Modal Symbolic Learning: A Tutorial

Giovanni Pagliarini[1,2] giovanni.pagliarini@unife.it
Guido Sciavicco[1] guido.sciavicco@unife.it

December 6-7-8th, 2023

[1]Applied Computational Logic and Artificial Intelligence (ACLAI) Laboratory,
Department of Mathematics and Computer Science, University of Ferrara, Italy

[2]Department of Mathematical, Physical, and Computer Sciences, University of Parma, Italy

Program

Program

- Day 1:
  - Basic propositional logic for learning
  - Basic modal logic for learning
  - Symbolic learning: decision trees and other models

Program

- Day 1:
  - Basic propositional logic for learning
  - Basic modal logic for learning

  - Symbolic learning: decision trees and other models
- Day 2:
  - Symbolic learning: modal decision trees
  - Unstructured data as modal data

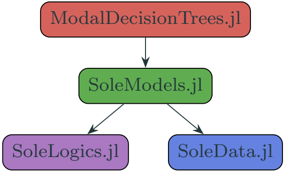
Program

- Day 1:
  - Basic propositional logic for learning
  - Basic modal logic for learning
  - Symbolic learning: decision trees and other models
- Day 2:
  - Symbolic learning: modal decision trees
  - Unstructured data as modal data
- Day 3:
  - Learning: symbolic feature selection
  - Learning: statistically solid, ensemble, and multi-modal models

Program

- Day 1:
  - Basic propositional logic for learning
  - Basic modal logic for learning
  - Practice: symbolic knowledge manipulation with SoleLogics.jl
  - Symbolic learning: decision trees and other models
- Day 2:
  - Symbolic learning: modal decision trees
  - Unstructured data as modal data
- Day 3:
  - Practice: learning with Sole.jl
  - Learning: symbolic feature selection
  - Learning: statistically solid, ensemble, and multi-modal models
  - Practice: experiment with Sole.jl

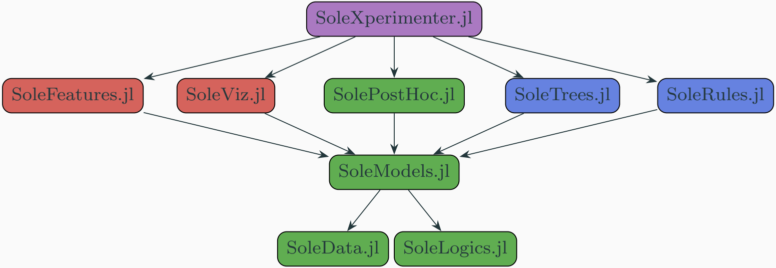
The Sole Framework



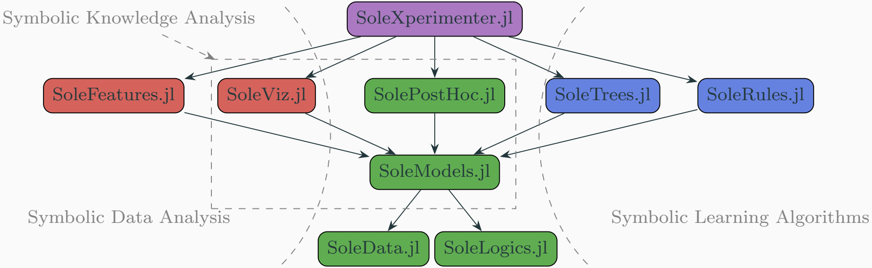
Currently released packages:

- `SoleLogics.jl` – logical formulas;
- `SoleData.jl` – multimodal, non-tabular datasets;
- `SoleModels.jl` – symbolic models (decision trees, association rules, ...);
- `ModalDecisionTrees.jl` – DT learning for multimodal, non-tabular data.

The Sole Framework tomorrow (tentative)



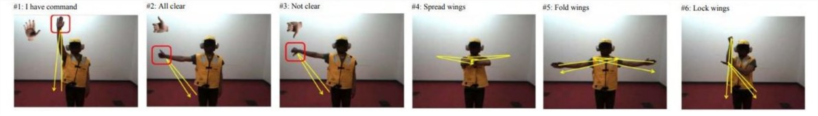
The Sole Framework tomorrow (tentative)



Appetizer

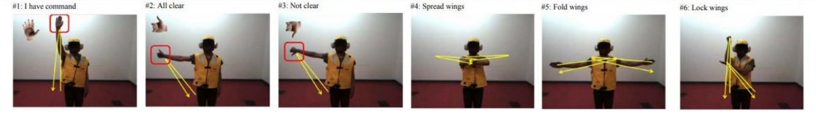
NATOPS: A gesture recognition problem

- **Dataset:** NATOPS, public benchmark for time series classification
  - 360 instances;
  - 24 variables;
  - 51 temporal points.
- **Task:** Gesture recognition from position sensors;
- **Input:** xyz coordinates for elbows/wrists/hands/thumbs (left+right) evolving through time;
- **Output:** gesture type (6 classes):



NATOPS: A gesture recognition problem

- **Dataset:** NATOPS, public benchmark for time series classification
  - 360 instances;
  - 24 variables;
  - 51 temporal points.
- **Task:** Gesture recognition from position sensors;
- **Input:** xyz coordinates for elbows/wrists/hands/thumbs (left+right) evolving through time;
- **Output:** gesture type (6 classes):

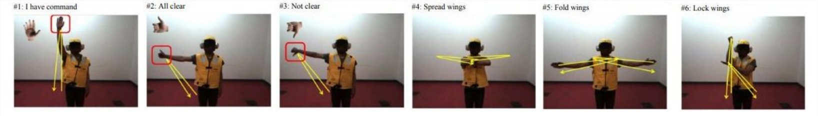


NN	SVM (non-NN state of the art)	CART Decision Tree
97.1%	88.5%	70.9%

Table 1: Accuracies (10-fold cv) for non-symbolic state-of-the-art and decision tree models.

NATOPS: A gesture recognition problem

- **Dataset:** NATOPS, public benchmark for time series classification
  - 360 instances;
  - 24 variables;
  - 51 temporal points.
- **Task:** Gesture recognition from position sensors;
- **Input:** xyz coordinates for elbows/wrists/hands/thumbs (left+right) evolving through time;
- **Output:** gesture type (6 classes):



NN	SVM	CART Decision Tree	**CART Modal Decision Tree**
97.1%	88.5%	70.9%	**89.7%**

Table 1: Accuracies (10-fold cv) for non-symbolic state-of-the-art and decision tree models.

Material

Material & instructions for this tutorial are available at:
`https://github.com/aclai-lab/modal-symbolic-learning-course/`

Day 1

A Gentle Introduction

What is Modal Symbolic Learning?

Modal symbolic learning is the generalization of symbolic learning to more-than-propositional languages.

What is Modal Symbolic Learning?

Modal symbolic learning is the generalization of symbolic learning to more-than-propositional languages. Symbolic learning is the set of techniques and methods that allow one to learn from data to produce a logical model.

Modal symbolic learning is the generalization of symbolic learning to more-than-propositional languages. Symbolic learning is the set of techniques and methods that allow one to learn from data to produce a logical model. Logical models are logical formulas, and they describe the data from which they have been learned; they are, in fact, a theory of the underlying phenomenon. It has been known for decades that different phenomena need different logics to be described;

What is Modal Symbolic Learning?

Modal symbolic learning is the generalization of symbolic learning to more-than-propositional languages. Symbolic learning is the set of techniques and methods that allow one to learn from data to produce a logical model. Logical models are logical formulas, and they describe the data from which they have been learned; they are, in fact, a theory of the underlying phenomenon. It has been known for decades that different phenomena need different logics to be described; here, we study how to bring the same concept to learning.

How Do We Use Logic?

Logic has three essential aspects

How Do We Use Logic?

Logic has three essential aspects

Syntax

How Do We Use Logic?

Logic has three essential aspects

Syntax

Semantics

How Do We Use Logic?

Logic has three essential aspects

Syntax

Semantics

Deduction

How Do We Use Logic?

Logic has three essential aspects

- Syntax
  - What can a language **express**?
- Semantics
- Deduction

How Do We Use Logic?

Logic has three essential aspects

Syntax

What can a language express?

Semantics

What does a formula mean in algebraic terms?

Deduction

How Do We Use Logic?

Logic has three essential aspects

Syntax

What can a language express?

Semantics

What does a formula mean in algebraic terms?

Deduction

How do we deduce true formulas from true formulas?

How Do We Use Logic?

Logic has three essential aspects

Syntax

What can a language **express**?

Semantics

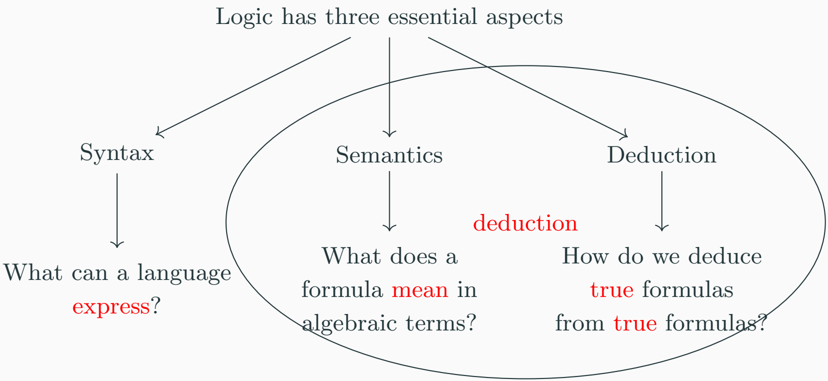
What does a formula **mean** in algebraic terms?

Deduction

How do we deduce **true** formulas from **true** formulas?

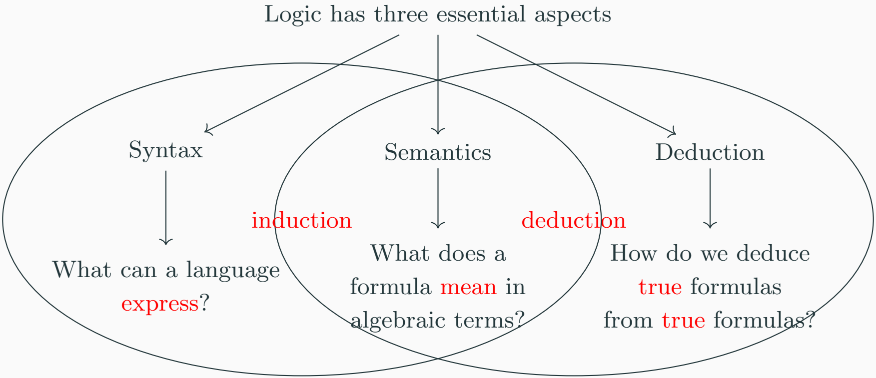
deduction

$? \Vdash \varphi$



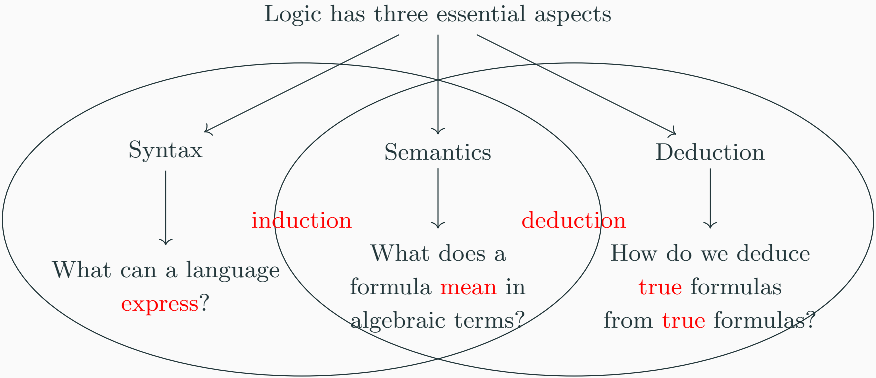
$$M_1, \ldots, M_m \Vdash \varphi$$

How Do We Use Logic?

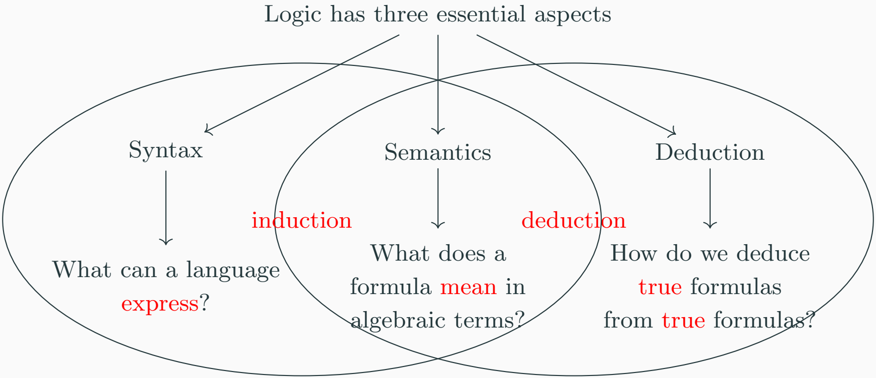


$\mathcal{M} = \{M_1, \ldots, M_m\} \Vdash ? \qquad M_1, \ldots, M_m \Vdash \varphi$

How Do We Use Logic?



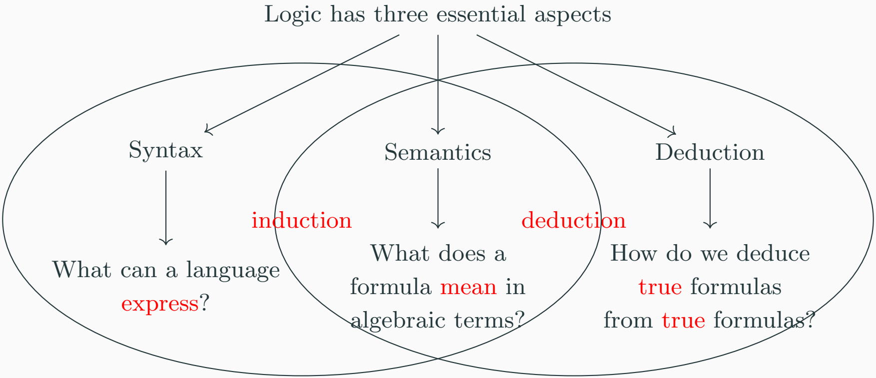
$\mathcal{M} = \{M_1, \ldots, M_m\} \Vdash \varphi \qquad M_1, \ldots, M_m \Vdash \varphi$



$\mathcal{M} = \{M_1, \ldots, M_m\} \Vdash \varphi \qquad M_1, \ldots, M_m \Vdash \varphi$

$M \,?\, \varphi$

How Do We Use Logic?



$$\mathcal{M} = \{M_1, \ldots, M_m\} \Vdash \varphi \qquad M_1, \ldots, M_m \Vdash \varphi$$

$$M \Vdash \varphi$$

Propositional Logic: Quick and Dirty

Propositional Logic: Syntax – 1

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

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi$$

in learning, atoms usually
belong to a deeper theory that allows,
at least, comparing variables with constants
hereafter, $\bowtie \in \{<, \leq, =, \geq, >\}$

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

in learning, we prefer non-minimal grammars, that enhance the learning phase at the expenses of the minimality

Propositional Logic: Syntax – 1

$$\varphi ::= (Fever > 38.5) \mid (BloodPr < 120) \mid \ldots \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

in learning, the alphabet
encompasses a set of attribute names
that coincides with the set of variables
of the propositional dataset from which we learn

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

How do we express. . . ?

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

How do we express...?

Fever is higher than 38 and
Blood pressure is lower than 120

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

How do we express...?

Fever is higher than 38 and
Blood pressure is lower than 120

$$Fever > 38 \wedge BloodPr < 120$$

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

How do we express...?

Fever is higher than 38 and
Blood pressure is lower than 120

$$Fever > 38 \wedge BloodPr < 120$$

Fever is higher than 38 implies
Blood pressure is lower than 120

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

How do we express. . . ?

Fever is higher than 38 and Blood pressure is lower than 120	$Fever > 38 \wedge BloodPr < 120$
Fever is higher than 38 implies Blood pressure is lower than 120	$Fever > 38 \rightarrow BloodPr < 120$

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

How do we express. . . ?

Fever is higher than 38 and
Blood pressure is lower than 120

$$Fever > 38 \wedge BloodPr < 120$$

Fever is higher than 38 implies
Blood pressure is lower than 120

$$Fever > 38 \rightarrow BloodPr < 120$$

Fever is higher than 38 and
Blood pressure is lower than 120 or
higher than 145

Propositional Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

How do we express. . . ?

Fever is higher than 38 and
Blood pressure is lower than 120

$$Fever > 38 \wedge BloodPr < 120$$

Fever is higher than 38 implies
Blood pressure is lower than 120

$$Fever > 38 \rightarrow BloodPr < 120$$

Fever is higher than 38 and
Blood pressure is lower than 120 or
higher than 145

$$Fever > 38 \wedge$$
$$(BloodPr < 120 \vee BloodPr > 145)$$

Definition

Given a set of n *names of attributes* $\mathcal{A} = \{A_1 \dots A_n\}$*, such that each attribute* A_i *is associated to a finite domain* $dom(A_i) \subset \mathbb{R}$*, the set of* *well-formed propositional (learning) formulas* *is obtained by the grammar*

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi,$$

where $A \in \mathcal{A}$*,* $v \in dom(A)$*, and* $\bowtie \in \{<, \leq, =, \geq, >\}$.

Definition

Given a set of n names of attributes $\mathcal{A} = \{A_1 \dots A_n\}$, such that each attribute A_i is associated to a finite domain $dom(A_i) \subset \mathbb{R}$, the set of well-formed propositional (learning) formulas is obtained by the grammar

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi,$$

where $A \in \mathcal{A}$, $v \in dom(A)$, and $\bowtie \in \{<, \leq, =, \geq, >\}$.

Here and in the rest of this notes, we use propositional letters $(p, q, \dots)$ and atoms $(A_1 \bowtie v_1, A_2 \bowtie v_2, \dots)$ interchangeably. We have to remember that from an inductive point of view, this is trivial; however, from a deductive one, the difference is substantial.

Definition

Given a set of n names of attributes $\mathcal{A} = \{A_1 \ldots A_n\}$, such that each attribute A_i is associated to a finite domain $dom(A_i) \subset \mathbb{R}$, the set of well-formed propositional (learning) formulas is obtained by the grammar

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi,$$

where $A \in \mathcal{A}$, $v \in dom(A)$, and $\bowtie \in \{<, \leq, =, \geq, >\}$.

Here and in the rest of this notes, we use propositional letters $(p, q, \ldots)$ and atoms $(A_1 \bowtie v_1, A_2 \bowtie v_2, \ldots)$ interchangeably. We have to remember that from an inductive point of view, this is trivial; however, from a deductive one, the difference is substantial.

Now, we need to specify how to formally interpret propositional formulas in this setting.

Propositional Logic: Semantics – 1

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

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

If I is a propositional interpretation, then

Propositional Logic: Semantics – 1

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

If I is a propositional interpretation, then

$$\begin{array}{lll} I \models p & \text{iff} & \text{p is true in } I \\ I \models \neg\varphi & \text{iff} & I \not\models \varphi \\ I \models \varphi \vee \psi & \text{iff} & I \models \varphi \text{ or } I \models \psi \end{array}$$

Propositional Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

Propositional Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

If we need to interpret atomic propositions such as $(A \bowtie v)$
we need I to be a set of pairs $(A_1, val), (A_2, val), \ldots$
instead of a set of truth assignments

Propositional Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

If we need to interpret atomic propositions such as $(A \bowtie v)$ we need I to be a set of pairs $(A_1, val), (A_2, val), \ldots$ instead of a set of truth assignments

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \lor \varphi \mid \varphi \land \varphi \mid \varphi \to \varphi$$

For example

Propositional Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

For example

$I_1 : Fever = 38, BloodPr = 120$

Propositional Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

For example

$I_1 : Fever = 38, BloodPr = 120$ satisfies $Fever > 37.5$

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

For example

$I_1 : Fever = 38, BloodPr = 120$ satisfies $Fever > 37.5$

$I_2 : Fever = 37, BloodPr = 110$

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

For example

$I_1 : Fever = 38, BloodPr = 120$	satisfies	$Fever > 37.5$
$I_2 : Fever = 37, BloodPr = 110$	satisfies	$\neg(Fever > 37.5) \wedge$ $(BloodPr > 100)$

Propositional Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

For example

$I_1 : Fever = 38, BloodPr = 120$	satisfies	$Fever > 37.5$
$I_2 : Fever = 37, BloodPr = 110$	satisfies	$\neg(Fever > 37.5) \wedge$ $(BloodPr > 100)$
$I_3 : Fever = 39, BloodPr = 130$		

Propositional Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi$$

For example

$I_1 : Fever = 38, BloodPr = 120$	satisfies	$Fever > 37.5$
$I_2 : Fever = 37, BloodPr = 110$	satisfies	$\neg(Fever > 37.5) \wedge$ $(BloodPr > 100)$
$I_3 : Fever = 39, BloodPr = 130$	satisfies	$Fever > 37.5 \wedge$ $(BloodPr < 140)$

Definition

Given a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ *of attributes, a propositional interpretation* I *is a function set of pairs*

$$I : \mathcal{A} \rightarrow \mathbb{R}$$

A propositional interpretation I *naturally induces the truth relation for a propositional formula* φ, *denoted* $I \models \varphi$, *obtained by applying the rules*

$$\begin{array}{lll} I \models (A \bowtie v) & \textit{iff} & A \bowtie v \textit{ in } I \\ I \models \neg\varphi & \textit{iff} & I \not\models \varphi \\ I \models \varphi \vee \psi & \textit{iff} & I \models \varphi \textit{ or } I \models \psi \\ I \models \varphi \wedge \psi & \textit{iff} & I \models \varphi \textit{ and } I \models \psi \\ I \models \varphi \rightarrow \psi & \textit{iff} & I \not\models \varphi \textit{ or } I \models \psi \end{array}$$

When $I \models \varphi$ *we say that* I *satisfies* φ. *An interpretation that satisfies a formula is said to be a model of that formula.*

Propositional Logic for Learning: Exercise

	A_1	A_2	A_3	A_4
I_1	5	7	10	2
I_2	3	7	12	2
I_3	10	3	6	3
I_4	9	3	1	7
I_5	12	4	6	9

Propositional Logic for Learning: Exercise

	A_1	A_2	A_3	A_4
I_1	5	7	10	2
I_2	3	7	12	2
I_3	10	3	6	3
I_4	9	3	1	7
I_5	12	4	6	9

Propositional Logic for Learning: Exercise

	A_1	A_2	A_3	A_4
I_1	5	7	10	2
I_2	3	7	12	2
I_3	10	3	6	3
I_4	9	3	1	7
I_5	12	4	6	9

$$I_1 \models (A_1 > 4) \wedge (A_3 < 7)$$

Propositional Logic for Learning: Exercise

	A_1	A_2	A_3	A_4
I_1	5	7	10	2
I_2	3	7	12	2
I_3	10	3	6	3
I_4	9	3	1	7
I_5	12	4	6	9

Propositional Logic for Learning: Exercise

	A_1	A_2	A_3	A_4
I_1	5	7	10	2
I_2	3	7	12	2
I_3	10	3	6	3
I_4	9	3	1	7
I_5	12	4	6	9

$$I_3 \not\models (A_1 > 4) \land (A_2 < 5) \land (A_3 > 7)$$

Propositional Logic for Learning: Exercise

	A_1	A_2	A_3	A_4
I_1	5	7	10	2
I_2	3	7	12	2
I_3	10	3	6	3
I_4	9	3	1	7
I_5	12	4	6	9

Propositional Logic for Learning: Exercise

	A_1	A_2	A_3	A_4
I_1	5	7	10	2
I_2	3	7	12	2
I_3	10	3	6	3
I_4	9	3	1	7
I_5	12	4	6	9

$$I_5 \models (A_3 = 1) \wedge (A_4 \geq 2)$$

Propositional logic expresses propositional knowledge. A single interpretation is a fact about the universe we want to learn from, and it establishes a static situation (e.g., *the fever is higher than 38*).

Propositional Logic for Learning: Summary

Propositional logic expresses propositional knowledge. A single interpretation is a fact about the universe we want to learn from, and it establishes a static situation (e.g., *the fever is higher than 38*). Propositional symbolic learning deals with algorithms and methods that learn from sets of propositional instances. The object of learning is both a propositional language (i.e., which are the most important atomic propositions?) and formulas (i.e., how do they combine to express the concept we want to learn?).

