always takes less time than a hundred times the time of training a corresponding single tree. This is could be due to the (previously observed) fact that, despite the forests having no pruning condition, in most cases, bagged trees are on average smaller than single trees trained on the same data. However, it is likely that the shared memoization policy has a much stronger impact, with regard to this point.



$V_{42} < 3367 \wedge V_{85} \geq 1296 \wedge V_7 \leq 1628 \rightarrowtail C_2$

$[\overline{NTPP}](\min(V_{43}) < 1978) \wedge \langle EC \rangle(\min(V_{83}) \geq 638 \wedge \max(V_6) \leq 1625) \rightarrowtail C_2$

FIGURE 6.2: Comparison between a propositional tree (left) and a HS-tree (right) extracted on *Salinas-A*. The highlighted paths show the rules that each model has extracted for the class C_2 (i.e., *Corn senesced green weeds*).

§6.6 Post-hoc Interpretation

As mentioned, the statistical performances of a symbolic method is a limited metric for evaluating its usefulness; given the interpretable nature of decision trees, on top of the above analysis, trained models can be inspected to gain insights into the extracted knowledge. A natural way to carry out this kind of analysis is to choose a performant model, analyze its structure, its classification rules and relating their accuracies to the performance of the model. Within our experimental context, we choose a propositional baseline tree, and a HS-tree trained on the same dataset, opting for pure approaches, which are more easily intepretable.

We begin this exploration by analyzing, for the propositional and the modal case, a pair of single trees trained on the same training-test split of Salinas-A, where the classification task is to distinguish six different types of vegetables. Fig. 6.2 (top) shows two trees extracted with the single-pixel, and with the RCC8 approach, respectively. As for their statistical performances, the single-pixel tree displays a validation κ of 92%; however, we found that the majority of the misclassifications, in this case, occur between two specific classes: 25% of the test samples belonging to class C_2 (*corn senesced green weeds*) are misclassified by the model as belonging to class C_6 (*lettuce romaine, 7 weeks*). The HS-tree, on the other hand, has an 100% accuracy (and κ), and, with respect to the single-pixel model, it increases the recall of class C_2 from 75% to 100%, while also reaching optimal performance for the other five classes. These trees have the same topology, and only differ by the split-decisions. The structural similarity allows for an additional comparison of the two trees, which reveals that eight out of ten split-decisions involve variables with indices that are within three units apart (e.g., in the two cases, the decisions at the root node depend on variables 42 and 43, respectively). Considering that hyperspectral channels with similar wavelengths are correlated, and given that Salinas-A has 200 variables, this suggests that the natural patterns they are capturing for each class are quite similar. Also, both trees have exactly one leaf (and, thus, one classification rule) for each of the six classes C_i; therefore, for each class, they encompass a single logical condition $\varphi_{C_i} = \varphi_{\ell_i}$ that is (in a statistical sense) both necessary and sufficient for C_i. The two conditions that emerge for class C_2 are shown in Fig. 6.2 (bottom) in their simplified form. At a glance, the formulas have overlapping semantics, different

TABLE 6.4: Confusion matrices on Indian Pines for pure single-pixel approach (top) and pure RCC5 approach (bottom). Each line shows the confusion of samples belonging to a single class, as well as the number of samples and the recall attained for that class. For the two models, precision and overall accuracy values are also shown. For each class, the highest recall and precision across the two models is highlighted, as well as the highest overall accuracy.

	single-pixel																	
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	**tot.**	**rec (%)**
C_1	0	0	0	0	0	0	0	45	0	0	1	0	0	0	0	0	46	0.0
C_2	0	484	41	32	0	15	0	1	0	0	828	27	0	0	0	0	1428	33.9
C_3	0	40	74	12	0	0	0	0	0	0	607	19	1	0	0	0	753	9.8
C_4	0	34	5	95	0	11	0	1	0	0	72	15	0	0	4	0	237	40.1
C_5	0	6	0	3	248	131	0	13	0	0	3	1	0	47	1	0	453	54.8
C_6	0	0	0	0	3	718	0	0	0	0	3	0	0	4	2	0	730	**98.4**
C_7	0	0	0	0	0	0	0	28	0	0	0	0	0	0	0	0	28	0.0
C_8	0	24	0	0	0	2	0	448	0	0	1	0	0	0	3	0	478	93.7
C_9	0	0	0	0	0	16	0	0	0	0	0	0	0	0	4	0	20	0.0
C_{10}	0	1	3	2	0	17	0	2	0	0	935	2	0	0	0	0	962	0.0
C_{11}	0	71	53	2	0	39	0	3	0	0	2216	6	0	0	2	0	2392	92.6
C_{12}	0	27	40	31	0	9	0	0	0	0	379	98	6	0	0	0	590	16.6
C_{13}	0	0	0	0	0	18	0	0	0	0	0	0	178	0	9	0	205	**86.8**
C_{14}	0	0	0	0	45	12	0	0	0	0	0	0	1	1187	20	0	1265	93.8
C_{15}	0	0	0	9	8	157	0	6	0	0	1	0	26	62	45	0	314	14.3
C_{16}	0	8	0	1	0	11	0	0	0	0	71	2	0	0	0	0	93	0.0
prec (%)	–	69.6	34.3	50.8	81.6	62.1	–	**81.9**	–	–	43.3	57.6	**84.0**	91.3	50.0	–	**acc** = 58.0	
# rules	0	5	3	4	2	1	0	1	0	0	3	3	1	3	2	0	$\mathbf{r_\emptyset}$ = 28	

(A)

	RCC5																	
	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	**tot.**	**rec (%)**
C_1	0	0	0	0	0	0	0	39	0	0	7	0	0	0	0	0	46	0.0
C_2	0	620	12	0	0	0	0	0	1	0	781	14	0	0	0	0	1428	**43.4**
C_3	0	14	138	23	0	0	0	0	0	0	557	21	0	0	0	0	753	**18.3**
C_4	0	54	9	97	0	1	0	0	5	0	43	15	13	0	0	0	237	**40.9**
C_5	0	3	0	4	381	14	0	7	11	0	17	0	2	13	1	0	453	**84.1**
C_6	0	0	0	0	5	711	0	9	0	0	1	0	0	4	0	0	730	97.4
C_7	0	0	0	0	0	0	0	27	0	0	0	0	0	0	1	0	28	0.0
C_8	0	0	0	0	0	0	0	476	0	0	0	0	0	0	2	0	478	**99.6**
C_9	0	0	0	0	2	8	0	0	7	0	3	0	0	0	0	0	20	**35.0**
C_{10}	0	5	4	0	0	0	0	4	0	0	949	0	0	0	0	0	962	0.0
C_{11}	0	18	25	4	0	0	0	3	9	0	2322	11	0	0	0	0	2392	**97.1**
C_{12}	0	37	15	7	0	0	0	2	4	0	319	193	8	0	5	0	590	**32.7**
C_{13}	0	0	0	0	0	0	0	0	6	0	13	0	174	0	12	0	205	84.9
C_{14}	0	0	0	0	17	17	0	0	0	0	0	0	0	1206	25	0	1265	**95.3**
C_{15}	0	0	0	0	19	68	0	1	0	0	0	0	9	63	154	0	314	**49.0**
C_{16}	0	17	1	0	0	0	0	21	3	0	49	0	2	0	0	0	93	0.0
prec (%)	–	**80.7**	**67.7**	**71.8**	**89.9**	**86.8**	–	80.8	15.2	–	**45.9**	**76.0**	83.6	**93.8**	**77.0**	–	**acc = 64.8**	
# rules	0	4	4	2	1	2	0	1	1	0	4	3	1	1	1	0	$\mathbf{r_\emptyset}$ = 25	

(B)

structure and use slightly different variables. The single-pixel, propositional condition plainly translates to *within the pixel, the value for variable* 42 *is less than 3367, the value for variable* 85 *is at least 1296, and the value for variable*7 *is at most 1628.* The modal one (in this case, a HS^2_{RCC8}-formula) describes a richer spatial scenario, which can be translated in natural language as follows: *within* 3×3 *window surrounding the pixel, the minimum value of variable* 43 *is less than* 1978*, and there exists at least one rectangle of neighboring pixels in which all of the pixels*

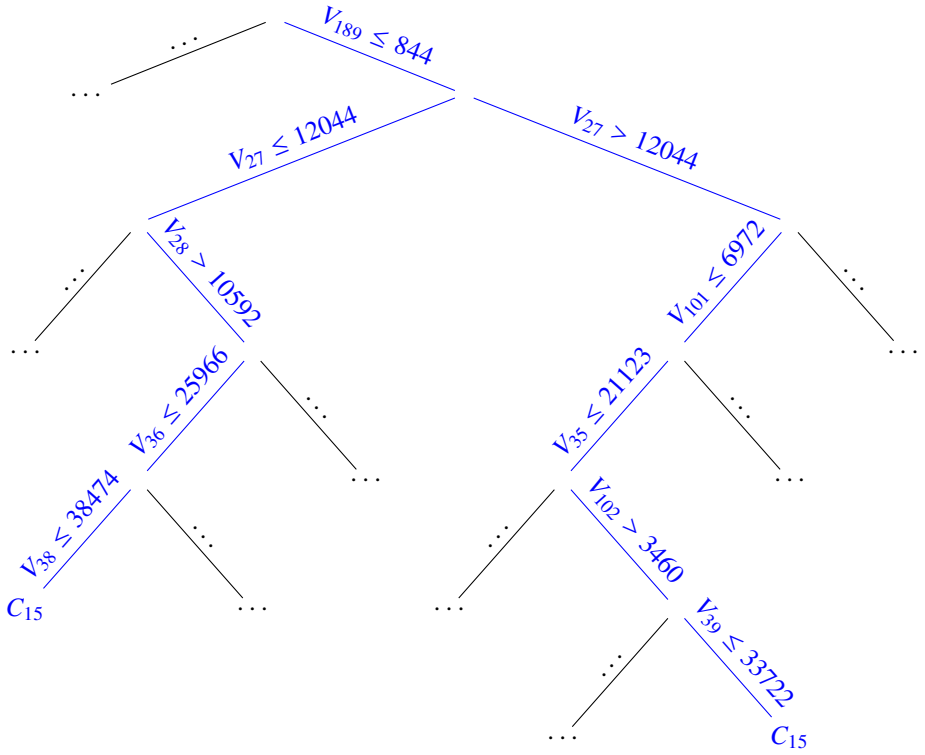


FIGURE 6.3: Propositional tree trained via the pure single-pixel approach on Indian Pines. While the tree comprehends 28 leaves, only the two paths for class C_{15} are shown.

have the value for variable 83 *greater than or equal to* 638 *and the value for variable* 6 *less than or equal to* 1625. In this case, the misclassifications of C_2 samples by the single-pixel tree can be attributed (at least) to the first single-pixel split; this split is performed on the characteristics of the pixel itself, while the class C_2, seemingly, requires spatial considerations on the neighboring pixels, to be correctly distinguished from C_6.

With the aim of both evaluating the generalization ability of these methods and further deepening their practical usefulness, as a last set of experiments, we perform a variation of this study, but considering, this time, the full extent of one of the datasets. We carry out this study on Indian Pines, which encloses the hardest of the five tasks, and, since for this dataset, RCC5 is, on average, more performant than RCC8, we use this approach for obtaining the HS-tree. While, so far, training the trees on Indian Pines involved a subset of 1600 labelled samples, the full set include 10249 samples; thus, a single-pixel tree and an RCC5 HS-tree are trained on a 20% slice of the whole dataset, which encompasses 1993 samples, with a class imbalance that approximates the original class distribution. Since the RCC5 model requires a 3×3 surrounding neighborhood, we ignore the labelled pixels that are on the edge of the scene, for which the neighborhood of pixels is not available; this leads to discarding 255 of the 10249 original labelled pixels, leading to a subdataset of 9994 samples. Starting, again, from the statistical performances, when tested

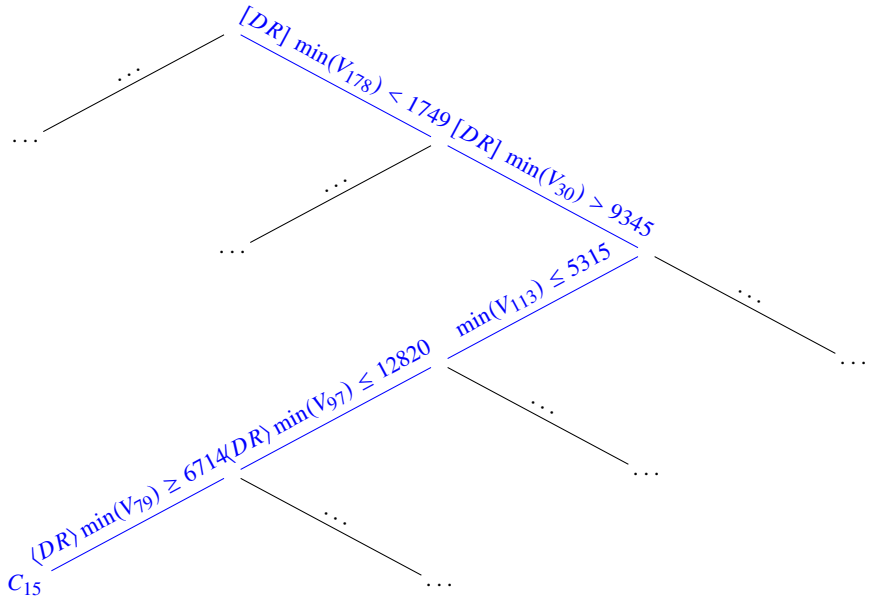


FIGURE 6.4: HS-tree trained via the pure RCC5 approach on Indian Pines. While the tree comprehends 25 leaves, the single path for class C_{15} is shown.

on the full set of samples, the two models achieved κ values equal to 50% and 58%, an overall accuracy of 58.0% and 64.8%, and an average precision of 64.2% and 72.4%, respectively. The relative improvement achieved by the RCC5 approach with respect to the single-pixel approach is still evident. However, with respect to the performances of the the previous set of experiments (Tab. 6.2), where the κ for the same models was 70.9% and 78.3%, respectively, a performance degradation affects both methods; this is probably due to the fact that the dataset is almost eight times larger (9994 samples, compared with the previous training size of 1280 samples), and the fraction of samples used for training is much smaller (20%, compared with 80%). Moreover, the training ad test sets are now both unbalanced, which ultimately makes these results uncomparable with the previous ones. Tab. 6.4 shows a detailed report of the performance of the two trees in terms of confusion matrices, per-class recalls and precisions. Across the whole dataset, the single-pixel tree scores a 0% recall on five classes (C_1, C_7, C_9, C_{10}, C_{16}); the HS-tree, for one of these classes (C_9) achieves a 35% recall, by correctly classifying 7 out of 20 instances, while it does not improve the recalls of the other four classes. As for the remaining classes, in ten out of twelve classes, the HS-tree achieves higher recall, with improvements that are sometimes as large as 34.7 (C_{15}) and 29.3 (C_5) percentage points; in the remaining two cases, the gap is smaller (less than 1.9 percentage points). Similar observations can be made by inspecting the per-class precisions.

Tab. 6.4 also reports the total and per-class number of rules. The total number of leaves/rules for the single-pixel and the RCC5 tree is 28 and 25, respectively, which is, again, in line with our considerations about the relation between statistical performance and structural complexity. For the two trees, it is the case that for those classes for which the tree scores a 0% recall (C_1, C_7,

(A) False color (B) Ground truth (C) single-pixel (50%) (D) RCC5 (58%)

FIGURE 6.5: Qualitative result comparison on Indian Pines: false color image (a), ground truth (b), and classification results obtained by the pure single-pixel approach (c) and the pure RCC5 approach (d). The two models scored a κ coefficient equal to 50% and 58%, respectively. A few regions where the RCC5 approach appears to improve over the single-pixel approach are highlighted.

C_{10}, C_{16} for both trees, and C_9 for the single-pixel case), the tree has exactly zero classification rules (and, therefore, an undefined precision). Additionally, for three classes (C_3, C_6, C_{11}) the single-pixel tree provides a smaller (non-zero) number of associated rules, and that for six classes (C_2, C_4, C_5, C_9, C_{14}, C_{15}), the HS-tree does. Focusing on C_{15}, the single-pixel tree and the RCC5 tree hove two rules and one rule for this class, respectively. By inspecting the trees we find that the two propositional, single-pixel rules for C_{15}, which corresponding paths are shown in Fig. 6.3, have a support, confidence and conviction of 0.3%, 53.6%, 1705% and 208.6%, and 0.6%, 48.4%, 1540% and 187.7%, respectively. The two rules are:

$$\begin{aligned} V_{189} &\leq 844 \wedge \\ V_{27} &\leq 12044 \wedge \\ V_{28} &> 10592 \wedge \\ V_{36} &\leq 25966 \wedge \\ V_{38} &\leq 38474 \rightarrowtail C_{15}, \end{aligned} \qquad \begin{aligned} V_{189} &\leq 844 \wedge \\ V_{27} &> 12044 \wedge \\ V_{101} &\leq 6972 \wedge \\ V_{35} &\leq 21123 \wedge \\ V_{102} &> 3460 \wedge \\ V_{39} &\leq 33722 \rightarrowtail C_{15}. \end{aligned}$$

On the other hand, the HS-tree has a single rule for the same class with a support of 2%, a confidence of 77.0%, a lift of 2451% and a conviction of 421.1%. The rule is:

$$\begin{aligned} [DR]\ \min(V_{178}) &< 1749 \wedge \\ [DR]\ \min(V_{30}) &> 9345 \wedge \\ \min(V_{113}) &\leq 5315 \wedge \\ \langle DR\rangle(\min(V_{97}) &\leq 12820 \wedge \quad \langle DR\rangle \min(V_{79}) \geq 6714) \quad \rightarrowtail C_{15}, \end{aligned}$$

and its corresponding path is shown in Fig. 6.4. The two propositional rules for C_{15} are responsible for the class precision and specificity of 50.0% and 14.3%, respectively, whereas the RCC5 rule for C_{15} displays a precision and a specificity of 77.0% (equal to its confidence) and 49.0%, respectively.

Finally, qualitative remarks on the behavior of the trained models can be made. Fig. 6.5

shows a visual depiction of the classification results achieved by the two trees on the full Indian Pines scene. The qualitative comparison shows that some of the areas in the image are classified with less confusion by the RCC5 HS-tree, with respect to the propositional, single-pixel one, while the opposite does not hold. From Fig. 6.5c and Fig. 6.5d, it can be observed that many misclassifications occur at the edges of the regions delimiting pixels of the same class. This could be due to several facts; for example, pixel mixing, which may hinder the predictive power of crisp scalar conditions.

CHAPTER 7

COVID-19 DIAGNOSIS FROM COUGH/BREATH AUDIO RECORDINGS

Do not draw your sword to kill a fly.

Korean proverb

As mentioned, interpretable time-series representations can be extracted from audio recordings. This section evaluates HS-based ModalCART and ModalCART-RF at the problem of recognizing COVID-19-positive and COVID-19-negative subjects from audio recordings of their cough and breath. Since a subject are simultaneously represented by a cough and a breath recording, the multimodal capabilities of these algorithms is leveraged [PSS21], allowing to use information from both samples in conjunction. In this case, the approach does not only outperform state-of-the-art models on the same dataset, but its results are also superior to those of most techniques applied on similar datasets. Ultimately, the extracted rules check temporal patterns in the powers of specific audio frequencies in the audio samples and, in principle, these rules can allow for a formal definition of *being positive (or negative)* to the disease. The presented results were published in Manzella et al. [Man+23a].

§7.1 Problem

COVID-19 is a respiratory disease caused by the *SARS-CoV2* virus. The disease was classified in 2019, and caused a pandemic that occurred during the years 2020–2023. The current scientific literature on COVID-19 is immense, and it ranges everywhere from medicine to economy, sociology, psychology, among many other fields. AI is no exception: as of April 2024, a simple Scopus search for `COVID-19 AND Artificial Intelligence` returns about 9500 entries. Perhaps one of the most appealing lines of research, in this regard, deals with the possibility of deriving computational models for the diagnosis of COVID-19 from respiratory sounds of human subjects, an idea already largely explored for diagnosing other respiratory diseases such as bronchitis or pneumonia. Diagnosis is usually enabled via coughs, breaths and oral speech audio samples, which can be easily recorded with smartphone hardware. This has a double advantage: it eases the availability of data, which can be quickly crowdsourced throughout smartphone applications, and it facilitates even more the actual diagnosis, which can be performed in real-time by the same applications. Together, the whole process provides

	Work	c	b	s	Model type	Dataset	Setting	Best
2020	Brown et al. [Bro+20]	✓	✓		LR, SVM	CAM	10-fold CV	AUC: 88
	Imran et al. [Imr+20]		✓		CNN	own	5-fold CV	acc: 92
	Hassan, Shahin, and Alsabek [HSA20]	✓	✓	✓	LSTM	own	train/test	acc: 98
	Laguarta, Hueto, and Subirana [LHS20]	✓			CNN	own	train/test	AUC: 97
	Chaudhari et al. [Cha+20]	✓		✓	CNN	V, CO, CW	train/test	AUC: 77
	Bansal, Pahwa, and Kannan [BPK20]	✓			CNN	own	train/test	AUC: 71
2021	Melek [Mel21]	✓			k-NN	V, NO	LOOCV	acc: 98
	Xia et al. [Xia+21a]	✓	✓	✓	CNN	own	10-fold CV	AUC: 74
	Pahar et al. [Pah+21]	✓			CNN, LSTM	own, CW	5-fold CV	acc: 95
	Despotovic et al. [Des+21]	✓	✓	✓	CNN, RF	own	5-fold CV	acc: 89
	Dash et al. [Das+21]	✓	✓	✓	SVM	CW, CAM	5-fold CV	acc: 85
	Stasak et al. [Sta+21]			✓	DT	own	5-fold CV	acc: 80
	Han et al. [Han+21]			✓	SVM	C19S	5-fold CV	acc: 79
	Muguli et al. [Mug+21]	✓	✓	✓	LR, RF	DI	5-fold CV	AUC: 75
	Coppock et al. [Cop+21]	✓	✓		CNN	CAM	3-fold CV	AUC: 91
	Fakhry et al. [Fak+21]	✓			CNN	CO	holdout	AUC: 99
	Das, Madhavi, and Li [DML21]	✓			LR, RF	DI	holdout	AUC: 81
	Xia et al. [Xia+21b]	✓	✓	✓	SVM, CNN	own	holdout	AUC: 75
	Casanova et al. [Cas+21]	✓		✓	CNN	CCS, CSS	holdout	bac: 76
	Schuller et al. [Sch+21]	✓		✓	SVM	own	holdout	bac: 74
	Deshpande and Schuller [DS21]	✓			LSTM	DI	train/test	AUC: 64
2022	Alkhodari et al. [AK22]		✓		CNN, LSTM	CW	LOOCV	acc: 95
	Dentamaro et al. [Den+22]	✓	✓		CNN	CAM, CW	10-fold CV	acc: 93
	Tena, Clarià, and Solsona [TCS22]	✓			RF	own	10-fold CV	acc: 90
	Chang et al. [Cha+21]	✓			CNN	CO	5-fold CV	acc: 72
	Nassif et al. [Nas+22]	✓	✓	✓	CNN, LSTM	own	holdout	acc: 98
	Aly, Rahouma, and Ramzy [ARR22]	✓	✓	✓	NN	CW	holdout	acc: 96
	Han et al. [Han+22]	✓	✓	✓	CNN	C19S	holdout	AUC: 71

TABLE 7.1: Overview of the existing work on COVID-19 diagnosis from audio recordings of *cough* (c), *breath* (b), or *speech* (s). Only works framing the problem as a binary classification one are listed, and for each work the type of model deployed, the dataset, the model evaluation setting, and an indication of the best performance attained is shown. Abbreviations were used for referring to datasets, CAM for *Cambridge* [Bro+20], V for *Virufy* [Cha+20], CO for *COUGHVID* [OTA21], CW for *Coswara* [Sha+20], NO for *NoCoCoDa* [CGK20], C19S for *COVID-19 Sounds* [Xia+21b], DI for *DiCOVA* [Mug+21], CCS and CSS for *ComParE2021 CCS* and *CSS* [Sch+21]. As for evaluation setting, CV stands for *cross-validation*, while LOOCV for *leave one out cross-validation*. As for performance evaluation, AUC stands for *area under the ROC curve*, acc for *accuracy*, and bac for *balanced accuracy*.

a compelling, zero-cost, non-invasive alternative to the widely used diagnostic (e.g., antigen or molecular) tests.

Tab. 7.1 lists relevant works published in 2020–2022 that attempts at the task of COVID-19 diagnosis via audio samples of cough, breaths, and/or speech, in various combinations. While different models are not easily comparable, not even in terms of statistical performances, this review reveals a unmistakable trend toward using functional methods for this task (with the exception of Stasak et al. [Sta+21], where decision tree models are used), and, in fact, the concepts of transparency and interpretability of the models are not even mentioned in any of

these contributions. This is in sharp contrast with the need for understanding the reasons that underly the decisions taken by intelligent systems, and in particular those related to medical diagnoses; such a need is widely shared in the community, and witnessed, for example, by a recent call for transparency of COVID-19 models appeared on Science [Sil+20]. The challenge arises, therefore, of devising a symbolic model for the diagnosis of COVID-19 based on the acoustic characteristics of a cough/breath sample, whose performances are at least comparable with those of non-symbolic ones.

§7.2 Data

The dataset in use, presented in Brown et al. [Bro+20], was originally crowdsourced and compiled by researchers at the University of Cambridge, and it is available upon request; in Tab. 7.1 it is referred to as CAM. 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. Beside recording sound samples, subjects were asked to fill in a very brief clinical history, plus information about their geographical location. In Brown et al. [Bro+20], this data was originally used 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 and 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* and 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* and 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. 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, here, as TA2+ and TA3+, respectively), obtained with

standard audio augmentation methods. In Brown et al. [Bro+20], the authors considered three versions of each task, which differ by how subjects are represented, that is, using only their cough sample, only their breath sample, or both; this results in two unimodal classification settings (with $q = 1$), and a multimodal classification one with $q = 2$ modalities, one for the cough sample, and one for the breath sample, respectively. Tab. 7.2 reports the state-of-the-art performances on the dataset in this form.