Propositional Logic for Learning: Summary

Propositional logic expresses propositional knowledge. A single interpretation is a fact about the universe we want to learn from, and it establishes a static situation (e.g., *the fever is higher than 38*). Propositional symbolic learning deals with algorithms and methods that learn from sets of propositional instances. The object of learning is both a propositional language (i.e., which are the most important atomic propositions?) and formulas (i.e., how do they combine to express the concept we want to learn?). Now, we move our attention on a very general way to express dynamic situations, in order to generalize learning to algorithms and methods that learn from sets of them.

Basic Modal Logic: Quicker and Dirtier

Modal logic generalizes propositional logic by extending its syntax and semantics.

Modal logic, Blackburn, De Rijke and Venema, 2001.

Modal logic generalizes propositional logic by extending its syntax and semantics. In the Kripke semantics, propositions are relativized to worlds, which in turn are connected by relations.

Modal logic, Blackburn, De Rijke and Venema, 2001.

Modal logic generalizes propositional logic by extending its syntax and semantics. In the Kripke semantics, propositions are relativized to worlds, which in turn are connected by relations. So asserting p does not mean p is true anymore; instead, it means p is true in the current world.

Modal logic, Blackburn, De Rijke and Venema, 2001.

Modal logic generalizes propositional logic by extending its syntax and semantics. In the Kripke semantics, propositions are relativized to worlds, which in turn are connected by relations. So asserting p does not mean p is true anymore; instead, it means p is true in the current world. New symbols are introduced to allow one to express facts about other worlds and, essentially, to move the focus of the discourse to other worlds.

Modal logic, Blackburn, De Rijke and Venema, 2001.

Modal logic generalizes propositional logic by extending its syntax and semantics. In the Kripke semantics, propositions are relativized to worlds, which in turn are connected by relations. So asserting p does not mean p is true anymore; instead, it means p is true in the current world. New symbols are introduced to allow one to express facts about other worlds and, essentially, to move the focus of the discourse to other worlds. The apparent abstractness of this setting is due to its generality; modal logic is, in fact, a prototype for many languages, in which the concept of world and the concept of relation(s) between worlds become concrete and useful to describe practical situations.

Modal logic, Blackburn, De Rijke and Venema, 2001.

Basic Modal Logic: Syntax – 1

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

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond\varphi$$

as before, atoms have a structure

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond\varphi$$

this is read
there exists a reachable world in which
φ holds true

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi$$

this is read

for every reachable world

φ holds true, and it is

definable in the original language

$$\Box\varphi \equiv \neg\Diamond\neg\varphi$$

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

as before we use a
non-minimal gramamr for formulas
in order to ease the learning phase

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

How do we express. . . ?

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

How do we express. . . ?

Fever is higher than 38
in the current world

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

How do we express...?

Fever is higher than 38
in the current world

$$Fever > 38$$

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

How do we express…?

Fever is higher than 38
in the current world

$Fever > 38$

There is a reachable world
in which the fever is higher than 38

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

How do we express. . . ?

Fever is higher than 38 in the current world	$Fever > 38$
There is a reachable world in which the fever is higher than 38	$\Diamond(Fever > 38)$

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

How do we express. . . ?

Fever is higher than 38 in the current world	$Fever > 38$
There is a reachable world in which the fever is higher than 38	$\Diamond(Fever > 38)$
For every reachable world there exists a reachable world in which the fever is higher than 38	

Basic Modal Logic: Syntax – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

How do we express...?

Fever is higher than 38 in the current world	$Fever > 38$
There is a reachable world in which the fever is higher than 38	$\Diamond(Fever > 38)$
For every reachable world there exists a reachable world in which the fever is higher than 38	$\Box\Diamond(Fever > 38)$

Basic Modal Logic: Syntax – 2

Definition

Given a set of n names of attributes $\mathcal{A} = \{A_1 \ldots A_n\}$, such that each attribute A_i is associated to a finite domain $dom(A_i) \subset \mathbb{R}$, the set of well-formed modal (learning) formulas is obtained by the grammar

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

where $A \in \mathcal{A}$, $v \in dom(A)$, and $\bowtie \in \{<, \leq, =, \geq, >\}$.

Definition

Given a set of n names of attributes $\mathcal{A} = \{A_1 \ldots A_n\}$, such that each attribute A_i is associated to a finite domain $dom(A_i) \subset \mathbb{R}$, the set of well-formed modal (learning) formulas is obtained by the grammar

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

where $A \in \mathcal{A}$, $v \in dom(A)$, and $\bowtie \in \{<, \leq, =, \geq, >\}$.

The so-called Kripke semantics formalizes the interpretation of a modal logic formula onto a directed graph.

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \Diamond\varphi \mid \Box\varphi$$

Basic Modal Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

$$I :$$

$$\begin{tikzcd}
 & w_1 : \begin{matrix} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{matrix} \arrow[r] & w_3 : \begin{matrix} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{matrix} \\
w_0 : \begin{matrix} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{matrix} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{matrix} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{matrix} &
\end{tikzcd}$$

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \Diamond\varphi \mid \Box\varphi$$

$$I:$$

$$\begin{tikzcd}
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_3 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}$$

$I, w_0 \Vdash A_1$ is higher than 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

$$I:$$

$$\begin{tikzcd}
& w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_3 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
& w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}$$

$$I, w_0 \Vdash A_1 > 1$$

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

$$I:$$

$$\begin{tikzcd}
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_3 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}$$

$I, w_0 \Vdash$ there exists a world with A_1 is higher than 1

Basic Modal Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

$$I:$$

$$\begin{tikzcd}
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_3 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}$$

$$I, w_0 \Vdash \Diamond(A_1 > 1)$$

Basic Modal Logic: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \Diamond\varphi \mid \Box\varphi$$

$$I :$$

$$\begin{tikzcd}
& w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_3 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
& w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}$$

$I, w_0 \Vdash$ for every world, every reachable world has A_2 less than 3

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \Diamond\varphi \mid \Box\varphi$$

$I:$

$$\begin{tikzcd}
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_3 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}$$

$$I, w_0 \Vdash \Box\Box(A_2 < 3)$$

Definition

Given a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ of attributes, a modal interpretation I is a directed graph $I = (W, R)$, where W is a set of worlds and $R \subseteq W \times W$; each world $w \in W$, in turn, is a function

$$w : \mathcal{A} \to \mathbb{R}.$$

A modal interpretation I and a world w in it naturally induce the truth relation for a modal formula φ, denoted $I, w \Vdash \varphi$, obtained by applying the rules

$$
\begin{array}{lll}
I, w \Vdash (A \bowtie v) & \textit{iff} & w(A) \bowtie v \\
I, w \Vdash \neg\varphi & \textit{iff} & I, w \nVdash \varphi \\
I, w \Vdash \varphi \vee \psi & \textit{iff} & I, w \Vdash \varphi \textit{ or } I \Vdash \psi \\
I, w \Vdash \varphi \wedge \psi & \textit{iff} & I, w \Vdash \varphi \textit{ and } I \Vdash \psi \\
I, w \Vdash \varphi \to \psi & \textit{iff} & I, w \nVdash \varphi \textit{ or } I \Vdash \psi \\
I, w \Vdash \Diamond\varphi & \textit{iff} & \textit{there is } w' \textit{ s.t. } wRw' \textit{ and } I, w' \Vdash \varphi \\
I, w \Vdash \Box\varphi & \textit{iff} & \textit{for every } w' \textit{ s.t. } wRw' \textit{ it happens } I, w' \Vdash \varphi
\end{array}
$$

Again, if $I, w \Vdash \varphi$, then I satisfies φ at w, and I is a model of φ.

Since modal logic is more complex than propositional one
a few observations are in order

Basic Modal Logic: Semantics – 3

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

are true on every world
with no successors

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

are true on every world
with no successors

$\Diamond\top$

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

are true on every world
with no successors

$\Diamond\top$

forces the existence
of a successors

Basic Modal Logic: Semantics – 3

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

are true on every world
with no successors

$\Box(p \to q) \to (\Box p \to \Box q)$

$\Diamond\top$

forces the existence
of a successors

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

are true on every world
with no successors

$\Box(p \to q) \to (\Box p \to \Box q)$

is true on every world
of every interpretation

$\Diamond\top$

forces the existence
of a successors

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

are true on every world
with no successors

$\Box(p \to q) \to (\Box p \to \Box q)$

is true on every world
of every interpretation

$\Diamond\top$

forces the existence
of a successors

$$\Diamond(p \wedge q) \leftrightarrow \Diamond p \wedge \Diamond q$$
$$\Box(p \vee q) \leftrightarrow \Box p \vee \Box q$$
$$\Box(p \to q) \leftrightarrow \Box p \to \Box q$$

Basic Modal Logic: Semantics – 3

Since modal logic is more complex than propositional one
a few observations are in order

$\Box\bot, \Box p$

are true on every world
with no successors

$\Box(p \to q) \to (\Box p \to \Box q)$

is true on every world
of every interpretation

$\Diamond\top$

forces the existence
of a successors

$$\Diamond(p \wedge q) \leftrightarrow \Diamond p \wedge \Diamond q$$
$$\Box(p \vee q) \leftrightarrow \Box p \vee \Box q$$
$$\Box(p \to q) \leftrightarrow \Box p \to \Box q$$

are false on some world
in some interpretation

Basic Modal Logic for Learning: Exercise – 1

$$I:$$

$$
\begin{tikzcd}
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_4 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}
$$

Is it true that

$$I:$$

$$
\begin{tikzcd}
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_4 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}
$$

Is it true that

$$I, w_0 \Vdash \Diamond\Diamond(A_4 > 1)?$$

Basic Modal Logic for Learning: Exercise – 1

$I:$

$$
\begin{tikzcd}[column sep=small]
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_4 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}
$$

Is it true that

$I, w_0 \Vdash \Diamond\Diamond(A_4 > 1)$? Yes

$$I:$$

$$
\begin{tikzcd}
& w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_4 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
& w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}
$$

Is it true that

$I, w_0 \Vdash \Box\Diamond\top$?

$I:$

$$w_1: \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \longrightarrow w_4: \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array}$$

$$\nearrow$$

$$w_0: \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array}$$

$$\searrow$$

$$w_2: \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array}$$

Is it true that
$I, w_0 \Vdash \Box\Diamond\top$?

No

$$I:$$

$$
\begin{array}{cccc}
 & & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} & \longrightarrow w_4 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
 & \nearrow & & \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} & & & \\
 & \searrow & & \\
 & & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{array}
$$

Is it true that

$$I, w_0 \Vdash \Box(A_1 > 6 \rightarrow \Diamond(A_4 > 3))?$$

I :

$$\begin{tikzcd}[column sep=small]
 & w_1 : \begin{array}{ll} A_1 = 2 & A_3 = 1 \\ A_2 = 5 & A_4 = 4 \end{array} \arrow[r] & w_4 : \begin{array}{ll} A_1 = 3 & A_3 = 6 \\ A_2 = 1 & A_4 = 2 \end{array} \\
w_0 : \begin{array}{ll} A_1 = 3 & A_3 = 4 \\ A_2 = 1 & A_4 = 6 \end{array} \arrow[ur] \arrow[dr] & & \\
 & w_2 : \begin{array}{ll} A_1 = 4 & A_3 = 2 \\ A_2 = 2 & A_4 = 5 \end{array} &
\end{tikzcd}$$

Is it true that

$I, w_0 \Vdash \Box(A_1 > 6 \rightarrow \Diamond(A_4 > 3))$? Yes

$$
\begin{tikzcd}
 & w_1 : A_1 = 3 \arrow[r] & w_3 : A_1 = 4 \\
I_1 : \; w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & \\
 & w_2 : A_1 = 1 &
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & w_1 : A_1 = 4 \arrow[r] & w_3 : A_1 = 1 \\
I_2 : \; w_0 : A_1 = 3 \arrow[ur] \arrow[dr] & & \\
 & w_2 : A_1 = 2 &
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & w_1 : A_1 = 1 \arrow[r] & w_3 : A_1 = 2 \\
I_3 : \; w_0 : A_1 = 4 \arrow[ur] \arrow[dr] & & \\
 & w_2 : A_1 = 5 &
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & w_1 : A_1 = 2 \arrow[r] & w_3 : A_1 = 3 \\
I_4 : \; w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & \\
 & w_2 : A_1 = 3 &
\end{tikzcd}
$$

$$\begin{array}{ll} & w_1 : A_1 = 3 \to w_3 : A_1 = 4 \\ & \nearrow \\ I_1 : w_0 : A_1 = 2 & \\ & \searrow \\ & w_2 : A_1 = 1 \end{array}$$

$$\begin{array}{ll} & w_1 : A_1 = 4 \to w_3 : A_1 = 1 \\ & \nearrow \\ I_2 : w_0 : A_1 = 3 & \\ & \searrow \\ & w_2 : A_1 = 2 \end{array}$$

$$\begin{array}{ll} & w_1 : A_1 = 1 \to w_3 : A_1 = 2 \\ & \nearrow \\ I_3 : w_0 : A_1 = 4 & \\ & \searrow \\ & w_2 : A_1 = 5 \end{array}$$

$$\begin{array}{ll} & w_1 : A_1 = 2 \to w_3 : A_1 = 3 \\ & \nearrow \\ I_4 : w_0 : A_1 = 2 & \\ & \searrow \\ & w_2 : A_1 = 3 \end{array}$$

$$I_1, w_0 \Vdash \Box\Box(A_1 > 3)$$

$$\begin{tikzcd}[row sep=small] & w_1 : A_1 = 3 \arrow[r] & w_3 : A_1 = 4 \\ I_1 : w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & \\ & w_2 : A_1 = 1 & \end{tikzcd}$$

$$\begin{tikzcd}[row sep=small] & w_1 : A_1 = 4 \arrow[r] & w_3 : A_1 = 1 \\ I_2 : w_0 : A_1 = 3 \arrow[ur] \arrow[dr] & & \\ & w_2 : A_1 = 2 & \end{tikzcd}$$

$$I_2, w_3 \Vdash \Box(A_1 < 4)$$

$$\begin{tikzcd}[row sep=small] & w_1 : A_1 = 1 \arrow[r] & w_3 : A_1 = 2 \\ I_3 : w_0 : A_1 = 4 \arrow[ur] \arrow[dr] & & \\ & w_2 : A_1 = 5 & \end{tikzcd}$$

$$\begin{tikzcd}[row sep=small] & w_1 : A_1 = 2 \arrow[r] & w_3 : A_1 = 3 \\ I_4 : w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & \\ & w_2 : A_1 = 3 & \end{tikzcd}$$

Basic Modal Logic for Learning: Exercise – 2

$$\begin{array}{ll} & w_1 : A_1 = 3 \to w_3 : A_1 = 4 \\ & \nearrow \\ I_1 : w_0 : A_1 = 2 & \\ & \searrow \\ & w_2 : A_1 = 1 \end{array}$$

$$\begin{array}{ll} & w_1 : A_1 = 4 \to w_3 : A_1 = 1 \\ & \nearrow \\ I_2 : w_0 : A_1 = 3 & \\ & \searrow \\ & w_2 : A_1 = 2 \end{array}$$

$$\begin{array}{ll} & w_1 : A_1 = 1 \to w_3 : A_1 = 2 \\ & \nearrow \\ I_3 : w_0 : A_1 = 4 & \\ & \searrow \\ & w_2 : A_1 = 5 \end{array}$$

$$I_3, w_0 \nVdash \Box\Diamond(A_1 = 2)$$

$$\begin{array}{ll} & w_1 : A_1 = 2 \to w_3 : A_1 = 3 \\ & \nearrow \\ I_4 : w_0 : A_1 = 2 & \\ & \searrow \\ & w_2 : A_1 = 3 \end{array}$$

Basic Modal Logic for Learning: Summary

Now we know how to formalize the idea that the knowledge we want to express is dynamic. In order to keep things general enough, we did not established what a Kripke graph represents.

Basic Modal Logic for Learning: Summary

Now we know how to formalize the idea that the knowledge we want to express is dynamic. In order to keep things general enough, we did not established what a Kripke graph represents. This problem will be considered in the next lesson, when we shall concretize basic modal logic into more useful languages that represent practical situations.

Basic Modal Logic for Learning: Summary

Now we know how to formalize the idea that the knowledge we want to express is dynamic. In order to keep things general enough, we did not established what a Kripke graph represents. This problem will be considered in the next lesson, when we shall concretize basic modal logic into more useful languages that represent practical situations. It is important, however, to understand that most of the important properties emerge already at the level of the basic language, which is why we focus on it at first.

Modal logic can be multi-modal. In multi-modal logic there is a set of diamonds $\langle R_1 \rangle, \langle R_2 \rangle, \ldots$, associated to a set of relations $R_1, R_2, \ldots$. Multi-modal logics are interpreted on multi-graphs, in which worlds are connected by more than one relation.

Modal logic can be multi-modal. In multi-modal logic there is a set of diamonds $\langle R_1\rangle, \langle R_2\rangle, \ldots$, associated to a set of relations $R_1, R_2, \ldots$. Multi-modal logics are interpreted on multi-graphs, in which worlds are connected by more than one relation. Interval temporal logic, which we shall see next as an example of useful modal logic, is, in fact, multi-modal; more in general, all useful modal logics are.

Practice!

Symbolic knowledge manipulation with SoleLogics.jl

- Manipulating formulas (composing, parsing, generating);
- Checking modal and propositional formulas on models;
- Using formulas for representing and checking symbolic knowledge.

Notebook

`Day1-symbolic-knowledge.ipynb`

Basic Symbolic Learning with Decision Trees

How Do We Learn Symbolically?

Modern AI

How Do We Learn Symbolically?

Machine Learning

How Do We Learn Symbolically?

$$\begin{tikzcd}
 & \text{Machine Learning} \arrow[dl] \arrow[dr] & \\
\text{Sub-Symbolic Learning} & & \text{Symbolic Learning}
\end{tikzcd}$$

How Do We Learn Symbolically?

Machine Learning

Sub-Symbolic Learning

Symbolic Learning

Regression

How Do We Learn Symbolically?

Machine Learning

Sub-Symbolic Learning

Symbolic Learning

Linear regression
Logistic regression
SVN
Neural networks
. . .

How Do We Learn Symbolically?

$$\begin{tikzcd}[column sep=small] & \text{Machine Learning} \arrow[dl] \arrow[dr] & \\ \text{Sub-Symbolic Learning} & & \text{Symbolic Learning} \end{tikzcd}$$

Sub-Symbolic Learning:

Linear regression
Logistic regression
SVN
Neural networks
...

Symbolic Learning:

Rule-based systems

How Do We Learn Symbolically?

Machine Learning

Sub-Symbolic Learning

Symbolic Learning

Linear regression
Logistic regression
SVN
Neural networks
...

Rule-based learning
Tree-based learning

How Do We Learn Symbolically?

Machine Learning

Sub-Symbolic Learning

- Linear regression
- Logistic regression
- SVN
- Neural networks
- ...

Symbolic Learning

- Rule-based systems
- Decision trees
- Random forests
- Boosted trees
- Association rules
- ...

How Do We Learn Symbolically?

Machine Learning

Sub-Symbolic Learning

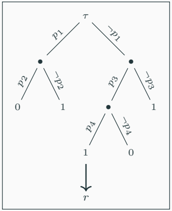
Symbolic Learning

Linear regression
Logistic regression
SVN
Neural networks
...

Rule-based systems
Decision trees
Random forests
Boosted trees
Association rules
...

Decision Trees as Representatives for Symbolic Learning

Decision tree



Decision Trees as Representatives for Symbolic Learning

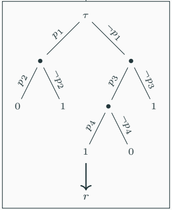
Decision list

$p_1 \wedge p_2$	$\Rightarrow$	0
$p_1 \wedge \neg p_2$	$\Rightarrow$	1
$\neg p_1 \wedge p_3 \wedge p_4$	$\Rightarrow$	1
$\neg p_1 \wedge p_3 \wedge \neg p_4$	$\Rightarrow$	0
$\neg p_1 \wedge \neg p_3$	$\Rightarrow$	1
$\top$	$\Rightarrow$	0

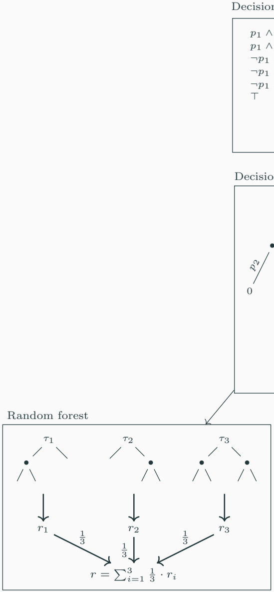
$\downarrow$

r

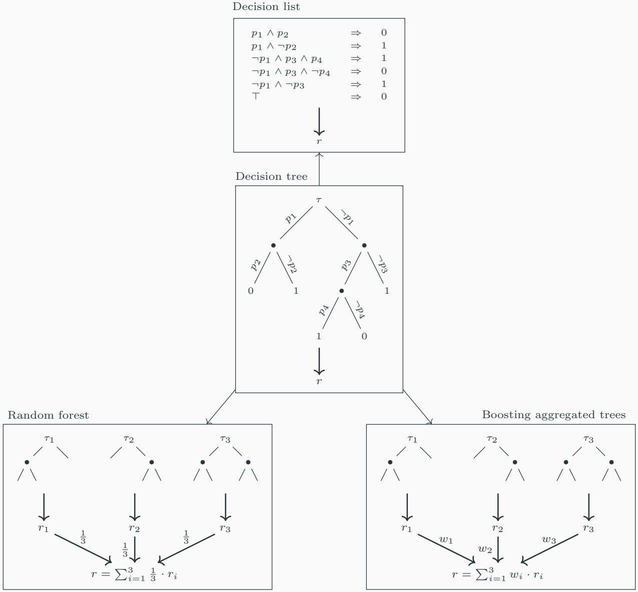
Decision tree



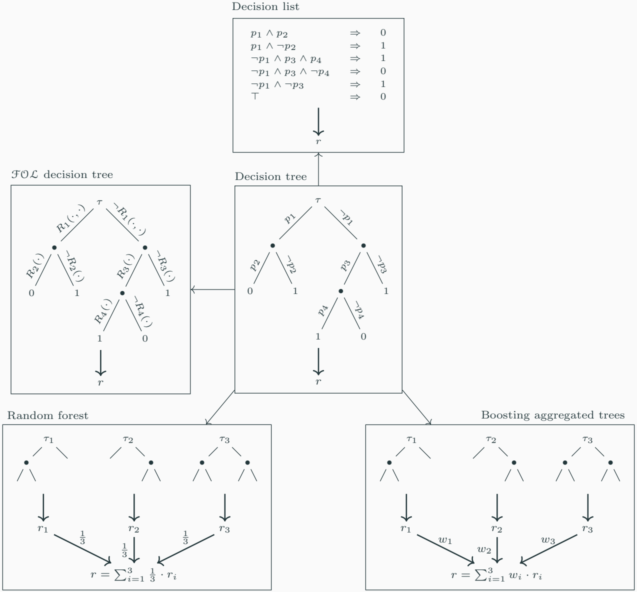
Decision Trees as Representatives for Symbolic Learning



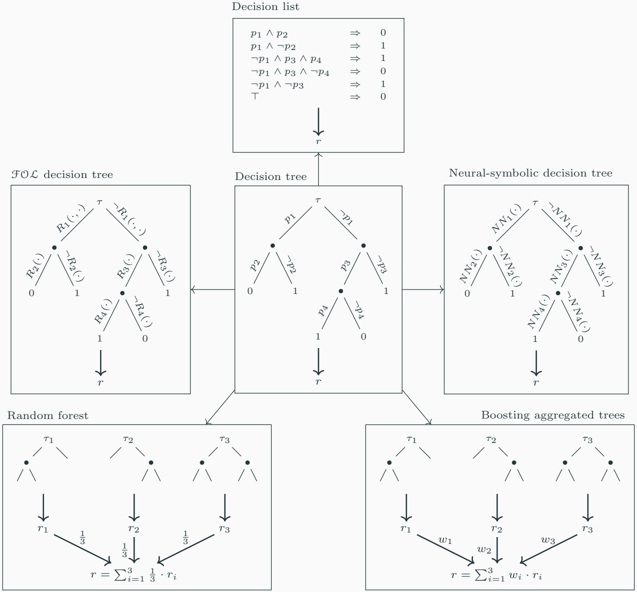
Decision Trees as Representatives for Symbolic Learning



Decision Trees as Representatives for Symbolic Learning



Decision Trees as Representatives for Symbolic Learning



Types of Learning

Learning

Types of Learning

$$\text{Learning} \longrightarrow \text{Supervised}$$

Types of Learning

Supervised — When datasets are labeled

Learning → Supervised

Types of Learning

$$\begin{tikzcd} & \text{Supervised} \\ \text{Learning} \arrow[ur] \arrow[r] & \text{Unsupervised} \end{tikzcd}$$

When datasets are labeled

$$
\begin{tikzcd}
 & \text{Supervised} \\
\text{Learning} \arrow[ur] \arrow[r] & \text{Unsupervised}
\end{tikzcd}
$$

Supervised: When datasets are labeled

Unsupervised: When datasets are not labeled or we ignore labels

Types of Learning

Learning → Supervised: When datasets are labeled

Learning → Unsupervised: When datasets are not labeled or we ignore labels

Learning → Semi-supervised

Types of Learning

Learning → Supervised: When datasets are labeled

Learning → Unsupervised: When datasets are not labeled or we ignore labels

Learning → Semi-supervised: When datasets are partially labeled e.g., one or more classes are missing

Types of Learning

Learning → Supervised: When datasets are labeled

decision trees are representative of this type

Learning → Unsupervised: When datasets are not labeled or we ignore labels

Learning → Semi-supervised: When datasets are partially labeled e.g., one or more classes are missing

Tabular/Propositional Dataset

Definition

Given a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ of attributes/variables, a tabular instance I is a function

$$I : \mathcal{A} \to \mathbb{R}$$

A tabular dataset, or propositional dataset is a set $\mathcal{I} = \{I_1, \ldots, I_m\}$ of tabular instances. A tabular dataset is labelled if and only if each instance is associated to a unique class or label from a set $\mathcal{L} = \{L_1, \ldots, L_k\}$.

Symbolic Classification Problem

Given a labelled dataset $\mathcal{I}$, the classification problem is the problem of devising an algorithm that correctly classifies the instances of a dataset $\mathcal{J}$ drawn from the same distribution as $\mathcal{I}$.

Symbolic Classification Problem

Given a labelled dataset $\mathcal{I}$, the classification problem is the problem of devising an algorithm that correctly classifies the instances of a dataset $\mathcal{J}$ drawn from the same distribution as $\mathcal{I}$. We say that the classifier has been learned from $\mathcal{I}$ (and tested on $\mathcal{J}$).

Symbolic Classification Problem

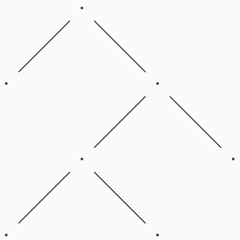
Given a labelled dataset $\mathcal{I}$, the classification problem is the problem of devising an algorithm that correctly classifies the instances of a dataset $\mathcal{J}$ drawn from the same distribution as $\mathcal{I}$. We say that the classifier has been learned from $\mathcal{I}$ (and tested on $\mathcal{J}$). The learning theory, orthogonal to the symbolic or sub-symbolic setting, offers a set of techniques to establish the quality of a classifier.

Symbolic Classification Problem

Given a labelled dataset $\mathcal{I}$, the classification problem is the problem of devising an algorithm that correctly classifies the instances of a dataset $\mathcal{J}$ drawn from the same distribution as $\mathcal{I}$. We say that the classifier has been learned from $\mathcal{I}$ (and tested on $\mathcal{J}$). The learning theory, orthogonal to the symbolic or sub-symbolic setting, offers a set of techniques to establish the quality of a classifier. Typical classification problems assume non-numerical classes; numerical classes are associated to regression problems but classification and regression can be dealt with in similar ways.

Here we focus on non-numerical symbolic classification via decision trees.

Propositional Decision Trees

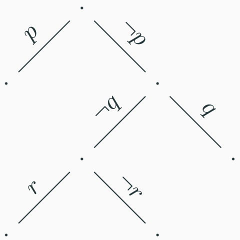


A propositional decision tree is a tree-shaped object

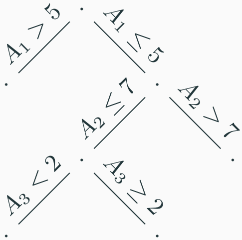
in which edges are propositional letters called

s

Propositional Decision Trees

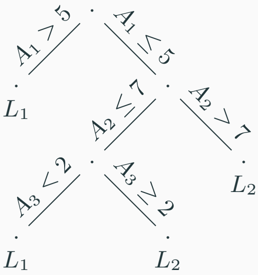


A propositional decision tree is a tree-shaped object
in which edges are propositional letters called decisions



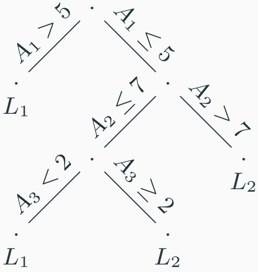
A propositional decision tree is a tree-shaped object in which edges are propostional atoms called decisions

Propositional Decision Trees



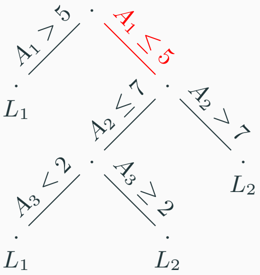
A propositional decision tree is a tree-shaped object in which edges are propostional atoms called decisions and leaves are classes

Propositional Decision Trees



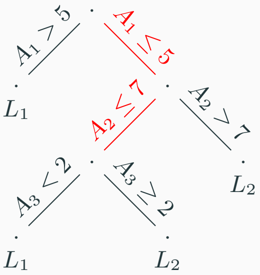
A tabular instance I is classified by a decision tree
by examining each decision progressively

Propositional Decision Trees



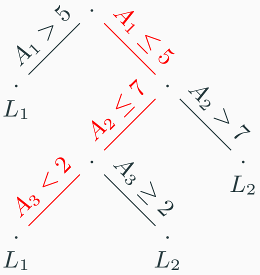
A tabular instance I is classified by a decision tree
by examining each decision progressively
thus building a propositional formula: $(A_1 \leq 5)$

Propositional Decision Trees



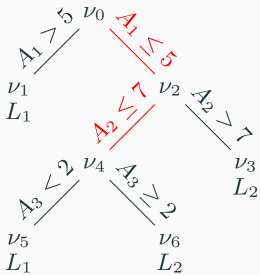
A tabular instance I is classified by a decision tree
by examining each decision progressively
thus building a propositional formula: $(A_1 \leq 5) \wedge (A_2 \leq 7)$

Propositional Decision Trees



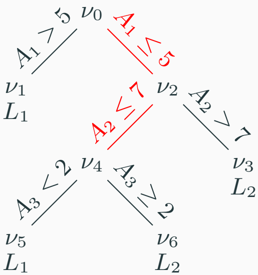
A tabular instance I is classified by a decision tree
by examining each decision progressively
thus building a propositional formula: $(A_1 \leq 5) \wedge (A_2 \leq 7) \wedge (A_3 > 2) \Rightarrow L_2$

Propositional Decision Trees



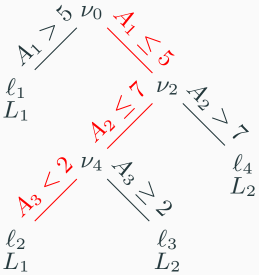
An object such as $\nu_0 \rightsquigarrow \nu_4$ is called
a path

Propositional Decision Trees



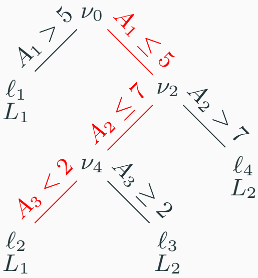
An object such as $\nu_0 \rightsquigarrow \nu_4$ is called
a path and it is associated to a path-formula;
in this example $\varphi_{\nu_0 \rightsquigarrow \nu_4} = (A_1 \leq 5) \wedge (A_2 \leq 7)$

Propositional Decision Trees



An object such as $\nu_0 \rightsquigarrow \ell_2$ is called
a branch

Propositional Decision Trees



An object such as $\nu_0 \rightsquigarrow \ell_2$ is called
a branch and it is associated to a branch-formula;
in this example $\varphi_{\nu_0 \rightsquigarrow \ell_2} = (A_1 \leq 5) \wedge (A_2 \leq 7) \wedge (A_2 < 2)$

Propositional Decision Trees, Formally – 1

Let $\tau = (\mathcal{V}, \mathcal{E})$ be a full directed binary tree. We denote by $\mathcal{V}^{\ell}$ the set of its leaves, by $\mathcal{V}^{\iota}$ the set of its internal nodes (i.e., non-root and non-leaf nodes), and by $\rho(\tau)$ its root.

Propositional Decision Trees, Formally – 1

Let $\tau = (\mathcal{V}, \mathcal{E})$ be a full directed binary tree. We denote by $\mathcal{V}^{\ell}$ the set of its leaves, by $\mathcal{V}^{\iota}$ the set of its internal nodes (i.e., non-root and non-leaf nodes), and by $\rho(\tau)$ its root. Each non-leaf node ν left (resp. right) child $\swarrow(\nu)$ (resp., $\searrow(\nu)$), and each non-root node ν has a parent $\uparrow(\nu)$. For a node ν, the set of its ancestors (ν included) is denoted by $\uparrow^{*}(\nu)$, where $\uparrow^{*}$ is the transitive and reflexive closure of $\uparrow$; we also define $\uparrow^{+}(\nu) = \uparrow^{*}(\nu) \setminus \{\nu\}$, and we say that if $\nu' \in \uparrow^{*}(\nu)$, then ν is a descendant of ν'.

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

Propositional Decision Trees, Formally – 2

Definition

Let $\mathcal{L}$ be a set of classes and $\mathcal{P}$ a finite set of propositional letters, and define the set of propositional decisions as

$$\Lambda = \{p, \neg p \mid p \in \mathcal{P}\}.$$

Then, a propositional decision tree (over $\mathcal{L}$) is a tuple

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

where $(\mathcal{V}, \mathcal{E})$ is a full binary directed tree, $l : \mathcal{V}^{\ell} \rightarrow \mathcal{L}$ is a leaf-labelling function that assigns a class from $\mathcal{L}$ to each leaf node in $\mathcal{V}^{\ell}$, and $e : \mathcal{E} \rightarrow \Lambda$ is an edge-labelling function that assigns a propositional decision from Λ to each edge in $\mathcal{E}$, such that $e(\nu, \swarrow(\nu)) \equiv \neg e(\nu, \searrow(\nu))$ for all non-leaf nodes ν. The family of propositional decision trees is denoted by $\mathcal{DT}$.

Pure Propositional Decision Trees: Note on the Alphabet

A decision tree is independent from the dataset from which it is learned. Thus it can be defined by fixing a set of abstract propositional letters that we call decisions.

A decision tree is independent from the dataset from which it is learned. Thus it can be defined by fixing a set of abstract propositional letters that we call decisions. If such letters are fixed before the learning phase, we say that to the dataset is a applied a filter (in machine learning terms);

Pure Propositional Decision Trees: Note on the Alphabet

A decision tree is independent from the dataset from which it is learned. Thus it can be defined by fixing a set of abstract propositional letters that we call decisions. If such letters are fixed before the learning phase, we say that to the dataset is a applied a filter (in machine learning terms); otherwise the letters (which are in fact propositional atoms) are learned during the learning phase.

Pure Propositional Decision Trees: Note on the Alphabet

A decision tree is independent from the dataset from which it is learned. Thus it can be defined by fixing a set of abstract propositional letters that we call decisions. If such letters are fixed before the learning phase, we say that to the dataset is a applied a filter (in machine learning terms); otherwise the letters (which are in fact propositional atoms) are learned during the learning phase. From the syntactical point of view of decision trees there is no difference. If we want to solve deduction problems with the learned formulas, however, we shall have to solve a satisfiability modulo theory problem.

Propositional Decision Trees, Formally – 3

Definition

For a path $\pi = \nu_0 \rightsquigarrow \nu_h$ in a propositional decision tree τ, the path-formula φ_π is defined as

$$\varphi_\pi = \bigwedge_{0 \leq i < h} e(\nu_i, \nu_{i+1}).$$

Similarly, for a leaf ℓ the leaf-formula φ_ℓ is defined as

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

Finally, for a class $L \in \mathcal{L}$ the class-formula φ_L is defined as

$$\varphi_L = \bigvee_{\{\ell \in \mathcal{V}^\ell \mid l(\ell) = L\}} \varphi_{\pi_\ell}.$$

Propositional Decision Trees, Formally – 4

Definition

Given a propositional decision tree τ, a node $\nu \in \tau$, a tabular dataset $\mathcal{I}$, and an instance $\mathfrak{I}$ in $\mathcal{I}$, the run of τ on $\mathfrak{I}$ from ν, denoted by $\tau(\mathfrak{I}, \nu)$, is defined as follows:

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

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

Definition

A decision tree τ is said to be optimal for a labelled structured dataset $\mathcal{I}$ if

$$\tau(\mathfrak{I}) = \mathcal{L}(\mathfrak{I})$$

for every instance $\mathfrak{I} \in \mathcal{I}$.

Definition

A family of decision trees is correct if and only if every tree classifies every instance of any labelled structured dataset into exactly one class. Furthermore, it is complete if and only if it contains an optimal decision tree for every labelled structured dataset. Finally, it is efficient if and only if there exists a polynomial-time algorithm that, given a labelled structured dataset, learns an optimal decision tree for it.

Propositional Decision Trees: Intuition – 1

Propositional decision trees are correct by definition of run: as edges are always opposite to each other, an instance can only be classified in a single way. They are also complete because every class in a propositional dataset can be captured by a propositional formula; in turn, every propositional formula can be expressed as a DNF, and every DNF corresponds to a propositional decision tree. How to prove that they are also efficient?

Propositional Decision Trees: Intuition – 2

It should be noticed that, in our context, optimal decision trees are not necessarily minimal. It is well-known that learning a minimal optimal decision tree is a NP-hard problem. The idea underlying optimality is observing that a decision tree has the ability to capture every propositionally-defined class. Learning a decision tree is an efficient, statistical process that aims at obtaining statistically well-behaved, although non-optimal, decision trees from a dataset; the fact that we can trivially modify such a learning algorithm to obtain an optimal (but not minimally so) tree is the important aspect that the concept of efficiency intends to highlight.

The most typical approach to 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 way that the quantity of information (e.g., the entropy) carried by each (dataset associated to a) node decreases.

Propositional Decision Trees: Learning – 1

The most typical approach to 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 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 based on the purity of the node (i.e., the measure of the distribution of classes in the node) is met, in which case the node becomes a leaf. For classification problems, 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; the process of devising an optimal decision tree being (well-knowingly) polynomial in the number of instances proves the fact that the family $\mathcal{DT}$ is efficient.

Propositional Decision Trees: Learning – 2

Algorithms based on this idea are generally called information-based algorithms, the most influential being CART. Despite having been introduced in the seventies, recent results show that its performances are still superior to those of more recent proposals. Their polynomial complexity (irrespective of the stopping criteria) is also based on assuming that the number n of attributes is fixed a priori, irrespective from the number m of instances.

Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e)$ be a propositional decision tree, ν a node of τ, and $\mathcal{I}$ a labelled dataset. Then, the ***ν-dataset*** *is defined as*

$$\mathcal{I}_\nu = \{\mathfrak{I} \in \mathcal{I} \mid \mathfrak{I} \Vdash \bigwedge_{\nu' \in \pi_\nu} \varphi_{\pi_{\nu'}}\}.$$

Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e)$ be a propositional decision tree, $\mathcal{I}$ a labelled dataset, ν a non-leaf node of τ. Then, the ***(binary) split*** *of $\mathcal{I}_\nu$ is the pair defined as*

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

Propositional Decision Trees: Learning, Formally – 2

Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e)$ be a modal decision tree, $\mathcal{I}$ a labelled modal dataset, ν a non-leaf node of τ whose ν-dataset is defined, and $(\mathcal{I}_{\swarrow(\nu)}, \mathcal{I}_{\searrow(\nu)})$ the split of $\mathcal{I}_\nu$. The split information of $\mathcal{I}_\nu$ is defined as:

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

Propositional Decision Trees: Learning, Formally – 3

The idea of the algorithm $CART$ is simple. Let $\mathcal{I}$ be a dataset with m instances. Then:

- If a stopping condition applies, then return;
- Find the best split $(\mathcal{I}_1, \mathcal{I}_2)$, and call $CART$ recursively on $\mathcal{I}_1$ and $\mathcal{I}_2$.

This process is polynomial and returns an optimal decision tree is the stopping conditions are precise.

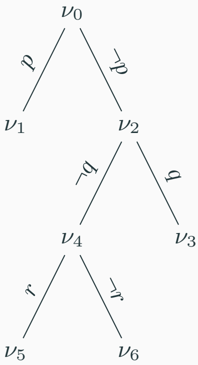
https://mlu-explain.github.io/decision-tree/

Propositional Decision Trees, Formally – 6

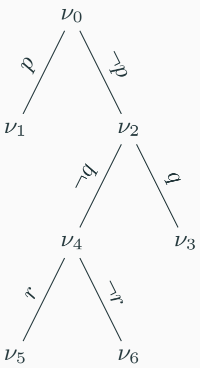
Theorem

The family $\mathcal{DT}$ of (pure) propositional decision trees is correct, complete, and efficient.

Pure Propositional Decision Trees

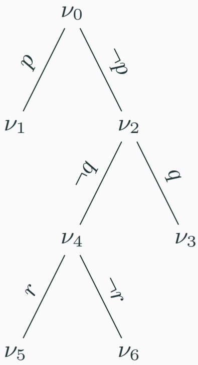


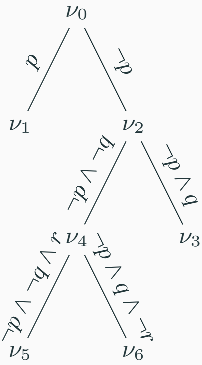
Pure Propositional Decision Trees



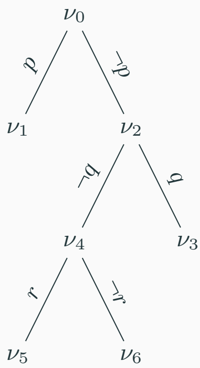
Path-formulas can be accumulated on the edges

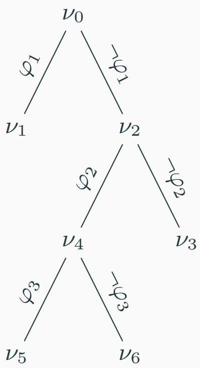
Pure Propositional Decision Trees





Pure Propositional Decision Trees





We call this a pure decision tree

Pure Propositional Decision Trees, Formally – 1

Definition

Let $\mathcal{L}$ be a set of classes and $\mathcal{P}$ a finite set of propositional letters, and define the set Φ of conjunctions that can be built on it. Then, a pure propositional decision tree (over $\mathcal{L}$) is a tuple

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

where $(\mathcal{V}, \mathcal{E})$ is a full binary directed tree, $l : \mathcal{V}^{\ell} \to \mathcal{L}$ is a leaf-labelling function that assigns a class from $\mathcal{L}$ to each leaf node in $\mathcal{V}^{\ell}$, and $e : \mathcal{E} \to \Phi$ is an edge-labelling function that assigns a propositional formula from Φ to each edge in $\mathcal{E}$, such that $e(\nu, \swarrow(\nu)) \equiv \neg e(\nu, \searrow(\nu))$ for all non-leaf nodes ν.

Pure Propositional Decision Trees, Formally – 2

We use pure decision trees as we use non-pure ones; the concepts of optimal tree, correct, complete, and efficient family of trees transfer immediately. But non-pure decision trees allow one to see the detail of learning algorithms, and appreciate the ideas under locally optimal ones. Starting from non-pure decision trees, stepping to pure ones is trivial.

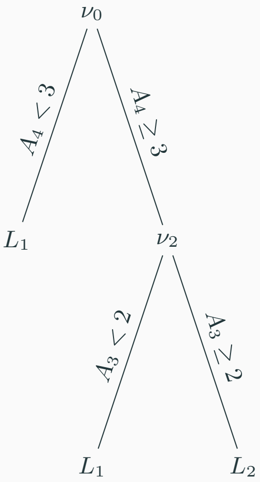
(Pure) Propositional Decision Trees: Exercise

	A_1	A_2	A_3	A_4	label
I_1	5	7	10	2	L_1
I_2	3	7	12	2	L_1
I_3	10	3	6	3	L_2
I_4	9	3	1	7	L_1
I_5	12	4	6	9	L_2

(Pure) Propositional Decision Trees: Exercise

	A_1	A_2	A_3	A_4	label
I_1	5	7	10	2	L_1
I_2	3	7	12	2	L_1
I_3	10	3	6	3	L_2
I_4	9	3	1	7	L_1
I_5	12	4	6	9	L_2

Draw an optimal decision tree for this tabular dataset



	A_1	A_2	A_3	A_4	label
I_1	5	7	10	2	L_1
I_2	3	7	12	2	L_1
I_3	10	3	6	3	L_2
I_4	9	3	1	7	L_1
I_5	12	4	6	9	L_2

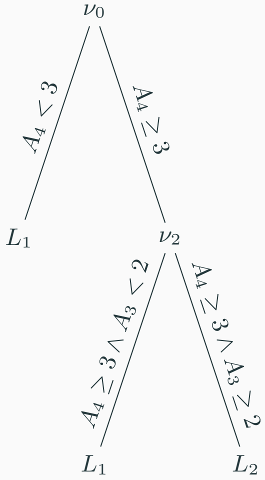
(Pure) Propositional Decision Trees: Exercise

	A_1	A_2	A_3	A_4	label
I_1	5	7	10	2	L_1
I_2	3	7	12	2	L_1
I_3	10	3	6	3	L_2
I_4	9	3	1	7	L_1
I_5	12	4	6	9	L_2

Draw the same tree in its pure version

(Pure) Propositional Decision Trees: Exercise

	A_1	A_2	A_3	A_4	label
I_1	5	7	10	2	L_1
I_2	3	7	12	2	L_1
I_3	10	3	6	3	L_2
I_4	9	3	1	7	L_1
I_5	12	4	6	9	L_2



Day 2

Modal Decision Trees, Not So Quick

Modal symbolic learning is a framework that encompasses the generalization of symbolic learning ideas and tools to the modal level.

Decision Trees with a Modal Flavor, *Proceedings of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA)*, Della Monica et al., 2022.

Modal symbolic learning is a framework that encompasses the generalization of symbolic learning ideas and tools to the modal level. We shall define modal data as a way to interpret non-tabular data (which itself is a generalization of tabular data), and, then, the concept of modal learning will emerge naturally.

Decision Trees with a Modal Flavor, *Proceedings of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA)*, Della Monica et al., 2022.

Modal symbolic learning is a framework that encompasses the generalization of symbolic learning ideas and tools to the modal level. We shall define modal data as a way to interpret non-tabular data (which itself is a generalization of tabular data), and, then, the concept of modal learning will emerge naturally. Within it, we shall focus on modal symbolic classification (as a supervised problem) and, more in particular, on modal decision trees.

Decision Trees with a Modal Flavor, *Proceedings of the 21st International Conference of the Italian Association for Artificial Intelligence (AIxIA)*, Della Monica et al., 2022.

Tabular data

Instance: a set of real values

Modal Symbolic Learning: Introduction – 2

propositional data

$\uparrow$

Tabular data

Instance: a set of real values

trivial intepretation

e.g.: $p = A \bowtie v$

Non-tabular data

Instance: a time series, an image, a graph,. . .

modal data

↑

non-trivial intepretation
obtained via a transformer
that depends on the specific
modal logic used

Non-tabular data
Instance: a time series, an image, a graph,. . .