With respect to the original work, we eliminated 14 instances that presented cough/breath labeling mistakes, empty audio recording, and/or a too noisy background; barring such a difference, it is possible to directly compare the presented results with the ones from the original paper and from other models trained on the same data. Moreover, because of the nature of modal decision trees, it also makes sense to explore learning from *single* coughs/breath cycles, instead of whole recordings which present several episodes; for each version of the dataset, we produced a *segmented* variant containing single episodes from the original ones.

§7.3 Data Preprocessing

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 in Davis and Mermelstein [DM80], is still the preferred technique for extracting sensible spectral representations of audio data, and its use in machine learning has been fruitful for tackling difficult 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;

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 (iv) does not preserve the spectral component, which makes this description not directly interpretable in terms of sound

	acc	*prec*	*rec*	*F*1	*AUC*	*acc*	*prec*	*rec*	*F*1	*AUC*
	Brown et al. [Bro+20]					[Den+22]'s implementation of [Cop+21]				
TA1	-	72	69	-	80	60	54	59	54	56
TA2	-	**80**	**72**	-	**82**	-	-	-	-	-
TA3	-	**69**	**69**	-	**80**	-	-	-	-	-
TA2+	-	70	90	-	87	72	82	85	90	66
TA3+	-	61	81	-	88	86	82	86	83	88
	Coppock et al. [Cop+21]					[Den+22]'s implementation of [BPK20]				
TA1	-	-	-	-	**83**	72	71	72	71	68
TA2	-	-	-	-	-	-	-	-	-	-
TA3	-	-	-	-	-	-	-	-	-	-
TA2+	-	-	-	-	68	88	86	88	84	66
TA3+	-	-	-	-	91	90	90	90	88	76
	Dentamaro et al. [Den+22]					[Den+22]'s implementation of [Imr+20]				
TA1	**80**	**80**	**80**	80	**83**	71	71	71	68	78
TA2	-	-	-	-	-	-	-	-	-	-
TA3	-	-	-	-	-	-	-	-	-	-
TA2+	**93**	**89**	**93**	90	**93**	85	82	85	90	88
TA3+	**93**	**89**	**93**	90	**93**	85	78	85	81	87

TABLE 7.2: State-of-the-art performances on the CAM [Bro+20] dataset.

frequencies; as such, in this work MFCC is only applied up to step (*iii*), ultimately producing a multivariate time-series representation where the n variables describe the power (in decibel) of the different sound frequencies. In order 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 by means of 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 [KSV96; SRM15]. On top of this, different techniques were used to clean and normalize the data prior to the MFCC step:

- a *noise gate* filter for removing signals below a given power threshold;
- a *peak normalization* filter where the amplitude is scaled on the highest signal level present in the recording granting consistent amplitude between audio tracks;
- silence removal filter to remove unwanted long silences.

§7.4 Experimental Setting

For each of the 30 datasets described above, a number of modal decision trees and random forests were trained via ModalCART and ModalCART-RF and evaluated via standard performance metrics for binary classification: overall accuracy (*acc*), precision (*prec*), precision (*rec*) and F1-score ($F1$). Similarly to Chapter 6, to minimize the bias, experiments were run in a randomized

10-folds cross-validation, and the average and standard deviation of the performance metrics were considered; at each fold, datasets were balanced by downsampling the majority class, and randomly split into two (balanced) sets for training (80%) and test (20%). Furthermore, for any training/test split, random forests were run 5 times with different initial random conditions, and their average performance was considered. 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. In all cases we let $\bowtie$ be 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 correspond 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 variables 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 samples with a sample rate lower than 16000Hz were discarded.

§7.5 Statistical Performance

As far as the suitability of our method is concerned, let us focus on Tab. 7.3 and Tab. 7.4. 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, modal decision trees and random forests can focus on the relevant acoustic aspects of positive versus negative samples with a single episode at the time; second, segmented datasets include more instances than non-segmented ones, which allowed us to train better models. We have therefore followed two different policies to highlight the results in Tab. 7.3 and Tab. 7.4: 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 having accuracy higher than 95%. A second, immediate observation is that the multimodal setting performs decidedly better than standard, single-audio learning; 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 in Brown et al. [Bro+20]. As a third consideration, we notice, as expected, that modal 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 [Bro+20], augmented datasets give rise to better models in

			acc	$prec$	rec	$F1$	m_{train}	m_{test}
ModalCART	cough	TA1	67.8 ± 8	68.9 ± 9	66.0 ± 9	67.2 ± 8	202	50
		TA2	79.0 ± 14	81.4 ± 18	80.0 ± 13	79.6 ± 12	40	10
		TA3	60.0 ± 21	65.0 ± 25	70.0 ± 26	63.7 ± 17	14	4
		TA2+	83.3 ± 6	82.1 ± 8	86.7 ± 11	83.7 ± 6	74	18
		TA3+	**90.6 ± 6**	96.4 ± 6	84.5 ± 11	89.7 ± 7	72	18
	breath	TA1	68.8 ± 6	70.7 ± 7	65.2 ± 10	67.4 ± 7	202	50
		TA2	82.0 ± 11	87.8 ± 14	76.0 ± 16	80.5 ± 12	36	10
		TA3	55.0 ± 16	53.7 ± 26	60.0 ± 39	51.0 ± 29	12	4
		TA2+	82.8 ± 9	86.1 ± 16	83.3 ± 12	83.2 ± 8	74	18
		TA3+	**85.0 ± 9**	93.5 ± 9	75.0 ± 16	82.6 ± 12	64	16
	cough+breath	TA1	66.4 ± 7	67.4 ± 8	64.4 ± 6	65.8 ± 6	202	50
		TA2	77.0 ± 15	81.5 ± 19	74.0 ± 16	76.4 ± 14	36	10
		TA3	65.0 ± 13	70.0 ± 22	75.0 ± 26	67.3 ± 11	12	4
		TA2+	**85.0 ± 9**	86.1 ± 10	84.5 ± 9	85.0 ± 9	74	18
		TA3+	84.4 ± 4	91.7 ± 7	76.3 ± 9	82.8 ± 5	64	16
ModalCART-RF	cough	TA1	76.6 ± 7	79.4 ± 9	72.4 ± 8	75.6 ± 8	202	50
		TA2	83.4 ± 12	85.3 ± 14	83.6 ± 18	83.2 ± 13	40	10
		TA3	70.5 ± 19	79.3 ± 23	70.0 ± 35	66.5 ± 28	14	4
		TA2+	**88.7 ± 6**	91.9 ± 7	85.3 ± 10	88.1 ± 7	74	18
		TA3+	**89.2 ± 7**	96.3 ± 6	81.6 ± 11	87.9 ± 8	72	18
	breath	TA1	74.5 ± 6	76.3 ± 9	72.3 ± 7	74.0 ± 6	202	50
		TA2	84.0 ± 10	88.9 ± 12	80.0 ± 19	82.7 ± 11	36	10
		TA3	62.0 ± 21	63.0 ± 32	63.0 ± 33	59.7 ± 27	12	4
		TA2+	**87.9 ± 6**	91.9 ± 8	84.0 ± 11	87.2 ± 6	74	18
		TA3+	**94.5 ± 4**	98.9 ± 4	90.3 ± 9	94.0 ± 5	64	16
	cough+breath	TA1	74.8 ± 6	76.1 ± 8	73.0 ± 7	74.3 ± 6	202	50
		TA2	84.2 ± 10	87.3 ± 10	81.2 ± 18	83.0 ± 11	36	10
		TA3	69.5 ± 19	76.0 ± 26	69.0 ± 25	68.6 ± 18	12	4
		TA2+	**89.8 ± 5**	93.5 ± 5	85.8 ± 11	89.1 ± 6	74	18
		TA3+	**89.5 ± 7**	95.4 ± 6	83.3 ± 11	88.5 ± 8	64	16

TABLE 7.3: Results obtained using non-segmented COVID-19 audio datasets.

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, which effects are worsened by the downsampling step; 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, modal random forests do not improve the accuracy (best accuracy in this case: 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; this result is already unmatched in the current literature (see Tab. 7.2). With modal random

			acc	$prec$	rec	$F1$	m_{train}	m_{test}
ModalCART	cough	TA1	72.8 ± 8	75.0 ± 8	68.4 ± 9	71.4 ± 8	340	86
		TA2	91.0 ± 10	96.7 ± 11	86.0 ± 14	90.3 ± 10	42	10
		TA3	72.5 ± 25	79.6 ± 26	70.0 ± 35	68.3 ± 32	12	4
		TA2+	93.2 ± 8	92.3 ± 8	94.6 ± 12	93.1 ± 9	92	22
		TA3+	90.0 ± 5	92.4 ± 7	87.5 ± 8	89.7 ± 6	64	16
	breath	TA1	74.3 ± 3	74.6 ± 4	74.1 ± 3	74.3 ± 2	982	246
		TA2	84.0 ± 5	84.0 ± 6	84.7 ± 11	83.9 ± 6	120	30
		TA3	61.7 ± 19	61.0 ± 30	66.7 ± 35	59.9 ± 27	20	6
		TA2+	88.2 ± 4	91.3 ± 4	84.6 ± 9	87.5 ± 5	326	82
		TA3+	91.9 ± 7	98.4 ± 3	85.4 ± 13	90.8 ± 9	104	26
	cough+breath	TA1	**95.9 ± 1**	96.5 ± 2	95.3 ± 2	95.9 ± 2	1338	334
		TA2	**98.8 ± 2**	100.0 ± 0	97.7 ± 4	98.8 ± 2	102	26
		TA3	90.0 ± 17	92.6 ± 15	90.0 ± 32	86.0 ± 31	12	4
		TA2+	**97.8 ± 2**	98.5 ± 1	97.0 ± 4	97.7 ± 2	430	108
		TA3+	84.4 ± 18	87.1 ± 21	86.3 ± 15	85.6 ± 16	64	16
ModalCART-RF	cough	TA1	80.4 ± 6	84.0 ± 6	75.2 ± 8	79.2 ± 6	340	86
		TA2	92.4 ± 7	99.3 ± 2	85.6 ± 13	91.4 ± 8	42	10
		TA3	73.5 ± 23	78.0 ± 25	77.0 ± 24	74.6 ± 20	12	4
		TA2+	**95.5 ± 4**	98.5 ± 3	92.6 ± 9	95.1 ± 5	92	22
		TA3+	92.9 ± 7	100.0 ± 0	85.8 ± 15	91.5 ± 10	64	16
	breath	TA1	81.9 ± 2	84.0 ± 3	79.0 ± 3	81.4 ± 2	982	246
		TA2	86.7 ± 7	91.5 ± 7	82.3 ± 16	85.4 ± 9	120	30
		TA3	66.3 ± 12	68.1 ± 19	67.3 ± 31	63.7 ± 19	20	6
		TA2+	90.5 ± 3	95.7 ± 3	84.9 ± 6	89.9 ± 3	326	82
		TA3+	92.0 ± 8	99.9 ± 0	84.2 ± 16	90.5 ± 11	104	26
	cough+breath	TA1	**98.0 ± 0**	99.4 ± 1	96.7 ± 1	98.0 ± 0	1338	334
		TA2	**97.2 ± 3**	100.0 ± 0	94.5 ± 6	97.0 ± 3	102	26
		TA3	86.5 ± 16	93.7 ± 9	81.0 ± 32	81.5 ± 28	12	4
		TA2+	**99.4 ± 1**	99.0 ± 1	99.9 ± 0	99.4 ± 1	430	108
		TA3+	**96.1 ± 5**	100.0 ± 0	92.3 ± 11	95.6 ± 6	64	16

TABLE 7.4: Results obtained using segmented COVID-19 audio datasets.

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 (see Tab. 7.1); in the segmented, multimodal setting, however, modal 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 multimodal approach, as it can be seen, also has the effect of lowering the standard deviation across the different sets of experiments, at least in the segmented case. In order to evaluate which of the sample types (cough, breath, or cough+breath) is most suited for this kind of prediction, and whether the multimodal approach is statistically more effective than the unimodal one, the accuracy results

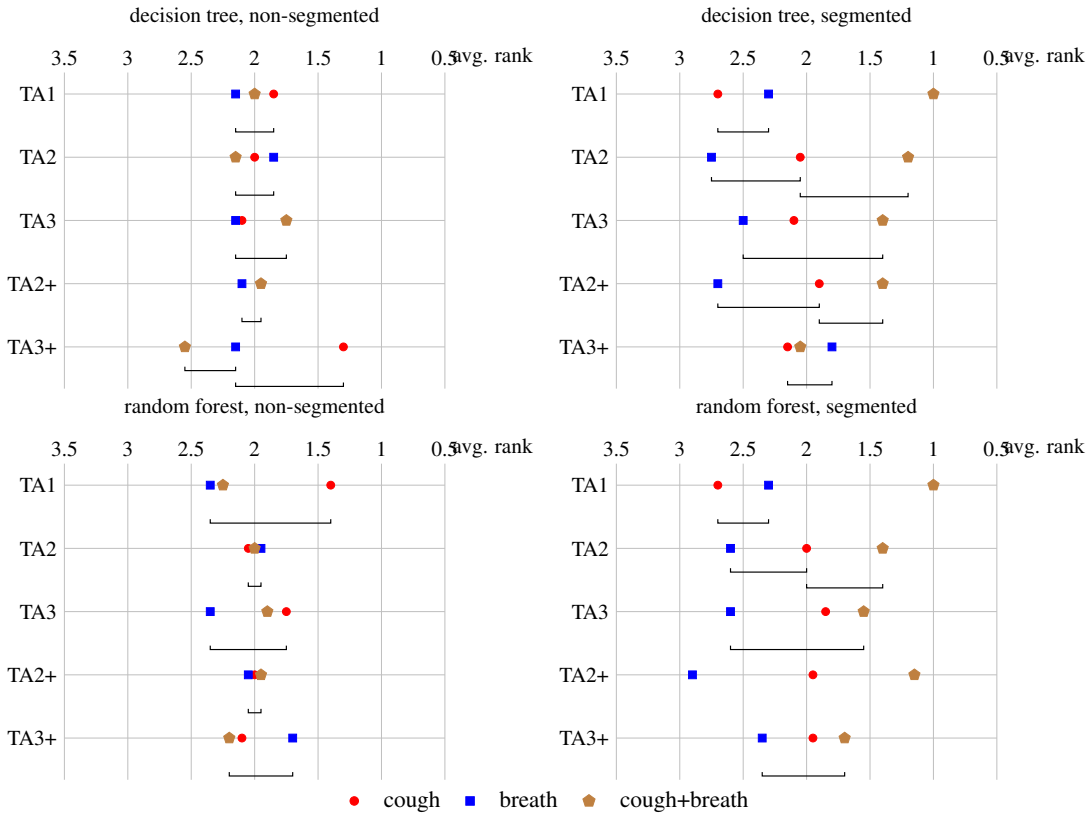


FIGURE 7.1: Critical difference diagrams comparing accuracies of the three sample types (cough, breath and cough+breath) for each task.

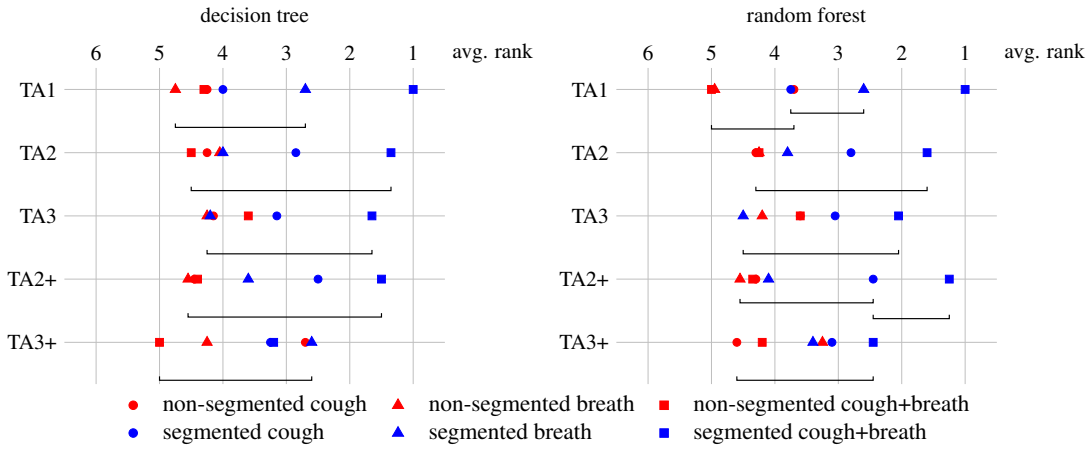


FIGURE 7.2: Critical difference diagrams comparing accuracies of the segmented and non-segmented versions for each task.

		non-segmented				segmented			
		train time (s)				train time (s)			
		ModalCART	ModalCART-RF	m_{train}	N	ModalCART	ModalCART-RF	m_{train}	N
cough	$TA1$	15.89	1,654.57	202	50	4.22	136.81	340	3
	$TA2$	1.05	53.34	40	37	0.18	2.06	42	3
	$TA3$	0.09	6.28	14	37	0.01	0.15	12	3
	$TA2+$	2.89	247.03	74	37	0.40	5.94	92	3
	$TA3+$	1.07	96.57	72	37	0.22	2.94	64	3
breath	$TA1$	110.30	5,184.86	202	50	122.47	8,271.27	982	44
	$TA2$	4.28	238.98	36	50	3.54	124.35	120	26
	$TA3$	0.10	80.99	12	50	0.23	5.03	20	26
	$TA2+$	10.91	1,079.55	74	50	13.50	947.30	326	26
	$TA3+$	15.51	598.27	64	50	2.78	101.68	104	26
cough+breath	$TA1$	56.07	5,355.72	202	50, 50	82.65	8,717.38	1338	3, 44
	$TA2$	1.09	170.95	36	37, 50	1.10	28.56	102	3, 26
	$TA3$	0.08	22.40	12	37, 50	0.01	0.67	12	3, 26
	$TA2+$	3.58	1,048.96	74	37, 50	14.08	905.60	430	3, 26
	$TA3+$	6.11	667.19	64	37, 50	0.14	21.10	64	3, 26

TABLE 7.5: Average computational time required to train COVID-19 models. For each experiment, the number of training instances and their length (i.e., number of points) is also shown. Note that, while the average time for decision tree models is computed over the 10 cross-validation repetitions, each random forest is computed 5 times with different random seeds, thus the average time is computed over a total of 50 experiments.

for the different versions can be compared using the *Friedman test* [Fri40]. In addition to this, a *Wilcoxon signed-rank test* [Wil92] can be performed on each pair of versions to check if there are significative differences between their results. The results of such tests, corrected using the *Holm–Bonferroni method* [Hol79], are shown in terms of *critical difference* [Dem06; BCM16] diagrams in Fig. 7.1. It is worth noting that in the segmented case, although the multimodal approach does not always show a critical difference with respect to the the single-sample cases, it shows a lower *average rank* in almost all tasks for the segmented dataset (that is, in all tasks except for TA3+ using the decision tree), indicating a better performing model. A similar trend is highlighted in Fig. 7.2, where the critical differences between decision trees and random forests across the two segmented and non-segmented versions are shown. The average time required to train the models for each task can be seen in Tab. 7.5. Note that unrelated experiments were simultaneously run on the same server, and the reported times may be affected by this.

§7.6 Post-hoc Interpretation

Similarly to Chapter 6, we now analyze the structure of the models themselves. Recall that we have extracted models from 30 variations of the original CAM dataset, with each extraction consisting of 10 models, which means that, in terms of single decision trees, we have a pool of 300 trained models. Among them, and in particular from those obtained for TA2+ in the segmented case (in which case we have reached, as already observed, an accuracy of 99.4% on average), we have selected a single tree with 100% test accuracy; it is shown in Fig. 7.3. As it can be seen, such a tree is in fact a very simple model, in which most of the examples fall in one

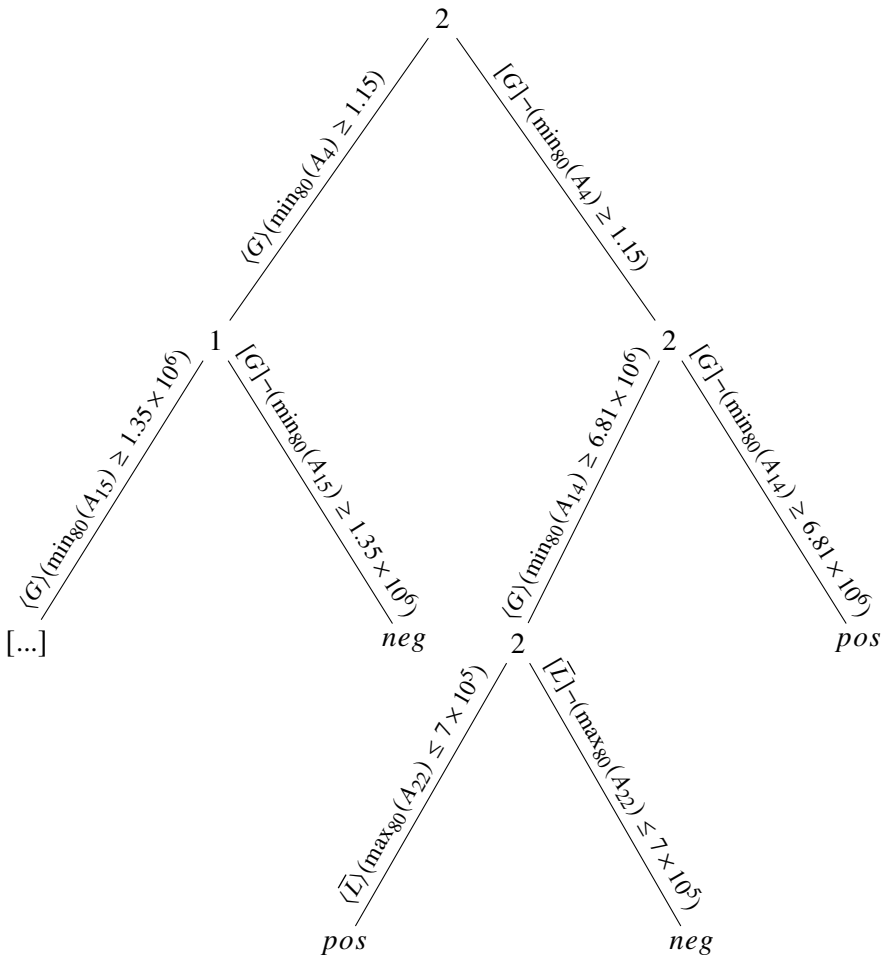


FIGURE 7.3: An HS-tree with 100% test accuracy for task TA2+ (segmented, cough+breath). Recall that the first modality represents the cough sample and the second one represents the breath sample; as such, the first split is performed according to a formula to be checked on the breath sample.

of two leaves. These induce two rules, one for positive cases (*pos*), with a support of 42%, and one negative ones (*neg*), with a support of 48%, as follows:

$$\begin{aligned}\mathfrak{r}_1: \quad & \{2\}\langle G\rangle(\min_{80}(V_4) \geq 1.15) \wedge \\ & \{1\}[G]\neg(\min_{80}(V_{15}) \geq 1.35 \times 10^6) \rightarrowtail \mathit{neg}\end{aligned}$$

$$\begin{aligned}\mathfrak{r}_2: \quad & \{2\}[G]\neg(\min_{80}(V_4) \geq 1.15) \wedge \\ & \{2\}\langle G\rangle(\min_{80}(V_{14}) \geq 6.81 \times 10^6) \wedge \\ & \{2\}\langle G\rangle(\min_{80}(V_{14}) \geq 6.81 \times 10^6 \wedge \langle \overline{L}\rangle(\max_{80}(V_{22}) \leq 7 \times 10^5)) \rightarrowtail \mathit{pos}\end{aligned}$$

The above classification rules describe modal, interval-based patterns in the cough and breath modalities, in terms of audio frequencies and their powers; in principle, they allow for a formal definition of COVID-19 positive traits in the audio footprint of cough and breath events. Recalling that the first modality represents the cough sample and the second one represents the breath sample, and considering that frequencies V_4, V_{14}, V_{15}, and V_{22} approximatively measure 43Hz, 299Hz, 363Hz, and 1405Hz, respectively, the first rule states that *if the power of frequency* 43Hz *in the breath sample reaches 1.15 decibel, and the power of frequency* 363Hz *in the cough sample is always less than* 1.35×10^6 *decibel, then the subject is negative to COVID-19* . The second rule disregards the cough modality, and states that *if in the breath sample the power of frequency* 43Hz *is always less than 1.15 decibel, and there exists an interval where the power of frequency* 299Hz *is always no less than* 6.81×10^6 *decibel, with an earlier interval where the power of frequency* 1405Hz *is always at most* 7×10^5 *decibel, then the subject is positive to COVID-19* .

CHAPTER 8

GAS TURBINE TRIP PREDICTION FROM SENSOR DATA

The growth and the excitement around neural networks has left symbols behind. Some people think that symbols are an evil invention of the old AI people. But symbols were invented because they're useful, necessary abstractions.

Dist. Prof. Dan Roth

Gas turbine trip is an operational event that arises when undesirable operating conditions are approached or exceeded, and predicting its onset is a largely unexplored area. In this section, ModalCART is applied to time-series, sensor data gathered from industrial gas turbines in operation, to predict the likelihood of a trip event occurring in the near future. Similarly to Section 7.1, HS is used as the underlying logical formalism. Inspite of relatively medium-low accuracies, the inherent *clustering nature* of symbolic models provides insights into the turbines' behaviour, as well as high-confidence rules. The presented results were published in Bechini et al. [Bec+23].

§8.1 Problem

Gas turbines have reached a primary position in the thermal power generation field due their efficiency and the availability of natural gas. Given the ever-growing amount of data produced and collected in industrial processes in general, and in turbine operations in particular, it is natural to consider machine learning as a primary tool to extract knowledge from such processes, with the aim of designing systems that may help to ensure a reliable predictive maintenance program. A *trip* event in a gas turbine is an unscheduled operational event during which a turbine abnormally shuts down from a certain operation state, thus leading to a direct impact on its lifespan and revenue [Bha17]. Many factors can favor the occurrence trip events, including, for example, abnormal vibrations, deviations and/or gradients of exhaust gas temperatures, and problems within the fuel spray nozzles. A trip can also occur when a turbine is unable to reach the self-sustained speed, in which case a restart is necessary. In any case, each trip occurrence entails an increase in costs, due to the subsequent necessary repair and maintenance, as well as due to the production interruption. Given the dynamic nature of such an event, a trip may also lead to compressor surge, so that a proper control must be activated to avoid unsafe consequences

for the whole machinery [MPV07; MPV09; PTA13]. Thus, a reliable method that is able to predict, in some form, the occurrence of a trip would be extremely beneficial for both the gas turbine manufacturer and its users.

Recording the values of several variables during a gas turbine operation gives rise to a multivariate time-series, and virtually all engineering tasks of interest in gas turbines, from the machine learning point of view, can be interpreted as classification or regression tasks. In the recent literature, the potential benefits of applying functional learning techniques, such as the ones above, to gas turbines diagnostics has been studied to some extent [LJH16; QC19]; more specifically, in De Giorgi, Campilongo, and Ficarella [DCF18], Zhong, Fu, and Lin [ZFL19], Amozeghar and Khorasani [AK16], and Bai et al. [Bai+20], the authors focus on applying several functional models to gas turbine monitoring, from support vector machines, to artificial neural networks, to nonlinear autoregressive models. Finally, in Naderi and Khorasani [NK18] the authors used a hidden Markov model applied to gas turbine sensors and actuators, and in Wong et al. [Won+14] a particular kind of neural network model, known as *extreme learning machine*, was used to perform diagnostics of gas turbines by employing features extracted from vibration signals. A common element to virtually all attempts at applying learning techniques to gas turbine diagnostics and predictive maintenance tasks, up to and including trip events, is their functional nature. However, when data do not support the construction of reliable models in absolute terms, functional models become less useful, as they cannot be inspected, and the partial knowledge that may have been extracted is hidden behind not always satisfactory statistical results. Symbolic approaches are less widespread in the engineering community, and . In this work we consider several measurable gas path variables recorded in a fleet of Siemens gas turbines located worldwide, and previously used by the same authors in other works [Los+21b; Los+21a; Los+22b]. Unlike previous attempts, we treat such data in the context of a regression problem, and we approach the question of predicting, given the present and past states of a turbine, how close the next trip event is. We apply ModalCART with different parametrizations, and extract rules that, to some extent, can detect anomalies. This is made possible by the symbolic nature of the method, which had not been applied before to predictive maintenance problems.

§8.2 Data

The data for this case study consists of 52 recordings of trip events and the respective measured variables (22 in total) taken during twenty-four hours of operation before trip occurrence. Such data were taken from 4 different gas turbines that belong to Siemens' fleet, which in the following will be referred to as turbine 1, 2, 3 and 4. Each recording consists of the values of 25 variables per minute, therefore entailing 1440 tuples of 25 values per recording. Each recording ends with a trip event; we use these data to model the problem of establishing *how far away* a certain operational time point is from the trip event. The considered variables, which are all raw measured values during gas turbine operation, are described in Tab. 8.1.

We first model the problem as a regression problem, using the distance (in minutes) from the trip event as target variable. To this end, we exclude the last 10 minutes before the trip event. During this 10-minutes time-frame, the trip event is already occurring, and the values of all variables change abruptly, therefore undermining any learning step. While the time point of

symbol	variable
AMB_H	ambient air humidity
AMB_T	ambient air temperature
GAS_P	gas fuel valve position
GAS_F	gas fuel mass flow rate
IGV_P	IGV compressors position
CD_T	compressor outlet temperature
CD_P	compressor outlet pressure
SPEED	rotational speed
POWER	power output
EX_T1	exhaust temperature thermocouple 1
EX_T2	exhaust temperature thermocouple 2
EX_T3	exhaust temperature thermocouple 3
EX_T4	exhaust temperature thermocouple 4
EX_T5	exhaust temperature thermocouple 5
EX_T6	exhaust temperature thermocouple 6
EX_T7	exhaust temperature thermocouple 7
EX_T8	exhaust temperature thermocouple 8
EX_T9	exhaust temperature thermocouple 9
EX_T10	exhaust temperature thermocouple 10
EX_T11	exhaust temperature thermocouple 11
EX_T12	exhaust temperature thermocouple 12
EX_T13	exhaust temperature thermocouple 13
EX_T14	exhaust temperature thermocouple 14
EX_T15	exhaust temperature thermocouple 15
EX_T16	exhaust temperature thermocouple 16

TABLE 8.1: Description of the variables used in this work.

the likely onset of trip symptoms can vary, as demonstrated in Losi et al. [Los+22a], the actual development of the trip event takes place in a very short period of time. Thus, a 10-minute truncation does not introduce any bias and contributes to data uniforming. From the remaining 1430 points of each recording we extract 138 one-hour series by sliding a moving window backwards from the last usable value, with a step of 10 minutes. We therefore obtain a dataset of 7176 one-hour multivariate time-series of 60 points each; in the following, they will be referred to as *instances*. Then, we separate the numeric classes into two bins: more than 4 hours (included) from the event (class *far from the event*, denoted F), and less than 4 hours from the event (*close to the event*, C); such a separation has been performed by computing the length of the interval that lies between the last point of the considered one-hour series and the point 1430, and the limits have been decided after a preliminary study of the informative content of the possible ones and taking into account a minimal amount of operative time in case of upcoming trip prediction. As a consequence, we derive a binary classification problem of discerning whether the trip event is close (C) or not (F). Tab. 8.4 shows the number of instances for the two classes for each of the turbines.

measure	symbol
mean	M
max	MAX
min	MIN
mode of z-scored distribution (5-bin histogram)	$Z5$
mode of z-scored distribution (10-bin histogram)	$Z10$
longest period of consecutive values above the mean	C
time intervals between successive extreme events above the mean	A
time intervals between successive extreme events below the mean	B
first $1/e$ crossing of autocorrelation function	FC
first minimum of autocorrelation function	FM
tot. power in lowest $1/5$ of frequencies in the Fourier power spectrum	TP
centroid of the Fourier power spectrum	CE
mean error from rolling 3-sample mean forecasting	ME
time-reversibility statistic $\langle(x_{t+1-x_t})^3\rangle t$	TR
automutual information ($m = 2$, $\tau = 5$)	AI
first minimum of the automutual information function	$FMAI$
proportion of successive differences exceeding 0.04σ	PD
longest period of successive incremental decreases	LP
Entropy of two successive letters in equiprobable 3-letter symbolization	EN
change in correlation length after iterative differencing	CC
exponential fit to successive distances in 2-d embedding space	EF
ratio of slower timescale fluctuations that scale with DFA (50% sampling)	$FDFA$
ratio of slower timescale fluctuations that scale with linearly rescaled range fits	FLF
trace of covariance of transition matrix between symbols in 3-letter alphabet	TC
periodicity measure	PM

TABLE 8.2: 25 statistical measures for time-series used in ModalCART (including 22 measures from Lubba et al. [Lub+19]).

§8.3 Experimental Setting

Several simulations of training via ModalCART and testing were run. In order to both reduce training time and achieve higher performance, each of the 60-points series is reduced via a moving average filter, which partitions the series into chunks of equal size s along the time axis, and aggregates each chunk by computing the average, ultimately producing a series of $\frac{60}{s}$ points. After a preliminary study in which different decision tree parametrizations are tested, two pruning limits were fixed, namely, a minimum number of 4 instances at the tree leaves, and a minimum entropy gain of 0.015 when selecting the best split condition at any internal node. The simulations are run using a variant of the typical cross-validation setting. In order to prevent data leakage and ensure a fair evaluation of performances, for each simulation, data from a single turbine is used for either training or testing, but not both. More specifically, for each parametrization the following protocol is adopted: 4 simulations are run, and each time the data from a single turbine is used for testing, while data from the other turbines, subsequent to a downsampling step which

	Test turb.	κ (%)	OA (%)	AA (%)	sens (%)	spec (%)	PPV (%)	NPV (%)	F1 (%)	# nodes	# leaves	height	time (s)
s = 4, 4 turbines	1	36	79	71	60	83	42	91	49	275	138	21	821
	2	20	72	62	48	77	30	87	37	203	102	17	686
	3	28	71	69	66	73	33	91	44	251	126	18	801
	4	9	71	55	32	79	24	85	27	213	107	16	823
	avg.	23.4	73.1	64.5	51.3	77.7	32.4	88.4	39.5	236	118	18	783
s = 15, 3 turbines	1	11	51	61	76	45	23	90	35	69	35	13	202
	2	**6**	**38**	**57**	**87**	**28**	**20**	**91**	**33**	**15**	**8**	**5**	**86**
	3	21	68	65	60	70	29	89	39	185	93	21	186
	avg.	12.8	52.2	60.9	74.4	47.5	24.1	90.1	35.7	90	45	13	158
s = 12, 3 turbines	1	21	70	64	54	73	30	88	38	209	105	19	249
	2	**28**	**75**	**67**	**56**	**79**	**35**	**89**	**43**	**177**	**89**	**15**	**141**
	3	15	65	61	56	66	26	88	36	145	73	15	170
	avg.	21.3	69.7	64	55.3	72.7	30.4	88.5	39.1	177	89	16	187

TABLE 8.3: Results obtained by the trained classification models.

turbine	# inst. for C	# inst. for F	# inst. (tot)
1	288	1368	1656
2	384	1824	2208
3	264	1254	1518
4	312	1482	1794

TABLE 8.4: Number of instances gathered from the four gas turbines.

ensures that the classes are balanced, is used for training. 25 feature functions from Tab. 8.2 were used, including the general-purpose feature function set Catch22 [Lub+19]

§8.4 Statistical Performance and Model Complexity

Tab. 8.3 displays some properties of the trained models, the training time required, and the performance obtained on the relative test data, for those simulations that achieved the most relevant results. The performance itself is measured in terms of κ coefficient (which relativizes the overall accuracy to the probability of a random correct answer), overall accuracy (OA), average accuracy (AA), sensitivity, specificity, precision (or positive predictive value, PPV), negative predictive value (NPV) and F1-score. It should be recalled that class C is considered as the positive class, while class F is considered as the negative class; as such, sensitivity and specificity represent the ability of the model to detect the presence or absence of trip events in the subsequent 4 hours, respectively. Additionally, the size of each tree can be assessed via the number of nodes, leaves, and via the tree's height. Note that the test sets, always consisting of instances from a single turbine, displays a high imbalance, with only 17% of the instances belonging to the positive class and 83% belonging to the negative class.

The results after the first round of simulations show that the overall accuracy, averaged over the four executions, is 73%. This results should be interpreted, however, taking into account the imbalance among classes; the average accuracy, which normalizes it in this sense, is about 64% – 65%. The κ coefficient is a measure of 'how well' the model has learned from the data (a value of 0 would mean that the model has the same predictive capacity of a random choice): on average, we reach 23%, which shows that we were able to extract, at least, *some* information from the data. Our ability to distinguish the class C is lower than the one to distinguish the class

F, which means that the models are more likely to extract good rules to *exclude* an upcoming trip event than to *warn* against one. This first round of simulations varying s also shows how turbine 4 behaves differently from the others: when turbine 4 is used for testing, a drop in performances occurs. More specifically, sensitivity experiences a dramatic drop from an average of 58% to 32% and precision from an average of 35% to 24%. This is likely due to the trip event being caused by different factors in the recordings from turbine 4. Native temporal data mining systems such as the ones used in this work search for temporal patterns; even if trip events in turbine 4 are similar to those in the other turbines, and could be detected with similar performances, they may show different patterns, thus worsening the accuracy of the entire system. When turbine 4 is ignored, the training phase is able to produce models with higher sensitivity, as well as models that are similar in performance, but more concise (i.e., with a lower number of nodes and leaves; observe that the smaller the tree, the more interpretable is). Tab. 8.3 shows the results for two representative parametrization, with $s = 15$ and $s = 12$ respectively. The first parametrization yields smaller trees (average of 45 leaves, compared with 118 in the previous round) with lower precision and F1-score, but a much higher sensitivity (average of 74%, compared with 51% in the previous round). The second parametrization yields trees with higher specificity and precision, namely, that perform better at detecting the absence of a trip event. The different behavior obtained with different but similar choices of s should be further investigated.

§8.5 Post-hoc Interpretation

From the first and second parametrization, we select and consider the two trees with highest sensitivity and precision, respectively, and denote them as τ_1 and τ_2. The rows relative to the two selected trees are highlighted in bold in Tab. 8.3. Incidentally, both trees are obtained by using turbine 2 for testing and turbine 1 and 3 for training, but they cover different aspects of providing *good* classifications. The two trees are displayed in textual and graphical form in Tabs. 8.5 and 8.6, respectively. Rows in the tables either correspond to an inner decisional node, or to a leaf, where the test instances are routed to and finally classified. For each leaf, the values of four performance metrics (support, confidence, lift, and conviction, see Section 3.4) are reported, for evaluating the corresponding classification rule. Note that thresholds have been made anonymous and denoted as $THRESH1$, $THRESH2$, etc., and that some uninteresting parts of the trees have been ellipsed for a clearer presentation.

Focusing on τ_1 (Tab. 8.5), a few considerations can be made. First, there exists a simple rule (r_1) for excluding the case of class C, that applies to about 5% – 6% of the cases, which can be paraphrased as *if there is not an interval where the ambient air humidity is lower than* $THRESH1$*, then the trip event is farther than 4 hours with a* 99% *confidence*. Another interesting rule (r_2) states that *if there exists an interval where the ambient air humidity is lower than* $THRESH1$*, as well as another larger interval that contains the first, where the exhaust temperature of thermocouple 4 is (i) on average higher than* $THRESH3$*, but (ii) always less than* $THRESH2$*, and (iii) less than* $THRESH4$ *in the final part of the interval, then the trip event is farther than 4 hours with a* 90% *confidence*. This second rule is more articulated, and has a lower confidence than the first. However, it covers 18% of the test instances, and 364 out of 1824 test instances is correctly classified by this rule, which means that this rule is responsible for a 20 of the 28

tree rules	supp	conf	lift	conv
⟨G⟩ MAX(AMB_H) ≤ THRESH1				
✓⟨D̄⟩ MAX(EX_T4) ≤ THRESH2				
✓ M(EX_T4) ≥ THRESH3				
✓⟨E⟩ MAX(EX_T4) ≤ THRESH4				
✓ F : 364/403	0.1825	0.90	1.09	1.80
✗ MIN(AMB_T) ≥ THRESH5				
✓ F : 11/11	0.0050	1.00	1.21	Inf
✗ C : 0/16	0.0072	0.00	0.00	0.00
✗ ...				
✗ C : 330/1610	0.7292	0.21	1.18	1.04
✗ F : 123/124	0.0562	0.99	1.20	21.57

TABLE 8.5: Decision tree τ_1. Three rules and paths with high confidence and/or support are highlighted.

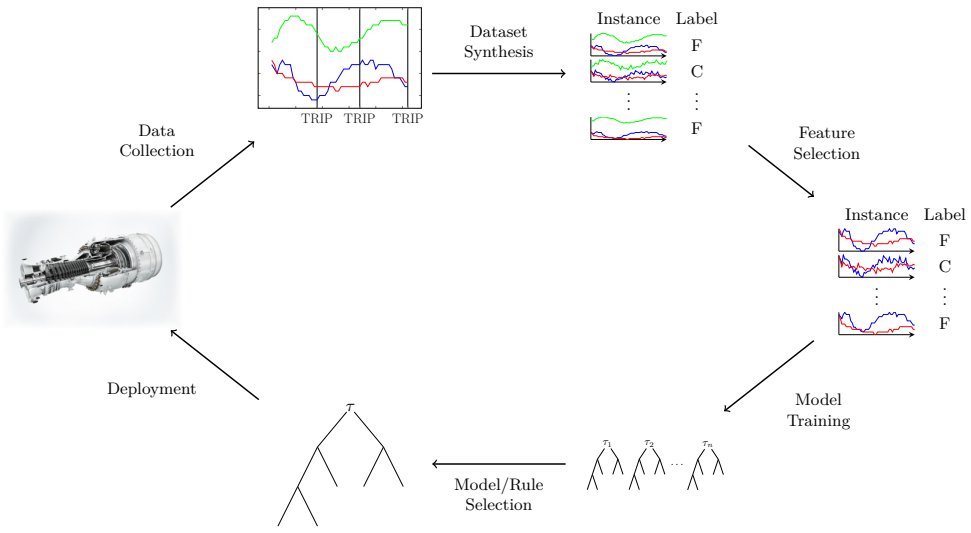


FIGURE 8.1: A workflow of the proposed method.

tree rules	supp	conf	lift	conv
⟨G⟩ MAX(AMB_H) ≤ THRESH6				
✓⟨D̄⟩ MAX(EX_T4) ≤ THRESH7				
✓⟨Ē⟩ MIN(EX_T4) ≥ THRESH8				
✓ F : 362/398	0.1803	0.91	1.10	1.92
✗ ...				
✗⟨Ō⟩ C(AMB_T) ≥ THRESH9				
✓ MAX(AMB_T) ≤ THRESH10				
✓⟨A⟩ MIN(AMB_H) ≥ THRESH11				
✓⟨L̄⟩ MAX(EX_T5) ≤ THRESH12				
✓⟨B̄⟩ M(SPEED) ≤ THRESH13				
✓ F : 137/158	0.0716	0.87	1.05	1.31
✗ ...				
✗ ...				
✗ MIN(EX_T5) ≥ THRESH14				
✓ ...				
✗ F : 25/25	0.0113	1.00	1.21	Inf
✗⟨A⟩ MIN(AMB_H) ≥ THRESH15				
✓⟨L̄⟩ MIN(SPEED) ≥ THRESH16				
✓⟨D̄⟩ MIN(SPEED) ≥ THRESH17				
✓ C : 6/14	0.0063	0.43	2.46	1.45
✗ ...				
✗ C : 12/20	0.0091	0.60	3.45	2.07
✗ MIN(AMB_H) ≥ THRESH18				
✓⟨Ē⟩ MIN(AMB_H) ≥ THRESH19				
✓⟨B̄⟩ MAX(AMB_T) ≤ THRESH20				
✓ ...				
✗ F : 39/39	0.0177	1.00	1.21	Inf
✗⟨O⟩ MIN(SPEED) ≥ THRESH23				
✓ F : 28/29	0.0131	0.97	1.17	5.04
✗ ...				
✗⟨B⟩ MIN(EX_T5) ≥ THRESH27				
✓⟨A⟩ MAX(SPEED) ≤ THRESH28				
✓⟨L̄⟩ MAX(SPEED) ≤ THRESH29				
✓ ...				
✗ C : 10/36	0.0163	0.28	1.60	1.14
✗ F : 59/61	0.0276	0.97	1.17	5.30
✗ F : 58/69	0.0312	0.84	1.02	1.09
✗⟨A⟩ MIN(EX_T5) ≥ THRESH30				
✓ MAX(AMB_H) ≤ THRESH31				
✓ C : 22/22	0.0100	1.00	5.75	Inf
✗ F : 16/16	0.0072	1.00	1.21	Inf
✗ MAX(EX_T4) ≤ THRESH32				
✓⟨Ā⟩ MIN(AMB_H) ≥ THRESH33				
✓⟨Ē⟩ MIN(AMB_T) ≥ THRESH34				
✓ C : 17/23	0.0104	0.74	4.25	3.17
✗⟨Ō⟩ MIN(AMB_H) ≥ THRESH35				
✓ C : 14/15	0.0068	0.93	5.37	12.39
✗⟨B̄⟩ ...				
✗ MIN(EX_T4) ≥ THRESH40				
✓ MIN(EX_T4) ≥ THRESH41				
✓ MAX(AMB_T) ≤ THRESH42				
✓ MIN(AMB_T) ≥ THRESH43				
✓ MIN(AMB_H) ≥ THRESH44				
✓ C : 17/36	0.0163	0.47	2.72	1.57
✗ ...				
✗ ...				
✗⟨L̄⟩ MAX(AMB_T) ≤ THRESH45				
✓ ...				
✗ MIN(EX_T4) ≥ THRESH46				
✓ C : 10/30	0.0136	0.33	1.92	1.24
✗ ...				
✗ F : 46/51	0.0231	0.90	1.09	1.77
✗ ...				
✗ FM(AMB_T) ≤ THRESH51				
✓⟨E⟩ MAX(AMB_H) ≤ THRESH52				
✓ ...				
✗⟨B⟩ MAX(AMB_T) ≤ THRESH53				
✓ MIN(AMB_T) ≥ THRESH54				
✓ F : 48/50	0.0226	0.96	1.16	4.35
✗ ...				
✗ ...				
✗ F : 62/62	0.0281	1.00	1.21	Inf
✗ F : 119/119	0.0539	1.00	1.21	Inf

TABLE 8.6: Decision tree τ_2. Three rules and paths with high confidence and/or support are highlighted.

percentage points of specificity (note that another 7 points is covered by r_1). Rules similar to r_1 and r_2, but with higher confidence and slightly lower support were extracted when using a different moving average filter; in fact, τ_2, shown in Tab. 8.6, provides a variant of r_1 that uses a stricter threshold ($THRESH6$) for the ambient air humidity, and achieves 100% confidence with a nearly equal support. Additionally, it provides a variant r_2 that states that *if there exists an interval where the ambient air humidity is lower than* $THRESH6$*, as well as another larger interval that contains the first, where the exhaust temperature of thermocouple 4 is (i) always less than* $THRESH7$*, (ii) but higher than* $THRESH8$ *just before, and throughout, the whole interval, then the trip event is farther than 4 hours with a* 91% *confidence*. As for class C, τ_1 provides the following rule: *if there exists an interval where the ambient air humidity is lower than* $THRESH1$*, but there does not exist an interval satisfying the properties presented in* r_2*, then the trip event is closer than 4 hours with a* 21% *confidence*. The peculiarity of this rule lies in the fact that 330 out of 384 instances for C are correctly classified; this entails that if a trip event is about to occur, it can be correctly predicted 86% of the times by solely applying this rule. While it is true that further investigation on rules, their semantics, and their significance is necessary, it is worth to observe that, in this approach, rules can work together even if they are extracted with different processes and in different moments; each rule adds new information, effectively improving the scalability of the whole system.

In order to highlight the ability of the presented method to deal with new operational data, Fig. 8.1 presents a generalized workflow for applying modal decision trees to trip prediction tasks on new gas turbines. The process starts with sensor data collection, proceeds to the synthesis of time-series instances (each associated with a class label), that are then reduced by a statistical analysis and feature selection step. The dataset is then used for training modal decision trees with different parametrization. Finally, a tree model can be synthesized by manual or automatic selection of good rules, and it can be deployed as a trip event warning system. As new data is collected, newer models can be trained and synthesized, making the process robust to changes in time (e.g., deterioration of a turbine's inner component). A different question concerns the robustness of the obtained models; while a model deployed on a given turbine is expected to yield higher performances when trained on data from the same turbine, testing a model on data taken from an entirely different environment allows one to estimate its generalization abilities. To this end, we performed a final external test by applying both decision trees τ_1 and τ_2 on data from six different turbines located in a different continent. The new dataset includes the recordings of 42 trip events, each with length of twenty-four hours. The observed change in overall accuracy was from 38% to 43% and from 75% to 65%, respectively. Interestingly, τ_1 decreases its sensitivity from 87% to 75%, but the gained test accuracy is due to a higher specificity, which increases from 28% to 36%, suggesting that the rules for class F may be more compelling than expected. As for τ_2, both sensitivity and specificity experience a decrease, respectively from 56% to 36% and from 79% to 71%, but, at least in the latter case, such a degradation of performance can be considered relatively contained. While more specific research must be performed in order to assess the performance change for the single rules, we can conclude that our results are a solid starting point for further research towards AI-driven trip prediction.

CHAPTER 9

PREDICTION OF THE EXPERIENCED LEVEL OF BEAUTY FROM EEG RECORDINGS

Everything not saved will be lost.

Nintendo "Quit Screen" message

The study of the electroencephalogram (EEG) signals recorded from subjects during an experience is a way to understand the brain processes that underlie their physical and emotional involvement. Using a proprietary dataset of 248 EEG recordings of subjects being exposed to art paintings, this work explores the application of HS-based ModalCART-RF at the task of predicting the liking or disliking of a painting. The extracted rules relate the liking to the voltage of electromagnetic frequencies at specific locations in the brain, and sometimes achieve near-perfect performances. Within the limits of the experiment, such rules (and their translation into natural language) may help to understand the brain process that drives liking or disliking experiences in human subjects. The presented results were published in Caselli et al. [Cas+23].

§9.1 Problem

Neuroaesthetics was defined by Zeki [Zek99] as the scientific approach to the study of the perception of beauty, that is, the study of *liking* (to be distinguished from *wanting*, as in Chatterjee and Vartanian [CV14]) in a broad sense. The approaches to neuroaesthetics vary very much in the recent scientific literature, but they can be fundamentally divided into *top-down* processes, in which the essence of beauty undergoes an axiomatic treatment in which the subjective feeling is broken down to its constituting elements, and *bottom-up* ones, in which some kind of objective data is analysed and related to the subjective expression of beauty. While the former is certainly fascinating from an epistemological point of view (it tries to answer the question of whether *beauty can be defined*), and it is, in essence, a philosophical exercise, the latter is a neurophysiological one, based on real data, systematic, and carried on with modern analysis techniques. Such an approach had already been suggested as early as the second half of the 19th century (the so-called Weber-Fechner Law), as an attempt to create a mathematical relationship between stimulus and perception, although not to quantify beauty, and it is largely the most common one.