Modal Dataset

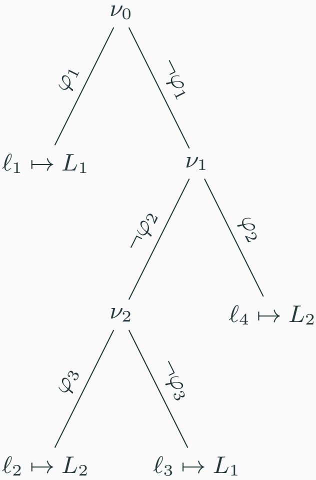
Definition

Given a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ of attributes, a modal instance I is a directed graph $I = (W, R)$, where W is a set of worlds and $R \subseteq W \times W$; each world $w \in W$, in turn, is a function

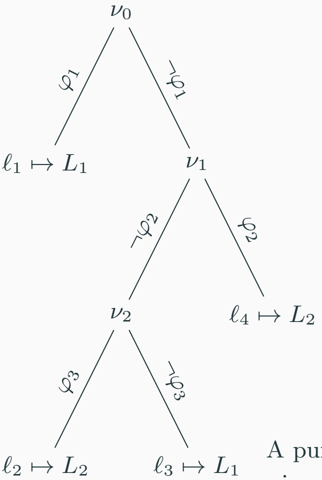
$$w : \mathcal{A} \rightarrow \mathbb{R}.$$

A modal dataset is a set $\mathcal{I} = \{I_1, \ldots, I_m\}$ of modal instances. A modal dataset is labelled if and only if each instance is associated to a unique class or label from a set $\mathcal{L} = \{L_1, \ldots, L_k\}$.

Pure Modal Decision Trees

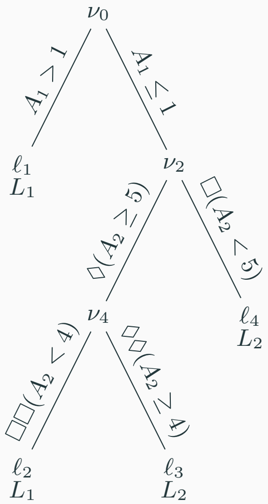


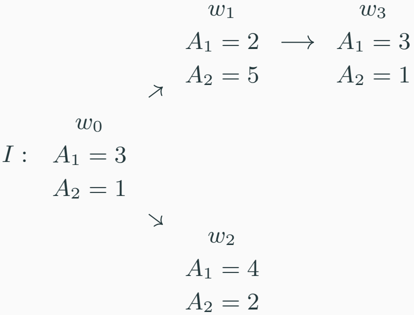
Pure Modal Decision Trees

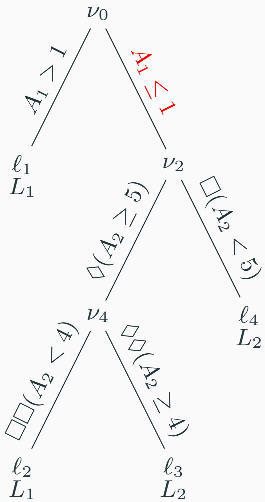


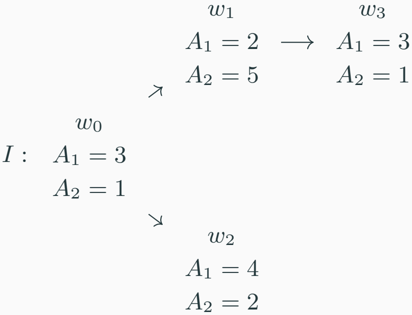
A pure modal decision tree is a tree-shaped object in which edges are labeled with modal formulas

Pure Modal Decision Trees

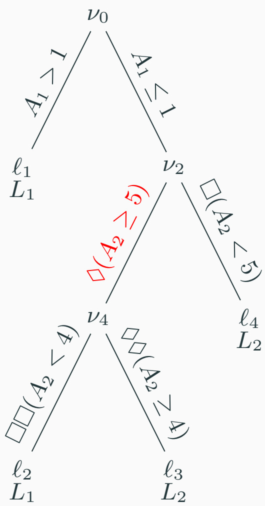


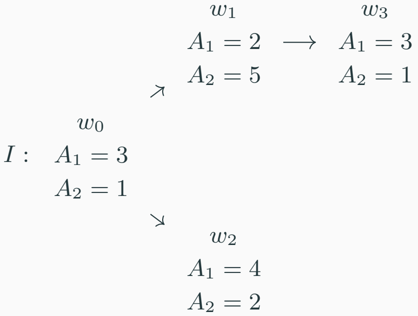




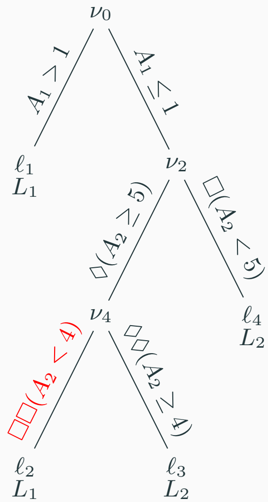


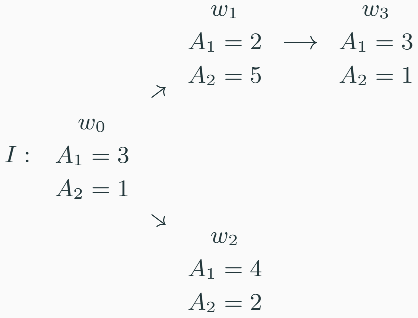
$$I, w_0 \Vdash A_1 \leq 1$$





$$I, w_0 \Vdash \Diamond(A_2 \geq 5)$$





$$I, w_0 \Vdash \Box\Box(A_2 < 3)$$

Pure Modal Decision Trees, Formally – 1

Definition

Let $\mathcal{L}$ *be a set of classes and* $\mathcal{P}$ *a finite set of propositional letters, and define the set* Φ *of* *modal formulas* *from it. Then, a* *pure modal decision tree* *(over* $\mathcal{L}$*) is a tuple*

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

where $(\mathcal{V}, \mathcal{E})$ *is a full binary directed tree,* $l : \mathcal{V}^{\ell} \to \mathcal{L}$ *is a* *leaf-labelling function* *that assigns a class from* $\mathcal{L}$ *to each leaf node in* $\mathcal{V}^{\ell}$*, and* $e : \mathcal{E} \to \Phi$ *is an* *edge-labelling function* *that assigns a propositional formula from* Φ *to each edge in* $\mathcal{E}$*, such that* $e(\nu, \swarrow(\nu)) \equiv \neg e(\nu, \searrow(\nu))$ *for all non-leaf nodes* ν*. The family of pure modal decision trees is denoted by* $\mathcal{MDT}$*.*

Pure Modal Decision Trees, Formally – 2

Observe now that within pure modal decision trees, the concepts of correctness of a tree, and completeness and efficiency of a family of trees can be inherited from the propositional case as is.

Observe now that within pure modal decision trees, the concepts of correctness of a tree, and completeness and efficiency of a family of trees can be inherited from the propositional case as is. Unfortunately, an efficient learning algorithm for pure modal decision trees cannot be immediately designed. To this end, it is necessary to build a framework for (non-pure) modal decision trees, in which atomic elements (not necessarily atoms) decorate the edges in such a way that path- and branch-formulas can be iteratively built in such a way to guarantee the correctness of each single tree and the completeness of the family.

Pure Modal Decision Trees, Formally – 2

Observe now that within pure modal decision trees, the concepts of correctness of a tree, and completeness and efficiency of a family of trees can be inherited from the propositional case as is. Unfortunately, an efficient learning algorithm for pure modal decision trees cannot be immediately designed. To this end, it is necessary to build a framework for (non-pure) modal decision trees, in which atomic elements (not necessarily atoms) decorate the edges in such a way that path- and branch-formulas can be iteratively built in such a way to guarantee the correctness of each single tree and the completeness of the family.

In a sense, we proceed backwards: we know that we want to obtain pure modal decision trees, and we go back to defining non-pure ones with this aim.

Definition

Let $\mathcal{L}$ be a set of classes and $\mathcal{P}$ a set of propositions, and define the set of modal decisions:

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

Then, a modal decision tree (over $\mathcal{L}$) is an object of the type:

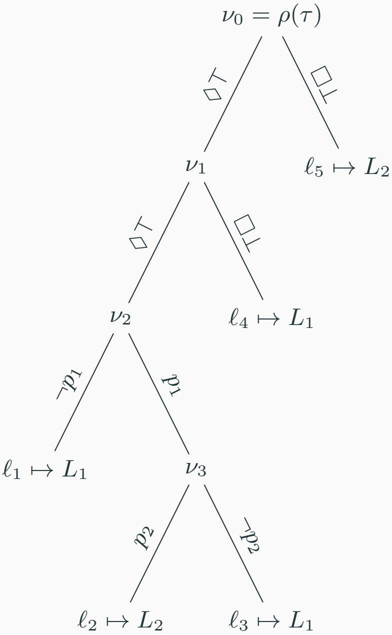
$$\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f),$$

where $(\mathcal{V}, \mathcal{E}, l, e)$ is a propositional decision tree, $b : \mathcal{V}^\iota \to \mathcal{V}^\iota$ 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 $\nu, \nu'\nu'' \in \mathcal{V}$:

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

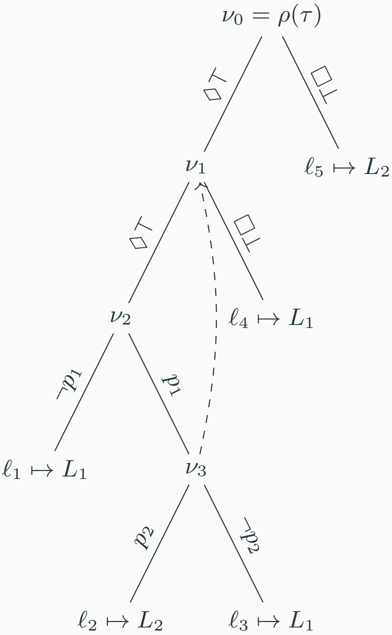
[1] Only listed here for completeness of treatment.

Modal Decision Trees

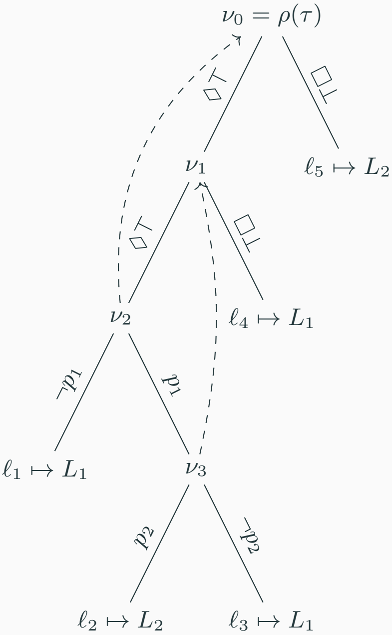


non-displayed backward- and forward-edges are self-loops

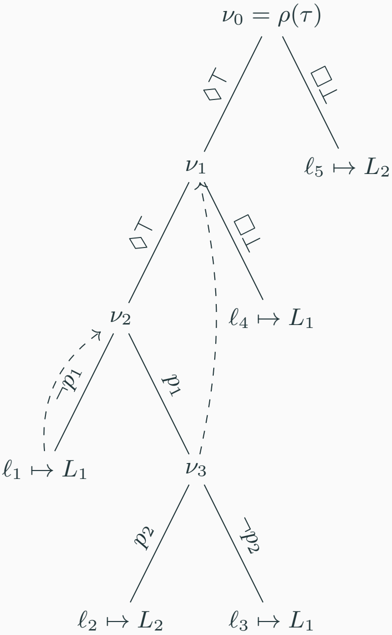
Modal Decision Trees



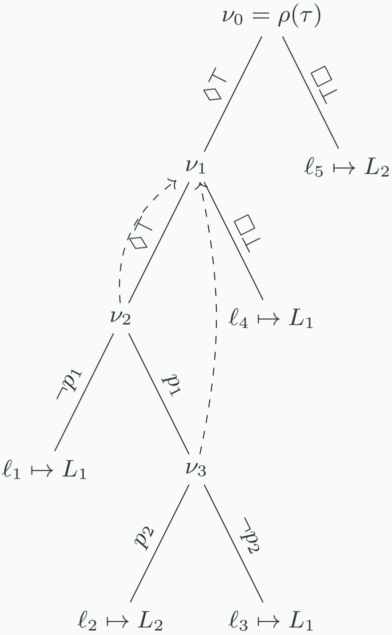
this is a non self-loop backward-edge



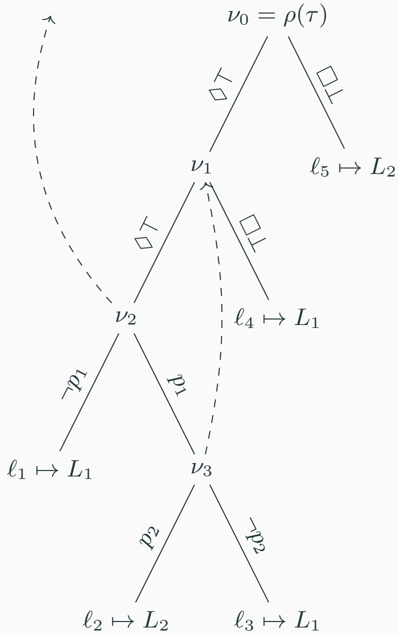
this is not allowed
(def. of backward-edge)



this is not allowed
(def. of backward-edge)



this is not allowed
(Condition 2)



this is not allowed
(Condition 3)

Modal Decision Trees, Formally – 2

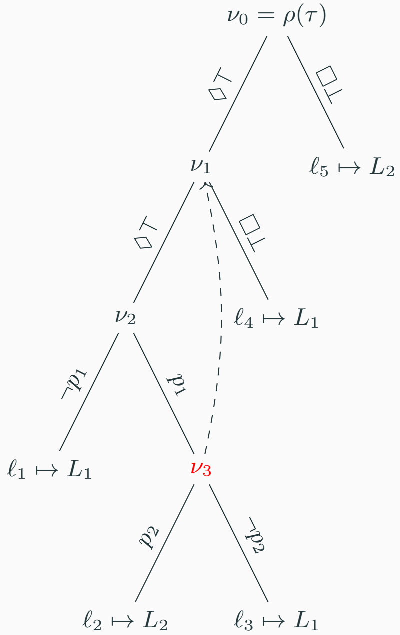
Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a modal decision tree, and $\pi = \nu_0 \rightsquigarrow \nu_h$ be a path in τ of length greater than 1. 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.

Definition

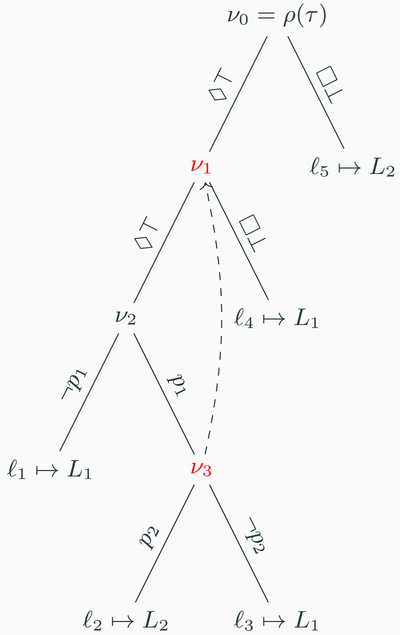
Let $\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a modal decision tree and $\pi = \nu_0 \rightsquigarrow \nu_h$ be a path in τ of length greater than 1. Then, given two nodes $\nu_i, \nu_j \in \pi$, with $i, j < h$, we say that they agree, denoted by $\mathfrak{A}(\nu_i, \nu_j)$, if $\nu_{i+1} = \swarrow(\nu_i)$ (resp., $\nu_{i+1} = \searrow(\nu_i)$) and $\nu_{j+1} = \swarrow(\nu_j)$ (resp., $\nu_{j+1} = \searrow(\nu_j)$); otherwise, we say that they disagree, denoted by $\mathfrak{D}(\nu_i, \nu_j)$.

Modal Decision Trees

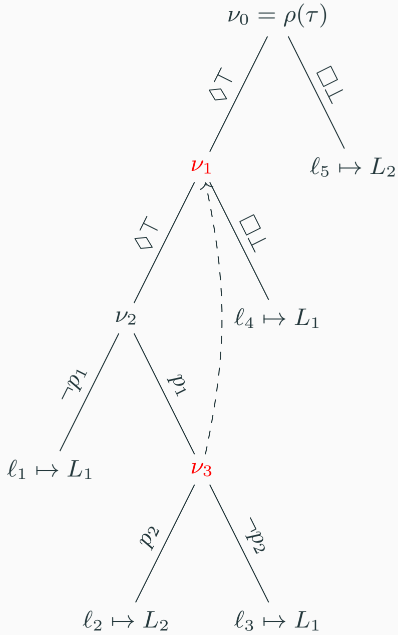


ν_3 is the contributor
of the path $\nu_0 \rightsquigarrow \ell_2$

Modal Decision Trees



ν_1 and ν_3
agree on the path
$\nu_0 \rightsquigarrow \ell_2$



ν_1 and ν_3
disagree on the path
$\nu_0 \rightsquigarrow \ell_3$

Modal Decision Trees, Formally – 3

Definition

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.

Definition

A modal formula φ *is* *implicative* *if it has the form* $\varphi_1 \to \varphi_2$ *or* $\Box(\varphi_1 \to \varphi_2)$*, and we denote by* Im *the set of* *implicative formulas*.

Implicative and non-implicative formulas are simply a tool to distinguish two types of modal formulas, and it will come handy as we start building path- and leaf-formulas.

In modal decision trees, we build path- and leaf-formulas from bottom up.

Modal Decision Trees: Intuition – 1

In modal decision trees, we build path- and leaf-formulas from bottom up. Agreement and disagreement, implicative and non-implicative formulas, and the type of decision λ on an edge can be combined into several combinations.

Modal Decision Trees: Intuition – 1

In modal decision trees, we build path- and leaf-formulas from bottom up. Agreement and disagreement, implicative and non-implicative formulas, and the type of decision λ on an edge can be combined into several combinations. Each combination will recursively give rise to a different modal formula, and we shall show that the combinations are enough to cover the entire space of semantically different modal formulas for a given alphabet.

Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a modal decision tree. Then, for each path $\pi^\tau = \nu_0 \rightsquigarrow \nu_h$ in τ, the path-formula φ^τ_π (or, simply, φ_π if τ is clear from the context) is defined inductively as:

- *if $h = 0$, then $\varphi^\tau_\pi = \top$;*
- *if $h = 1$, then $\varphi^\tau_\pi = e(\nu_0, \nu_1)$;*
- *if $h > 1$, $\lambda = e(\nu_0, \nu_1), \pi^\tau_1 = \nu_1 \rightsquigarrow \zeta(\pi^\tau)$, and $\pi^\tau_2 = \zeta(\pi^\tau) \rightsquigarrow \nu_h$, then:*

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

Modal Decision Trees, Formally – 5

Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a modal decision tree. For each leaf $\ell \in \mathcal{V}^{\ell}$, the leaf-formula φ_{ℓ}^{τ} (or, simply, φ_{ℓ} if τ is clear from the context) is defined as:

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

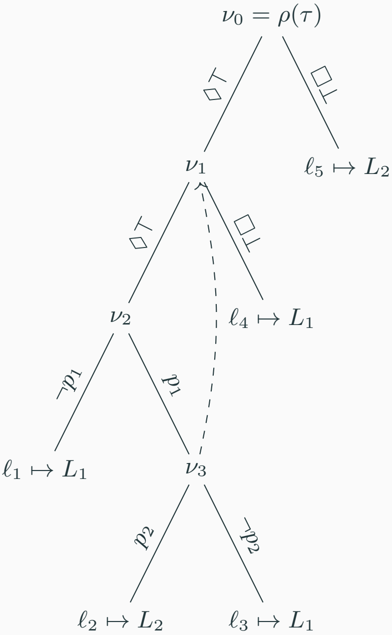
Then, for each class L, the class-formula φ_{L}^{τ} (or, simply, φ_{L} if τ is clear from the context) is defined as:

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

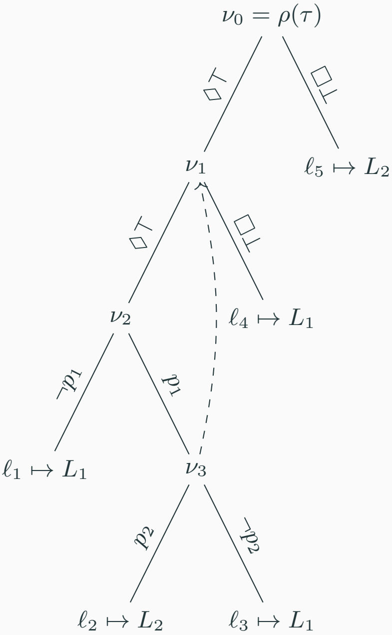
Resulting grammar for path- (φ_{π}), leaf- (φ_{ℓ}), and class-formulas (φ_{L}):

$$\varphi_{\pi} = \begin{cases} \varphi_{\pi} ::= & \lambda \wedge (\varphi_{\pi} \wedge \varphi_{\pi}) \quad | \\ & \lambda \rightarrow (\varphi_{\pi} \rightarrow \varphi_{\pi}) \quad | \\ & \Diamond(\varphi_{\pi} \wedge \varphi_{\pi}) \quad | \\ & \Box(\varphi_{\pi} \rightarrow \varphi_{\pi}) \\ \varphi_{\ell} ::= & \varphi_{\pi} \mid \varphi_{\pi} \wedge \varphi_{\pi} \\ \varphi_{L} ::= & \varphi_{\ell} \mid \varphi_{\ell} \vee \varphi_{\ell} \end{cases}$$

Modal Decision Trees

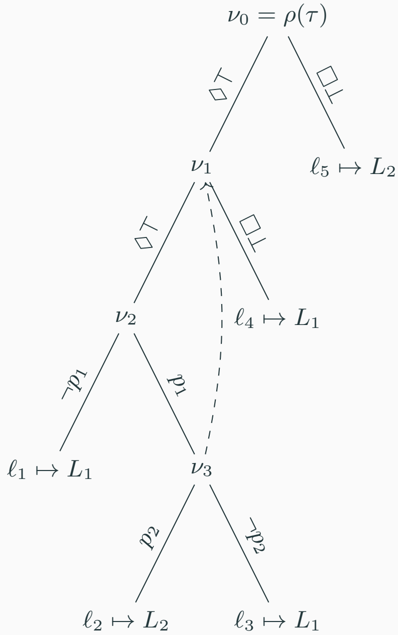


Modal Decision Trees

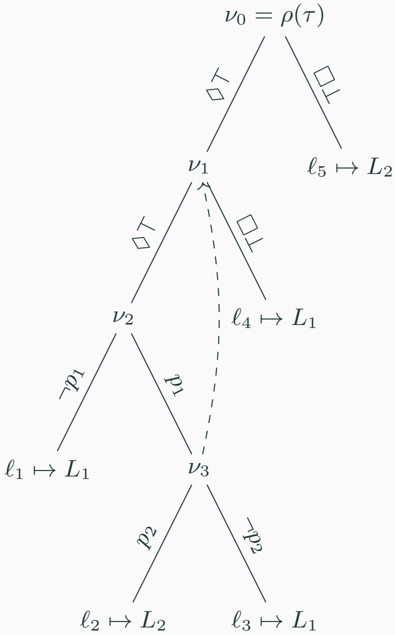


$$\varphi_{\nu \rightsquigarrow \nu} \quad = \quad \top, \text{ for all } \nu \in \mathcal{V},$$

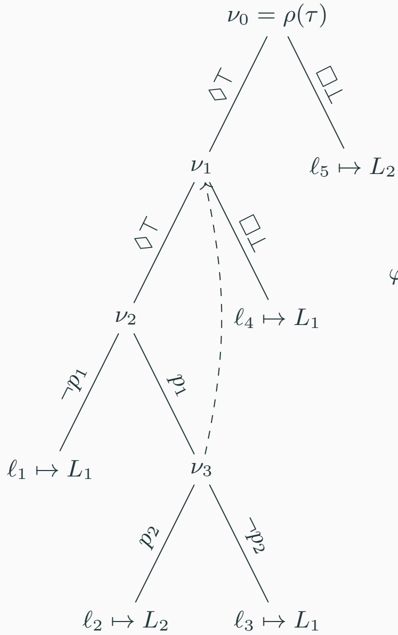
Modal Decision Trees



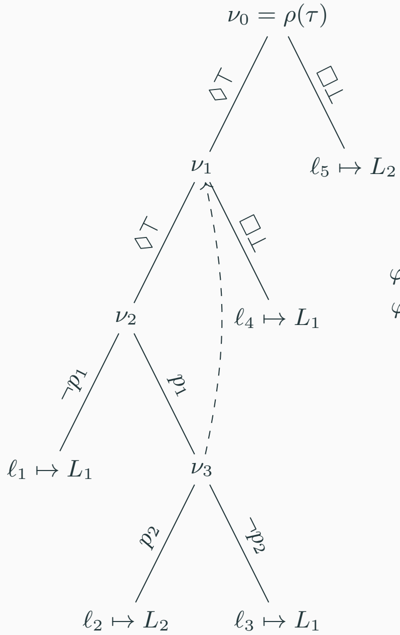
$$
\begin{aligned}
\varphi_{\nu \rightsquigarrow \nu} &= \top, \text{ for all } \nu \in \mathcal{V}, \\
\varphi_{\rotatebox{90}{$\rightsquigarrow$}(\nu) \rightsquigarrow \nu} &= e(\rotatebox{90}{$\rightsquigarrow$}(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau)
\end{aligned}
$$



$$\begin{aligned}
\varphi_{\nu \rightsquigarrow \nu} &= \top, \text{ for all } \nu \in \mathcal{V}, \\
\varphi_{\wr(\nu) \rightsquigarrow \nu} &= e(\wr(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau) \\
\varphi_{\nu_1 \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \neg p_1),
\end{aligned}$$