Bottom-up strategies aimed to understand the mechanisms of the brain can be broadly classified into (functional) MRI (*magnetic resonance*) image processing and interpretation, and EEG (*electroencephalogram*) signal processing; in some cases the approaches are combined, or other brain analysis techniques are adopted. Even focusing on EEG only, the relevant work is huge. It includes, among other elements: experiments aimed to understand human emotions (of which liking is just one example, often not considered as a single emotion but as a combination of several ones [BW03]); experiments designed with own or public datasets; case studies in laboratory conditions or real-life situations; approaches focused on qualitative or quantitative descriptions, in some cases time-related; experiments in which the stimuli vary from images, to audio, to movies, sometimes computer-generated. Performing a complete review of such a body of work is beyond the scope of this work; just the most recent review paper of this field [Liu+21] surveys as much as 102 contributions selected from a pool of 654 potential ones. Such an high interest in the subject is also witnessed by the availability of several public datasets, which include SEED, DEAP, MANHOB-HCI, DREAMER, AMIGOS, DECAF, INTERFACES, among many others (see [Rah+21] for a complete list of public datasets). From 1985 to 2020, several authors have worked, in particular, on the problem of classifying emotions from EEG signals using statistical/machine learning techniques with approaches that are comparable with ours. To begin with, Ray and Cole [RC85] approached the problem of examining the effect of attentional demand during cognitive and emotional tasks, in a pair of experiments with 18 and 40 subjects, respectively; the experiments were carried out in a clinical context and, in terms of emotions, the focus was on distinguishing between happiness and sadness. In Musha et al. [Mus+97] the authors focused on specific features extracted from 10-electrode EEGs to be correlated with anger, sadness, joy, and relaxation; their experiment involved 5 subjects. Schmidt and Trainor [ST01] found that the pattern of asymmetrical frontal EEG activity distinguished valence of the musical excerpts, in an experiment with 59 subjects, 4-electrode EEG, and music-induced stimulation. In Murugappan et al. [Mur+08] the authors discussed how the changes in the electrical activity of the human brain related to distinct emotions, in an experiment with 6 subjects and 63-electrode EEG, focusing on anger, disgust, fear, happiness, sadness, and surprise, elicited via the exposition of the subjects to movie clips; three of the same authors later focused on signal preprocessing, and designed a classifier for 5 emotions, that is, disgust, happy, surprise, fear and neutral, from an experiment with 20 subjects [Mur+10]. Only two emotions, instead, were the subject of an experiment by Li and Lu [LL09], carried out on 10 subjects, stimulated, in a clinical context, by being exposed to pictures displaying facial expressions; while in this case very high accuracies were reached, the total number of trials was relatively low. Schaaff and Schultz [SS09] performed an experiment with 5 subjects, 4-electrode EEG, with the aim of building a classifier to recognize positive, neutral, and negative emotions in humans, with relatively low success. In Khosrowabadi and Rahman [KR10] the authors studied EEG correlates on emotions using features extracted by Gaussian mixtures of EEG spectrograms; their experiment involved 31 subjects, 8-electrode EEG, two emotions (positive and negative), and two arousal states (calm and excited); in Khosrowabadi, Quek, and Ang [KQA10], in a similar experiment with 26 subjects, the performance of a EEG-based emotion recognition system based on a self-organizing map to identify the boundaries between separable regions were studied. Petrantonakis and Hadjileontiadis [PH09] proposed a novel emotion evocation and EEG-based

feature extraction technique, combined with higher order crossings analysis, for feature extraction and robust classification; their experiment involved 16 subjects, and 3-channel EEG. In Frantzidis et al. [Fra+10], instead, the authors propose a methodology for robust classification of EEG biosignals into four emotional states; their experiment involved 28 subjects and 19-electrode EEG, and emotions were elicited with pictures observation. Liu and Sourina [LS13; LS14] worked on a real-time fractal dimension based valence level recognition algorithm from EEG signals, which allows a sort of continuous classification of up to 16 different emotions; their work is based on a public dataset. Nie, Wang, Shi, and Lu [Nie+11] considered the problem of classifying positive and negative emotions from 62-electrode EEG, stimulated by the action of watching movie clips from relatively famous, and emotional movies; the high accuracies reached in this case by using support vector machines is to be considered in the perspective of a very reduced number of subject (3) and trials (12 per subject). Later on, the same authors considered, the problem of classifying several types of emotions including amusement, anger, contentment, disgust, fear, neutral, sadness, and surprise [WNL14], using, again, support vector machines; their experiment involved 6 subjects, and movies were, once more, used to generate emotions in laboratory conditions. In Hadjidimitriou and Hadjileontiadis [HH12], Hadjidimitriou and Hadjileontiadis used different feature extraction approaches and classifiers, and focused on the discrimination between subjects' EEG responses to self-assessed liked or disliked music, in an experiment with 9 subjects and 14-channel EEG. Jenke, Peer, and Buss proposed a statistical method to select electrodes and features that separate classes well in the problem of emotion classification; their experiment was carried on 16 subjects with a 64-channel EEG [JPB13]; at the same conference, Rozgić, Vitaladevuni, and Prasad addressed single-trial binary classification of emotion dimensions, using a public EEG dataset [RVP13]. In Alhagry, Fahmy, and El-Khoribi [AFEK17], the authors proposed a deep learning method is proposed to recognize emotion from raw EEG signals using LSTM neural networks and a public dataset. Neural networks were also applied to the same problem in Schirrmeister et al. [Sch+17], with several public and private datasets, and also in Garg et al. [Gar+19], again on a public dataset, as well as in Luo et al. [Luo+20].

Several considerations can be drawn from reviewing the literature. First, most of the existing work on computational neuroaesthetics focuses on artificial, 2D images and 3D shapes designed with experimental purposes, or movies, and experiments are almost never carried on in real contexts. Moreover, EEG signals have been analysed with functional machine learning techniques that do not allow, in general, the interpretation of the resulting models, nor the extraction of explicit rules, and can only be judged by their statistical performances. The most common approaches vary from univariate/multivariate inferential and descriptive statistical approaches, to support vector machines, to neural networks; in some other cases, EEG signals have been classified with other methods, e.g., hidden Markov models, but not in the case of emotions. Furthermore, even if liking is one of the labels explicitly present in public datasets (e.g. DEAP), it has been rarely singled out in the experiments; quickly surveying among the authors that do mention it [JPB14; RVP13; AFEK17; Gar+19], we may observe that the average accuracy of classification ranges from 60% to 86%, although the peculiarities of the single experiments and their conditions hardly allow for a definitive comparison. Finally, most of the effort has been directed to build complete classification models, and only in a few cases to understand, in a

systematic fashion, the role of each electrode placement, frequency, and feature in relation with the class to be learned. In this work, we consider data collected during a real-world experiment, analysing the explicit and implicit reactions of participants, using different kinds of sensors, during their visit to an art exhibition. The data considered are part of a larger biosignals dataset recorded to evaluate the participants' reactions during the observation of paintings, which included ECG (*electrocardiogram*), EDA (*electrodermal activity*), two different tools for EEG recording, the *gaze pattern*, the participants' main characteristics (age, gender, education, familiarity with art, etc.) and their explicit judgments about paintings.

§9.2 Data

The dataset used in this work was collected during the exhibition *Painting affections: sacred painting in Ferrara between the '500 and the '700*, set up at the Estense Castle in Ferrara (Italy) from the 26th of January to the 26th of December 2019 [Coc+19; Coc+20]. The original exhibition included 54 paintings, of which 18 have selected according to some exclusion criteria, to avoid the risk that people may be distracted when approaching to the observation point (too small paintings), by details unrelated to the artwork itself (damaged paintings) or by the difficulties in focusing on single paintings (paintings that are part of a series). The experiment included two different ways to collect EEG data, to analyse benefit and boundaries of different tools, procedures and final results. However, in this work we analyse only the outcomes coming from subjects wearing a dry electrode EEG cap *Waveguard™touch* by *ANT Neuro* in the 64-channel variant during the exhibition. To minimise the influence of movements, the 16 people involved in this particular setting attended to the trial while using a wheelchair pushed by a researcher. Each experiment produced a single long recording per subject, ranging from the beginning to the end of the exhibition; therefore all data required a preliminary slicing operation, based on the actual time at which the subject was standing in front of each painting, in order to be able to analyse the data related to each specific painting. To better take into account the strictly personal reactions of people to emotions and the environment, each subject has been exposed to a blank image (a white wall) for 60 seconds before his/her trial, which consisted of a 60-seconds observation of each of the 18 paintings. Being recorded at a sampling rate of 512Hz, each data slice consisted of a time-series of 30720 points. Each subject was asked to express an integer in the range $[0, 50]$ as an expression of their subjective liking score of the painting, using a *Visual Analog Scale* (*VAS*) scale [WL90]. Since not all subject-painting pairs were recorded, the resulting dataset is composed of 248 instances, each instance being described by 64 time-series (one for each electrode) of 30720 points and labelled by the relative liking score. A previous study provided importance measurements for each of the 64 electrodes, as well as importance measurements for feature functions (or *measures*) taken from Tab. 8.2.

Statistically-wise, subjects were analysed by their gender, age, nationality, education level, and subjective interest in the painting arts. The subjects' set is composed of 10 males and 6 females with an average age of 32.19 ± 15.81, with a subjective interest in the painting arts of 25.93 ± 7.58 on a scale of $[0, 50]$. Moreover, 14 subjects (that is, the 87.5%) were Italian, 1 (6.25%) was Portuguese and 1 (6.25%) was Russian, and the education level was distributed as follows: in the 62.5% (10 subjects) of the cases, the highest level of education was high

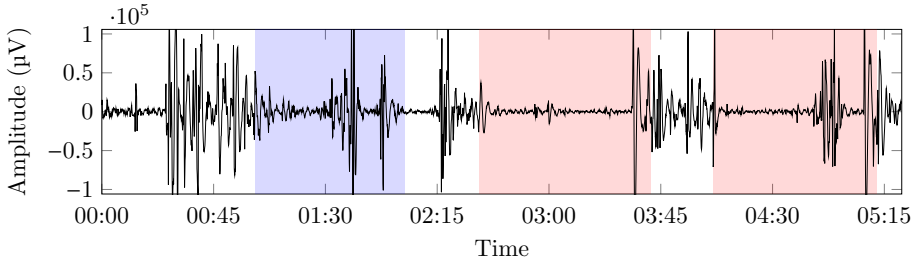


FIGURE 9.1: Example of a raw EEG signal of a single subject, from a single electrode; the first highlighted area corresponds to the blank portion of the observation, while the others correspond, each, to the observation of different paintings (here, only two paintings are shown in the picture).

school diploma, in the 12.5% (2 subjects) was bachelor's degree, in the 12.5% (2 subjects) it was master's degree, and in the remaining 12.5% (2 subjects) cases it was PhD, or other type of postgraduate. In terms of sample size, and other relevant hypothesis on the distribution of data, no pre-qualifying tests (e.g., power analysis) were performed, as they are generally unnecessary when data are analyzed with machine learning techniques; this is particularly relevant in observational studies that require complex preparation and whose sample size and distribution cannot be easily controlled at design time.

The raw EEG signals (see an example in Fig. 9.1), originally sampled at 512Hz were, first, re-sampled at 104Hz in order to lower the Nyquist frequency to 52Hz [Sha49] while removing possible noisy frequency bands in the high range, and, then, processed with a Short Time Fourier Transform (STFT), in order to retain both the frequency and time domains. The $(0, 52]$Hz spectrum (0 excluded, 52 included) was divided into 13 equally wide bands of frequencies $(F_1, \ldots, F_{13})$, with a resulting width of 4Hz, as per the result of the initial screening. The STFT time window size was set to 50ms with a step time of 20ms, and the resulting time-series per each trial were 3379 points long. For reference and comparison with existing work, such 13 bands can be grouped into the five standard relevant wave patterns (see Fig. 9.2 for an example): δ, ranging in 0−4Hz (usually associated with slow-wave sleep), θ, ranging in 4−8Hz (usually associated with phase 1 and 2 of non-REM sleep and with REM sleep), α, ranging in 8−12Hz (usually associated with waking state with closed eyes and instants prior to fall asleep), β, ranging in 12−32Hz (usually associated with intense mental activity), and γ, ranging in 32−52Hz (usually associated with states of particular stress). Observe that frequencies F_i with $1 \leq i \leq 13$ measure finer bands than a normal band $b \in \{\alpha, \beta, \gamma, \delta\}$; the resulting dataset is, thus, richer than those obtained in more classic approaches (e.g., using the four bands directly), but not so rich to generate learning noise (e.g., using 1Hz bands). This makes it possible to extract more statistically reliable models at the expenses of a more computationally demanding extraction phase. Our information extraction approach being not distribution-dependent and based on simple variance renders the typical eye blinking and movement noise cancellation unnecessary [Gwi+10], effectively simplifying the whole procedure; moreover, in this way, we expect to extract a model that is, at least in principle, robust to eye blinking and neck muscle contraction noise, to be used in a real-world environment.

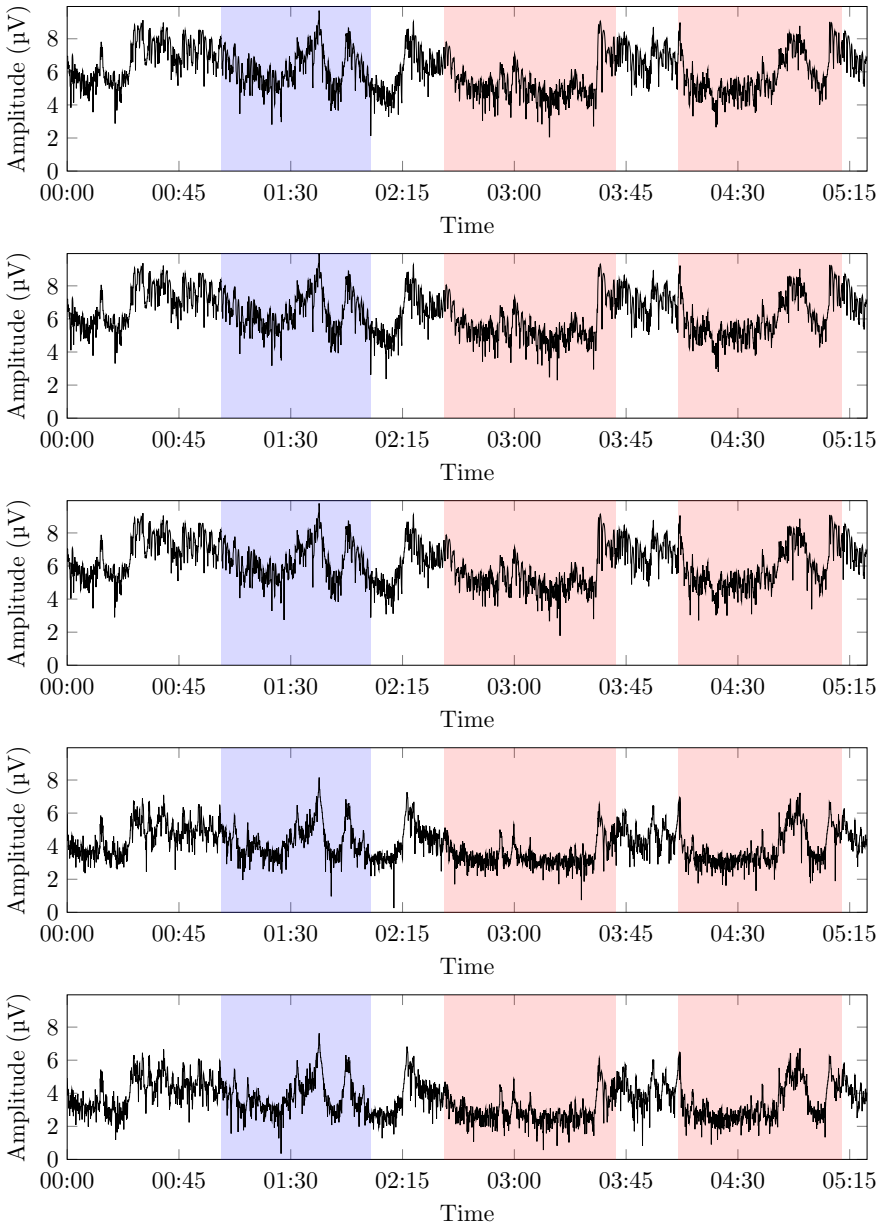


FIGURE 9.2: From top to bottom, the intensity of voltage at bands δ, θ, α, β (specifically, the interval of β included in F_6), and γ (specifically, the interval of γ included in F_{11}) during the trial of example in Fig. 9.1, after preprocessing. All graphics are $\log_{10}$-normalized. As before, the first highlighted area corresponds to the blank portion of the observation, while the others correspond, each, to the observation of different paintings (two, in this case).

The scores among the dataset, in the range $[0, 50]$, were non-normally distributed ($p = 10^{-56}$, with a Kolmogorov-Smirnov normality test). For the purpose of learning, we treated the resulting problem as a classification one, by discretizing the liking scores into three categories (dislike, neutral, like). For this *data binning* step, the three different pairs of thresholds $\langle < 25, \geq 25 \rangle$, $\langle < 17, \geq 34 \rangle$, and $\langle < 10, \geq 41 \rangle$ were considered, giving rise to three datasets ($\mathcal{D}_{25}$, $\mathcal{D}_{34}$, $\mathcal{D}_{41}$) encompassing, respectively, 248 instances (125 dislike versus 123 like), 167 instances (86 dislike versus 81 like), and 91 instances (46 dislike versus 45 like). Binning was equal-length despite the distribution not being normal, as it still shows a symmetrical behaviour with respect to the median value.

§9.3 Experimental Setting

Three sub-groups of 1, 5, and 10 electrodes, respectively, were selected accourding to their importance, giving rise to three datasets. For each dataset, ModalCART-RF was applied, and the models were trained on the most important feature functions, using all frequency bands $F_1, \ldots, F_{13}$ (from δ to γ), in different combinations. Since the three datasets are different in terms of number of instances, in order to ensure comparability of the results we ran all the experiments in *leave-p-out* cross-validation [CR08], with $p = 10$; the leave-p-out method consists in training the model leaving p instances as validation set and repeating this process over all the possible combination of p instances with different random seeds. This cross-validation method is mostly adopted when dealing with small datasets; to maintain consistency across our experiments we limited ourselves to 10 repetions out of all possible combinations; leave-p-out generalizes the more standard cross-validation method and ensure the comparability of results between datasets of very different cardinalities, and choosing $p = 10$ is mostly accepted in the literature in terms of number of folds [KJ13].

For each of the three datasets, we explored 4 different configurations of frequency bands, *(i)* all the 13 frequency bands *(ii)* only those corresponding to β, *(iii)* only those corresponding to γ and *(iv)* only those corresponding to β and γ. From each of the signals recorded by the electrodes, modal random forests were trained on the 5 best measures; we performed three different groups of experiments, using the best 10, the best 5 and the single best electrode(s), respectively, as explained in the previous section. As a consequence, we have a total of 36 experiments composed of 10 seeds each.

Tab. 9.1 reports a summary of the validation performances; for each experiment, the table reports the average and standard deviation of four performance metrics, namely, the overall accuracy (the fraction of correctly classified instances), the balanced accuracy (the average between sensitivity and specificity), the sensitivity (the fraction of likes that were correctly classified), and the specificity (the fraction of dislikes that were correctly classified); our datasets not being perfectly balanced makes the balanced accuracy slightly more informative than the overall accuracy, the latter being, however, a more standard metric, usually preferred in terms of comparison among different experiments. To ensure a deeper exploration of the solution space, we also run all experiments in the same conditions, but selecting only the best measure (instead of the best 5 ones); the results of this version of the experiments are shown in Tab. 9.2.

		nelectrodes = 1				*nelectrodes* = 5				*nelectrodes* = 10			
		acc	*macc*	*sens*	*spec*	*acc*	*macc*	*sens*	*spec*	*acc*	*macc*	*sens*	*spec*
$\mathcal{D}_{25}$	*all*	52 ± 19	53 ± 19	37 ± 21	66 ± 26	63 ± 16	62 ± 19	48 ± 22	76 ± 21	66 ± 11	68 ± 10	68 ± 17	69 ± 17
	β	**70** ± 15	67 ± 17	53 ± 24	82 ± 14	56 ± 15	55 ± 16	54 ± 18	56 ± 25	60 ± 17	59 ± 18	52 ± 26	66 ± 19
	γ	59 ± 15	58 ± 17	46 ± 26	71 ± 15	64 ± 14	64 ± 16	60 ± 25	68 ± 19	56 ± 13	57 ± 15	49 ± 18	65 ± 19
	$\beta+\gamma$	63 ± 20	65 ± 20	50 ± 23	79 ± 30	60 ± 14	61 ± 14	56 ± 21	65 ± 21	66 ± 17	65 ± 20	58 ± 33	72 ± 19
$\mathcal{D}_{34}$	*all*	58 ± 12	62 ± 12	44 ± 17	80 ± 19	63 ± 19	62 ± 18	51 ± 28	72 ± 19	59 ± 17	61 ± 19	50 ± 30	71 ± 12
	β	66 ± 14	66 ± 14	53 ± 16	80 ± 20	65 ± 12	67 ± 10	51 ± 15	83 ± 13	65 ± 16	67 ± 16	55 ± 20	80 ± 19
	γ	67 ± 13	68 ± 9	56 ± 23	80 ± 18	62 ± 10	61 ± 8	53 ± 19	70 ± 16	60 ± 11	60 ± 13	64 ± 13	56 ± 21
	$\beta+\gamma$	66 ± 10	65 ± 10	47 ± 18	82 ± 20	66 ± 17	69 ± 15	63 ± 24	74 ± 22	57 ± 16	57 ± 16	48 ± 23	66 ± 20
$\mathcal{D}_{41}$	*all*	**74** ± 13	75 ± 9	57 ± 18	92 ± 10	**71** ± 9	70 ± 11	57 ± 25	83 ± 14	**76** ± 10	80 ± 7	72 ± 16	89 ± 15
	β	**76** ± 13	82 ± 9	64 ± 18	100 ± 2	**71** ± 13	73 ± 14	64 ± 15	83 ± 19	**77** ± 11	79 ± 11	69 ± 17	89 ± 13
	γ	**77** ± 12	78 ± 12	71 ± 12	85 ± 15	**74** ± 6	76 ± 7	68 ± 11	83 ± 14	**72** ± 11	68 ± 15	59 ± 26	78 ± 19
	$\beta+\gamma$	**78** ± 9	81 ± 9	70 ± 14	92 ± 10	69 ± 13	68 ± 12	61 ± 17	75 ± 12	**78** ± 11	78 ± 11	66 ± 17	89 ± 12

TABLE 9.1: Results of the experiments in the original conditions; all values are expressed in percentage points.

		nelectrodes = 1				*nelectrodes* = 5				*nelectrodes* = 10			
		acc	*macc*	*sens*	*spec*	*acc*	*macc*	*sens*	*spec*	*acc*	*macc*	*sens*	*spec*
$\mathcal{D}_{25}$	*all*	53 ± 21	55 ± 20	39 ± 23	68 ± 27	59 ± 9	58 ± 11	48 ± 19	68 ± 19	57 ± 13	61 ± 13	58 ± 22	64 ± 23
	β	**70** ± 14	67 ± 17	54 ± 25	80 ± 14	57 ± 19	55 ± 20	52 ± 24	59 ± 28	53 ± 14	53 ± 16	47 ± 25	60 ± 19
	γ	61 ± 16	60 ± 17	46 ± 25	73 ± 17	64 ± 16	64 ± 19	59 ± 31	69 ± 24	54 ± 11	55 ± 13	49 ± 18	61 ± 17
	$\beta+\gamma$	66 ± 22	66 ± 21	54 ± 26	78 ± 30	58 ± 14	58 ± 13	55 ± 16	61 ± 21	62 ± 16	62 ± 20	52 ± 37	72 ± 20
$\mathcal{D}_{34}$	*all*	61 ± 12	64 ± 11	45 ± 18	83 ± 16	64 ± 23	65 ± 23	57 ± 31	73 ± 22	61 ± 18	62 ± 19	57 ± 21	67 ± 22
	β	67 ± 14	68 ± 15	52 ± 16	85 ± 17	66 ± 15	67 ± 14	54 ± 17	80 ± 14	68 ± 14	70 ± 14	60 ± 15	80 ± 18
	γ	66 ± 13	67 ± 9	54 ± 24	80 ± 13	62 ± 9	61 ± 8	53 ± 15	68 ± 21	64 ± 12	65 ± 12	64 ± 19	67 ± 14
	$\beta+\gamma$	66 ± 9	64 ± 9	48 ± 19	80 ± 19	67 ± 16	68 ± 13	64 ± 23	73 ± 14	63 ± 17	63 ± 16	54 ± 26	72 ± 15
$\mathcal{D}_{41}$	*all*	**73** ± 15	73 ± 12	57 ± 18	90 ± 11	**76** ± 8	74 ± 12	69 ± 28	79 ± 16	**77** ± 10	82 ± 6	72 ± 15	92 ± 15
	β	**76** ± 13	82 ± 9	64 ± 18	100 ± 0	**77** ± 13	78 ± 14	77 ± 15	80 ± 19	**77** ± 13	78 ± 14	71 ± 21	85 ± 13
	γ	**75** ± 11	74 ± 11	71 ± 12	77 ± 17	**79** ± 7	79 ± 9	78 ± 19	81 ± 14	**76** ± 10	73 ± 15	67 ± 25	78 ± 22
	$\beta+\gamma$	**76** ± 11	79 ± 12	70 ± 14	88 ± 16	**70** ± 14	67 ± 13	64 ± 19	70 ± 18	**81** ± 12	81 ± 13	75 ± 16	86 ± 15

TABLE 9.2: Results of the experiments after selecting only the best measure; all values are expressed in percentage points.