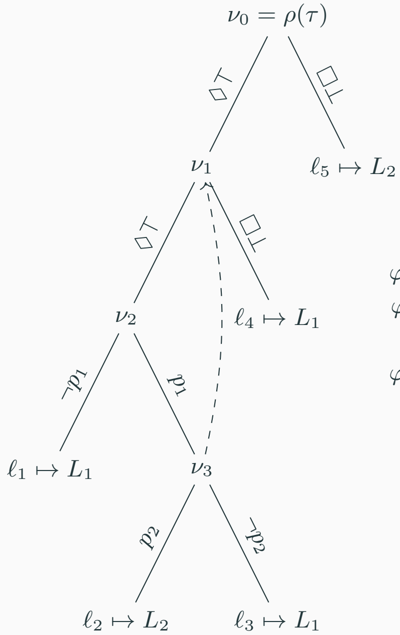


$$
\begin{aligned}
\varphi_{\nu \rightsquigarrow \nu} &= \top, \text{ for all } \nu \in \mathcal{V}, \\
\varphi_{\wr(\nu) \rightsquigarrow \nu} &= e(\wr(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau) \\
\varphi_{\nu_1 \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \neg p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_2} &= \Diamond(\top \wedge \Diamond\top),
\end{aligned}
$$



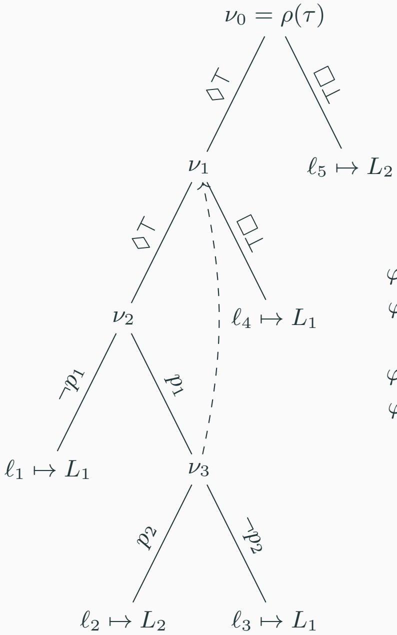
$$
\begin{aligned}
\varphi_{\nu \rightsquigarrow \nu} &= \top, \text{ for all } \nu \in \mathcal{V}, \\
\varphi_{\wr(\nu) \rightsquigarrow \nu} &= e(\wr(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau) \\
\varphi_{\nu_1 \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \neg p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_2} &= \Diamond(\top \wedge \Diamond\top), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \Diamond(\top \wedge \neg p_1)), \\
\varphi_{\nu_1 \rightsquigarrow \nu_3} &= \Box(\top \rightarrow p_1),
\end{aligned}
$$

Modal Decision Trees

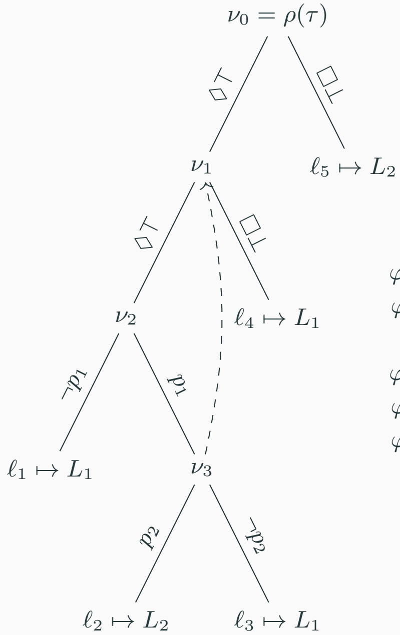


$$
\begin{aligned}
\varphi_{\nu \rightsquigarrow \nu} &= \top, \text{ for all } \nu \in \mathcal{V}, \\
\varphi_{\wr(\nu) \rightsquigarrow \nu} &= e(\wr(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau) \\
\varphi_{\nu_1 \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \neg p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_2} &= \Diamond(\top \wedge \Diamond\top), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \Diamond(\top \wedge \neg p_1)), \\
\varphi_{\nu_1 \rightsquigarrow \nu_3} &= \Box(\top \rightarrow p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_3} &= \Box(\top \rightarrow \Box(\top \rightarrow p_1)),
\end{aligned}
$$

Modal Decision Trees

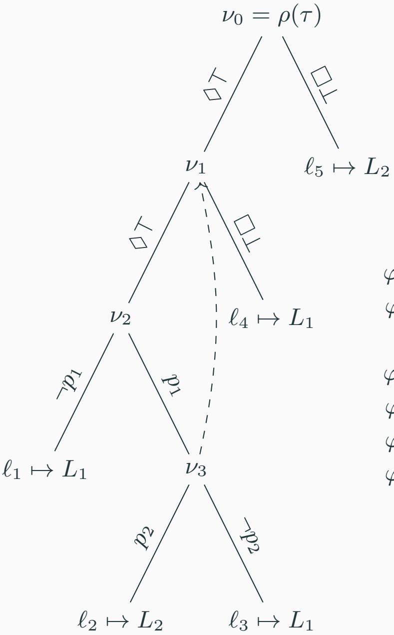


$$
\begin{aligned}
\varphi_{\nu \rightsquigarrow \nu} &= \top, \text{ for all } \nu \in \mathcal{V}, \\
\varphi_{\wr(\nu) \rightsquigarrow \nu} &= e(\wr(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau)\} \\
\varphi_{\nu_1 \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \neg p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_2} &= \Diamond(\top \wedge \Diamond\top), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \Diamond(\top \wedge \neg p_1)), \\
\varphi_{\nu_1 \rightsquigarrow \nu_3} &= \Box(\top \to p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_3} &= \Box(\top \to \Box(\top \to p_1)), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_2} &= \Diamond(\Box(\top \to p_1) \wedge p_2),
\end{aligned}
$$

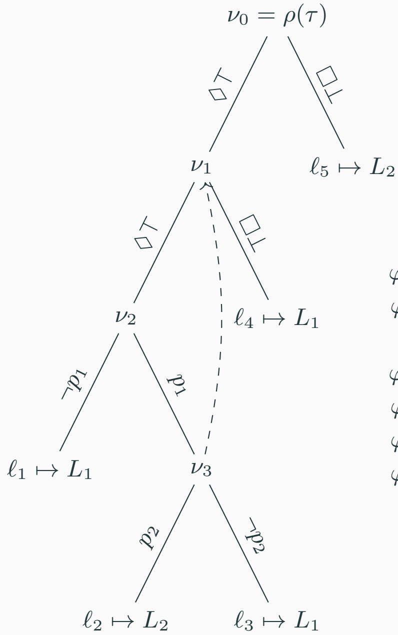


$$
\begin{aligned}
\varphi_{\nu\rightsquigarrow\nu} &= \top, \text{ for all } \nu \in \mathcal{V},\\
\varphi_{\uparrow(\nu)\rightsquigarrow\nu} &= e(\uparrow(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau)\\
\varphi_{\nu_1\rightsquigarrow\ell_1} &= \Diamond(\top \wedge \neg p_1),\\
\varphi_{\rho(\tau)\rightsquigarrow\nu_2} &= \Diamond(\top \wedge \Diamond\top),\\
\varphi_{\rho(\tau)\rightsquigarrow\ell_1} &= \Diamond(\top \wedge \Diamond(\top \wedge \neg p_1)),\\
\varphi_{\nu_1\rightsquigarrow\nu_3} &= \Box(\top \rightarrow p_1),\\
\varphi_{\rho(\tau)\rightsquigarrow\nu_3} &= \Box(\top \rightarrow \Box(\top \rightarrow p_1)),\\
\varphi_{\rho(\tau)\rightsquigarrow\ell_2} &= \Diamond(\Box(\top \rightarrow p_1) \wedge p_2),\\
\varphi_{\rho(\tau)\rightsquigarrow\ell_3} &= \Box(\Box(\top \rightarrow p_1) \rightarrow \neg p_2),
\end{aligned}
$$

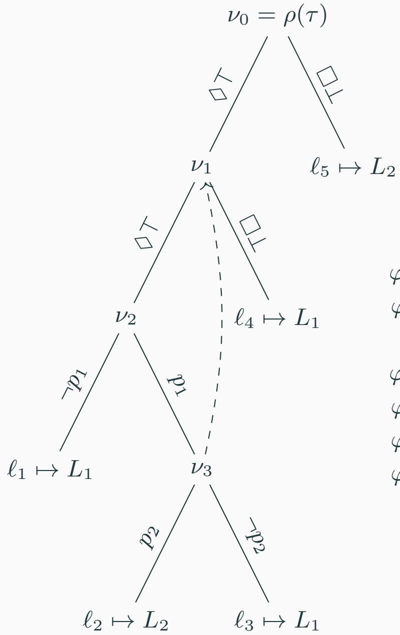
Modal Decision Trees



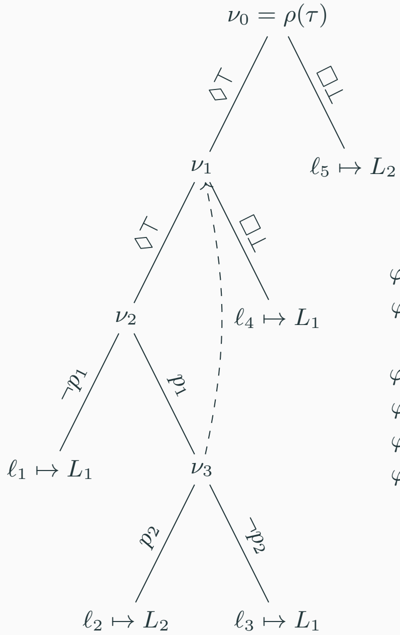
$$
\begin{aligned}
\varphi_{\nu \rightsquigarrow \nu} &= \top, \text{ for all } \nu \in \mathcal{V}, \\
\varphi_{\wr(\nu) \rightsquigarrow \nu} &= e(\wr(\nu), \nu), \text{ for all } \nu \in \mathcal{V} \setminus \{\rho(\tau)\}, \\
\varphi_{\nu_1 \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \neg p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_2} &= \Diamond(\top \wedge \Diamond\top), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_1} &= \Diamond(\top \wedge \Diamond(\top \wedge \neg p_1)), \\
\varphi_{\nu_1 \rightsquigarrow \nu_3} &= \Box(\top \rightarrow p_1), \\
\varphi_{\rho(\tau) \rightsquigarrow \nu_3} &= \Box(\top \rightarrow \Box(\top \rightarrow p_1)), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_2} &= \Diamond(\Box(\top \rightarrow p_1) \wedge p_2), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_3} &= \Box(\Box(\top \rightarrow p_1) \rightarrow \neg p_2), \\
\varphi_{\rho(\tau) \rightsquigarrow \ell_4} &= \Box(\top \rightarrow \Box\bot),
\end{aligned}
$$



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



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



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

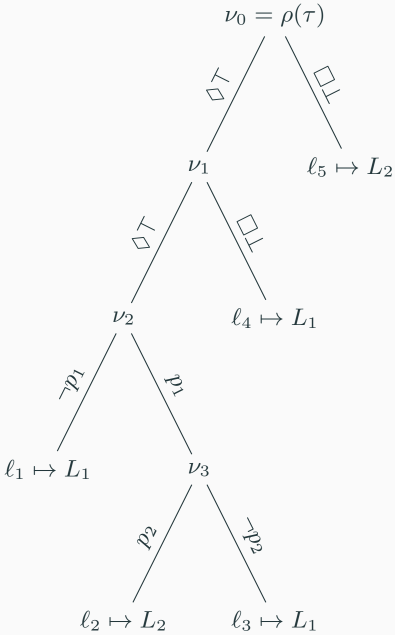
Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ *be a modal decision tree,* ν *a node in* τ*, and* $\mathfrak{I}$ *an instance. Then, the run of* τ *on* $\mathfrak{I}$ *from* ν*, denoted by* $\tau(\mathfrak{I}, \nu)$*, is defined as:*

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

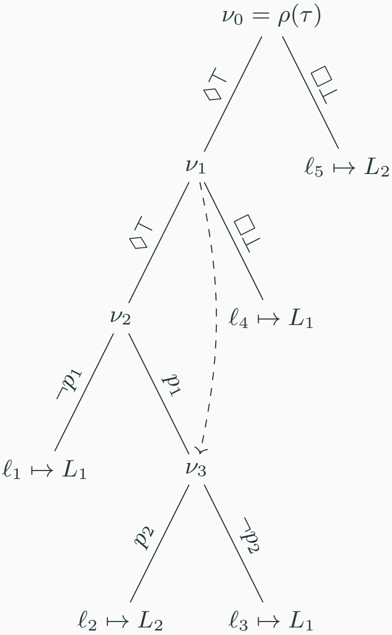
The run of τ *on* $\mathfrak{I}$*, denoted by* $\tau(\mathfrak{I})$*, is defined as* $\tau(\mathfrak{I}, \rho(\tau))$*. Moreover,* $\mathfrak{I}$ *is* classified into $L \in \mathcal{L}$ *by* τ *if and only if* $\tau(\mathfrak{I}) = L$*.*

Modal Decision Trees

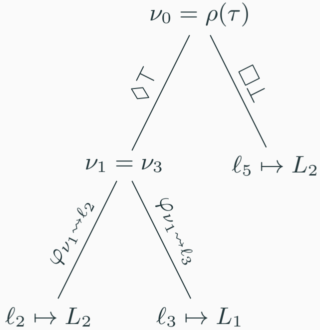


non-displayed backward-
and forward-edges
are self-loops

Modal Decision Trees

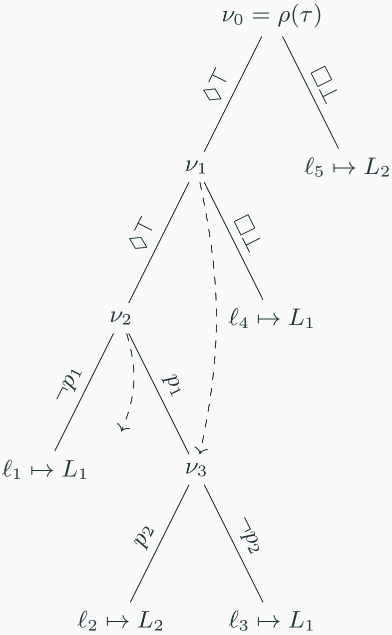


this is a non self-loop
forward-edge

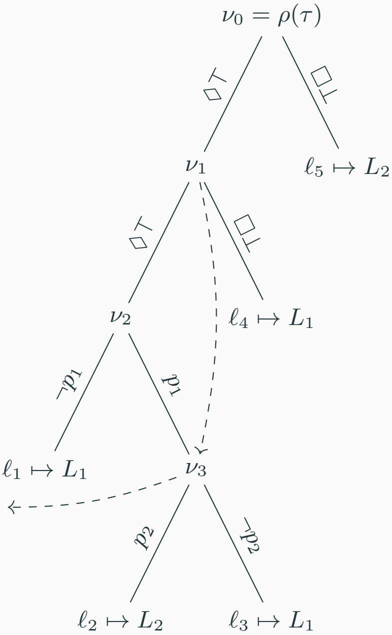


this is its semantics

Modal Decision Trees



this is not allowed
(Condition 4)



this is not allowed
(Condition 4)

Modal Decision Trees: Intuition – 2

In modal decision trees, we use backward-edges to build formulas with conjuncts at the same modal depth (e.g., $\Diamond p \wedge q$), and forward-edges to modify the flow of decisions. Each decision is a new modal formula; by following a non-trivial forward-edge, one can skip one or more decisions.

In modal decision trees, we use backward-edges to build formulas with conjuncts at the same modal depth (e.g., $\Diamond p \wedge q$), and forward-edges to modify the flow of decisions. Each decision is a new modal formula; by following a non-trivial forward-edge, one can skip one or more decisions. As we shall see, this is essential for guaranteeing the completeness of the method.

Modal Decision Trees: Exercise

$$\begin{tikzcd}[row sep=small, column sep=small]
 & w_1 : A_1 = 3 \arrow[r] & w_3 : A_1 = 4 \\
I_1 :\ w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & L_1 \\
 & w_2 : A_1 = 1 &
\end{tikzcd}$$

$$\begin{tikzcd}[row sep=small, column sep=small]
 & w_1 : A_1 = 4 \arrow[r] & w_3 : A_1 = 1 \\
I_2 :\ w_0 : A_1 = 3 \arrow[ur] \arrow[dr] & & L_2 \\
 & w_2 : A_1 = 2 &
\end{tikzcd}$$

$$\begin{tikzcd}[row sep=small, column sep=small]
 & w_1 : A_1 = 1 \arrow[r] & w_3 : A_1 = 2 \\
I_3 :\ w_0 : A_1 = 4 \arrow[ur] \arrow[dr] & & L_1 \\
 & w_2 : A_1 = 5 &
\end{tikzcd}$$

$$\begin{tikzcd}[row sep=small, column sep=small]
 & w_1 : A_1 = 2 \arrow[r] & w_3 : A_1 = 3 \\
I_4 :\ w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & L_2 \\
 & w_2 : A_1 = 3 &
\end{tikzcd}$$

$$I_1 : w_0 : A_1 = 2 \quad \begin{array}{l} \nearrow w_1 : A_1 = 3 \rightarrow w_3 : A_1 = 4 \\ \searrow w_2 : A_1 = 1 \end{array} \qquad L_1$$

$$I_2 : w_0 : A_1 = 3 \quad \begin{array}{l} \nearrow w_1 : A_1 = 4 \rightarrow w_3 : A_1 = 1 \\ \searrow w_2 : A_1 = 2 \end{array} \qquad L_2$$

$$I_3 : w_0 : A_1 = 4 \quad \begin{array}{l} \nearrow w_1 : A_1 = 1 \rightarrow w_3 : A_1 = 2 \\ \searrow w_2 : A_1 = 5 \end{array} \qquad L_1$$

$$I_4 : w_0 : A_1 = 2 \quad \begin{array}{l} \nearrow w_1 : A_1 = 2 \rightarrow w_3 : A_1 = 3 \\ \searrow w_2 : A_1 = 3 \end{array} \qquad L_2$$

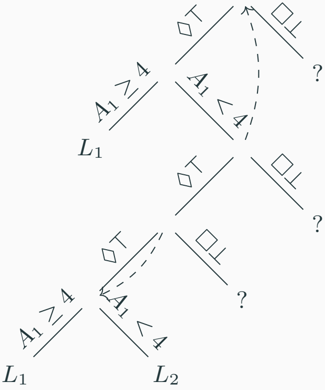
draw an optimal modal decision tree for this dataset

$$\begin{tikzcd} & w_1 : A_1 = 3 \arrow[r] & w_3 : A_1 = 4 \\ I_1 : w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & L_1 \\ & w_2 : A_1 = 1 & \end{tikzcd}$$

$$\begin{tikzcd} & w_1 : A_1 = 4 \arrow[r] & w_3 : A_1 = 1 \\ I_2 : w_0 : A_1 = 3 \arrow[ur] \arrow[dr] & & L_2 \\ & w_2 : A_1 = 2 & \end{tikzcd}$$

$$\begin{tikzcd} & w_1 : A_1 = 1 \arrow[r] & w_3 : A_1 = 2 \\ I_3 : w_0 : A_1 = 4 \arrow[ur] \arrow[dr] & & L_1 \\ & w_2 : A_1 = 5 & \end{tikzcd}$$

$$\begin{tikzcd} & w_1 : A_1 = 2 \arrow[r] & w_3 : A_1 = 3 \\ I_4 : w_0 : A_1 = 2 \arrow[ur] \arrow[dr] & & L_2 \\ & w_2 : A_1 = 3 & \end{tikzcd}$$



Modal Decision Trees, Formally – 7

Learning a modal decision tree is done exactly the same way as in the propositional case.

Learning a modal decision tree is done exactly the same way as in the propositional case. The idea is to substitute the concept of split with a more general one.

Definition

Given a modal decision tree $\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ and a path π_ν^τ from the root to a node ν, let $\hat{\pi}_\nu^\tau = \nu_0, \nu_1, \ldots, \nu$ (or, simply, $\hat{\pi}_\nu$, when τ is clear from the context) be the shortened path *from $\rho(\tau)$ to ν defined as the unique sub-sequence, if any, of the path π_ν in which $\nu_0 = \rho(\tau)$ and, for each i, ν_{i+1} is the successor of $f(\nu_i)$ in π_ν^τ. If such a sub-sequence does not exist, then the shortened path is undefined.*

Definition

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

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

Definition

Let $\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f)$ be a modal decision tree, $\mathcal{I}$ a labelled modal dataset, ν a non-leaf node of τ whose ν-dataset is defined. Then, the *(binary) (sub-tree) split* *of $\mathcal{I}_\nu$ is the pair defined as*

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

Modal Decision Trees, Formally – 9

Considering that the amount of information can be computed as in the propositional case, the algorithm $ModalCART$, for a modal dataset with m instances, works as follow:

- If a stopping condition applies, then return;
- Find the best sub-tree split $(\mathcal{I}_1, \mathcal{I}_2)$, and call $ModalCART$ recursively on $\mathcal{I}_1$ and $\mathcal{I}_2$.

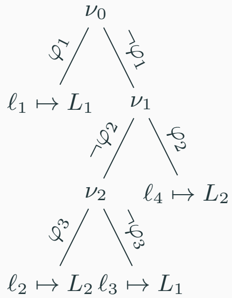
Efficient Modal Decision Trees, *Proceedings of the 22nd International Conference of the Italian Association for Artificial Intelligence (AIxIA)*, Manzella et al., 2023.

Theorem

The family $\mathcal{MDT}$ of (pure) modal decision trees is correct, complete, and efficient.

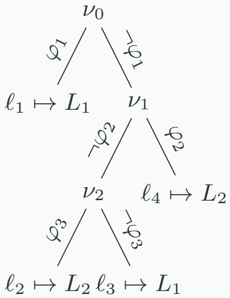
Modal Decision Trees: Summary

Pure modal decision trees, obtained from modal decision trees, do not have forward- and backward-edges, and behave exactly as pure propositional ones (except edges are labeled with modal formulas to be checked on w_0). Barring the technical complexity of their detailed definition, the overall concepts are very simple.



Modal Decision Trees: Summary

Pure modal decision trees, obtained from modal decision trees, do not have forward- and backward-edges, and behave exactly as pure propositional ones (except edges are labeled with modal formulas to be checked on w_0). Barring the technical complexity of their detailed definition, the overall concepts are very simple.



One question remains to be answered, and it may come in many forms. One of them is: why unstructured data are modal data?

Modal Symbolic Learning at Work

Interval Logic: Introduction

A first example of how modal logics can be made concrete so that they can express practical situations is that of temporal logic.

A first example of how modal logics can be made concrete so that they can express practical situations is that of temporal logic. Temporal logic comes in many flavours: point-based ones, in which we describe a discrete process (such as the evolution of the states of a machine or a program), and in which time can be linear (i.e., a linear sequence) or branching (i.e., a tree), and properties express states whose temporal relation we want to model;

A first example of how modal logics can be made concrete so that they can express practical situations is that of temporal logic. Temporal logic comes in many flavours: point-based ones, in which we describe a discrete process (such as the evolution of the states of a machine or a program), and in which time can be linear (i.e., a linear sequence) or branching (i.e., a tree), and properties express states whose temporal relation we want to model; or interval-based ones, implicitly defined to describe a continuous process (such as time series, as we shall see in detail), and properties express situations that are meaningful over a period of time.

A first example of how modal logics can be made concrete so that they can express practical situations is that of temporal logic. Temporal logic comes in many flavours: point-based ones, in which we describe a discrete process (such as the evolution of the states of a machine or a program), and in which time can be linear (i.e., a linear sequence) or branching (i.e., a tree), and properties express states whose temporal relation we want to model; or interval-based ones, implicitly defined to describe a continuous process (such as time series, as we shall see in detail), and properties express situations that are meaningful over a period of time. Let us focus on the latter case.

Interval Temporal Logic: Intuition

$$\begin{array}{ccccc} p & \neg p & p & \neg p & p \\ q & q & \neg q & \neg q & \neg q \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{array}$$

in point-based temporal logics
worlds are linearly ordered
and related to each other directly

Interval Temporal Logic: Intuition

p	$\neg p$	p	$\neg p$	p	in point-based temporal logics
q	q	$\neg q$	$\neg q$	$\neg q$	worlds are linearly ordered
$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	and related to each other directly

Typical point-based temporal logics allow one to refer to the next world
future world(s), and/or past worlds

Interval Temporal Logic: Intuition

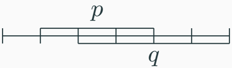
Interval Temporal Logic: Intuition

in interval-based temporal logics
worlds are intervals, that is
pairs of points
and their relationships are not trivial

Interval Temporal Logic: Intuition

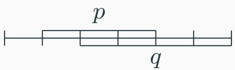
in interval-based temporal logics
worlds are intervals, that is
pairs of points
and their relationships are not trivial

Interval Temporal Logic: Intuition



in interval-based temporal logics
worlds are intervals, that is
pairs of points
and their relationships are not trivial

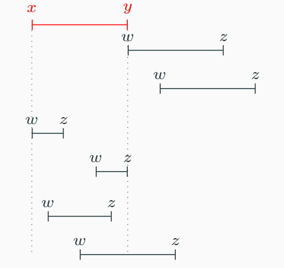
Interval Temporal Logic: Intuition



in interval-based temporal logics
worlds are intervals, that is
pairs of points
and their relationships are not trivial

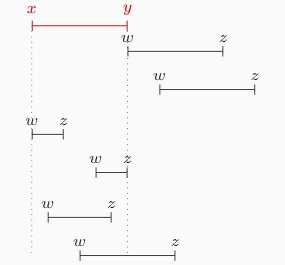
Typical interval-based temporal logics allow one to refer to inside intervals overlapping ones, starting/ending ones, and so on

Interval Temporal Logic for Learning: Syntax – 1

Modality	**Definition**	**Example**
$\langle A \rangle$ (after)	$[x, y]R_A[w, z] \quad \Leftrightarrow \quad y = w$	
$\langle L \rangle$ (later)	$[x, y]R_L[w, z] \quad \Leftrightarrow \quad y < w$	
$\langle B \rangle$ (begins)	$[x, y]R_B[w, z] \quad \Leftrightarrow \quad x = w \wedge z < y$	
$\langle E \rangle$ (ends)	$[x, y]R_E[w, z] \quad \Leftrightarrow \quad y = z \wedge x < w$	
$\langle D \rangle$ (during)	$[x, y]R_D[w, z] \quad \Leftrightarrow \quad x < w \wedge z < y$	
$\langle O \rangle$ (overlaps)	$[x, y]R_O[w, z] \quad \Leftrightarrow \quad x < w < y < z$	

A Propositional Modal Logic of Time Intervals, *Journal of the ACM*, Halpern and Shoham, 1991.

Modality	**Definition**	**Example**
$\langle A \rangle$ (after)	$[x,y]R_A[w,z] \quad \Leftrightarrow \quad y = w$	
$\langle L \rangle$ (later)	$[x,y]R_L[w,z] \quad \Leftrightarrow \quad y < w$	
$\langle B \rangle$ (begins)	$[x,y]R_B[w,z] \quad \Leftrightarrow \quad x = w \wedge z < y$	
$\langle E \rangle$ (ends)	$[x,y]R_E[w,z] \quad \Leftrightarrow \quad y = z \wedge x < w$	
$\langle D \rangle$ (during)	$[x,y]R_D[w,z] \quad \Leftrightarrow \quad x < w \wedge z < y$	
$\langle O \rangle$ (overlaps)	$[x,y]R_O[w,z] \quad \Leftrightarrow \quad x < w < y < z$	



Interval Temporal Logic (i.e., $\mathcal{HS}$) has 13 (12+equality) relations, that correspond to 12 modal operators (i.e, 12 diamonds and 12 corresponding boxes). The relations are in the set $\mathcal{O} = \{A, L, B, E, D, O, \overline{A}, \overline{L}, \overline{B}, \overline{E}, \overline{D}, \overline{O}, \}$. For $X \in \mathcal{O}$, $\overline{X}$ is the inverse relation.

A Propositional Modal Logic of Time Intervals, *Journal of the ACM*, Halpern and Shoham, 1991.

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

when $X = A$, for example, this is read
there exists an interval after the current one
with the endind point in common with
the current one, in which φ holds true

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X\rangle\varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

when $X = B$, again for example, this is read there exists an interval that starts the current one, in which φ holds true

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

the box version of each operator
allows one to capture all intervals
in a specific relation

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

as always, we use a
non-minimal grammar for formulas
in order to ease the learning phase

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

How do we express. . . ?

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

How do we express. . . ?

The fever is over 38
in the current interval

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle\varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

How do we express...?

The fever is over 38 in the current interval	$Fever > 38$

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

How do we express. . . ?

The fever is over 38 in the current interval

$Fever > 38$

In the current interval, while the fever is over 38, there is an interval during which pain¡3

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

How do we express...?

The fever is over 38 in the current interval	$Fever > 38$
In the current interval, while the fever is over 38, there is an interval during which pain¡3	$(Fever > 38) \wedge \langle D \rangle (Pain < 3)$

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

How do we express...?

The fever is over 38 in the current interval	$Fever > 38$
In the current interval, while the fever is over 38, there is an interval during which pain¡3	$(Fever > 38) \wedge \langle D \rangle (Pain < 3)$
The fever is always below 37 during the current interval	

Interval Temporal Logic for Learning: Syntax – 2

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle\varphi \mid [X]\varphi, \quad X \in \mathcal{O}$$

How do we express. . . ?

The fever is over 38 in the current interval	$Fever > 38$
In the current interval, while the fever is over 38, there is an interval during which pain¡3	$(Fever > 38) \wedge \langle D \rangle (Pain < 3)$
The fever is always below 37 during the current interval	$[D]\neg(Fever < 37)$

Interval Logic for Learning: Summary

Interval temporal logic is one way to concretize modal logic into a useful language for learning.

Interval Logic for Learning: Summary

Interval temporal logic is one way to concretize modal logic into a useful language for learning. The paradigm of modal symbolic learning is general, but the temporal case is the case in which the ideas emerge more easily. As in the cases of propositional and modal logic, in the case of interval logic too we are defining the language modulo theory;

Interval Logic for Learning: Summary

Interval temporal logic is one way to concretize modal logic into a useful language for learning. The paradigm of modal symbolic learning is general, but the temporal case is the case in which the ideas emerge more easily. As in the cases of propositional and modal logic, in the case of interval logic too we are defining the language modulo theory; this causes a difference between induction and deduction properties that must be addressed.

Definition

Given a set of n names of attributes $\mathcal{A} = \{A_1 \ldots A_n\}$, such that each attribute A_i is associated to a finite domain $dom(A_i) \subset \mathbb{R}$, the set of well-formed modal (learning) formulas is obtained by the grammar

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi,$$

where $A \in \mathcal{A}$, $v \in dom(A)$, $\bowtie \in \{<, \leq, =, \geq, >\}$, and $X \in \mathcal{O}$.

Interval Temporal Logic for Learning: Syntax – 2

Definition

Given a set of n names of attributes $\mathcal{A} = \{A_1 \ldots A_n\}$, such that each attribute A_i is associated to a finite domain $dom(A_i) \subset \mathbb{R}$, the set of well-formed modal (learning) formulas is obtained by the grammar

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi,$$

where $A \in \mathcal{A}$, $v \in dom(A)$, $\bowtie \in \{<, \leq, =, \geq, >\}$, and $X \in \mathcal{O}$.

How do we concretize the Kripke semantics for this case? What are worlds? What are their relationships?

Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi$$

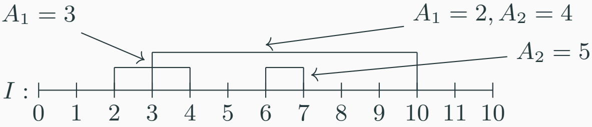
Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi$$

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle\varphi \mid [X]\varphi$$

Given a linearly ordered set $\mathbb{D}$

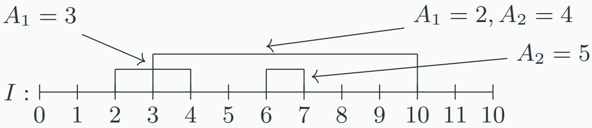
$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X\rangle\varphi \mid [X]\varphi$$



Given a linearly ordered set $\mathbb{D}$
the set of all intervals $W = \mathbb{I}(\mathbb{D}) = \{[x, y] \mid x, y \in \mathbb{D}, x < y\}$
is the set of worlds of an interval logic interpretation

Interval Temporal Logic for Learning: Semantics – 1

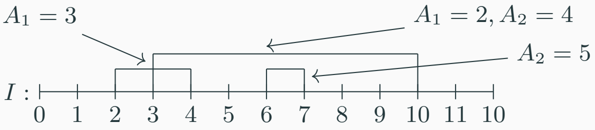
$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle\varphi \mid [X]\varphi$$



Given a linearly ordered set $\mathbb{D}$
the set of all intervals $W = \mathbb{I}(\mathbb{D}) = \{[x, y] \mid x, y \in \mathbb{D}, x < y\}$
is the set of worlds of an interval logic interpretation
and the set of relations is the set of Allen's relations: $R_A, R_L, \ldots$

Interval Temporal Logic for Learning: Semantics – 1

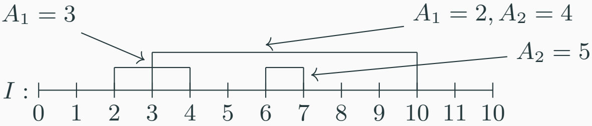
$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle\varphi \mid [X]\varphi$$



Suppose that in I there are only two attributes A_1, A_2, and that everywhere not specified, $A_i = 0$ for $i \in \{1, 2\}$ then:

Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X\rangle\varphi \mid [X]\varphi$$

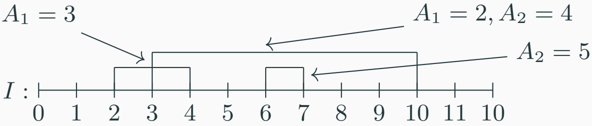


Suppose that in I there are only two attributes A_1, A_2, and that everywhere not specified, $A_i = 0$ for $i \in \{1, 2\}$ then:

$$I, [0, 1] \Vdash \text{sometimes in the future } A_1 > 1$$

Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \to \varphi \mid \langle X \rangle \varphi \mid [X]\varphi$$

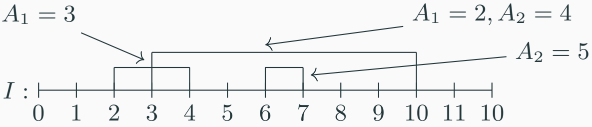


Suppose that in I there are only two attributes A_1, A_2, and that everywhere not specified, $A_i = 0$ for $i \in \{1, 2\}$ then:

$$I, [0, 1] \Vdash \langle L \rangle (A_1 > 1)$$

Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi$$

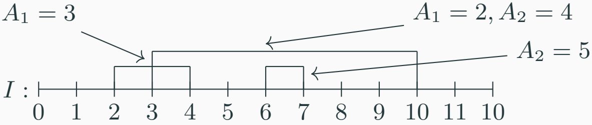


Suppose that in I there are only two attributes A_1, A_2, and that everywhere not specified, $A_i = 0$ for $i \in \{1, 2\}$ then:

$I, [0, 1] \Vdash$ sometimes in the future $A_1 > 1$ overlapped by $A_2 > 1$

Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X \rangle \varphi \mid [X]\varphi$$

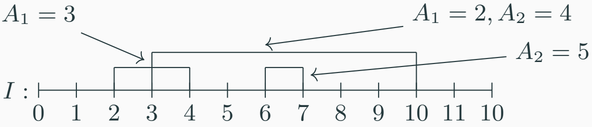


Suppose that in I there are only two attributes A_1, A_2, and that everywhere not specified, $A_i = 0$ for $i \in \{1, 2\}$ then:

$$I, [0, 1] \Vdash \langle L \rangle ((A_1 > 1) \wedge \langle O \rangle (A_2 > 1))$$

Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X\rangle\varphi \mid [X]\varphi$$

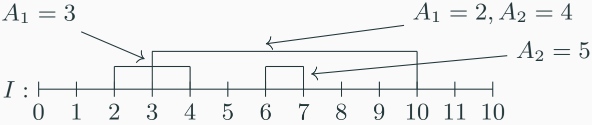


Suppose that in I there are only two attributes A_1, A_2, and that everywhere not specified, $A_i = 0$ for $i \in \{1, 2\}$ then:

$I, [3, 10] \Vdash$ it is never true within me that $A_2 > 6$

Interval Temporal Logic for Learning: Semantics – 1

$$\varphi ::= (A \bowtie v) \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \rightarrow \varphi \mid \langle X\rangle\varphi \mid [X]\varphi$$



Suppose that in I there are only two attributes A_1, A_2, and that everywhere not specified, $A_i = 0$ for $i \in \{1, 2\}$ then:

$$I, [3, 0] \Vdash [D](A_2 \leq 5)$$

Definition

Given a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ of attributes, an interval temporal interpretation I is tuple $I = (\mathbb{I}(\mathbb{D}), \mathcal{R})$, where $\mathbb{D}$ is a linearly ordered set, $\mathbb{I}(\mathbb{D})$ is the set of all intervals on $\mathbb{D}$, $\mathcal{R}$ is the set of Allen's relations, and each world w, denoted as an interval $[x, y]$, is a valuation function:

$$w : \mathcal{A} \rightarrow \mathbb{R}.$$

A interval temporal interpretation I and a world $w = [x, y]$ in it naturally induce the truth relation for a formula φ ($I, w \Vdash \varphi$), obtained by

$$
\begin{array}{lll}
I, w \Vdash (A \bowtie v) & \textit{iff} & w(V) \bowtie v \\
I, w \Vdash \neg\varphi & \textit{iff} & I, w \nVdash \varphi \\
I, w \Vdash \varphi \vee \psi & \textit{iff} & I, w \Vdash \varphi \textit{ or } I \Vdash \psi \\
I, w \Vdash \varphi \wedge \psi & \textit{iff} & I, w \Vdash \varphi \textit{ and } I \Vdash \psi \\
I, w \Vdash \varphi \rightarrow \psi & \textit{iff} & I, w \nVdash \varphi \textit{ or } I \Vdash \psi \\
I, w \Vdash \langle X \rangle\varphi & \textit{iff} & \textit{there is } w\textit{' s.t. } wR_X w' \textit{ and } I, w' \Vdash \varphi \\
I, w \Vdash [X]\varphi & \textit{iff} & \textit{for every } w' \textit{ s.t. } wR_X w' \textit{ it happens } I, w' \Vdash \varphi
\end{array}
$$

If $I, w \Vdash \varphi$ for some w, then I satisfies φ at w, and I is a model of φ.

Interval Temporal Logic for Learning: Semantics – 3

Some relevant truths:

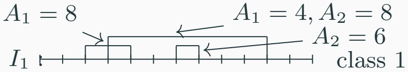
$$\langle L\rangle\varphi \leftrightarrow \langle A\rangle\langle A\rangle\varphi \qquad \langle \overline{L}\rangle\varphi \leftrightarrow \langle \overline{A}\rangle\langle \overline{A}\rangle\varphi$$

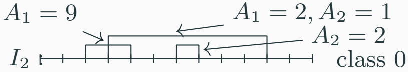
$$\langle D\rangle\varphi \leftrightarrow \langle B\rangle\langle E\rangle\varphi \qquad \langle \overline{D}\rangle\varphi \leftrightarrow \langle \overline{B}\rangle\langle \overline{E}\rangle\varphi$$

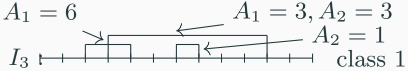
$$\langle O\rangle\varphi \leftrightarrow \langle E\rangle\langle \overline{B}\rangle\varphi \qquad \langle \overline{O}\rangle\varphi \leftrightarrow \langle B\rangle\langle \overline{E}\rangle\varphi$$

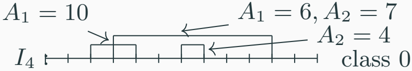
$$\ldots$$

Interval Temporal Logic for Learning: Exercise

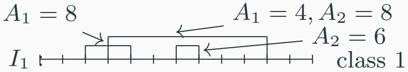


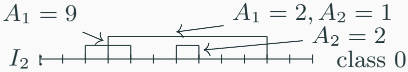


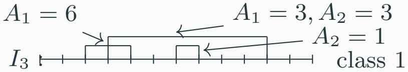


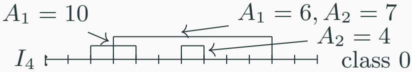


Interval Temporal Logic for Learning: Exercise



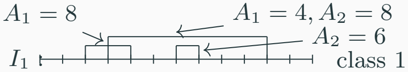


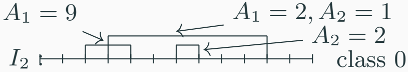


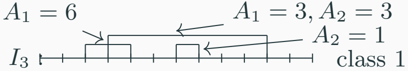


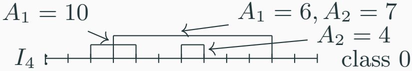
What formula on $[0, 1]$ separates class 0 from class 1?

Interval Temporal Logic for Learning: Exercise









$$[L](A_1 > 7 \rightarrow \langle O \rangle (A_2 > 5))$$

Transformers: Introduction – 1

Unstructured data comes in many forms. An (incomplete) list includes:

- Dimensional data. Typically numerical data with 0,1, or more dimensions. Examples are tabular data (dimension 0), time series (dimension 1), images (dimensions 2), videos (dimension 3).
- Textual data.
- Graph-based data.
- Multimodal data, that is data whose instances are described by more than one data type (e.g., a patient described by tabular data plus a diagnostic image).

Unstructured data comes in many forms. An (incomplete) list includes:

- Dimensional data. Typically numerical data with 0,1, or more dimensions. Examples are tabular data (dimension 0), time series (dimension 1), images (dimensions 2), videos (dimension 3).
- Textual data.
- Graph-based data.
- Multimodal data, that is data whose instances are described by more than one data type (e.g., a patient described by tabular data plus a diagnostic image).

The underlying idea to modal symbolic learning is that an unstructured instance can be always seen as a modal one.

Dimensional data can be seen as modal data with a quite small abstraction step. As a consequence, they are ideal to introduce the concept of transformer, as well as the idea of feature extraction.

The Voice of COVID-19: Breath and Cough Recording Classification with Temporal Decision Trees and Random Forests. *Artificial Intelligence in Medicine*, Manzella et al., 2023.

Statistical Rule Extraction for Gas Turbine Trip Prediction, *Journal of Engineering for Gas Turbines and Power*, Bechini et al., 2023.

Towards an Objective Theory of Subjective Liking: a First Step in Understanding the Sense of Beauty, *PLOS One*, Mazzacane et al., 2023.

Dimensional data can be seen as modal data with a quite small abstraction step. As a consequence, they are ideal to introduce the concept of transformer, as well as the idea of feature extraction. Seeing textual and graph-based data as modal data is not immediate, and we shall limit ourselves to simple examples.

The Voice of COVID-19: Breath and Cough Recording Classification with Temporal Decision Trees and Random Forests. *Artificial Intelligence in Medicine*, Manzella et al., 2023.

Statistical Rule Extraction for Gas Turbine Trip Prediction, *Journal of Engineering for Gas Turbines and Power*, Bechini et al., 2023.

Towards an Objective Theory of Subjective Liking: a First Step in Understanding the Sense of Beauty, *PLOS One*, Mazzacane et al., 2023.

Dimensional data can be seen as modal data with a quite small abstraction step. As a consequence, they are ideal to introduce the concept of transformer, as well as the idea of feature extraction. Seeing textual and graph-based data as modal data is not immediate, and we shall limit ourselves to simple examples. Once unimodal unstructured data can be treated, dealing with multimodal data is immediate.

The Voice of COVID-19: Breath and Cough Recording Classification with Temporal Decision Trees and Random Forests. *Artificial Intelligence in Medicine*, Manzella et al., 2023.

Statistical Rule Extraction for Gas Turbine Trip Prediction, *Journal of Engineering for Gas Turbines and Power*, Bechini et al., 2023.

Towards an Objective Theory of Subjective Liking: a First Step in Understanding the Sense of Beauty, *PLOS One*, Mazzacane et al., 2023.

Dimensional data can be seen as modal data with a quite small abstraction step. As a consequence, they are ideal to introduce the concept of transformer, as well as the idea of feature extraction. Seeing textual and graph-based data as modal data is not immediate, and we shall limit ourselves to simple examples. Once unimodal unstructured data can be treated, dealing with multimodal data is immediate.

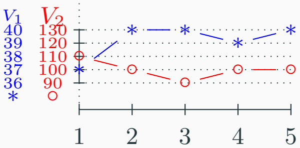
Let us focus on the case of time series (dimension 1 data).

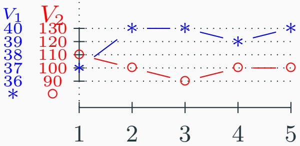
The Voice of COVID-19: Breath and Cough Recording Classification with Temporal Decision Trees and Random Forests. *Artificial Intelligence in Medicine*, Manzella et al., 2023.

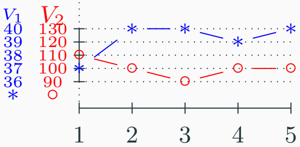
Statistical Rule Extraction for Gas Turbine Trip Prediction, *Journal of Engineering for Gas Turbines and Power*, Bechini et al., 2023.

Towards an Objective Theory of Subjective Liking: a First Step in Understanding the Sense of Beauty, *PLOS One*, Mazzacane et al., 2023.

Transformers: The Case of Time Series

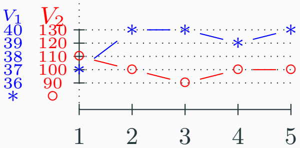


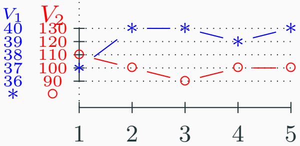


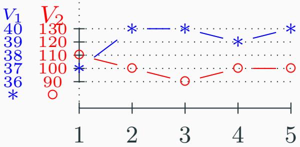


...

Transformers: The Case of Time Series



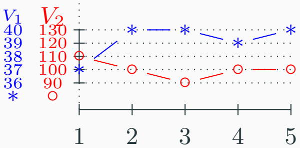


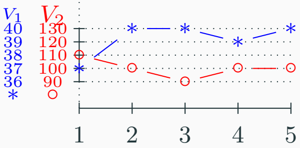


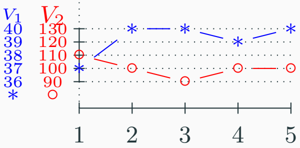
...

concrete representation:

	V_1	V_2
I_1	37,40,40,39,40;	110,100,90,100,100;
I_2	...	...
I_3	...	...
...	...	...



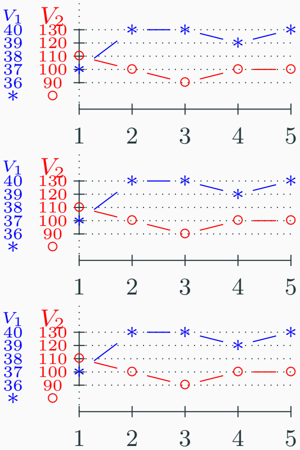




...

abstract representation:

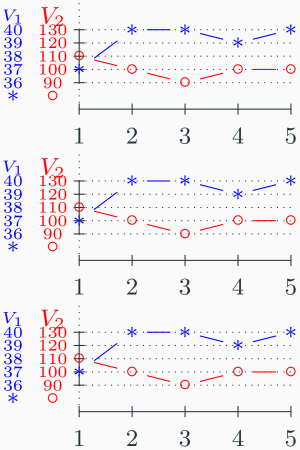
I_1	$\mathbb{I}([1,2,3,4,5]),\mathcal{R})$
I_2	$\mathbb{I}([1,2,3,4,5]),\mathcal{R})$
I_3	...
...	...