§9.4 Statistical Performance

Upon observing Tab. 9.1, a few observations can be made. Varying the dataset from $\mathcal{D}_{25}$ to $\mathcal{D}_{34}$, and then to $\mathcal{D}_{41}$, one observes how the validation accuracy tends to improve, which shows that, as expected, training a model with clearer like/dislike instances (that is, instances with more extreme grades) improves the ability of the model itself to pick up the underlying temporal patterns; the same happens when varying the number of electrodes from 1 to 5, and, then, to 10, although not as clearly. Another general observation is that all the trained models tend to be more specific than sensitive, that is, the models have a greater capacity to distinguish a dislike than a like. Moreover, focusing on the frequencies β and γ (and their combination) tends to improve the performance, with respect to when all frequencies are used. Thus, the best result in terms of balanced accuracy corresponds to the case of the model learned with the frequencies within $\beta+\gamma$ on $\mathcal{D}_{41}$, with a value of 78%, with a specificity of 89%, which means that it correctly recognizes

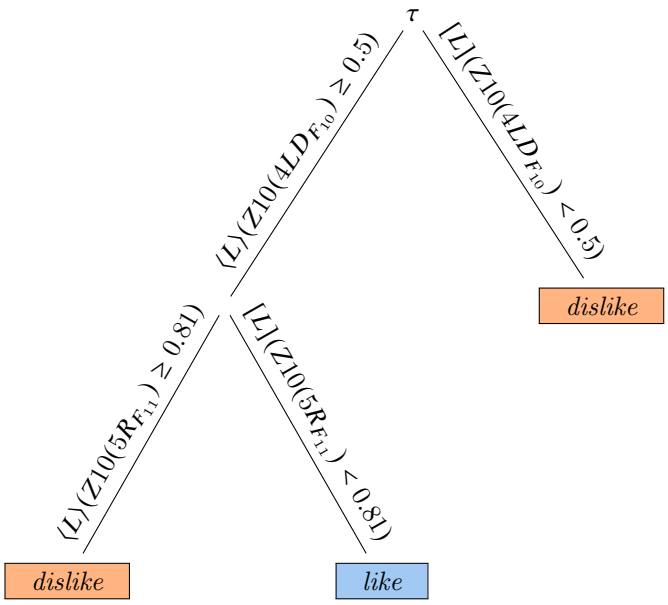


FIGURE 9.3: An HS-tree extracted with the best 5 measures.

almost 9 dislikes out of 10 while retaining a sensitivity of 66%. We can also observe how the difference between $\mathcal{D}_{25}$ and $\mathcal{D}_{34}$ tends to be small, which may suggest how the subjects are unclear in their own judgments when the judgment itself is not extreme. Perhaps unexpectedly, using only one measure instead of five has a positive effect on the performances of the models, as it can be seen in Tab. 9.2. While all the above considerations remain valid, one observes that the models extracted from $\mathcal{D}_{41}$, using 10 electrodes, show very good performances across all frequency bands, and in the case of $\beta + \gamma$ it reaches an balanced accuracy of 81%, with a good balance between specificity (86%) and sensitivity (75%). As we have seen, only a very few works in the recent literature addressed the problem of identifying the areas of the brain, and the frequencies, in which a certain experience seems to manifest itself; in most cases the electrodes are chosen on the basis of the previous neurophysiological literature, and the signals are separated in well-known bands, as opposed to the presented approach in which important electrodes and measures are identified from the data, and frequencies are explored with finer granularity. Concerning the accuracy with which the liking experience can be recognized by a model, it is, again, very hard to compare the results of very different experiments; it could be said, however, that these results (up to 81% of balanced accuracy, as we have seen) are very much comparable with the accuracies that emerges across the literature that mentions addressing liking directly (that is, 60%−86%), with two important *caveat*s: these experiment was carried on in a real-world environment, and the trained models are interpretable.

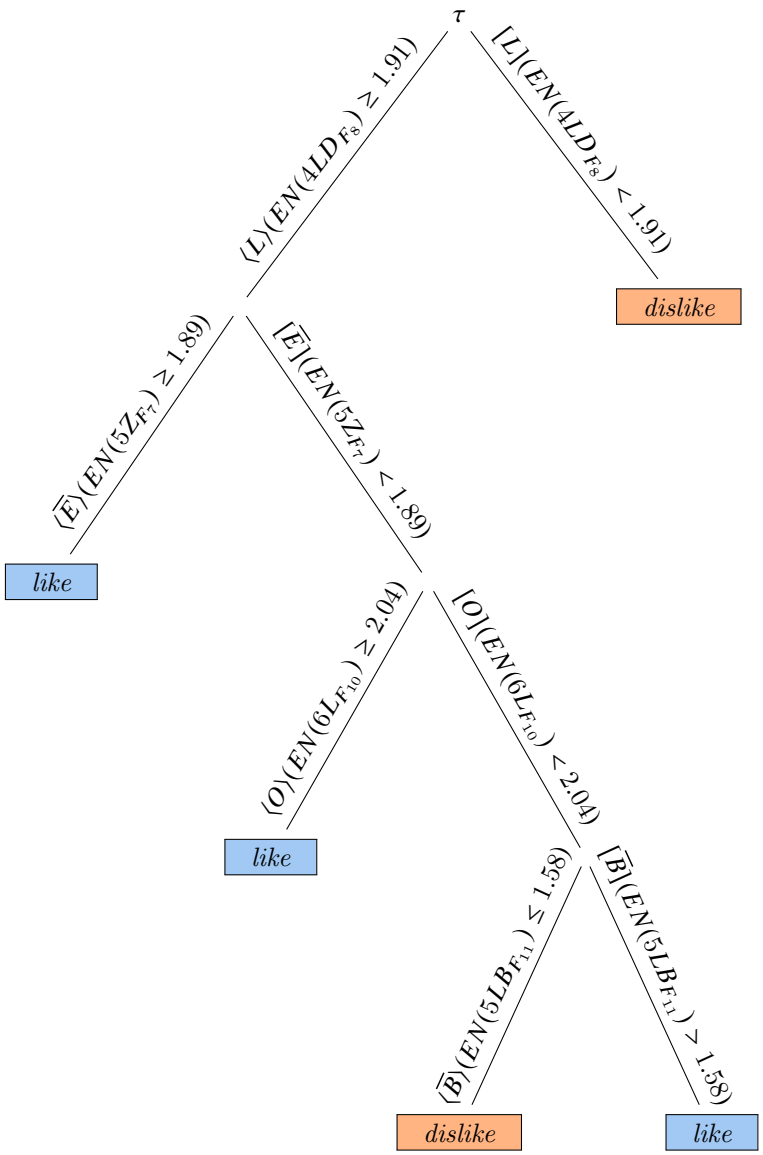


FIGURE 9.4: An HS-tree extracted with the single best measure.

§9.5 Post-hoc Interpretation

Building on the same ideas from previous chapters, we can now extract explicit trees from the models obtained, and discuss them from the point of view of working toward understanding how the EEG signal behaves in subjects in our case. The HS-tree shown in Fig. 9.3 has been learned from $\mathcal{D}_{41}$, using 10 electrodes, the bands in the γ spectrum only, and the best 5 measures; its random forest counterpart, as per Tab. 9.1, shows 72% of accuracy, on average. This particular tree has, however, 100% validation accuracy, and it reaches such a performance with only five nodes, of which, three are leaves. From it, one can easily extract two rules:

$$\begin{aligned} \mathfrak{r}_1: \quad & \langle L\rangle(Z10(4LD_{F_{10}}) \geq 0.5 \wedge [L](Z10(5R_{F_{11}}) < 0.81) \quad \rightarrowtail \quad \textit{like}, \\ \mathfrak{r}_2: \quad & [L](Z10(4LD_{F_{10}}) < 0.5 \vee \langle L\rangle(Z10(5R_{F_{11}}) \geq 0.81) \quad \rightarrowtail \quad \textit{dislike}. \end{aligned}$$

Although interpreting such a rule system from a neurobiological point of view may be misleading, one can observe that, as per this model, the sensation of like is witnessed by the presence of an interval of time (sometimes during the observation) in which the most frequent value of amplitude of voltage at the frequency F_{10}, having discretized the possible values in 10 bins, is greater than 0.5 in the electrode $4LD$, while it is never the case that such value is greater than 0.81 in the electrode $5R$ at the frequency F_{11}. In the same way the model seems to suggest that if the most frequent value of amplitude of voltage at the frequency F_{10} in the electrode $4LD$ is always less than 0.5 or it reaches a peak greater than 0.81 in the electrode $5R$ at the frequency F_{11} sometimes during the observation, the subject should be experimenting a feeling of dislike. As one may see, explicit models such as those extracted by K-trees and forests are qualitative in temporal terms (for example there is no indication on the length of such spikes) but quantitative in event terms. Such an analysis is emblematic of the difference between functional and symbolic models; in the former case, the statistical value is the only indication of the performance of a model, while in the latter case models can be inspected. Even in an ideal situation such as the one under analysis, with 100% validation accuracy, however, precise rules such as the above ones should be taken with precaution, and the experiments should be repeated in different conditions. For comparison, a similar exercise can be done for models extracted using only the best measure. In particular, in Fig. 9.4, we show a tree learned, again, from $\mathcal{D}_{41}$ using 10 electrodes; this time, however, we have chosen a tree learned using bands in the $\beta + \gamma$ spectrum, to be compared, therefore, with a random forest model with 81% of accuracy. The tree alone shows, once more, 100% validation accuracy, but, this time, patterns are temporally more complex; the most important electrode is still $4LD$, over which the first decision is taken, while the other are, $5Z$, $6L$, and $5LB$.

It should be noticed, finally, that using symbolic models such as K-trees brings computational advantages. As a matter of fact, after the (offline) learning phase, models are, as shown above, simply sets of rules. As such, they can be implemented in a very simple way. All preprocessing steps, including Fourier transforms, can be efficiently implemented, which contributes to obtain a potentially online tool that provides a real-time, nearly-instantaneous classification. Our post-hoc analysis, specifically, the after-shuffling model learning, shows that, as expected, our models seem to be robust against potential noises of a real-world environment, such as eye movements and neck muscle contractions.

CHAPTER 10

CONCLUSION & FUTURE DIRECTIONS

Even the darkest night will end and the sun will rise.

Victor Hugo, Les Misérables

The contribution of this thesis is manifold. First, it highlights the importance of interpretable modelling, proposing Symbolic Learning as a possibly valuable alternative to black-box learning. After ponting out the need for both an abstract general framework and a programming framework for symbolic learning, it presents SOLE, the first effort to unify data-driven symbolic methods for AI, and a first implementation in the Julia programming framework. Three symbolic methods covering three steps of a standard AI workflow are, then, introduced: from a feature selection algorithm, to a model learning algorithm, up to an algorithm for post-hoc knowledge extraciton from a learned model. All three methods are devised as based on an underlying modal logical formalism, which highlights the suitability of these formalisms for AI purposes. Ultimately, the thesis introduces the reader to the potential of Modal Symbolic Learning techniques, by presenting the results achieved by one of these methods on four different applicative domains. In each of the four study case, significant effort was made for extracting relevant and human-understandable representations of the data involved. While this might seem like a drawback, it is, in fact, in line with the undesirability of delegating all steps of an AI workflow (from data processing, to learning, and up to automation of a given task) to an unsound, black-box technology.

As AI evolves from an emerging technology to a limitless and established practice, the aim of this work is to promote interest, research and implementation of symbolic methods. A few directions emerge as future steps of this work:

- New algorithms for learning modal symbolic models from non-tabular data can now easily be devised and prototyped in Sole.jl. This may involve extending existing propositional algorithms for decision tree learning, decision list learning, and association rule mining (as first proposed in [Sta+22]). Of particular interest is the possibility of learning neuro-symbolic hybrids, combining the strengths of both symbolic and black-box learning technologies (as proposed in Pagliarini et al. [Pag+22]);
- Post-hoc analysis of learned models opens up a wide range of possibilities that are yet to uncover. For example, algorithms for simplifying and/or approximating (modal) symbolic models can draw from the literature of formula minimization (or formula minimization modulo theory), as it was first investigated in [Pag+23];

- The extension from crisp (Boolean) logic to many-valued logics (e.g., fuzzy or many-expert), where the truth of logical statements can be declined via different shades that go beyond simple truth or falsehood; is likely to provide a sound methodology for dealing with uncertain, unclear or missing information.

Bibliography

[Agg+15] Charu C Aggarwal et al. *Data mining: the textbook*. Vol. 1. Springer, 2015.

[AS94] R. Agrawal and R. Srikant. "Fast Algorithms for Mining Association Rules in Large Databases". In: *Proceedings of 20th International Conference on Very Large Data Bases (VLDB)*. 1994, pp. 487–499.

[AIS93] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. "Mining Association Rules between Sets of Items in Large Databases". In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, May 26-28, 1993*. Ed. by Peter Buneman and Sushil Jajodia. ACM Press, 1993, pp. 207–216. URL: https://doi.org/10.1145/170035.170072.

[Agr+96] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, A Inkeri Verkamo, et al. "Fast discovery of association rules." In: *Advances in knowledge discovery and data mining* 12.1 (1996), pp. 307–328.

[AKH17] Muhammad Ahmad, Adil Mehmood Khan, and Rasheed Hussain. "Graph-based spatial-spectral feature learning for hyperspectral image classification". In: *IET Image Processing* 11.12 (2017), pp. 1310–1316.

[Ala+21] S. Alaniz, D. Marcos, B. Schiele, and Z. Akata. "Learning Decision Trees Recurrently Through Communication". In: *Proceedings of the 34th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 13518–13527.

[AR00] Juan M Ale and Gustavo H Rossi. "An approach to discovering temporal association rules". In: *Proceedings of the 2000 ACM Symposium on Applied computing-Volume 1*. 2000, pp. 294–300.

[AFEK17] S. Alhagry, A.A. Fahmy, and R. El-Khoribi. "Emotion Recognition based on EEG using LSTM Recurrent Neural Network". In: *International Journal of Advanced Computer Science and Applications* 8 (2017), pp. 355–358.

[AK22] Mohanad Alkhodari and Ahsan H. Khandoker. "Detection of COVID-19 in smartphone-based breathing recordings: a pre-screening deep learning tool". In: *PLOS ONE* 17.1 (Jan. 2022), pp. 1–25.

[All83] J. F. Allen. "Maintaining Knowledge about Temporal Intervals". In: *Communication of the ACM* 26.11 (1983), pp. 832–843.

[ARR22] Mahmoud Aly, Kamel H. Rahouma, and Safwat M. Ramzy. “Pay attention to the speech: COVID-19 diagnosis using machine learning and crowdsourced respiratory and speech recordings”. In: *Alexandria Engineering Journal* 61.5 (2022), pp. 3487–3500.

[AK16] M. Amozeghar and K. Khorasani. “An ensemble of dynamic neural network identifiers for fault detection and isolation of gas turbine engines”. In: *Neural Networks* 76 (Jan. 2016), pp. 106 –121.

[Ang+17] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo I. Seltzer, and Cynthia Rudin. “Learning Certifiably Optimal Rule Lists for Categorical Data”. In: *J. Mach. Learn. Res.* 18 (2017), 234:1–234:78. URL: https://jmlr.org/papers/v18/17-716.html.

[AZZ98] B. Apolloni, G. Zamponi, and A. M. Zanaboni. “Learning fuzzy decision trees”. In: *Neural Networks* 11.5 (1998), pp. 885–895.

[ALL19] N. Audebert, B. Le Saux, and S. Lefèvre. “Deep Learning for Classification of Hyperspectral Data: A Comparative Review”. In: *IEEE Geoscience and Remote Sensing Magazine* 7.2 (2019), pp. 159–173.

[ARB19] Mohamed Azmi, George C Runger, and Abdelaziz Berrado. “Interpretable regularized class association rules algorithm for classification in a categorical data space”. In: *Information Sciences* 483 (2019), pp. 313–331.

[Bab+21] Boris Babic, Sara Gerke, Theodoros Evgeniou, and I. Glenn Cohen. “Beware explanations from AI in health care”. In: *Science* 373.6552 (2021), pp. 284–286. URL: https://www.science.org/doi/abs/10.1126/science.abg1834.

[Bai+20] M. Bai, J. Liu, J. Chai, X. Zhao, and D. Yu. “Anomaly detection of gas turbines based on normal pattern extraction”. In: *Applied Thermal Engineering* 166 (2020), p. 114664.

[BCF98] P. Balbiani, J.F. Condotta, and L. Fariñas del Cerro. “A Model for Reasoning about Bidimensional Temporal Relations”. In: *Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning (KR’98)*. Morgan Kaufmann, 1998, pp. 124–130.

[BPK20] V. Bansal, G. Pahwa, and N. Kannan. “Cough Classification for COVID-19 based on audio MFCC features using Convolutional Neural Networks”. In: *Proceedings of the 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*. 2020, pp. 604–608.

[BBS14] Ezio Bartocci, Luca Bortolussi, and Guido Sanguinetti. “Data-driven statistical learning of temporal logic properties”. In: *International conference on formal modeling and analysis of timed systems*. Springer. 2014, pp. 23–37.

[Bar+22] Ezio Bartocci, Cristinel Mateis, Eleonora Nesterini, and Dejan Nickovic. "Survey on mining signal temporal logic specifications". In: *Inf. Comput.* 289.Part (2022), p. 104957. URL: https://doi.org/10.1016/j.ic.2022.104957.

[Bec+23] G. Bechini, E. Losi, L. Manservigi, G. Pagliarini, G. Sciavicco, I. E. Stan, and M. Venturini. "Statistical Rule Extraction for Gas Turbine Trip Prediction". In: *Journal of Engineering for Gas Turbines and Power* 145.5 (2023).

[BDS14] M. Belgiu, L. Drăguţ, and J. Strobl. "Quantitative evaluation of variations in rule-based classifications of land cover in urban neighbourhoods using WorldView-2 imagery". In: *The ISPRS Journal of Photogrammetry and Remote Sensing* 87 (2014), pp. 205–215.

[Bel56] W. A. Belson. "A Technique for Studying the Effects of Television Broadcast". In: *Journal of the Royal Statistical Society* 5.3 (1956), pp. 195–202.

[Bén+21] Clément Bénard, Gérard Biau, Sébastien Da Veiga, and Erwan Scornet. "Interpretable Random Forests via Rule Extraction". In: *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*. Ed. by Arindam Banerjee and Kenji Fukumizu. Vol. 130. Proceedings of Machine Learning Research. PMLR, 2021, pp. 937–945. URL: https://proceedings.mlr.press/v130/benard21a.html.

[BCM16] Alessio Benavoli, Giorgio Corani, and Francesca Mangili. "Should we really use post-hoc tests based on mean-ranks?" In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 152–161.

[Ber+18] T. Berhane, C. Lane, Q. Wu, B. Autrey, O. Anenkhonov, V. Chepinoga, and H. Liu. "Decision-Tree, Rule-Based, and Random Forest Classification of High-Resolution Multispectral Imagery for Wetland Mapping and Inventory". In: *Remote Sensing* 10.580 (2018), 1–26.

[BW03] Kent Berridge and Piotr Winkielman. "What is an unconscious emotion? (The case for unconscious "liking")". In: *Cognition and Emotion* 17.2 (2003), pp. 181–211.

[Bha17] R.K. Bhargava. *Technical dictionary on the gas turbine technology*. Innovative Turbomachinery Technologies Corp., 2017.

[Bie09] Meghyn Bienvenu. "Prime implicates and prime implicants: From propositional to modal logic". In: *Journal of Artificial Intelligence Research* 36 (2009), pp. 71–128.

[Bis+16] Bernd Bischl, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones. "mlr: Machine Learning in R". In: *Journal of Machine Learning Research* 17.170 (2016), pp. 1–5. URL: https://jmlr.org/papers/v17/15-066.html.

[Bla+19] Anthony Blaom, Franz Kiraly, Thibaut Lienart, and Sebastian Vollmer. *MLJ: A pure Julia machine learning framework*. Alan Turing Institute. 2019. URL: https://doi.org/10.5281/zenodo.3541505.

[BD98] H. Blockeel and L. De Raedt. "Top-Down Induction of First-Order Logical Decision Trees". In: *Artificial Intelligence* 101.1-2 (1998), pp. 285–297.

[BDR96] Hendrik Blockeel and Luc De Raedt. "Relational knowledge discovery in databases". In: *International Conference on Inductive Logic Programming*. Springer. 1996, pp. 199–211.

[BB21] Giuseppe Bombara and Calin Belta. "Offline and online learning of signal temporal logic formulae using decision trees". In: *ACM Transactions on Cyber-Physical Systems* 5.3 (2021), pp. 1–23.

[Bom+16] Giuseppe Bombara, Cristian-Ioan Vasile, Francisco Penedo, Hirotoshi Yasuoka, and Calin Belta. "A decision tree approach to data classification using signal temporal logic". In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*. 2016, pp. 1–10.

[Bon+21] Gloria Bonaccorsi, Melchiore Giganti, Maxim Nitsenko, Giovanni Pagliarini, Giacomo Piva, and Guido Sciavicco. "Predicting treatment recommendations in postmenopausal osteoporosis". In: *J. Biomed. Informatics* 118 (2021), p. 103780. URL: https://doi.org/10.1016/j.jbi.2021.103780.

[Bor10] Christian Borgelt. "Simple algorithms for frequent item set mining". In: *Advances in Machine Learning II: Dedicated to the Memory of Professor Ryszard S. Michalski*. Springer, 2010, pp. 351–369.

[BW99] X. Boyen and L. Wehenkel. "Automatic induction of fuzzy decision trees and its application to power system security assessment". In: *Fuzzy Sets and Systems* 102.1 (1999), pp. 3–19.

[Bre96] L. Breiman. "Bagging Predictors". In: *Machine Learning* 24.2 (1996), pp. 123–140.

[Bre01] L. Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32.

[Bre+84] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees*. Wadsworth Publishing Company, 1984.

[Bre91] R. P. Brent. "Fast training algorithms for multilayer neural nets". In: *IEEE Transactions on Neural Networks* 2.3 (1991), pp. 346–354.

[Bre+10] D. Bresolin, P. Sala, D. Della Monica, A. Montanari, and G. Sciavicco. "A Decidable Spatial Generalization of Metric Interval Temporal Logic". In: *Proceedings of the 17th International Symposium on Temporal Representation and Reasoning (TIME 2016)*. 2010, pp. 95–102.

[Bri+97] Sergey Brin, Rajeev Motwani, Jeffrey D Ullman, and Shalom Tsur. "Dynamic itemset counting and implication rules for market basket data". In: *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. 1997, pp. 255–264.

[BNZ11] Björn Bringmann, Siegfried Nijssen, and Albrecht Zimmermann. "Pattern-based classification: A unifying perspective". In: *arXiv preprint arXiv:1111.6191* (2011).

[Bro+20] C. Brown, J. Chauhan, A. Grammenos, J. Han, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo. "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)*. 2020, pp. 3474–3484.

[Bru+23] Andrea Brunello, Dario Della Monica, Angelo Montanari, Nicola Saccomanno, and Andrea Urgolo. "Monitors that learn from failures: Pairing STL and genetic programming". In: *IEEE Access* (2023).

[Bry22] E. Brynjolfsson. "The Turing Trap: The Promise & Peril of Human-Like Artificial Intelligence". In: *Daedalus* 151.2 (2022), pp. 272–287.

[Cao+20] Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. "A comprehensive survey on geometric deep learning". In: *IEEE Access* 8 (2020), pp. 35929–35949.

[Cao+18] X. Cao, F. Zhou, L. Xu, D. Meng, Z. Xu, and J. W. Paisley. "Hyperspectral Image Classification With Markov Random Fields and a Convolutional Neural Network". In: *IEEE Transactions on Image Processing* 27.5 (2018), pp. 2354–2367.

[CC87] C. Carter and J. Catlett. "Assessing Credit Card Applications Using Machine Learning". In: *IEEE Expert* 2.3 (1987), pp. 71–79.

[Cas+21] E. Casanova, A. Candido Jr., R. Corso Fernandes Junior, M. Finger, L. R. Stefanel Gris, M. Antonelli Ponti, and D. Peixoto Pinto da Silva. "Transfer Learning and Data Augmentation Techniques to the COVID-19 Identification Tasks in ComParE 2021". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2021, pp. 446–450.

[Cas+23] E. Caselli, M. Coccagna, A. Gatti, F. Manzella, S. Mazzacane, G. Pagliarini, V.A. Sironi, and G. Sciavicco. "Towards an Objective Theory of Subjective Liking: a First Step in Understanding the Sense of Beauty". In: *Plos ONE* 8.6 (2023), pp. 1–20.

[Cav+23] Patrik Cavina, Federico Manzella, Giovanni Pagliarini, Guido Sciavicco, and Ionel Eduard Stan. "(Un)supervised Univariate Feature Extraction and Selection for Dimensional Data". In: *Proceedings of the 2nd Italian Conference on Big Data and Data Science (ITADATA)*. Vol. 3606. CEUR Workshop Proceedings. CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3606/paper51.pdf.

[CR08] Alain Celisse and Stéphane Robin. “Nonparametric density estimation by exact leave-p-out cross-validation”. In: *Comput. Stat. Data Anal.* 52.5 (2008), pp. 2350–2368. URL: https://doi.org/10.1016/j.csda.2007.10.002.

[Cen87] Jadzia Cendrowska. “PRISM: An Algorithm for Inducing Modular Rules”. In: *Int. J. Man Mach. Stud.* 27.4 (1987), pp. 349–370. URL: https://doi.org/10.1016/S0020-7373(87)80003-2.

[CP77] R. L. P. Chang and T. Pavlidis. “Fuzzy Decision Tree Algorithms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* 7.1 (1977), pp. 28–35.

[Cha+21] Yi Chang, Xin Jing, Zhao Ren, and Björn W. Schuller. “CovNet: A Transfer Learning Framework for Automatic COVID-19 Detection From Crowd-Sourced Cough Sounds”. In: *Frontiers in Digital Health* 3 (2021), p. 799067.

[CA21] Bahzad Charbuty and Adnan Abdulazeez. “Classification Based on Decision Tree Algorithm for Machine Learning”. In: *Journal of Applied Science and Technology Trends* 2.01 (2021), pp. 20 –28. URL: https://jastt.org/index.php/jasttpath/article/view/65.

[CV14] A. Chatterjee and O. Vartanian. “Neuroaesthetics”. In: *Trends in Cognitive Sciences* 18.7 (2014), pp. 370–375.

[Cha+20] G. Chaudhari, X. Jiang, A. Fakhry, A. Han, J. Xiao, S. Shen, and A. Khanzada. “Virufy: Global Applicability of Crowdsourced and Clinical Datasets for AI Detection of COVID-19 from Cough”. In: *CoRR* abs/2011.13320 (2020). arXiv: 2011.13320.