...

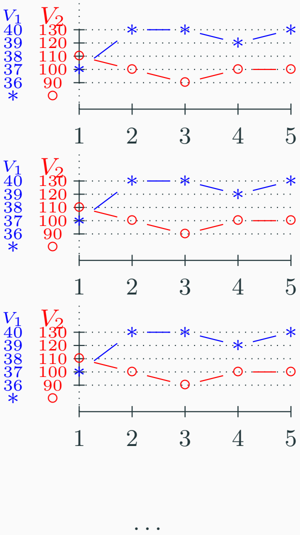
But what are the attributes?

How are they extracted from the variables?



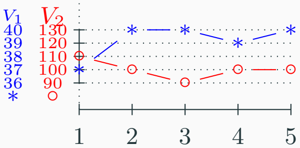
...

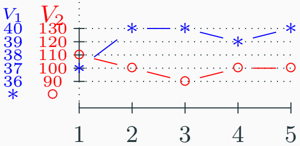
We take each variable V_i
and we apply a feature extraction
function $f \in \mathcal{F}$

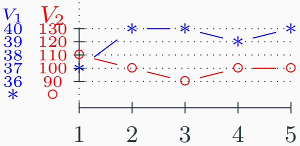


Functions are taken from
the classic function vocabulary
including average, min, max,
as well as much more complex ones.

Transformers: The Case of Time Series

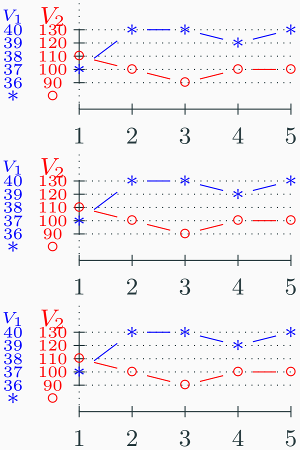






...

So, a function f applied to a variable V (that is, $f(V)$) becomes an attribute A.



...

$$w_0: \begin{array}{c} min(V_1) = 37 \\ max(V_2) = 110 \\ \ldots \\ [1,2] \end{array}$$

after

$$w_1: \begin{array}{c} max(V_1) = 40 \\ min(V_2) = 90 \\ \ldots \\ [2,4] \end{array}$$

begins

after

$$w_2: \begin{array}{c} min(V_2) = 100 \\ max(V_2) = 110 \\ \ldots \\ [2,3] \end{array}$$

Transformers, Formally

Definition

Let $\mathfrak{I}$ be the set of all possible datasets, $\mathfrak{F}$ the set of all possible sets of feature extraction functions, and $\mathfrak{L}$ the set of all possible (unary) (multi-)modal logics, then a transformer is a function

$$T : \mathfrak{I} \times \mathfrak{F} \times \mathfrak{L} \to \mathfrak{M}$$

where $\mathfrak{M}$ is the set of all possible modal datasets.

Definition

Let $\mathfrak{I}$ be the set of all possible datasets, $\mathfrak{F}$ the set of all possible sets of feature extraction functions, and $\mathfrak{L}$ the set of all possible (unary) (multi-)modal logics, then a transformer is a function

$$T : \mathfrak{I} \times \mathfrak{F} \times \mathfrak{L} \to \mathfrak{M}$$

where $\mathfrak{M}$ is the set of all possible modal datasets.

While the above definition is not very precise, the ingredients are clear:

Definition

Let $\mathfrak{I}$ be the set of all possible datasets, $\mathfrak{F}$ the set of all possible sets of feature extraction functions, and $\mathfrak{L}$ the set of all possible (unary) (multi-)modal logics, then a transformer is a function

$$T : \mathfrak{I} \times \mathfrak{F} \times \mathfrak{L} \to \mathfrak{M}$$

where $\mathfrak{M}$ is the set of all possible modal datasets.

While the above definition is not very precise, the ingredients are clear: given a dataset of (un)structured instances, given the specific modal logic that one wants to use to extract information from it, and given a suitable set of feature extraction functions, one immediately obtains the modal version of the original dataset, as we did in the temporal case.

Definition

Let $\mathfrak{I}$ be the set of all possible datasets, $\mathfrak{F}$ the set of all possible sets of feature extraction functions, and $\mathfrak{L}$ the set of all possible (unary) (multi-)modal logics, then a transformer is a function

$$T : \mathfrak{I} \times \mathfrak{F} \times \mathfrak{L} \rightarrow \mathfrak{M}$$

where $\mathfrak{M}$ is the set of all possible modal datasets.

While the above definition is not very precise, the ingredients are clear: given a dataset of (un)structured instances, given the specific modal logic that one wants to use to extract information from it, and given a suitable set of feature extraction functions, one immediately obtains the modal version of the original dataset, as we did in the temporal case. What if the information is not purely temporal?

Transformers: the Dimensional Case

It so happens that d-dimensional data can be interpreted in the same way as 1-dimensional ones. There is an obvious generalization of the logic $\mathcal{HS}$ to d dimensions that allows one to deal with hyper-rectangles instead of intervals.

Decision Tree Learning with Spatial Modal Logics, *12th International Symposium on Games, Automata, Logics and Formal Verification (GANDALF)*, Pagliarini and Sciavicco, 2021.

Transformers: the Dimensional Case

It so happens that d-dimensional data can be interpreted in the same way as 1-dimensional ones. There is an obvious generalization of the logic $\mathcal{HS}$ to d dimensions that allows one to deal with hyper-rectangles instead of intervals. The operators can be extended accordingly. Observe that the tabular case (0-dimensional) is now a special case of dimensional data.

Decision Tree Learning with Spatial Modal Logics, *12th International Symposium on Games, Automata, Logics and Formal Verification (GANDALF)*, Pagliarini and Sciavicco, 2021.

Transformers: the Dimensional Case

It so happens that d-dimensional data can be interpreted in the same way as 1-dimensional ones. There is an obvious generalization of the logic $\mathcal{HS}$ to d dimensions that allows one to deal with hyper-rectangles instead of intervals. The operators can be extended accordingly. Observe that the tabular case (0-dimensional) is now a special case of dimensional data. So, images, videos, and so on can be now treated in a completely uniform way.

Decision Tree Learning with Spatial Modal Logics, *12th International Symposium on Games, Automata, Logics and Formal Verification (GANDALF)*, Pagliarini and Sciavicco, 2021.

Transformers: the Dimensional Case

It so happens that d-dimensional data can be interpreted in the same way as 1-dimensional ones. There is an obvious generalization of the logic $\mathcal{HS}$ to d dimensions that allows one to deal with hyper-rectangles instead of intervals. The operators can be extended accordingly. Observe that the tabular case (0-dimensional) is now a special case of dimensional data. So, images, videos, and so on can be now treated in a completely uniform way.Is this the only way to modalize dimensional data?

Decision Tree Learning with Spatial Modal Logics, *12th International Symposium on Games, Automata, Logics and Formal Verification (GANDALF)*, Pagliarini and Sciavicco, 2021.

Transformers: the Dimensional Case

It so happens that d-dimensional data can be interpreted in the same way as 1-dimensional ones. There is an obvious generalization of the logic $\mathcal{HS}$ to d dimensions that allows one to deal with hyper-rectangles instead of intervals. The operators can be extended accordingly. Observe that the tabular case (0-dimensional) is now a special case of dimensional data. So, images, videos, and so on can be now treated in a completely uniform way.Is this the only way to modalize dimensional data? No. Using different modal languages gives rise to very different modal datasets that contain the same information but in which different details are highlighted.

Decision Tree Learning with Spatial Modal Logics, *12th International Symposium on Games, Automata, Logics and Formal Verification (GANDALF)*, Pagliarini and Sciavicco, 2021.

Transformers: the Dimensional Case

It so happens that d-dimensional data can be interpreted in the same way as 1-dimensional ones. There is an obvious generalization of the logic $\mathcal{HS}$ to d dimensions that allows one to deal with hyper-rectangles instead of intervals. The operators can be extended accordingly. Observe that the tabular case (0-dimensional) is now a special case of dimensional data. So, images, videos, and so on can be now treated in a completely uniform way.Is this the only way to modalize dimensional data? No. Using different modal languages gives rise to very different modal datasets that contain the same information but in which different details are highlighted. In the dimensional case, examples include $\mathcal{HS}^d$ (hyper-rectangles, all qualitatively different directional relations), $\mathcal{HS}^2_{RCC8}, \mathcal{HS}^2_{RCC5}$ (in the 2-dimensional case, hyper-rectangles related by different flavours of topological relations), among others.

Decision Tree Learning with Spatial Modal Logics, *12th International Symposium on Games, Automata, Logics and Formal Verification (GANDALF)*, Pagliarini and Sciavicco, 2021.

Interval temporal logic and its variants/extensions cover the whole sub-case of dimensional data; it must be stressed, however, that this is not the only way to transform dimensional data into modal data.

Towards Symbolic Metaphor Identification, *Proceedings of the 9th Italian Conference on Computational Linguistics (CLIC-IT)*, Manzella et al., 2023.

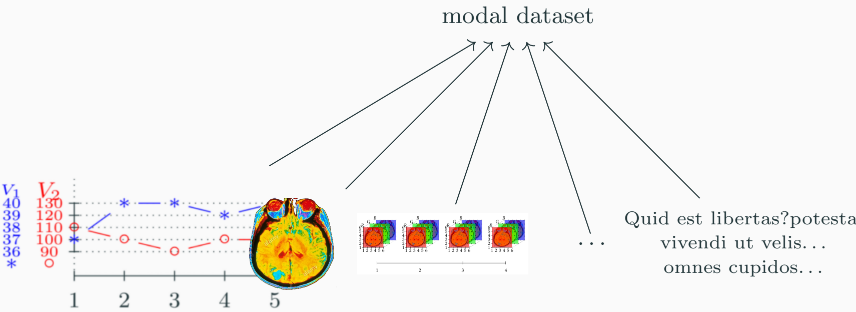
Interval temporal logic and its variants/extensions cover the whole sub-case of dimensional data; it must be stressed, however, that this is not the only way to transform dimensional data into modal data. The case of textual data is currently under study. Yet, the principle can be the same: text is linearly ordered, words (or other tokens) could be the points on this linear order; intervals are determined by pairs of tokens, and feature extraction function can be NLP functions of any sort. Thus, interval temporal logics such as $\mathcal{HS}$ could cover the case of text data as well. Other characterizations are also possible.

Towards Symbolic Metaphor Identification, *Proceedings of the 9th Italian Conference on Computational Linguistics (CLIC-IT)*, Manzella et al., 2023.

Interval temporal logic and its variants/extensions cover the whole sub-case of dimensional data; it must be stressed, however, that this is not the only way to transform dimensional data into modal data. The case of textual data is currently under study. Yet, the principle can be the same: text is linearly ordered, words (or other tokens) could be the points on this linear order; intervals are determined by pairs of tokens, and feature extraction function can be NLP functions of any sort. Thus, interval temporal logics such as $\mathcal{HS}$ could cover the case of text data as well. Other characterizations are also possible. Graph-based data has not been studied, but initial observations seem to indicate that it may make sense to treat graph-based data in a similar way.

Towards Symbolic Metaphor Identification, *Proceedings of the 9th Italian Conference on Computational Linguistics (CLIC-IT)*, Manzella et al., 2023.

Transformers have the ability providing a uniform treatment of (un)structured data.



Day 3

Practice: Learning with Sole.jl

Modal Decision Trees, restricted version

Definition

Let $\mathcal{L}$ be a set of classes and $\mathcal{P}$ a set of propositions, and define the set of *modal decisions:*

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

Then, a modal decision tree (over $\mathcal{L}$) is an object of the type:

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

where $(\mathcal{V}, \mathcal{E}, l, e)$ is a propositional decision tree, $b : \mathcal{V}^\iota \to \mathcal{V}^\iota$ 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 $\nu, \nu' \nu'' \in \mathcal{V}$:

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

[2] Only listed here for completeness of treatment.

Definition

Let $\mathcal{L}$ be a set of classes and $\mathcal{P}$ a set of propositions, and define the set of modal decisions:

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

Then, a modal decision tree (over $\mathcal{L}$) is an object of the type:

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

where $(\mathcal{V}, \mathcal{E}, l, e)$ is a propositional decision tree, $b : \mathcal{V}^{\iota} \to \mathcal{V}^{\iota}$ 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 $\nu, \nu'\nu'' \in \mathcal{V}$:

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

[2] Only listed here for completeness of treatment.

Definition

Let $\mathcal{L}$ be a set of classes and $\mathcal{P}$ a set of propositions, and define the set of modal decisions:

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

Then, a modal decision tree (over $\mathcal{L}$) is an object of the type:

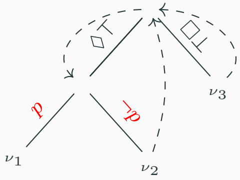
$$\tau = (\mathcal{V}, \mathcal{E}, l, e, b, f),$$

where $(\mathcal{V}, \mathcal{E}, l, e)$ is a propositional decision tree, $b : \mathcal{V}^{\iota} \to \mathcal{V}^{\iota}$ 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 $\nu, \nu' \nu'' \in \mathcal{V}$:

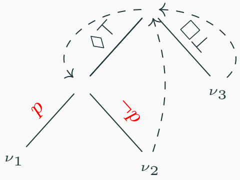
1. *if $\nu \notin \mathcal{V}^{\ell}$, then $e(\nu, \swarrow(\nu)) \equiv \neg e(\nu, \searrow(\nu))$*[2]*;*
2. *if $e(\nu, \nu') = \Diamond\top$, then: $\nu' = \swarrow(\nu), f(\nu) = \nu', e(\nu', \swarrow(\nu')) \in \mathcal{P}$.*
3. *if $\searrow(\nu) = \nu'$ and $e(\updownarrow(\nu), \nu) = \Diamond\top$,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

[2] Only listed here for completeness of treatment.

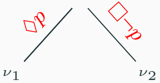
Modal Decision Trees, restricted version



$$\Diamond\top \wedge \Diamond(\top \wedge p)$$



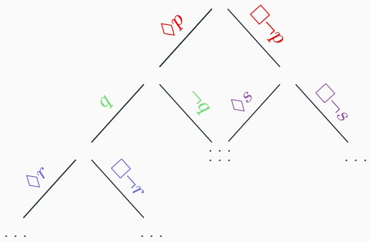
$\Diamond\top \wedge \Diamond(\top \wedge p)$



$\Diamond p$

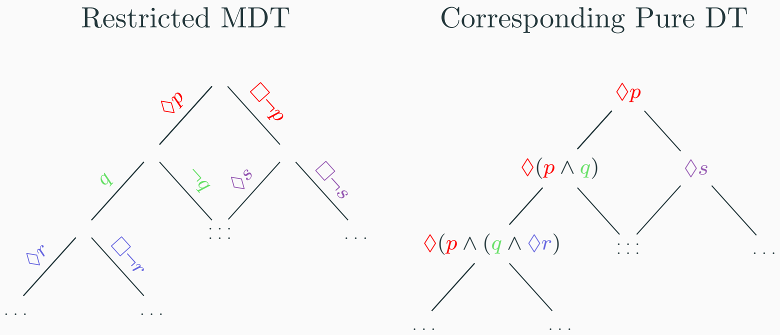
Modal Decision Trees, restricted version

Restricted MDT



Only checks non-alternating, existential formulas of the kind:

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



Only checks non-alternating, existential formulas of the kind:

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

Grammars for path- (φ_π), leaf- (φ_ℓ), and class-formulas (φ_L):

MDT

$$
\begin{aligned}
\varphi_\pi ::= \quad & \lambda \wedge (\varphi_\pi \wedge \varphi_\pi) & | \\
& \lambda \rightarrow (\varphi_\pi \rightarrow \varphi_\pi) & | \\
& \Diamond(\varphi_\pi \wedge \varphi_\pi) & | \\
& \Box(\varphi_\pi \rightarrow \varphi_\pi) & \\
\varphi_\ell ::= \quad & \varphi_\pi \mid \varphi_\pi \wedge \varphi_\pi & \\
\varphi_L ::= \quad & \varphi_\ell \mid \varphi_\ell \vee \varphi_\ell &
\end{aligned}
$$

Grammars for path- (φ_π), leaf- (φ_ℓ), and class-formulas (φ_L):

MDT

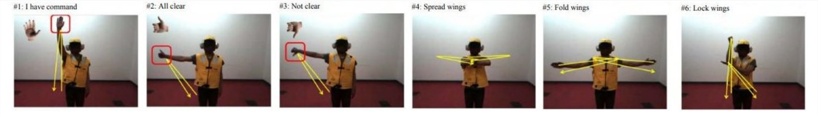
$$
\begin{aligned}
\varphi_\pi ::= \quad & \lambda \wedge (\varphi_\pi \wedge \varphi_\pi) \quad | \\
& \lambda \rightarrow (\varphi_\pi \rightarrow \varphi_\pi) \quad | \\
& \Diamond(\varphi_\pi \wedge \varphi_\pi) \quad | \\
& \Box(\varphi_\pi \rightarrow \varphi_\pi) \\
\varphi_\ell ::= \quad & \varphi_\pi \mid \varphi_\pi \wedge \varphi_\pi \\
\varphi_L ::= \quad & \varphi_\ell \mid \varphi_\ell \vee \varphi_\ell
\end{aligned}
$$

Restricted DT

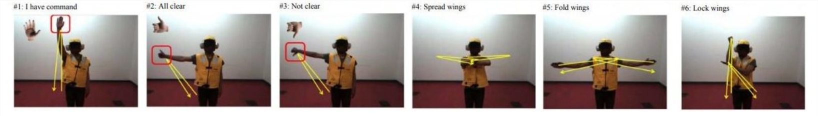
$$
\begin{aligned}
\varphi_{\swarrow} ::= \quad & p \mid (p \wedge \varphi_{\swarrow}) \mid \Diamond p \mid \Diamond(p \wedge \varphi_{\swarrow}) \\
\varphi_{\searrow} ::= \quad & \neg p \mid (\neg p \rightarrow \varphi_{\searrow}) \mid \Box\neg p \mid \Box(\neg p \rightarrow \varphi_{\searrow}) \\
\varphi_\pi ::= \quad & \varphi_{\swarrow} \mid \varphi_{\searrow} \\
\varphi_\ell ::= \quad & \varphi_\pi \mid \varphi_\pi \wedge \varphi_\pi \\
\varphi_L ::= \quad & \varphi_\ell \mid \varphi_\ell \vee \varphi_\ell
\end{aligned}
$$

Gesture recognition

- **Dataset:** NATOPS, public benchmark for time series classification
  - 360 instances;
  - 51 points;
  - 24 variables.
- **Task:** Gesture recognition from position sensors;
- **Input:** xyz coordinates for elbows/wrists/hands/thumbs (left+right) evolving through time;
- **Output:** gesture type (6 classes):



- **Dataset:** NATOPS, public benchmark for time series classification
  - 360 instances;
  - 51 points: using a full, interval-based frame results in $\frac{51 \times 52}{2} = 1326$ intervals;
  - 24 variables: using $\mathcal{F} = \{\min, \max\}$ results in 24 attributes.
- **Task:** Gesture recognition from position sensors;
- **Input:** xyz coordinates for elbows/wrists/hands/thumbs (left+right) evolving through time;
- **Output:** gesture type (6 classes):



NN	SVM (non-NN SOTA)	CART Decision Tree	**CART Modal Decision Tree**
97.1%	88.5%	70.9%	**89.7%**

Table 2: Accuracies (10-fold cv) for non-symbolic state-of-the-art and decision tree models.

What Does It Mean to Perform an Experiment?

Data

What Does It Mean to Perform an Experiment?

Data $\longrightarrow$ standard pre-process

What Does It Mean to Perform an Experiment?

$$\text{Data} \xrightarrow{\qquad\qquad} \begin{array}{c}\text{standard}\\ \text{pre-process}\end{array} \longrightarrow \begin{array}{c}\text{identify}\\ \text{modal logic}\\ \text{set of functions}\end{array}$$

What Does It Mean to Perform an Experiment?

Data $\longrightarrow$ standard pre-process $\longrightarrow$ identify modal logic set of functions

$\downarrow$

apply feature extraction and selection

What Does It Mean to Perform an Experiment?

$$\begin{tikzcd} \text{Data} \arrow[r] & \begin{array}{c}\text{standard}\\ \text{pre-process}\end{array} \arrow[r] & \begin{array}{c}\text{identify}\\ \text{modal logic}\\ \text{set of functions}\end{array} \arrow[d] \\ & \begin{array}{c}\text{apply}\\ \text{transformer}\end{array} & \begin{array}{c}\text{apply}\\ \text{feature extraction}\\ \text{and selection}\end{array} \arrow[l] \end{tikzcd}$$

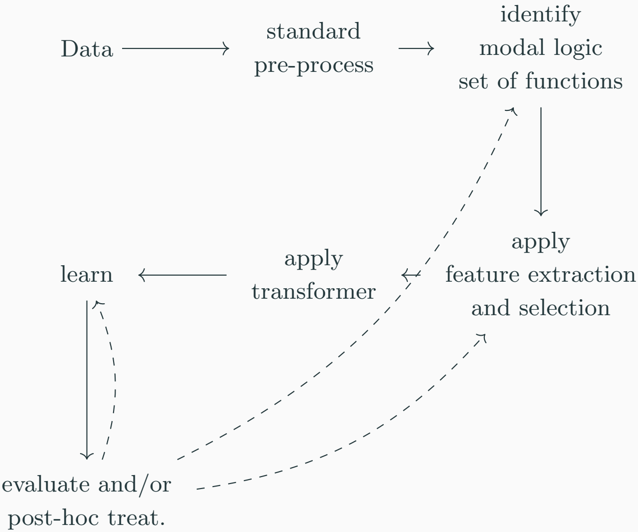
What Does It Mean to Perform an Experiment?

Data $\longrightarrow$ standard pre-process $\longrightarrow$ identify modal logic set of functions

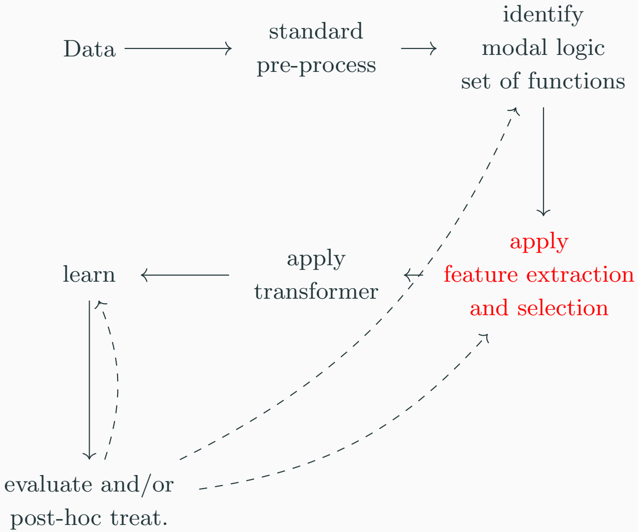
$\downarrow$

learn $\longleftarrow$ apply transformer $\longleftarrow$ apply feature extraction and selection

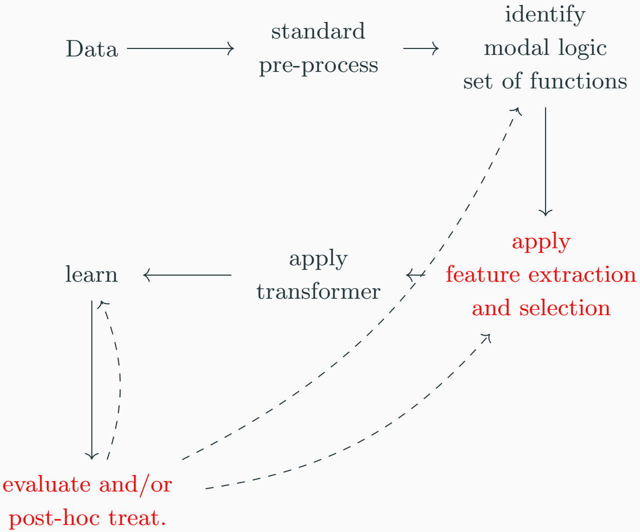
What Does It Mean to Perform an Experiment?