[Che95] Hsinchun Chen. “Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms”. In: *J. Am. Soc. Inf. Sci.* 46.3 (1995), pp. 194–216. URL: https://doi.org/10.1002/(SICI)1097-4571(199504)46:3<194::AID-ASI4>3.0.CO;2-S.

[CG16] T. Chen and C. Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2016, pp. 785–794.

[Che+21] Haitao Cheng, Peng Li, Ruchuan Wang, and He Xu. “Dynamic spatio-temporal logic based on RCC-8”. In: *Concurrency and Computation: Practice and Experience* 33.22 (2021), e5900.

[CY96] Z. Chi and H. Yan. “ID3-derived fuzzy rules and optimized defuzzification for handwritten numeral recognition”. In: *IEEE Transactions on Fuzzy Systems* 4.1 (1996), pp. 24–31.

[CR21] Nuri Cingillioglu and Alessandra Russo. "pix2rule: End-to-end Neuro-symbolic Rule Learning". In: *Proceedings of the 15th International Workshop on Neural-Symbolic Learning and Reasoning as part of the 1st International Joint Conference on Learning & Reasoning (IJCLR 2021), Virtual conference, October 25-27, 2021*. Ed. by Artur S. d'Avila Garcez and Ernesto Jiménez-Ruiz. Vol. 2986. CEUR Workshop Proceedings. CEUR-WS.org, 2021, pp. 15–56. URL: https://ceur-ws.org/Vol-2986/paper3.pdf.

[CS92] K. J. Cios and L. M. Sztandera. "Continuous ID3 algorithm with fuzzy entropy measures". In: *Proceedings of the 1992 IEEE International Conference on Fuzzy Systems*. 1992, pp. 469–476.

[CN89] P. Clark and T. Niblett. "The CN2 Induction Algorithm". In: *Machine Learning* 3 (1989), pp. 261–283.

[CES86] E. M. Clarke, E. A. Emerson, and A. P. Sistla. "Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications". In: *ACM Transactions on Programming Languages and Systems* 8.2 (1986), pp. 244–263.

[Cla+18] E. M. Clarke, O. Grumberg, D. Kroening, D. A. Peled, and H. Veith. *Model Checking*. 2nd. MIT Press, 2018.

[Coc+19] Maddalena Coccagna, Pietro Avanzini, Giovanni Vecchiato, Maddalena FABBRI DESTRO, Vittorio Sironi, Raffaella Folgieri, Anna Banzi, Sante Mazzacane, et al. "Environment and people perceptions: the experience of nevart, neuroestethics of the art vision". In: *Proceedings of Global Challenges in Assistive Technology: Research, Policy & Practice*. 2019, pp. 204 –205.

[Coc+22] Maddalena Coccagna, Federico Manzella, S. Mazzacane, Giovanni Pagliarini, and Guido Sciavicco. "Statistical and Symbolic Neuroaesthetics Rules Extraction from EEG Signals". In: *Artificial Intelligence in Neuroscience: Affective Analysis and Health Applications - 9th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2022, Puerto de la Cruz, Tenerife, Spain, May 31 - June 3, 2022, Proceedings, Part I*. Ed. by José Manuel Ferrández de Vicente, José Ramón Álvarez Sánchez, Félix de la Paz López, and Hojjat Adeli. Vol. 13258. Lecture Notes in Computer Science. Springer, 2022, pp. 536–546. URL: https://doi.org/10.1007/978-3-031-06242-1_53.

[Coc+20] Maddalena Coccagna, Avanzini Pietro, Portera Mariagrazia, Vecchiato Giovanni, Maddalena FABBRI DESTRO, Sironi Vittorio Alessandro, Salvi Fabrizio, Andrea Gatti, Filippo Domenicali, Folgieri Raffaella, et al. "Neuroaesthetics of art vision: an experimental approach to the sense of beauty". In: *Journal of Clinical Trials* 10 (2020), pp. 1–8.

[Coe20] M. Coeckelbergh. *AI Ethics*. MIT Press, 2020.

[Coh60] J. Cohen. "A coefficient of agreement for nominal scales". In: *Educational and psychological measurement* 20.1 (1960), pp. 37–46.

[Coh93] William W. Cohen. “Efficient Pruning Methods for Separate-and-Conquer Rule Learning Systems”. In: *International Joint Conference on Artificial Intelligence*. 1993. URL: https://api.semanticscholar.org/CorpusID:17358040.

[Coh95a] William W. Cohen. “Fast Effective Rule Induction”. In: *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*. Ed. by Armand Prieditis and Stuart Russell. Morgan Kaufmann, 1995, pp. 115–123. URL: https://doi.org/10.1016/b978-1-55860-377-6.50023-2.

[CGK20] Madison Cohen-McFarlane, Rafik Goubran, and Frank Knoefel. “Novel Coronavirus Cough Database: NoCoCoDa”. In: *IEEE Access* 8 (2020), pp. 154087–154094.

[Coh95b] A.G. Cohn. “A Hierarchical Representation of Qualitative Shape based on Connection and Convexity”. In: *Proceedings of th International Conference on Spatial Information Theory: A Theoretical Basis for GIS (COSIT)*. Vol. 988. Lecture Notes in Computer Science. 1995, pp. 311–326.

[CS13] Yann Collette and Patrick Siarry. *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2013.

[Cop+21] Harry Coppock, Alex Gaskell, Panagiotis Tzirakis, Alice Baird, Lyn Jones, and Björn Schuller. “End-to-end convolutional neural network enables COVID-19 detection from breath and cough audio: a pilot study”. In: *BMJ Innovations* 7.2 (2021).

[CS95] M. W. Craven and J. W. Shavlik. “Extracting Tree-Structured Representations of Trained Networks”. In: *Proceedings of the 8th Advances in Neural Information Processing Systems (NIPS)*. 1995, pp. 24–30.

[DMB04] D. Dancey, D. McLean, and Z. Bandar. “Decision Tree Extraction from Trained Neural Networks”. In: *Proceedings of the 7th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*. 2004, pp. 515–519.

[DML21] Rohan Kumar Das, Maulik C. Madhavi, and Haizhou Li. “Diagnosis of COVID-19 Using Auditory Acoustic Cues”. In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2021, pp. 921–925.

[Das+21] T. K. Dash, S. Mishra, G. Panda, and S. C. Satapathy. “Detection of COVID-19 from speech signal using bio-inspired based cepstral features”. In: *Pattern Recognition* 117 (2021), p. 107999.

[Das22] Jeffrey Dastin. “Amazon scraps secret AI recruiting tool that showed bias against women”. In: *Ethics of data and analytics*. Auerbach Publications, 2022, pp. 296–299.

[Dav18] M. Davis. *The Universal Computer: The Road from Leibniz to Turing*. 3rd. CRC Press, Inc., 2018.

[DM80] S. B. Davis and P. Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences". In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 28.4 (1980), pp. 357–366.

[DCF18] M.G. De Giorgi, S. Campilongo, and A. Ficarella. "A diagnostics tool for aero-engines health monitoring using machine learning technique". In: *Energy Procedia* 148 (2018), pp. 860 –867.

[de 13] B. de Ville. "Decision trees". In: *WIREs Computational Statistics* 5.6 (2013), pp. 448–455.

[Deb01] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms.* Wiley-Interscience series in systems and optimization. Wiley, 2001, pp. I–XIX, 1–497.

[Deb+02] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Trans. Evol. Comput.* 6.2 (2002), pp. 182–197. URL: https://doi.org/10.1109/4235.996017.

[Ded+23] Klest Dedja, Felipe Kenji Nakano, Konstantinos Pliakos, and Celine Vens. "BELLATREX: Building Explanations Through a LocaLly AccuraTe Rule EXtractor". In: *IEEE Access* 11 (2023), pp. 41348–41367. URL: https://doi.org/10.1109/ACCESS.2023.3268866.

[DM+22] Dario Della Monica, Giovanni Pagliarini, Guido Sciavicco, and Ionel Eduard Stan. "Decision Trees with a Modal Flavor". In: *Proceedings of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA)*. Ed. by Agostino Dovier, Angelo Montanari, and Andrea Orlandini. Vol. 13796. Lecture Notes in Computer Science. Springer, 2022, pp. 47–59. URL: https://doi.org/10.1007/978-3-031-27181-6_4.

[Dem06] Janez Demšar. "Statistical comparisons of classifiers over multiple data sets". In: *The Journal of Machine learning research* 7.1 (2006), pp. 1–30.

[Dem+13] Janez Demsar, Tomaz Curk, Ales Erjavec, Crtomir Gorup, Tomaz Hocevar, Mitar Milutinovic, Martin Mozina, Matija Polajnar, Marko Toplak, Anze Staric, Miha Stajdohar, Lan Umek, Lan Zagar, Jure Zbontar, Marinka Zitnik, and Blaz Zupan. "Orange: data mining toolbox in python". In: *J. Mach. Learn. Res.* 14.1 (2013), pp. 2349–2353. URL: https://dl.acm.org/doi/10.5555/2567709.2567736.

[Den19] H. Deng. "Interpreting tree ensembles with inTrees". In: *International Journal of Data Science and Analytics* 7.4 (2019), pp. 277–287.

[Den+22] Vincenzo Dentamaro, Paolo Giglio, Donato Impedovo, Luigi Moretti, and Giuseppe Pirlo. "AUCO ResNet: an end-to-end network for COVID-19 pre-screening from cough and breath". In: *Pattern Recognition* 127 (2022), p. 108656.

[DS21] Gauri Deshpande and Björn W. Schuller. "The DiCOVA 2021 Challenge – An Encoder-Decoder Approach for COVID-19 Recognition from Coughing Audio". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2021, pp. 931–935.

[Des+21] V. Despotovic, M. Ismael, M. Cornil, R. M. Call, and G. Fagherazzi. "Detection of COVID-19 from voice, cough and breathing patterns: Dataset and preliminary results". In: *Computers in Biology and Medicine* 138 (2021), p. 104944.

[Dev+11] C Lakshmi Devasena, T Sumathi, VV Gomathi, and M Hemalatha. "Effectiveness evaluation of rule based classifiers for the classification of iris data set". In: *Bonfring International Journal of Man Machine Interface* 1 (2011), p. 5.

[Did90] Edwin Diday. "Knowledge representation and symbolic data analysis". In: *Knowledge, Data and Computer-Assisted Decisions*. Springer, 1990, pp. 17–34.

[Don+99] Guozhu Dong, Xiuzhen Zhang, Limsoon Wong, and Jinyan Li. "CAEP: Classification by aggregating emerging patterns". In: *Discovery Science: Second International Conference, DS'99 Tokyo, Japan, December 6–8, 1999 Proceedings 2*. Springer. 1999, pp. 30–42.

[DF18] Julia Dressel and Hany Farid. "The accuracy, fairness, and limits of predicting recidivism". In: *Science Advances* 4.1 (2018), eaao5580. URL: https://www.science.org/doi/abs/10.1126/sciadv.aao5580.

[DRB01] Kurt Driessens, Jan Ramon, and Hendrik Blockeel. "Speeding Up Relational Reinforcement Learning through the Use of an Incremental First Order Decision Tree Learner". In: *Machine Learning: EMCL 2001, 12th European Conference on Machine Learning, Freiburg, Germany, September 5-7, 2001, Proceedings*. Ed. by Luc De Raedt and Peter A. Flach. Vol. 2167. Lecture Notes in Computer Science. Springer, 2001, pp. 97–108. URL: https://doi.org/10.1007/3-540-44795-4_9.

[Dze06] Saso Dzeroski. "From Inductive Logic Programming to Relational Data Mining". In: *Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15, 2006, Proceedings*. Ed. by Michael Fisher, Wiebe van der Hoek, Boris Konev, and Alexei Lisitsa. Vol. 4160. Lecture Notes in Computer Science. Springer, 2006, pp. 1–14. URL: https://doi.org/10.1007/11853886_1.

[Dze10] Saso Dzeroski. "Relational Data Mining". In: *Data Mining and Knowledge Discovery Handbook, 2nd ed*. Ed. by Oded Maimon and Lior Rokach. Springer, 2010, pp. 887–911. URL: https://doi.org/10.1007/978-0-387-09823-4_46.

[ESM11] M. Egenhofer, J. Sharma, and D. Mark. "A critical comparison of the 4-intersection and 9-intersection models for spatial relations: formal analysis". In: *Proceedings of the 11th International Symposium on Computer-Assisted Cartography (Auto-Carto)-11*. 2011, pp. 1–13.

[EK01] T. Elomaa and M. Kaariainen. “An Analysis of Reduced Error Pruning”. In: *Journal of Artificial Intelligence Research* 15 (2001), pp. 163–187.

[Fak+21] Ahmed Fakhry, Xinyi Jiang, Jaclyn Xiao, Gunvant Chaudhari, and Asriel Han. “A Multi-Branch Deep Learning Network for Automated Detection of COVID-19”. In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2021, pp. 4139–4143.

[Fis88] R. A. Fisher. *Iris*. UCI Machine Learning Repository. 1988. URL: https://doi.org/10.24432/C56C76.

[Fra96] A. Frank. “Qualitative Spatial Reasoning: Cardinal Directions as an Example”. In: *International Journal of Geographical Information Science* 10 (1996), pp. 269–290.

[FW98] Eibe Frank and Ian H. Witten. “Generating Accurate Rule Sets Without Global Optimization”. In: *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*. Ed. by Jude W. Shavlik. Morgan Kaufmann, 1998, pp. 144–151.

[Fra+10] C.A. Frantzidis, C. Bratsas, C.L. Papadelis, E. Konstantinidis, C. Pappas, and P.D. Bamidis. “Toward emotion aware computing: an integrated approach using multichannel neurophysiological recordings and affective visual stimuli”. In: *IEEE transactions on information technology in biomedicine* 14.3 (2010), pp. 589–597.

[Fre92] C. Freksa. “Using orientation information for qualitative spatial reasoning”. In: *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*. Vol. 639. Lecture Notes in Computer Science. Springer, 1992, pp. 162–178.

[FB97] M. A. Friedl and C. E. Brodley. “Decision tree classification of land cover from remotely sensed data”. In: *Remote Sensing of Environment* 61.3 (1997), pp. 399–409.

[Fri01] Jerome H Friedman. “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics* (2001), pp. 1189–1232.

[FF99] Jerome H Friedman and Nicholas I Fisher. “Bump hunting in high-dimensional data”. In: *Statistics and computing* 9.2 (1999), pp. 123–143.

[FP08] Jerome H. Friedman and Bogdan E. Popescu. “Predictive learning via rule ensembles”. In: *The Annals of Applied Statistics* 2.3 (2008), pp. 916–954. URL: https://doi.org/10.1214/07-AOAS148.

[FP+03] Jerome H Friedman, Bogdan E Popescu, et al. “Importance sampled learning ensembles”. In: *Journal of Machine Learning Research* 94305 (2003), pp. 1–32.

[Fri40] Milton Friedman. “A comparison of alternative tests of significance for the problem of m rankings”. In: *The Annals of Mathematical Statistics* 11.1 (1940), pp. 86–92.

[FGL12] Johannes Fürnkranz, Dragan Gamberger, and Nada Lavrac. *Foundations of Rule Learning*. Cognitive Technologies. Springer, 2012. URL: https://doi.org/10.1007/978-3-540-75197-7.

[FW94] Johannes Fürnkranz and Gerhard Widmer. "Incremental Reduced Error Pruning". In: *Machine Learning, Proceedings of the Eleventh International Conference, Rutgers University, New Brunswick, NJ, USA, July 10-13, 1994*. Ed. by William W. Cohen and Haym Hirsh. Morgan Kaufmann, 1994, pp. 70–77. URL: https://doi.org/10.1016/b978-1-55860-335-6.50017-9.

[GC95] Brian R. Gaines and Paul Compton. "Induction of Ripple-Down Rules Applied to Modeling Large Databases". In: *J. Intell. Inf. Syst.* 5.3 (1995), pp. 211–228. URL: https://doi.org/10.1007/BF00962234.

[GL96] Dragan Gamberger and Nada Lavrac. "Noise Detection and Elimination Applied to Noise Handling in a KRK Chess Endgame". In: *Inductive Logic Programming, 6th International Workshop, ILP-96, Stockholm, Sweden, August 26-28, 1996, Selected Papers*. Ed. by Stephen H. Muggleton. Vol. 1314. Lecture Notes in Computer Science. Springer, 1996, pp. 72–88. URL: https://doi.org/10.1007/3-540-63494-0_49.

[GH15] A. Gandomi and M. Haider. "Beyond the hype: Big data concepts, methods, and analytics". In: *International Journal of Information Management* 35.2 (2015), pp. 137–144.

[Gar+19] A. Garg, A. Kapoor, A. Bedi, and R. Sunkaria. "Merged LSTM Model for emotion classification using EEG signals". In: *Proceedings of the 5th International Conference on Data Science and Engineering*. 2019, pp. 139–143.

[Ge+17] Jiaqi Ge, Yuni Xia, Jian Wang, Chandima Hewa Nadungodage, and Sunil Prabhakar. "Sequential pattern mining in databases with temporal uncertainty". In: *Knowl. Inf. Syst.* 51.3 (2017), pp. 821–850. URL: https://doi.org/10.1007/s10115-016-0977-1.

[GH06] Liqiang Geng and Howard J. Hamilton. "Interestingness measures for data mining: A survey". In: *ACM Comput. Surv.* (2006).

[GT19] Seyed Mohssen Ghafari and Christos Tjortjis. "A survey on association rules mining using heuristics". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 9.4 (2019), e1307.

[Ghi+23] Michele Ghiotti, Federico Manzella, Giovanni Pagliarini, Guido Sciavicco, and Ionel Eduard Stan. "Evolutionary Explainable Rule Extraction from (Modal) Random Forests". In: *ECAI 2023 - 26th European Conference on Artificial Intelligence, September 30 - October 4, 2023, Kraków, Poland - Including 12th Conference on Prestigious Applications of Intelligent Systems (PAIS 2023)*. Ed. by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu. Vol. 372. Frontiers in Artificial Intelligence and Applications. IOS Press, 2023, pp. 827–834. URL: https://doi.org/10.3233/FAIA230350.

[Gin21] Corrado Gini. "Measurement of inequality of incomes". In: *The economic journal* 31.121 (1921), pp. 124–125.

[Gla+22] Claire Glanois, Zhaohui Jiang, Xuening Feng, Paul Weng, Matthieu Zimmer, Dong Li, Wulong Liu, and Jianye Hao. "Neuro-Symbolic Hierarchical Rule Induction". In: *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 7583–7615. URL: https://proceedings.mlr.press/v162/glanois22a.html.

[GPW21] M. Gnewuch, H. Pasing, and C. Weiß. "A generalized Faulhaber inequality, improved bracketing covers, and applications to discrepancy". In: *Mathematics of Computation* 90.332 (2021), pp. 2873–2898.

[Goe+03] P.K. Goel, S.O. Prasher, R.M. Patel, J.A. Landry, R.B. Bonnell, and A.A. Viau. "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 (2003), pp. 67–93.

[GOV22] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. "Why do tree-based models still outperform deep learning on typical tabular data?" In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*. Ed. by Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh. 2022.

[Gun+19] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. "XAI - Explainable artificial intelligence". In: *Science Robotics* 4.37 (2019).

[GG92a] H. Guo and S. B. Gelfand. "Classification trees with neural network feature extraction". In: *IEEE Transactions on Neural Networks* 3.6 (1992), pp. 923–933.

[GG92b] H. Guo and S. B. Gelfand. "Classification Trees with Neural Network Feature Extraction". In: *IEEE Transactions on Neural Networks* 3.6 (1992), pp. 923–933.

[Gwi+10] J. T. Gwin, K. Gramann, S. Makeig, and D. P. Ferris. "Removal of Movement Artifact from High-Density EEG Recorded During Walking and Running". In: *Journal of Neurophysiology* 103.6 (2010), pp. 3526–3534.

[HH12] S.K. Hadjidimitriou and L.J. Hadjileontiadis. "Toward an EEG-based recognition of music liking using time-frequency analysis". In: *IEEE Transactions on Biomedical Engineering* 59.12 (2012), pp. 3498–3510.

[Hah06] Michael Hahsler. "A Model-Based Frequency Constraint for Mining Associations from Transaction Data". In: *Data Min. Knowl. Discov.* 13.2 (2006), pp. 137–166. URL: https://doi.org/10.1007/s10618-005-0026-2.

[HGH05] Michael Hahsler, Bettina Grün, and Kurt Hornik. "arules-A computational environment for mining association rules and frequent item sets". In: *Journal of statistical software* 14.15 (2005), pp. 1–25.

[Hah+19] Michael Hahsler, Ian Johnson, Tomáš Kliegr, and Jaroslav Kuchař. "Associative Classification in R: arc, arulesCBA, and rCBA". In: *The R Journal* 11.2 (2019), pp. 254–267. URL: https://doi.org/10.32614/RJ-2019-048.

[Háj13] P. Hájek. *Metamathematics of fuzzy logic*. Springer, 2013.

[Hal+09] Mark A. Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update". In: *SIGKDD Explor.* 11.1 (2009), pp. 10–18. URL: https://doi.org/10.1145/1656274.1656278.

[HS91a] J. Y. Halpern and Y. Shoham. "A Propositional Modal Logic of Time Intervals". In: *Journal of the ACM* 38.4 (1991), pp. 935–962.

[HS91b] J.Y. Halpern and Y. Shoham. "A Propositional Modal Logic of Time Intervals". In: 38.4 (1991), pp. 935–962.

[Han+21] J. Han, C. Brown, J. Chauhan, A. Grammenos, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, and C. Mascolo. "Exploring Automatic COVID-19 Diagnosis via Voice and Symptoms from Crowdsourced Data". In: *Proc of. the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021, pp. 8328–8332.

[HPY00] Jiawei Han, Jian Pei, and Yiwen Yin. "Mining Frequent Patterns without Candidate Generation". In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA*. Ed. by Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein. ACM, 2000, pp. 1–12. URL: https://doi.org/10.1145/342009.335372.

[Han+22] Jing Han, Tong Xia, Dimitris Spathis, Erika Bondareva, Chloë Brown, Jagmohan Chauhan, Ting Dang, Andreas Grammenos, Apinan Hasthanasombat, Andres Floto, et al. "Sounds of COVID-19: exploring realistic performance of audio-based digital testing". In: *NPJ Digital Medicine* 5.1 (2022), pp. 1–9.

[HSA20] A. Hassan, I. Shahin, and M. B. Alsabek. "COVID-19 Detection System using Recurrent Neural Networks". In: *Proceedings of the 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI)*. 2020, pp. 1–5.

[HTF09] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. Springer Series in Statistics. Springer, 2009.

[HP15] Ioannis Hatzilygeroudis and Jim Prentzas. "Symbolic-neural rule based reasoning and explanation". In: *Expert Syst. Appl.* 42.9 (2015), pp. 4595–4609. URL: https://doi.org/10.1016/j.eswa.2015.01.068.

[Hay+98] I. Hayashi, T. Maeda, A. Bastian, and L. C. Jain. "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*. 1998, pp. 681–685.

[HVD15] G. Hinton, O. Vinyals, and J. Dean. *Distilling the Knowledge in a Neural Network*. 2015. URL: https://doi.org/10.48550/arXiv.1503.02531.

[Hol79] Sture Holm. "A simple sequentially rejective multiple test procedure". In: *Scandinavian journal of statistics* (1979), pp. 65–70.

[Hol93] Robert C. Holte. "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets". In: *Mach. Learn.* 11 (1993), pp. 63–91. URL: https://doi.org/10.1023/A:1022631118932.

[Hon+22] D. Hong, Z. Han, J. Yao, L. Gao, B. Zhang, A. Plaza, and J. Chanussot. "SpectralFormer: Rethinking Hyperspectral Image Classification With Transformers". In: *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), pp. 1–15.

[Hu+12] Po Hu, Minlie Huang, Peng Xu, Weichang Li, Adam K Usadi, and Xiaoyan Zhu. "Finding nuggets in IP portfolios: core patent mining through textual temporal analysis". In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2012, pp. 1819–1823.

[Hu+15] W. Hu, Y. Huang, W. Li, F. Zhang, and H.-C. Li. "Deep Convolutional Neural Networks for Hyperspectral Image Classification". In: *Journal of Sensors* 2015 (2015), 258619:1–258619:12.

[HH09] Jens Christian Hühn and Eyke Hüllermeier. "FURIA: an algorithm for unordered fuzzy rule induction". In: *Data Min. Knowl. Discov.* 19.3 (2009), pp. 293–319. URL: https://doi.org/10.1007/s10618-009-0131-8.

[HBH23] Rik Huijzer, Frank Blaauw, and Ruud JR den Hartigh. "SIRUS.jl: Interpretable Machine Learning via Rule Extraction". In: *Journal of Open Source Software* 8.90 (2023), p. 5786. URL: https://doi.org/10.21105/joss.05786.

[HSD01] Geoff Hulten, Laurie Spencer, and Pedro M. Domingos. "Mining time-changing data streams". In: *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*. Ed. by Doheon Lee, Mario Schkolnick, Foster J. Provost, and Ramakrishnan Srikant. ACM, 2001, pp. 97–106. URL: https://doi.org/10.1145/502512.502529.

[HSM66] E. B. Hunt, P. J. Stone, and J. Marin. *Experiments in Induction*. Academic Press New York, 1966.

[HR76] L. Hyafil and R. L. Rivest. "Constructing Optimal Binary Decision Trees is NP-Complete". In: *Information Processing Letters* 5.1 (1976), pp. 15–17.

[Ich+96] Hidetomo Ichihashi, T. Shirai, Kazunori Nagasaka, and Tetsuya Miyoshi. "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 (1996), pp. 157–167.

[Imr+20] A. Imran, I. Posokhova, H. N. Qureshi, U. Masood, M. Sajid Riaz, K. Ali, C. N. John, I. Hussain, and M. Nabeel. "AI4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app". In: *Informatics in Medicine Unlocked* 20 (2020), pp. 1–14.

[IK95] I. Ivanova and M. Kubat. "Initialization of neural networks by means of decision trees". In: *Knowledge-Based Systems* 8.6 (1995), pp. 333–344.

[Jan94] J.-S. R. Jang. "Structure determination in fuzzy modeling: a fuzzy CART approach". In: *Proceedings of the 3rd IEEE International Fuzzy Systems Conference*. 1994, pp. 480–485.

[Jan98] C. Z. Janikow. "Fuzzy decision trees: issues and methods". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 28.1 (1998), pp. 1–14.

[JM20] Mikolás Janota and António Morgado. "SAT-Based Encodings for Optimal Decision Trees with Explicit Paths". In: *Theory and Applications of Satisfiability Testing - SAT 2020 - 23rd International Conference, Alghero, Italy, July 3-10, 2020, Proceedings*. Ed. by Luca Pulina and Martina Seidl. Vol. 12178. Lecture Notes in Computer Science. Springer, 2020, pp. 501–518. URL: https://doi.org/10.1007/978-3-030-51825-7_35.

[JJL97] B. Jeng, Y.-M. Jeng, and T.-P. Liang. "FILM: a fuzzy inductive learning method for automated knowledge acquisition". In: *Decision Support Systems* 21.2 (1997), pp. 61–73.

[JPB13] R. Jenke, A. Peer, and M. Buss. "Effect-size-based electrode and feature selection for emotion recognition from EEG". In: *Proceedings of the 25th International Conference on Acoustics, Speech, and Signal Processing*. 2013, pp. 1217–1221.

[JPB14] Robert Jenke, Angelika Peer, and Martin Buss. "Feature extraction and selection for emotion recognition from EEG". In: *IEEE Trans. Affect. Comput.* 5.3 (2014), pp. 327–339. URL: https://doi.org/10.1109/TAFFC.2014.2339834.

[Jia+19] J. Jiang, J. Ma, Z. Wang, C. Chen, and X. Liu. "Hyperspectral Image Classification in the Presence of Noisy Labels". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.2 (2019), pp. 851–865.

[Jia+12] Z. Jiang, S. Shekhar, P. Mohan, J. Knight, and J. Corcoran. "Learning spatial decision tree for geographical classification: A summary of results". In: *Proceedings of the 12th International Conference on Advances in Geographic Information Systems*. ACM, 2012, pp. 390–393.

[Jor94] Michael I. Jordan. “A Statistical Approach to Decision Tree Modeling”. In: *Proceedings of the 11th International Conference on Machine Learning*. 1994, pp. 363–370.

[Jul+12] Andreea Julea, Nicolas Méger, Christophe Rigotti, Emmanuel Trouvé, Romain Jolivet, Philippe Bolon, et al. “Efficient Spatio-temporal Mining of Satellite Image Time Series for Agricultural Monitoring.” In: *Trans. Mach. Learn. Data Min.* 5.1 (2012), pp. 23–44.

[KMD95] B. Kartikeyan, K.L. Majumder, and A.R. Dasgupta. “An expert system for land cover classification”. In: *IEEE Transactions on Geoscience and Remote Sensing* 33.1 (1995), pp. 58–66.

[Kas80] G. V. Kass. “An Exploratory Technique for Investigating Large Quantities of Categorical Data”. In: *Journal of the Royal Statistical Society* 29.2 (1980), pp. 119–127.

[Ke+17] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Proceedings of the 31st Advances in Neural Information Processing Systems (NIPS)*. 2017, pp. 3146–3154.

[KV94] M. J. Kearns and L. G. Valiant. “Cryptographic Limitations on Learning Boolean Formulae and Finite Automata”. In: *Journal of the ACM* 41.1 (1994), pp. 67–95.

[KQA10] R. Khosrowabadi, C. Quek, and K. Ang. “EEG-based Emotion Recognition Using Self-Organizing Map for Boundary Detection”. In: *Proceedings of the 20th International Conference on Pattern Recognition*. 2010, pp. 4242–4245.

[KR10] R. Khosrowabadi and A.W. Abdul Rahman. “Classification of EEG correlates on emotion using features from Gaussian mixtures of EEG spectrogram”. In: *Proceedings of the 3rd International Conference on Information and Communication Technology for the Muslim World*. 2010, pp. 1–6.

[KI23] Tomáš Kliegr and Ebroul Izquierdo. “QCBA: improving rule classifiers learned from quantitative data by recovering information lost by discretisation”. In: *Applied Intelligence* (2023), pp. 1–31.

[Koh95] Ron Kohavi. “The power of decision tables”. In: *European conference on machine learning*. Springer. 1995, pp. 174–189.

[Kon84] Igor Kononenko. “Experiments in automatic learning of medical diagnostic rules”. In: *Technical report, Jozef Stefan Institute* (1984).

[Kon+15] P. Kontschieder, M. Fiterau, A. Criminisi, and S. Rota Bulò. “Deep Neural Decision Forests”. In: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1467–1475.

[KSV96] J. Korpáš, J. Sadloňová, and .M Vrabec. “Analysis of the cough sound: an overview”. In: *Pulmonary pharmacology* 9.5-6 (1996), pp. 261–268.

[Kot13] Sotiris B. Kotsiantis. "Decision trees: a recent overview". In: *Artif. Intell. Rev.* 39.4 (2013), pp. 261–283. URL: https://doi.org/10.1007/s10462-011-9272-4.

[KW01] Stefan Kramer and Gerhard Widmer. "Inducing Classification and Regression Trees in First Order Logic". In: *Relational Data Mining*. Ed. by Sašo Džeroski and Nada Lavrač. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 140–159. URL: https://doi.org/10.1007/978-3-662-04599-2_6.

[Kri63] S. A. Kripke. "Semantical Analysis of Modal Logic I. Normal Propositional Calculi". In: *Zeitschrift fur mathematische Logik und Grundlagen der Mathematik* 9.5-6 (1963), pp. 67–96.

[KSB99] R. Krishnan, G. Sivakumar, and P. Bhattacharya. "Extracting decision trees from trained neural networks". In: *Pattern Recognition* 32.12 (1999), pp. 1999–2009.

[Kub98] M. Kubat. "Decision Trees Can Initialize Radial-Basis Function Networks". In: *IEEE Transactions on Neural Networks* 9.5 (1998), pp. 813–821.

[Kuc+20] M. Kucharczyk, Geoffrey J. Hay, S. Ghaffarian, and C. H. Hugenholtz. "Geographic Object-Based Image Analysis: A Primer and Future Directions". In: *Remote Sensing* 12.12 (2020). URL: https://doi.org/10.3390/rs12122012.

[KJ13] M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer, 2013.

[Kuh+13] Max Kuhn, Kjell Johnson, Max Kuhn, and Kjell Johnson. "Regression trees and rule-based models". In: *Applied predictive modeling* (2013), pp. 173–220.

[KS17] A. Kulkarni and A. Shrestha. "Multispectral Image Analysis using Decision Trees". In: *International Journal of Advanced Computer Science and Applications* 8.6 (2017), 11–18.

[Kwo20] Tom Kwong. *Hands-On Design Patterns and Best Practices with Julia: Proven solutions to common problems in software design for Julia 1. x*. Packt Publishing Ltd, 2020.

[LHS20] J. Laguarta, F. Hueto, and B. Subirana. "COVID-19 Artificial Intelligence Diagnosis Using Only Cough Recordings". In: *IEEE Open Journal of Engineering in Medicine and Biology* 1 (2020), pp. 275–281.

[LHT14] Guo-Cheng Lan, Tzung-Pei Hong, and Vincent S Tseng. "An efficient projection-based indexing approach for mining high utility itemsets". In: *Knowledge and information systems* 38 (2014), pp. 85–107.

[LHF05] Niels Landwehr, Mark A. Hall, and Eibe Frank. "Logistic Model Trees". In: *Mach. Learn.* 59.1-2 (2005), pp. 161–205. URL: https://doi.org/10.1007/s10994-005-0466-3.

[LK17] H. Lee and H. Kwon. "Going Deeper With Contextual CNN for Hyperspectral Image Classification". In: *IEEE Transactions on Image Processing* 26.10 (2017), pp. 4843–4855.

[Len95] Douglas B. Lenat. "CYC: A Large-Scale Investment in Knowledge Infrastructure". In: *Commun. ACM* 38.11 (1995), pp. 32–38. URL: https://doi.org/10.1145/219717.219745.

[Lew18] C. I. Lewis. *A Survey of Symbolic Logic*. University of California Press, 1918.

[LL09] Mu Li and Bao-Liang Lu. "Emotion classification based on gamma-band EEG". In: *Proceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2009, pp. 1223–1226.

[LFJ92] T. Li, L. Fang, and A. Jennings. "Structurally Adaptive Self-Organizing Neural Trees". In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 1992, pp. 329–334.

[LHP01] Wenmin Li, Jiawei Han, and Jian Pei. "CMAR: Accurate and efficient classification based on multiple class-association rules". In: *Proceedings 2001 IEEE international conference on data mining*. IEEE. 2001, pp. 369–376.

[LZS17] Y. Li, H. Zhang, and Q. Shen. "Spectral-Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network". In: *Remote Sensing* 9.1 (2017), p. 67.

[LHM98] Bing Liu, Wynne Hsu, and Yiming Ma. "Integrating classification and association rule mining". In: *Proceedings of the fourth international conference on knowledge discovery and data mining*. 1998, pp. 80–86.

[LMW00] Bing Liu, Yiming Ma, and Ching Kian Wong. "Improving an association rule based classifier". In: *Principles of Data Mining and Knowledge Discovery: 4th European Conference, PKDD 2000 Lyon, France, September 13–16, 2000 Proceedings 4*. Springer. 2000, pp. 504–509.

[LXY00] Bing Liu, Yiyuan Xia, and Philip S. Yu. "Clustering Through Decision Tree Construction". In: *Proceedings of the 2000 ACM CIKM International Conference on Information and Knowledge Management, McLean, VA, USA, November 6-11, 2000*. ACM, 2000, pp. 20–29. URL: https://doi.org/10.1145/354756.354775.

[Liu+21] H. Liu, Y. Zhang, Y. Li, and X. Kong. "Review on Emotion Recognition Based on Electroencephalography". In: *Frontiers on Computational Neuroscience* 15 (2021), pp. 1–15.

[LS13] Y. Liu and O. Sourina. "Real-Time Fractal-Based Valence Level Recognition from EEG". In: *Transactions on Computational Sciences* 18 (2013), pp. 101–120.

[LS14] Y. Liu and O. Sourina. "Real-Time Subject-Dependent EEG-Based Emotion Recognition Algorithm". In: *Transactions on Computational Sciences* 23 (2014), pp. 199–223.

[Loh11] W. Y. Loh. “Classification and regression trees”. In: *WIREs Data Mining and Knowledge Discovery* 1.1 (2011), pp. 14–23.

[Loh14] W. Y. Loh. “Fifty Years of Classification and Regression Trees”. In: *International Statistical Review* 82.3 (2014), pp. 329–348.

[Los+22a] E. Losi, M. Venturini, L. Manservigi, and G. Bechini. “Detection of the Onset of Trip Symptoms Embedded in Gas Turbine Operating Data”. In: *Proceedings of ASME Turbo Expo, Virtual Conference and Exhibition*. ASME Paper GT2022-80666. 2022, pp. 1–15.

[Los+21a] E. Losi, M. Venturini, L. Manservigi, G.F. Ceschini, G. Bechini, G. Cota, and F. Riguzzi. “Data selection and feature engineering for the application of machine learning to the prediction of gas turbine trip”. In: *Proceedings of ASME Turbo Expo, Virtual Conference and Exhibition*. 2021, pp. 1–11.

[Los+21b] E. Losi, M. Venturini, L. Manservigi, G.F. Ceschini, G. Bechini, G. Cota, and F. Riguzzi. “Structured methodology for clustering gas turbine transients by means of multivariate time series”. In: *Journal of Engineering for Gas Turbines and Power* 143.3 (2021), pp. 1–13.

[Los+22b] Enzo Losi, Mauro Venturini, Lucrezia Manservigi, Giuseppe Fabio Ceschini, Giovanni Bechini, Giuseppe Cota, and Fabrizio Riguzzi. “Prediction of Gas Turbine Trip: A Novel Methodology Based on Random Forest Models”. In: *Journal of Engineering for Gas Turbines and Power* 144.3 (2022). 031025. URL: https://doi.org/10.1115/1.4053194.

[LJH16] F. Lu, H. Ju, and J. Huang. “An improved extended Kalman filter with inequality constraints for gas turbine engine health monitoring”. In: *Aerospace Science and Technology* 58 (2016), pp. 36–47.

[Lub+19] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones. “catch22: CAnonical Time-series CHaracteristics - Selected through highly comparative time-series analysis”. In: *Data Mining and Knowledge Discovery* 33.6 (2019), pp. 1821–1852.

[Luo+20] Y. Luo, Q. Fu, J. Xie, Y. Qin, G. Wu, J. Liu, F. Jiang, Y. Cao, and X. Ding. “EEG-Based Emotion Classification Using Spiking Neural Networks”. In: *IEEE Access* 8 (2020), pp. 46007–46016.

[LW06] C. Lutz and F. Wolter. “Modal logics of topological relations”. In: *Logical Methods in Computer Science* 2.2 (June 2006), pp. 1–41.

[Mak+15] K. Makantasis, K. Karantzalos, A. D. Doulamis, and N. D. Doulamis. “Deep supervised learning for hyperspectral data classification through convolutional neural networks”. In: *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2015, pp. 4959–4962.

[Man+21] F. Manzella, G. Pagliarini, G. Sciavicco, and I. E. Stan. "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, 2021, 7:1–7:18.

[Man+23a] F. Manzella, G. Pagliarini, G. Sciavicco, and I.E. Stan. "The voice of COVID-19: Breath and cough recording classification with temporal decision trees and random forests". In: *Artifcial Intelligence in Medicine* 137 (2023), p. 102486.

[Man+23b] Federico Manzella, Giovanni Pagliarini, Guido Sciavicco, and Ionel Eduard Stan. "Efficient Modal Decision Trees". In: *AIxIA 2023 - Advances in Artificial Intelligence - XXIInd International Conference of the Italian Association for Artificial Intelligence, AIxIA 2023, Rome, Italy, November 6-9, 2023, Proceedings*. Ed. by Roberto Basili, Domenico Lembo, Carla Limongelli, and Andrea Orlandini. Vol. 14318. Lecture Notes in Computer Science. Springer, 2023, pp. 381–395. URL: https://doi.org/10.1007/978-3-031-47546-7_26.

[MR99] M. Marx and M. Reynolds. "Undecidability of Compass Logic". In: *Journal of Logic and Computation* 9 (1999), pp. 897–914.

[MF00] Hirofumi Matsuzawa and Takeshi Fukuda. "Mining structured association patterns from databases". In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2000, pp. 233–244.

[Mat+13] Stephen G Matthews, Mario A Gongora, Adrian A Hopgood, and Samad Ahmadi. "Web usage mining with evolutionary extraction of temporal fuzzy association rules". In: *Knowledge-Based Systems* 54 (2013), pp. 66–72.

[MMS91] Clayton McMillan, Michael C Mozer, and Paul Smolensky. "Rule induction through integrated symbolic and subsymbolic processing". In: *Advances in neural information processing systems* 4 (1991).

[Mel21] M. Melek. "Diagnosis of COVID-19 and non-COVID-19 patients by classifying only a single cough sound". In: *Neural Computing and Applications* 33.24 (2021), pp. 17621–17632.

[MM72] R. Messenger and L. Mandell. "A Modal Search Technique for Predictive Nominal Scale Multivariate Analysis". In: *Journal of the American Statistical Association* 67.340 (1972), pp. 768–772.

[Mic69] Ryszard S Michalski. "On the quasi-minimal solution of the general covering problem". In: *V International Symposium on Information Processing (FCIP 69)(Switching Circuits)*. Vol. A3. 1969, pp. 125–128.

[Mic80] Ryszard S. Michalski. "Pattern Recognition as Rule-Guided Inductive Inference". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 2.4 (1980), pp. 349–361. URL: https://doi.org/10.1109/TPAMI.1980.4767034.

[Mic83] Ryszard S. Michalski. “A Theory and Methodology of Inductive Learning”. In: *Artif. Intell.* 20.2 (1983), pp. 111–161. URL: https://doi.org/10.1016/0004-3702(83)90016-4.

[Mic+86] Ryszard S. Michalski, Igor Mozetic, Jiarong Hong, and Nada Lavrac. “The Multi-Purpose Incremental Learning System AQ15 and Its Testing Application to Three Medical Domains”. In: *Proceedings of the 5th National Conference on Artificial Intelligence. Philadelphia, PA, USA, August 11-15, 1986. Volume 2: Engineering*. Ed. by Tom Kehler and Stanley J. Rosenschein. Morgan Kaufmann, 1986, pp. 1041–1047. URL: https://www.aaai.org/Library/AAAI/1986/aaai86-171.php.

[Mic+12] C. Micheloni, A. Rani, S. Kumar, and G. L. Foresti. “A balanced neural tree for pattern classification”. In: *Neural Networks* 27 (2012), pp. 81–90.

[Min91] Marvin Minsky. “Logical Versus Analogical or Symbolic Versus Connectionist or Neat Versus Scruffy”. In: *AI Mag.* 12.2 (1991), pp. 34–51. URL: https://doi.org/10.1609/aimag.v12i2.894.

[Mit97] T. M. Mitchell. *Machine learning*. McGraw-Hill, 1997.

[MPR22] Abdelkader Mokkadem, Mariane Pelletier, and Louis Raimbault. “A recursive algorithm for mining association rules”. In: *SN Computer Science* 3.5 (2022), p. 384.

[MPS09] A. Montanari, G. Puppis, and P. Sala. “A Decidable Spatial Logic with Cone-Shaped Cardinal Directions”. In: *Proceedings of the 18th Annual Conference of the European Association for Computer Science Logic (CSL 2009)*. Vol. 5771. Lecture Notes in Computer Science. Springer, 2009, pp. 394–408.

[MM11] S. T. Monteiro and R. J. Murphy. “Embedded feature selection of hyperspectral bands with boosted decision trees”. In: *Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2011, pp. 2361–2364.

[MNNS07] A. Morales Nicolás, I. Navarrete, and G. Sciavicco. “A new modal logic for reasoning about space: Spatial propositional neighborhood logic”. In: *Annals of Mathematics and Artificial Intelligence* 51 (2007), pp. 1–25.

[MS63] J. N. Morgan and J. A. Sonquist. “Problems in the Analysis of Survey Data, and a Proposal”. In: *Journal of the American Statistical Association* 58.302 (1963), pp. 415–434.

[MPV07] M. Morini, M. Pinelli, and M. Venturini. “Development of a one-dimensional modular dynamic model for the simulation of surge in compression systems”. In: *Journal of Turbomachinery* 129 (2007), pp. 437 –447.

[MPV09] M. Morini, M. Pinelli, and M. Venturini. “Analysis of biogas compression system dynamics”. In: *Applied Energy* 86 (Nov. 2009), pp. 2466 –2475.

[Mos+23] Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. "Human-in-the-loop machine learning: a state of the art". In: *Artif. Intell. Rev.* 56.4 (2023), pp. 3005–3054. URL: https://doi.org/10.1007/s10462-022-10246-w.

[MGZ17] L. Mou, P. Ghamisi, and X. X. Zhu. "Deep Recurrent Neural Networks for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.7 (2017), pp. 3639–3655.

[Mug91] S. H. Muggleton. "Inductive Logic Programming". In: *New Generation Computing* 8.4 (1991), pp. 295–318.

[Mug95] Stephen H. Muggleton. "Inverse Entailment and Progol". In: *New Gener. Comput.* 13.3&4 (1995), pp. 245–286. URL: https://doi.org/10.1007/BF03037227.

[MR94] Stephen H. Muggleton and Luc De Raedt. "Inductive Logic Programming: Theory and Methods". In: *J. Log. Program.* 19/20 (1994), pp. 629–679. URL: https://doi.org/10.1016/0743-1066(94)90035-3.

[Mug+21] Ananya Muguli, Lancelot Pinto, Nirmala R, Neeraj Sharma, Prashant Krishnan, Prasanta Kumar Ghosh, Rohit Kumar, Shrirama Bhat, Srikanth Raj Chetupalli, Sriram Ganapathy, Shreyas Ramoji, and Viral Nanda. "DiCOVA Challenge: Dataset, Task, and Baseline System for COVID-19 Diagnosis Using Acoustics". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2021, pp. 901–905.

[Muñoz-V+19] E. Muñoz-Velasco, M. Pelegrín-Garcí, P. Sala, G. Sciavicco, and I. E. Stan. "On coarser interval temporal logics". In: *Artificial Intelligence* 266 (2019), pp. 1–26.

[Mur+16a] C. Murdock, Z. Li, H. Zhou, and T. Duerig. "Blockout: Dynamic Model Selection for Hierarchical Deep Networks". In: *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2583–2591.

[MKS94] Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. "A System for Induction of Oblique Decision Trees". In: *J. Artif. Intell. Res.* 2 (1994), pp. 1–32. URL: https://doi.org/10.1613/jair.63.

[Mur+16b] V. N. Murthy, V. Singh, T. Chen, R. Manmatha, and D. Comaniciu. "Deep Decision Network for Multi-class Image Classification". In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2240–2248.

[Mur+10] M. Murugappan, M.Rizon, R. Nagarajan, and S. Yaacob. "Classification of human emotion from EEG using discrete wavelet transform". In: *Journal of Biomedical Science and Engineering* 3.4 (2010), pp. 390–396.

[Mur+08] M. Murugappan, M.Rizon, R. Nagarajan, S. Yaacob, I.Zunaidi, and D.Hazry. "EEG feature extraction for classifying emotions using FCM and FKM". In: *International Journal of Computers and Communnications* 2 (2008), pp. 299–304.

[Mus+97] T. Musha, Y. Terasaki, H.A. Haque, and G.A. Ivamitsky. "Feature extraction from EEGs associated with emotions". In: *Artificial Life and Robotics* 1 (1997), pp. 15–19.

[NK18] E. Naderi and K. Khorasani. "Data-driven fault detection, isolation and estimation of aircraft gas turbine engine actuator and sensors". In: *Mechanical Systems and Signal Processing* 100 (2018), pp. 415 –438.

[Nar+18] Nina Narodytska, Alexey Ignatiev, Filipe Pereira, and João Marques-Silva. "Learning Optimal Decision Trees with SAT". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. Ed. by Jérôme Lang. ijcai.org, 2018, pp. 1362–1368. URL: https://doi.org/10.24963/ijcai.2018/189.

[Nas+22] Ali Bou Nassif, Ismail Shahin, Mohamed Bader, Abdelfatah Hassan, and Naoufel Werghi. "COVID-19 detection systems using deep-learning algorithms based on speech and image data". In: *Mathematics* 10.4 (2022), p. 564.

[Nav+13] I. Navarrete, A. Morales Nicolás, G. Sciavicco, and M. A. Cárdenas-Viedma. "Spatial reasoning with rectangular cardinal relations: The convex tractable subalgebra". In: *Annals of Mathematics and Artificial Intelligence* 67.1 (2013), pp. 31–70.

[Nie+11] D. Nie, X.W. Wang, L.C. Shi, and B.L. Lu. "EEG-based emotion recognition during watching movies". In: *Proceedings of the 5th International IEEE/EBMS Conference on Neural Engineering (NER)*. 2011, pp. 667–670.

[Obe+19] Z. Obermeyer, B. Powers, C. Vogeli, and S. Mullainathan. "Dissecting racial bias in an algorithm used to manage the health of populations". In: *Science* 366.6464 (2019), pp. 447–453.

[Obr19] Josue Obregon. *RuleCOSI - Rule extraction COmbination and SImplification from classification tree ensembles*. https://github.com/jobregon1212/rulecosi. 2019.

[OJ23] Josué Obregon and Jae-Yoon Jung. "RuleCOSI+: Rule extraction for interpreting classification tree ensembles". In: *Inf. Fusion* 89 (2023), pp. 355–381. URL: https://doi.org/10.1016/j.inffus.2022.08.021.

[OKJ19] Josué Obregon, Aekyung Kim, and Jae-Yoon Jung. "RuleCOSI: Combination and simplification of production rules from boosted decision trees for imbalanced classification". In: *Expert Syst. Appl.* 126 (2019), pp. 64–82. URL: https://doi.org/10.1016/j.eswa.2019.02.012.

[OC23] Siobhan O'Connor and ChatGPT. "Open Artificial Intelligence Platforms in Nursing Education: Tools for Academic Progress or Abuse?" English. In: *Nurse Education in Practice* 66 (Jan. 2023). URL: https://doi.org/10.1016/j.nepr.2022.103537.

[OW03] C. Olaru and L. Wehenkel. "A complete fuzzy decision tree technique". In: *Fuzzy Sets Syst.* 138.2 (2003), pp. 221–254.

[OTA21] Lara Orlandic, Tomás Teijeiro, and David Atienza. "The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms". In: *Scientific Data* 8 (June 2021).

[OF16] F.E.B. Otero and A.A. Freitas. "Improving the Interpretability of Classification Rules Discovered by an Ant Colony Algorithm: Extended Results". In: *Evolutionary Computation* 24.3 (2016), pp. 385–409.

[PH90] Giulia Pagallo and David Haussler. "Boolean Feature Discovery in Empirical Learning". In: *Mach. Learn.* 5 (1990), pp. 71–99. URL: https://doi.org/10.1007/BF00115895.

[Pag+24] G. Pagliarini, F. Manzella, G. Sciavicco, and I. E. Stan. *ModalDecisionTrees.jl: Interpretable Models for Native Time-series & Image Classification*. https://github.com/aclai-lab/ModalDecisionTrees.jl. 2024.

[Pag+22] G. Pagliarini, S. Scaboro, G. Serra, G. Sciavicco, and I. E. Stan. "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, 2022, 13:1–13:15.

[PS21] G. Pagliarini and G. Sciavicco. "Decision Tree Learning with Spatial Modal Logics". In: *Proceedings of the 12th International Symposium on Games, Automata, Logics, and Formal Verification (GANDALF)*. Vol. 346. EPTCS. 2021, pp. 273–290.

[PS23] G. Pagliarini and G. Sciavicco. "Interpretable Land Cover Classification with Modal Decision Trees". In: *European Journal of Remote Sensing* 56.1 (2023), p. 2262738. URL: https://doi.org/10.1080/22797254.2023.2262738.

[PSS21] G. Pagliarini, G. Sciavicco, and I. E. Stan. "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, 2021, pp. 37–41.

[Pag+23] Giovanni Pagliarini, Andrea Paradiso, Sasha Rubin, Guido Sciavicco, and Ionel Eduard Stan. "Heuristic Minimization Modulo Theory of Modal Decision Trees Class-Formulas". In: *Short Paper Proceedings of the 5th Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis hosted by the 22nd International Conference of the Italian Association for*

Artificial Intelligence (AIxIA 2023), Rome, Italy, November 7, 2023. Vol. 3629. CEUR Workshop Proceedings. CEUR-WS.org, 2023, pp. 49–53. URL: https://ceur-ws.org/Vol-3629/paper8.pdf.

[Pah+21] M. Pahar, M. Klopper, R. Warren, and T. Niesler. "COVID-19 cough classification using machine learning and global smartphone recordings". In: *Computers in Biology and Medicine* 135 (2021), p. 104572.

[PM03] M. Pal and P. M. Mather. "An assessment of the effectiveness of decision tree methods for land cover classification". In: *Remote Sensing of Environment* 86.4 (2003), pp. 554–565.

[PC24] Giovanni Pagliarini Guido Sciavicco Ionel Eduard Stan Patrik Cavina Federico Manzella. "Beyond Tabular: Feature Extraction and Selection for High-Dimensional Data". In: *Journal of Big Data* (2024). To appear.

[Ped+11a] F. Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Ped+11b] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

[Pei+04] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Mei-Chun Hsu. "Mining sequential patterns by pattern-growth: The prefixspan approach". In: *IEEE Transactions on knowledge and data engineering* 16.11 (2004), pp. 1424–1440.

[PH09] P. Petrantonakis and L. Hadjileontiadis. "Emotion Recognition From EEG Using Higher Order Crossings". In: *IEEE transactions on information technology in biomedicine* 14 (2009), pp. 186–197.

[PTA13] P. Pezzini, D. Tucker, and T. Alberto. "Avoiding compressor surge during emergency shut-down hybrid turbine systems". In: *Journal of Engineering for Gas Turbines and Power* 135 (2013), pp. 1 –9.

[Phi+20] D. Phiri, M. Simwanda, V. Nyirenda, Y. Murayama, and M. Ranagalage. "Decision Tree Algorithms for Developing Rulesets for Object-Based Land Cover Classification". In: *ISPRS International Journal of Geo-Information* 9.5 (2020), p. 329.

[PMN21] Federico Pigozzi, Eric Medvet, and Laura Nenzi. "Mining road traffic rules with signal temporal logic and grammar-based genetic programming". In: *Applied Sciences* 11.22 (2021), p. 10573.

[Pnu77] A. Pnueli. "The Temporal Logic of Programs". In: *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FOCS)*. 1977, pp. 46–57.

[PH11] Jim Prentzas and Ioannis Hatzilygeroudis. "Neurules – A type of neuro-symbolic rules: An overview". In: *Combinations of Intelligent Methods and Applications: Proceedings of the 2nd International Workshop, CIMA 2010, France, October 2010*. Springer. 2011, pp. 145–165.

[PDN21] Aniruddh G. Puranic, Jyotirmoy V. Deshmukh, and Stefanos Nikolaidis. "Learning From Demonstrations Using Signal Temporal Logic in Stochastic and Continuous Domains". In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 6250–6257. URL: https://doi.org/10.1109/LRA.2021.3092676.

[QC19] S.J. Qin and L.H. Chiang. "Advances and opportunities in machine learning for process data analytics". In: *Computers and Chemical Engineering* 126 (2019), pp. 465–473.

[Qui79] J. R. Quinlan. "Discovering rules by induction from large collections of examples". In: *Expert systems in the micro electronics age* (1979).

[Qui86] J. R. Quinlan. "Induction of Decision Trees". In: *Machine Learning* 1 (1986), pp. 81–106.

[Qui87] J. R. Quinlan. "Generating Production Rules from Decision Trees". In: *Proceedings of the 10th International Joint Conference on Artificial Intelligence. Milan, Italy, August 23-28, 1987*. Ed. by John P. McDermott. Morgan Kaufmann, 1987, pp. 304–307. URL: https://ijcai.org/Proceedings/87-1/Papers/063.pdf.

[Qui90] J. R. Quinlan. "Learning logical definitions from relations". In: *Machine learning* 5 (1990), pp. 239–266.

[Qui93] J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, 1993.

[Qui99] J. R. Quinlan. "Simplifying decision trees". In: *International Journal of Human-Computer Studies* 51.2 (1999), pp. 497–510.

[Qui04] J. R. Quinlan. "Data mining tools See5 and C5.0". In: *https://www.rulequest.com/see5-info.html* (2004).

[Qui+92] John R Quinlan et al. "Learning with continuous classes". In: *5th Australian joint conference on artificial intelligence*. Vol. 92. World Scientific. 1992, pp. 343–348.

[R C21] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: https://www.R-project.org/.

[Rae08] Luc De Raedt. *Logical and relational learning*. Cognitive Technologies. Springer, 2008. URL: https://doi.org/10.1007/978-3-540-68856-3.

[RB97] Luc De Raedt and Hendrik Blockeel. "Using Logical Decision Trees for Clustering". In: *Inductive Logic Programming, 7th International Workshop, ILP-97, Prague, Czech Republic, September 17-20, 1997, Proceedings*. Ed. by Nada Lavrac and Saso Dzeroski. Vol. 1297. Lecture Notes in Computer Science. Springer, 1997, pp. 133–140. URL: https://doi.org/10.1007/3540635149_41.

[Rah+21] M.M. Rahman, A.K. Sarkar, M.A. Hossain, M.S. Hossain, M.R. Islam, M.B. Hossain, J.M.W. Quinn, and M.A. Moni. "Recognition of human emotions using EEG signals: A review". In: *Computers in Biology and Medicine* 136 (2021), pp. 1 –18.

[RCC92] D. Randell, Z. Cui, and A.G. Cohn. "A Spatial Logic Based on Regions and Connection". In: *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*. 1992, pp. 165–176.

[Ras18] Sebastian Raschka. "MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack". In: *The Journal of Open Source Software* 3.24 (Apr. 2018). URL: https://doi.org/10.21105/joss.00638.

[RC85] W J Ray and H W Cole. "EEG alpha activity reflects attentional demands, and beta activity reflects emotional and cognitive processes." In: *Science* 228.4700 (1985), pp. 750–2.

[Gdp] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)*. 2016. URL: https://data.europa.eu/eli/reg/2016/679/oj.

[RN07] J. Renz and B. Nebel. "Qualitative Spatial Reasoning Using Constraint Calculi". In: *Handbook of Spatial Logics*. Springer, 2007, 161–215.

[Riv87] R. L. Rivest. "Learning Decision Lists". In: *Machine Learning* 2.3 (1987), pp. 229–246.

[Riz+15] Lala Septem Riza, Christoph Bergmeir, Francisco Herrera, and José M Benítez. "frbs: Fuzzy rule-based systems for classification and regression in R". In: *Journal of statistical software* 65 (2015), pp. 1–30.

[RM05] Lior Rokach and Oded Maimon. "Top-down induction of decision trees classifiers - a survey". In: *IEEE Trans. Syst. Man Cybern. Part C* 35.4 (2005), pp. 476–487. URL: https://doi.org/10.1109/TSMCC.2004.843247.

[Roy+20] S. K. Roy, G. Krishna, S. R. Dubey, and B. B. Chaudhuri. "HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification". In: *IEEE Geoscience and Remote Sensing Letters* 17.2 (2020), pp. 277–281.

[RVP13] V. Rozgic, S. Vitaladevuni, and R. Prasad. "Robust EEG emotion classification using segment level decision fusion". In: *Proceedings of the 25th International Conference on Acoustics, Speech, and Signal Processing*. 2013, pp. 1286–1290.

[Rud19] C. Rudin. "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead". In: *Nature Machine Intelligence* 1.5 (2019), pp. 206–215.

[RN20] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. 4th. Pearson, 2020.

[Sad+22] Ben Sadeghi, Poom Chiarawongse, Kevin Squire, Daniel C. Jones, Andreas Noack, Cédric St-Jean, Rik Huijzer, Roland Schätzle, Ian Butterworth, Yu-Fong Peng, and Anthony Blaom. *DecisionTree.jl - A Julia implementation of the CART Decision Tree and Random Forest algorithms*. Version 0.11.3. Nov. 2022. URL: https://doi.org/10.5281/zenodo.7359268.

[SL91] S. Rasoul Safavian and David A. Landgrebe. "A survey of decision tree classifier methodology". In: *IEEE Trans. Syst. Man Cybern.* 21.3 (1991), pp. 660–674. URL: https://doi.org/10.1109/21.97458.

[San+17] A. Santara, K. Mani, P. Hatwar, A. Singh, A. Garg, K. Padia, and P. Mitra. "BASS Net: Band-Adaptive Spectral-Spatial Feature Learning Neural Network for Hyperspectral Image Classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.9 (2017), pp. 5293–5301.

[SS09] K. Schaaff and T. Schultz. "Towards emotion recognition from electroencephalographic signals". In: *Proceedings of the 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*. 2009, pp. 1–6.

[SS21] André Schidler and Stefan Szeider. "SAT-based Decision Tree Learning for Large Data Sets". In: *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 3904–3912. URL: https://doi.org/10.1609/aaai.v35i5.16509.

[Sch+17] R.T. Schirrmeister, J.T. Springenberg, L.D.J. Fiederer, M. Glasstetter, K. Eggensperger, M. Tangermann, F. Hutter, W. Burgard, and T. Ball. "Deep Learning With Convolutional Neural Networks for EEG Decoding and Visualization". In: *Human Brain Mapping* 38 (2017), pp. 5391–5420.

[ST01] L.A. Schmidt and L.J. Trainor. "Frontal brain electrical activity (EEG) distinguishes valence and intensity of musical emotions". In: *Cognition and Emotion* 15.4 (2001), pp. 487–500.

[SAG99] G. P. J. Schmitz, C. Aldrich, and F. S. Gouws. "ANN-DT: An Algorithm for Extraction of Decision Trees from Artificial Neural Networks". In: *IEEE Transactions on Neural Networks* 10.6 (1999), pp. 1392–1401.

[Sch+21] B. W. Schuller, A. Batliner, C. Bergler, C. Mascolo, J. Han, I. Lefter, H. Kaya, S. Amiriparian, A. Baird, L. Stappen, S. Ottl, M. Gerczuk, P. Tzirakis, C. Brown, J. Chauhan, A. Grammenos, A. Hasthanasombat, D. Spathis, T. Xia, P. Cicuta, L. J. M. Rothkrantz, J. A. Zwerts, J. Treep, and C. S. Kaandorp. "The INTERSPEECH 2021 Computational Paralinguistics Challenge: COVID-19 Cough, COVID-19 Speech, Escalation & Primates". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2021, pp. 431–435.

[Sen+22] Prithviraj Sen, Breno W. S. R. de Carvalho, Ryan Riegel, and Alexander G. Gray. "Neuro-Symbolic Inductive Logic Programming with Logical Neural Networks". In: *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*. AAAI Press, 2022, pp. 8212–8219. URL: https://doi.org/10.1609/aaai.v36i8.20795.

[Ser21] Sefik Ilkin Serengil. *ChefBoost: A Lightweight Boosted Decision Tree Framework*. https://doi.org/10.5281/zenodo.5576203. Oct. 2021. URL: https://doi.org/10.5281/zenodo.5576203.

[Set90] I. K. Sethi. "Entropy nets: from decision trees to neural networks". In: *Proceedings of the IEEE* 78.10 (1990), pp. 1605–1613.

[SL99a] R. Setiono and W. K. Leow. "On mapping decision trees and neural networks". In: *Knowledge-Based Systems* 12.3 (1999), pp. 95–99.

[SL99b] R. Setiono and H. Liu. "A Connectionist Approach to Generating Oblique Decision Trees". In: *IEEE Transactions on Systems, Man and Cybernetics – Part B* 29.3 (1999), pp. 440–444.

[Sha49] C. E. Shannon. "Communication in the presence of noise". In: *Proceedings of the Institute of Radio Engeneers* 37.1 (1949), pp. 10–21.

[Sha+20] Neeraj Sharma, Prashant Krishnan, Rohit Kumar, Shreyas Ramoji, Srikanth Raj Chetupalli, Nirmala R., Prasanta Kumar Ghosh, and Sriram Ganapathy. "Coswara – A Database of Breathing, Cough, and Voice Sounds for COVID-19 Diagnosis". In: *Proceedings of the 21st Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2020, pp. 4811–4815.

[SCM21] Pouya Shati, Eldan Cohen, and Sheila A. McIlraith. "SAT-Based Approach for Learning Optimal Decision Trees with Non-Binary Features". In: *27th International Conference on Principles and Practice of Constraint Programming, CP 2021, Montpellier, France (Virtual Conference), October 25-29, 2021*. Ed. by Laurent D. Michel. Vol. 210. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, 50:1–50:16. URL: https://doi.org/10.4230/LIPIcs.CP.2021.50.

[SA22] R. Shwartz-Ziv and A. Armon. "Tabular data: Deep learning is not all you need". In: *Information Fusion* 81 (2022), pp. 84–90.

[Sil+20] Jennifer Sills, C. Michael Barton, Marina Alberti, Daniel Ames, Jo-An Atkinson, Jerad Bales, Edmund Burke, Min Chen, Saikou Y Diallo, David J. D. Earn, Brian Fath, Zhilan Feng, Christopher Gibbons, Ross Hammond, Jane Heffernan, Heather Houser, Peter S. Hovmand, Birgit Kopainsky, Patricia L. Mabry, Christina Mair, Petra Meier, Rebecca Niles, Brian Nosek, Nathaniel Osgood, Suzanne Pierce, J. Gareth Polhill, Lisa Prosser, Erin Robinson, Cynthia Rosenzweig, Shankar Sankaran, Kurt Stange, and Gregory Tucker. "Call for transparency of COVID-19 models". In: *Science* 368.6490 (2020), pp. 482–483.

[SRM15] V. P. Singh, J. M. S. Rohith, and V. K. Mittal. "Preliminary analysis of cough sounds". In: *Proceedings of the 2015 Annual IEEE India Conference (INDICON)*. 2015, pp. 1–6.

[SK04] S. Skiadopoulos and M. Koubarakis. "Composing cardinal direction relations". In: *Artificial Intelligence* 152.2 (2004), pp. 143–171.

[Ski+07] S. Skiadopoulos, N. Sarkas, T.K. Sellis, and M. Koubarakis. "A Family of Directional Relation Models for Extended Objects". In: *IEEE Transactions on Knowledge and Data Engineering* 19.8 (2007), pp. 1116–1130.

[Sön+08] C. Sönströd, U. Johansson, R. König, and L. Niklasson. "Genetic Decision Lists for Concept Description". In: *Proceedings of The 2008 International Conference on Data Mining, DMIN 2008, July 14-17, 2008, Las Vegas, USA, 2 Volumes*. Ed. by Robert Stahlbock, Sven F. Crone, and Stefan Lessmann. CSREA Press, 2008, pp. 450–456.

[SG19] Zenon A. Sosnowski and Lukasz Gadomer. "Fuzzy trees and forests - Review". In: *WIREs Data Mining Knowl. Discov.* 9.6 (2019). URL: https://doi.org/10.1002/widm.1316.

[SA96] Ramakrishnan Srikant and Rakesh Agrawal. "Mining Sequential Patterns: Generalizations and Performance Improvements". In: *Advances in Database Technology - EDBT'96, 5th International Conference on Extending Database Technology, Avignon, France, March 25-29, 1996, Proceedings*. Ed. by Peter M. G. Apers, Mokrane Bouzeghoub, and Georges Gardarin. Vol. 1057. Lecture Notes in Computer Science. Springer, 1996, pp. 3–17. URL: https://doi.org/10.1007/BFb0014140.

[Sri01] Ashwin Srinivasan. *The aleph manual*. 2001.

[SS13] N. Srivastava and R. Salakhutdinov. "Discriminative Transfer Learning with Tree-based Priors". In: *Proceedings of the 26th Advances In Neural Information Processing Systems (NIPS)*. 2013, pp. 2094–2102.

[Sta+22] I. E. Stan, G. Sciavicco, E. Muñoz-Velasco, Giovanni Pagliarini, Mauro Milella, and Andrea Paradiso. "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, 2022, pp. 53–65.

[Sta+21] B. Stasak, Z. Huang, S. Razavi, D. Joachim, and J. Epps. "Automatic Detection of COVID-19 Based on Short-Duration Acoustic Smartphone Speech Analysis". In: *Journal of Healthcare Informatics Research* 5.2 (2021), pp. 201–217.

[Ste+20] Paul Stey, Mary McGrath, Maria Isabel Restrepo, Dilum Aluthge, and BCBI Bot. *bcbi/ARules.jl: v0.0.2*. Version v0.0.2. Feb. 2020. URL: https://doi.org/10.5281/zenodo.3660120.

[SL99c] A. Suárez and J. F. Lutsko. "Globally Optimal Fuzzy Decision Trees for Classification and Regression". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.12 (1999), pp. 1297–1311.

[TST92] T. Tani, M. Sakoda, and K. Tanaka. "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*. 1992, pp. 923–930.

[TGS20] Akbar Telikani, Amir H Gandomi, and Asadollah Shahbahrami. "A survey of evolutionary computation for association rule mining". In: *Information Sciences* 524 (2020), pp. 318–352.

[TCS22] Alberto Tena, Francesc Clarià, and Francesc Solsona. "Automated detection of COVID-19 cough". In: *Biomedical Signal Processing and Control* 71.Part (2022), p. 103175.

[TC96] Hendrik Theron and Ian Cloete. "BEXA: A Covering Algorithm for Learning Propositional Concept Descriptions". In: *Mach. Learn.* 24.1 (1996), pp. 5–40. URL: https://doi.org/10.1007/BF00117830.

[TF06] Sébastien Thomassey and Antonio Fiordaliso. "A hybrid sales forecasting system based on clustering and decision trees". In: *Decis. Support Syst.* 42.1 (2006), pp. 408–421. URL: https://doi.org/10.1016/j.dss.2005.01.008.

[TWY00] E. C. C. Tsang, X. Wang, and D. S. Yeung. "Improving learning accuracy of fuzzy decision trees by hybrid neural networks". In: *IEEE Transactions on Fuzzy Systems* 8.5 (2000), pp. 601–614.

[Tsu+96] T. Tsuchiya, T. Maeda, Y. Matsubara, and M. Nagamachi. "A fuzzy rule induction method using genetic algorithm". In: *International Journal of Industrial Ergonomics* 18.2 (1996), pp. 135–145.

[Uno+03] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura. "LCM: An Efficient Algorithm for Enumerating Frequent Closed Item Sets." In: *Fimi*. Vol. 90. 2003.

[UKA+04] Takeaki Uno, Masashi Kiyomi, Hiroki Arimura, et al. "LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets". In: *Fimi*. Vol. 126. 2004.

[Val79] Leslie G. Valiant. "The Complexity of Enumeration and Reliability Problems". In: *SIAM J. Comput.* 8.3 (1979), pp. 410–421. URL: https://doi.org/10.1137/0208032.

[VRDJ95] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[WZ19] P. A. Walega and M. Zawidzki. "A Modal Logic for Subject-Oriented Spatial Reasoning". In: *Proceedings of the 26th International Symposium on Temporal Representation and Reasoning (TIME'19)*. Vol. 147. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl, 2019, 4:1–4:22.

[Wan+21] A. Wan, L. Dunlap, D. Ho, J. Yin, S. Lee, S. Petryk, S. Adel Bargal, and J. E. Gonzalez. "NBDT: Neural-Backed Decision Tree". In: *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. 2021.

[Wan+18] Ling Wang, Jianyao Meng, Peipei Xu, and Kaixiang Peng. "Mining temporal association rules with frequent itemsets tree". In: *Applied Soft Computing* 62 (2018), pp. 817–829.

[WS87] Q. R. Wang and C. Y. Suen. "Large Tree Classifier with Heuristic Search and Global Training". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 9.1 (1987), pp. 91–102.

[Wan+00] X. Wang, B. Chen, G. Qian, and F. Ye. "On the optimization of fuzzy decision trees". In: *Fuzzy Sets and Systems* 112.1 (2000), pp. 117–125.

[WNL14] Xiao-Wei Wang, Dan Nie, and Bao-Liang Lu. "Emotional state classification from EEG data using machine learning approach". In: *Neurocomputing* 129 (2014), pp. 94–106. URL: https://doi.org/10.1016/j.neucom.2013.06.046.

[WW97] Yong Wang and Ian H Witten. "Inducing model trees for continuous classes". In: *Proceedings of the ninth European conference on machine learning*. Vol. 9. 1. 1997, pp. 128–137.

[Web95] Geoffrey I. Webb. "OPUS: An Efficient Admissible Algorithm for Unordered Search". In: *J. Artif. Intell. Res.* 3 (1995), pp. 431–465. URL: https://doi.org/10.1613/jair.227.

[WL90] M.E. Wewers and N.K. Lowe. "A critical review of visual analogue scales in the measurement of clinical phenomena". In: *Research in Nursing and Health* 13 (1990), pp. 227–236.

[Wil92] Frank Wilcoxon. "Individual comparisons by ranking methods". In: *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[WF99] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999. URL: https://www.cs.waikato.ac.nz/\%7Eml/weka/book.html.

[Won+14] P.K. Wong, Z. Yang, C.M. Vong, and J. Zhong. "Real-time fault diagnosis for gas turbine generator systems using extreme learning machine". In: *Neurocomputing* 128 (2014), pp. 249 –257.

[Xia+21a] T. Xia, J. Han, L. Qendro, T. Dang, and C. Mascolo. "Uncertainty-Aware COVID-19 Detection from Imbalanced Sound Data". In: *Proceedings of the 22nd Annual Conference of the International Speech Communication Association (INTERSPEECH)*. 2021, pp. 2951–2955.

[Xia+21b] Tong Xia, Dimitris Spathis, Chloë Brown, J Ch, Andreas Grammenos, Jing Han, Apinan Hasthanasombat, Erika Bondareva, Ting Dang, Andres Floto, Pietro Cicuta, and Cecilia Mascolo. "COVID-19 Sounds: A Large-Scale Audio Dataset for Digital COVID-19 Detection". In: *Proceedings of the 35th Conference on Neural Information Processing Systems (NIPS) Datasets and Benchmarks Track (Round 2)*. 2021.

[Xu+05] M. Xu, P. Watanachaturaporn, P.K. Varshney, and M.K. Arora. "Decision tree regression for soft classification of remote sensing data". In: *Remote Sensing of Environment* 97.3 (2005), pp. 322–336.

[Yan04] G. Yang. "The complexity of mining maximal frequent itemsets and maximal frequent patterns". In: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 2004, pp. 344–353.

[YK16] Pinar Yazgana and Ali Osman Kusakci. "A literature survey on association rule mining algorithms". In: *Southeast Europe Journal of soft computing* 5.1 (2016).

[YH03] Xiaoxin Yin and Jiawei Han. "CPAR: Classification based on predictive association rules". In: *Proceedings of the 2003 SIAM international conference on data mining*. SIAM. 2003, pp. 331–335.

[YS95] Y. Yuan and M. J. Shaw. "Induction of fuzzy decision trees". In: *Fuzzy Sets and Systems* 69.2 (1995), pp. 125–139.

[Zaf+17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi. "Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment". In: *Proceedings of the 26th International Conference on World Wide Web*. WWW '17. Perth, Australia: International World Wide Web Conferences Steering Committee, 2017, 1171–1180. URL: https://doi.org/10.1145/3038912.3052660.

[Zah+16] Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. "Apache spark: a unified engine for big data processing". In: *Communications of the ACM* 59.11 (2016), pp. 56–65.

[Zak00] Mohammed J Zaki. "Sequence mining in categorical domains: incorporating constraints". In: *Proceedings of the ninth international conference on Information and knowledge management*. 2000, pp. 422–429.

[Zak+97] Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li, et al. "New algorithms for fast discovery of association rules." In: *KDD*. Vol. 97. 1997, pp. 283–286.

[Zek99] S. Zeki. *Inner vision: an exploration of art and the brain*. Oxford University Press, 1999.

[Zem+08] Abdelhamid Zemirline, Laurent Lecornu, Basel Solaiman, and Ahmed Ech-Cherif. "An Efficient Association Rule Mining Algorithm for Classification". In: *Artificial Intelligence and Soft Computing - ICAISC 2008, 9th International Conference, Zakopane, Poland, June 22-26, 2008, Proceedings*. Ed. by Leszek Rutkowski, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada. Vol. 5097. Lecture Notes in Computer Science. Springer, 2008, pp. 717–728. URL: https://doi.org/10.1007/978-3-540-69731-2_69.

[ZW03] Q. Zhang and J. Wang. "A rule-based urban land use inferring method for fine-resolution multispectral imagery". In: *Canadian Journal of Remote Sensing* 29 (2003), pp. 1–13.

[Zha+21] A. Zharmagambetov, S. Hada, M. Gabidolla, and M. Carreira-Perpinan. "Non-Greedy Algorithms for Decision Tree Optimization: An Experimental Comparison". In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. 2021, pp. 1–8.

[ZFL19] S. Zhong, S. Fu, and L. Lin. "A novel gas turbine fault diagnosis method based on transfer learning with CNN". In: *Measurement* 137 (2019), pp. 435 –453.

[ZC02] Z.-H. Zhou and Z. Chen. "Hybrid Decision Tree". In: *Knowledge-Based Systems* 15.8 (2002), pp. 515–528.

[ZJ04] Z.-H. Zhou and Y. Jiang. "NeC4.5: Neural Ensemble Based C4.5". In: *IEEE Transactions on Knowledge and Data Engineering* 16.6 (2004), pp. 770–773.