What Does It Mean to Perform an Experiment?



What Does It Mean to Perform an Experiment?



Symbolic Feature Extraction and Selection

Symbolic Feature Extraction and Selection: Introduction

Feature selection in tabular data is the process of selecting a subset of the variables, with the aim of reducing the amount of data, ease the learning process, and gain initial information on the problem at hand.

Feature selection in tabular data is the process of selecting a subset of the variables, with the aim of reducing the amount of data, ease the learning process, and gain initial information on the problem at hand. By stepping up to the modal case, this process can be generalized, and we can ask the question of which variables, which functions (that is, which variables), and which worlds contain the most information for the problem at hand. This is symbolic feature selection.

Symbolic Feature Extraction and Selection: Dimensional Case - 1

Feature selection methods in the tabular case are usually separated into:

- filters – to be applied independently from the learning model,
- wrappers – that work in strict connection with a learning model, and
- embedded – which are part of the learning model itself.

Symbolic Feature Extraction and Selection: Dimensional Case - 1

Feature selection methods in the tabular case are usually separated into:

- filters – to be applied independently from the learning model,
- wrappers – that work in strict connection with a learning model, and
- embedded – which are part of the learning model itself.

Designing a wrapper method requires working with a learning model; wrapper are usually well-behaved, but computationally expensive. Embedded models are new learning models, in a sense, and the selection problem in this case is not separated from the learning one.

Symbolic Feature Extraction and Selection: Dimensional Case - 1

Feature selection methods in the tabular case are usually separated into:

- filters – to be applied independently from the learning model,
- wrappers – that work in strict connection with a learning model, and
- embedded – which are part of the learning model itself.

Designing a wrapper method requires working with a learning model; wrapper are usually well-behaved, but computationally expensive. Embedded models are new learning models, in a sense, and the selection problem in this case is not separated from the learning one. We cover, here, the filter case, which is the simplest one, and easiest to understand and implement.

Symbolic Feature Extraction and Selection: Dimensional Case - 2

Two steps can be easily identified, already at dimension 0:

- Unsupervised: select variables with methods that are independent from the class (if it exists), and validate the selection;
- Supervised: select variables with methods that use the class information (assuming it exists), and validate the selection (in this case the validation could be, but it does not need to, a model learning step).

Symbolic Feature Extraction and Selection: Dimensional Case - 3

For higher dimensions, the schema can be generalized:

- Unsupervised: select variables/feature extraction functions/worlds with methods that are independent from the class (if it exists), and validate the selection;
- Supervised: select variables/feature extraction functions/worlds with methods that use the class information (assuming it exists), and validate the selection (again, in this case the validation could be, but it does not need to, a model learning step).

(Un)supervised Univariate Feature Extraction and Selection for Dimensional Data, *Proceedings of the 2nd Italian Conference on Big Data and Data Science (ITADATA)*, Cavina et al., 2023.

Symbolic Feature Extraction and Selection: Temporal Case

	V_1	V_2
I_1	37,40,40,39,40;	110,100,90,100,100;
I_2	36,41,40,38,41;	100,110,95,95,100;
I_2	37,40,38,38,39;	120,120,100,100,95;
...	...	...

Symbolic Feature Extraction and Selection: Temporal Case

	V_1	V_2
I_1	37,40,40,39,40;	110,100,90,100,100;
I_2	36,41,40,38,41;	100,110,95,95,100;
I_2	37,40,38,38,39;	120,120,100,100,95;
...	...	...

Starting from the concrete representation
of a dataset of time series
(dimensional case, dimension 1)

Symbolic Feature Extraction and Selection: Temporal Case

	V_1	V_2
I_1	37,40,40,39,40;	110,100,90,100,100;
I_2	36,41,40,38,41;	100,110,95,95,100;
I_2	37,40,38,38,39;	120,120,100,100,95;
$\dots$	$\dots$	$\dots$

we consider the set of feature
extraction functions $\mathcal{F} = \{f_1, f_2, \dots\}$

Symbolic Feature Extraction and Selection: Temporal Case

	V_1	V_2
I_1	37,40,40,39,40;	110,100,90,100,100;
I_2	36,41,40,38,41;	100,110,95,95,100;
I_2	37,40,38,38,39;	120,120,100,100,95;
$\ldots$	$\ldots$	$\ldots$

and we apply each one of them
to each interval of each instance

Symbolic Feature Extraction and Selection: Temporal Case

	$f_1(V_1)_{[1,2]}$	$f_1(V_2)_{[3,5]}$	$\dots$	$f_2(V_1)_{[1,3]}$	$f_2(V_2)_{[3,4]}$	$\dots$
I_1	37	110	$\dots$	40	100	$\dots$
I_2	38	100	$\dots$	41	110	$\dots$
I_2	37.5	120	$\dots$	39	105	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

Symbolic Feature Extraction and Selection: Temporal Case

	$f_1(V_1)_{[1,2]}$	$f_1(V_2)_{[3,5]}$	$\ldots$	$f_2(V_1)_{[1,3]}$	$f_2(V_2)_{[3,4]}$	$\ldots$
I_1	37	110	$\ldots$	40	100	$\ldots$
I_2	38	100	$\ldots$	41	110	$\ldots$
I_2	37.5	120	$\ldots$	39	105	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$

then we apply the standard feature selection
at both the unsupervised and supervised
level (if applicable)

Symbolic Feature Extraction and Selection: Temporal Case

	$f_1(V_1)_{[1,2]}$	$f_1(V_2)_{[3,5]}$	$\dots$	$f_2(V_1)_{[1,3]}$	$f_2(V_2)_{[3,4]}$	$\dots$
I_1	37	110	$\dots$	40	100	$\dots$
I_2	38	100	$\dots$	41	110	$\dots$
I_2	37.5	120	$\dots$	39	105	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

to select a set of triples
variable-feature-interval

Symbolic Feature Extraction and Selection: Temporal Case

	$f_1(V_1)_{[1,2]}$	$f_1(V_2)_{[3,5]}$	$\dots$	$f_2(V_1)_{[1,3]}$	$f_2(V_2)_{[3,4]}$	$\dots$
I_1	37	110	$\dots$	40	100	$\dots$
I_2	38	100	$\dots$	41	110	$\dots$
I_2	37.5	120	$\dots$	39	105	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

then we aggregate them
in order to interpret the result
and answer the question of which
variables/feature/intervals carry the most information

Symbolic Feature Extraction and Selection: Temporal Case

	V_2
I_1	110,100,90,100,100;
I_2	100,110,95,95,100;
I_2	120,120,100,100,95;
$\dots$	$\dots$

Symbolic Feature Extraction and Selection: Temporal Case

	V_2
I_1	110,100,90,100,100;
I_2	100,110,95,95,100;
I_2	120,120,100,100,95;
$\ldots$	$\ldots$

variables found to carry too
few information can be excluded
from the dataset

Symbolic Feature Extraction and Selection: Temporal Case

	V_2
I_1	110,100,90,100,100;
I_2	100,110,95,95,100;
I_2	120,120,100,100,95;
...	...

functions found to extract too
few information can be excluded
from the learning process

Symbolic Feature Extraction and Selection: Temporal Case

	V_2
I_1	110,100,90,100,100;
I_2	100,110,95,95,100;
I_2	120,120,100,100,95;
...	...

to ensure completeness, however
intervals are kept in any case

	V_2
I_1	110,100,90,100,100;
I_2	100,110,95,95,100;
I_2	120,120,100,100,95;
$\dots$	$\dots$

nevertheless, the information gained during the process is relevant on its own, and can be validated via standard statistical univariated methods and techniques

Symbolic Feature Extraction and Selection: Dimensional Case - 4

For higher dimensions, the schema can be generalized by simply using hyper-windows instead of intervals.

Symbolic Feature Extraction and Selection: Dimensional Case - 4

For higher dimensions, the schema can be generalized by simply using hyper-windows instead of intervals. It does not always make the same intuitive sense, and other symbolic techniques can be used.

For higher dimensions, the schema can be generalized by simply using hyper-windows instead of intervals. It does not always make the same intuitive sense, and other symbolic techniques can be used. Systematic selection following the same idea has not been studied or tried yet for other types of unstructured data.

Symbolic Feature Extraction and Selection: Summary

Feature extraction and selection has the same role in modal symbolic learning as it has in standard (symbolic) learning. It should part of every serious data science/information extraction exercise, and its results are relevant on its own.

Symbolic Feature Extraction and Selection: Summary

Feature extraction and selection has the same role in modal symbolic learning as it has in standard (symbolic) learning. It should part of every serious data science/information extraction exercise, and its results are relevant on its own. At the modal level, the feature extraction functions assume a more central role, and this phase's importance grows accordingly.

Solid Results, Multimodal Models, and Ensembles

Statistically Solid Results

Modal symbolic learning is about extracting models of the data. Models are subject to the same evaluation as in any other case.

Modal symbolic learning is about extracting models of the data. Models are subject to the same evaluation as in any other case. Let us focus, in particular, on classification problems, and let us go over classical concepts of statistical evaluation of classification models that immediately apply here.

On which instances is the learned model tested?

https://mlu-explain.github.io/cross-validation/

Classical Performance Indicators and Test Modes - 1

On which instances is the learned model tested?

the same ones from which it has been learned from

this is called full training evaluation mode, and it is not reliable

https://mlu-explain.github.io/cross-validation/

Classical Performance Indicators and Test Modes - 1

On which instances is the learned model tested?

the dataset has been used partly for learning and partly for testing

↑

this is called training+test evaluation mode, and it is partially reliable, but it strongly depends on chance

https://mlu-explain.github.io/cross-validation/

Classical Performance Indicators and Test Modes - 1

On which instances is the learned model tested?

the dataset has been sliced into several subsets, and each one of them has played, in different moments, the role of learning and testing dataset; the results are averaged

↑

this is called cross-validation evaluation mode, and it is very reliable, not depending at all on chance

https://mlu-explain.github.io/cross-validation/

What do we measure?

Error
if it is a classification
problem

What do we measure?

Classical Performance Indicators and Test Modes - 2

What do we measure? $\longrightarrow$

Error if it is a classification problem

Distance if it is a regression problem

Classical Performance Indicators and Test Modes - 2

What do we measure?

- → **Error** if it is a classification problem
- → **Distance** if it is a regression problem
- → Other statistical indicators if it is an unsupervised problem

Classical Performance Indicators and Test Modes - 2

What do we measure?

- **Errors** if it is a classification problem
- **Distance** if it is a regression problem
- **Other statistical indicators** if it is an unsupervised problem

confusion matrix:

ground truth yes predicted yes true positive (TP)	

confusion matrix:

ground truth yes predicted yes true positive (TP)	ground truth yes predicted no false negative (FN)

confusion matrix:

ground truth yes predicted yes true positive (TP)	ground truth yes predicted no false negative (FN)
ground truth no predicted yes false positive (FP)	

confusion matrix:

ground truth yes predicted yes true positive (TP)	ground truth yes predicted no false negative (FN)
ground truth no predicted yes false positive (FP)	ground truth no predicted no true negative (TN)

TP, TN, FP, FN are combined
into several metrics
that give us several
indications of performances

Statistically Solid Results: Summary

Models are usual learned in full training for an initial exploration of data. Then, whenever possible, experiments are run in training+test or in cross-validation to evaluate the solidity through all essential performance indicators. Finally, if applicable, models learned in full training are deployed.

Multimodal Models: Introduction

Multimodal models are models designed to learn from dataset whose instances present the information in more than one way at the same time.

Multimodal models are models designed to learn from dataset whose instances present the information in more than one way at the same time. Typical examples include instances with temporal and image information (e.g., voice+video), but also medical cases with patients described by more than one diagnostic exam (e.g., EEG+CT SCAN).

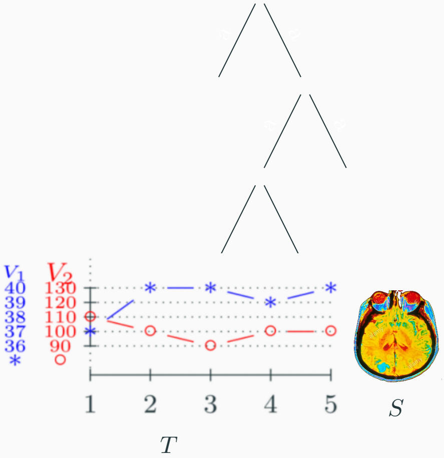
Multimodal Models: Introduction

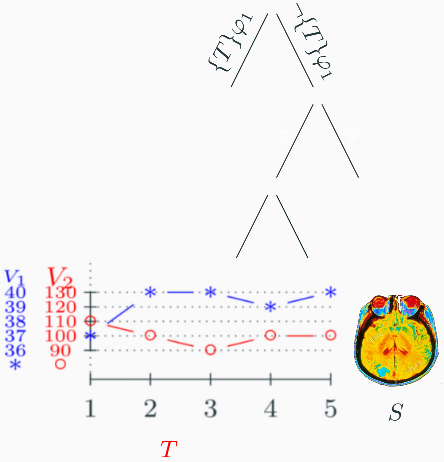
Multimodal models are models designed to learn from dataset whose instances present the information in more than one way at the same time. Typical examples include instances with temporal and image information (e.g., voice+video), but also medical cases with patients described by more than one diagnostic exam (e.g., EEG+CT SCAN).

Can a single algorithm (e.g., CART) learn from more than one source at the same time?

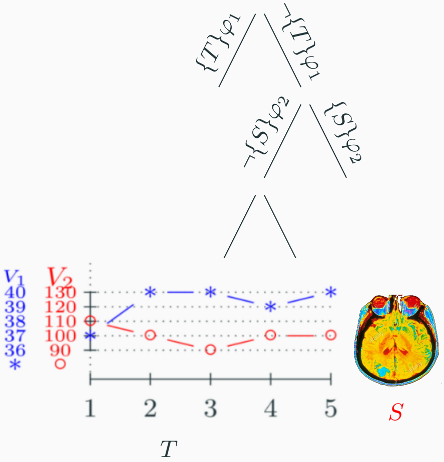
Multimodal models are models designed to learn from dataset whose instances present the information in more than one way at the same time. Typical examples include instances with temporal and image information (e.g., voice+video), but also medical cases with patients described by more than one diagnostic exam (e.g., EEG+CT SCAN).

Can a single algorithm (e.g., CART) learn from more than one source at the same time? Since modal decision trees have shown the ability of learning from unstructured data, this question is even more interesting than it was before.

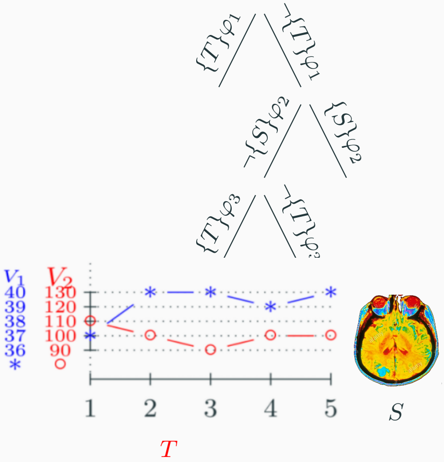

V_1
V_2
40
39
38
37
36
130
120
110
100
90
1
2
3
4
5
T
S


$\{T\}\varphi_1$
$\neg\{T\}\varphi_1$
V_1
V_2
40 130
39 120
38 110
37 100
36 90
1 2 3 4 5
S
T

Multimodal Trees - 1



Multimodal Trees - 1



Multimodal Trees: Summary

Multimodal trees learn from multimodal instances; here the word multimodal does not refer to the fact that the logic is multi-modal: each single ‘mode’ in which the instance is presented is described, in general, with a multi-modal logic (e.g., in the temporal ‘mode’ we use $\mathcal{HS}$ with 12 modalities, in the spatial ‘mode’ we use $\mathcal{HS}^2$, with 168).

Multi-Frame Modal Symbolic Learning. Proceedings of the 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY). Pagliarini et al., 2021.

Multimodal trees learn from multimodal instances; here the word multimodal does not refer to the fact that the logic is multi-modal: each single 'mode' in which the instance is presented is described, in general, with a multi-modal logic (e.g., in the temporal 'mode' we use $\mathcal{HS}$ with 12 modalities, in the spatial 'mode' we use $\mathcal{HS}^2$, with 168). The learning algorithm can be adapted to optimize (greedily) also the mode from which we perform a single split.

Multi-Frame Modal Symbolic Learning. Proceedings of the 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY). Pagliarini et al., 2021.

Multimodal Trees: Summary

Multimodal trees learn from multimodal instances; here the word multimodal does not refer to the fact that the logic is multi-modal: each single ‘mode’ in which the instance is presented is described, in general, with a multi-modal logic (e.g., in the temporal ‘mode’ we use $\mathcal{HS}$ with 12 modalities, in the spatial ‘mode’ we use $\mathcal{HS}^2$, with 168). The learning algorithm can be adapted to optimize (greedily) also the mode from which we perform a single split. The resulting tree can be read in terms of formulas in the same way, only each formula is referred to a specific ‘mode’.

Multi-Frame Modal Symbolic Learning. Proceedings of the 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis (OVERLAY). Pagliarini et al., 2021.

All learning models display a statistical behaviour. As such, as it is well-known, different ‘copies’ of the same (or even different) models learned on the same dataset, but under slightly different conditions, can be grouped together and then aggregated to take a final decision.

https://mlu-explain.github.io/random-forest/

Ensembles

All learning models display a statistical behaviour. As such, as it is well-known, different 'copies' of the same (or even different) models learned on the same dataset, but under slightly different conditions, can be grouped together and then aggregated to take a final decision. The most famous way to do this is to use random forests with several trees (usually in the hundreds); the corresponding concept at the modal level is that of modal random forests.

https://mlu-explain.github.io/random-forest/

Practice: experiments with Sole.jl

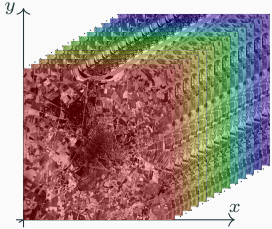
Interpretable COVID-19 Diagnosis

- **Dataset:** COVID-19 Sounds
  - 198 instances;
  - 2 modalities: cough & breath audio recordings;
  - 800 & 90 temporal points for cough & breath, respectively;
  - 30 variables.
- **Task:** COVID-19 diagnosis from breath+cough;
- **Input:** Frequency components (via Fourier transform) evolving through time during a single cough/breath event;
- **Output:** NEGATIVE/POSITIVE (2 classes):

Exploring automatic diagnosis of COVID-19 from crowdsourced respiratory sound data, *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining.* Brown et al., 2020.

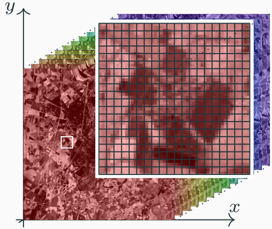
Land Cover Classification

- **Dataset:** Pavia University, benchmark for land cover classification (LCC)
  - An 610×340 image (*scene*);
  - $42,776$ pixel labels;
  - 103 variables (*hyperspectral channels*).
- **Task:** Image segmentation;
- **Input:** A pixel's coordinates;
- **Output:** Class label (2 classes):



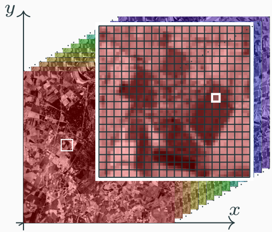
Land Cover Classification

- **Dataset:** Pavia University, benchmark for land cover classification (LCC)
  - An 610×340 image (*scene*);
  - $42,776$ pixel labels;
  - 103 variables (*hyperspectral channels*).
- **Task:** Image segmentation;
- **Input:** A pixel's coordinates;
- **Output:** Class label (2 classes):



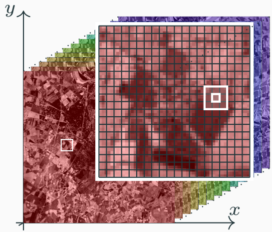
Land Cover Classification

- **Dataset:** Pavia University, benchmark for land cover classification (LCC)
  - An 610×340 image (*scene*);
  - $42,776$ pixel labels;
  - 103 variables (*hyperspectral channels*).
- **Task:** Image segmentation;
- **Input:** A pixel's coordinates;
- **Output:** Class label (2 classes):



Land Cover Classification

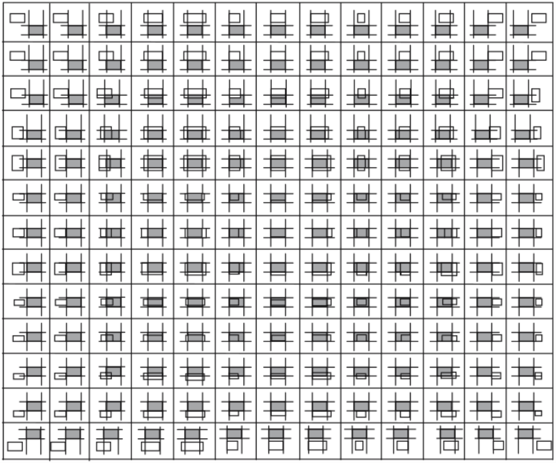
- **Dataset:** Pavia University, benchmark for land cover classification (LCC)
  - An 610×340 image (*scene*);
  - $42,776$ pixel labels;
  - 103 variables (*hyperspectral channels*).
- **Task:** Image segmentation;
- **Input:** A pixel's coordinates;
- **Output:** Class label (2 classes):



Land Cover Classification via a rectangle logic ($\mathcal{HS}^{\mathbf{2}}$)

Meet $\mathcal{HS}^{\mathbf{2}}$, a 2D spatial logic of *rectangles* with 169 *directional relations*:

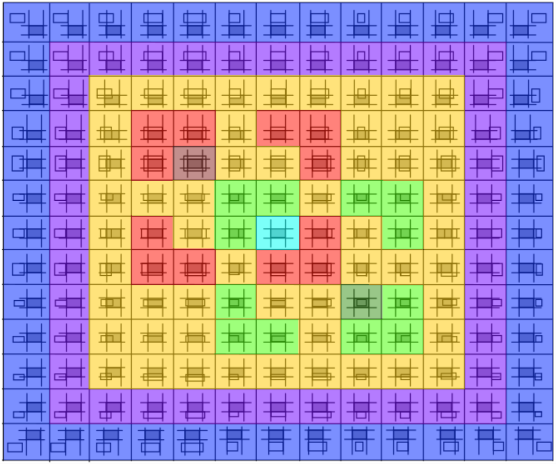
$\mathcal{HS}^{\mathbf{2}}$



$$\langle G \rangle \, (\neg p \wedge \langle LL \rangle q \wedge \langle AL \rangle r)$$

These relations can be combined to capture *topological* aspects:

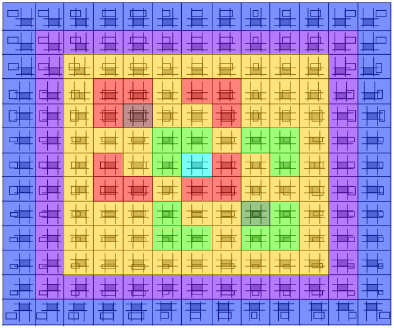
$\mathcal{HS}^{\mathbf{2}}{}_{RCC8}$



$\langle G \rangle (\neg p \wedge \langle PO \rangle q \wedge \langle TPP \rangle r)$

These relations can be combined to capture *topological* aspects:

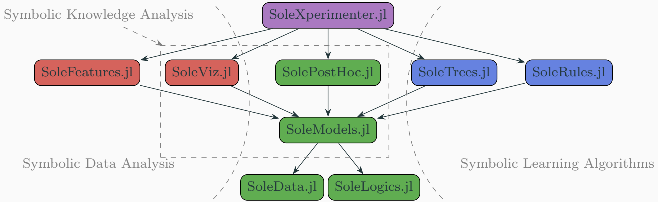
$$\mathcal{HS}^{\mathbf{2}}{}_{RCC8}$$



$\mathcal{HS}^{\mathbf{2}}_{\mathbf{RCC8}}$ **operator**
$\langle DC \rangle$ (disconnected)
$\langle EC \rangle$ (externally connected)
$\langle PO \rangle$ (partially overlapping)
$\langle TPP \rangle$ (tangential proper part)
$\langle TPPi \rangle$ (tangential proper part inverse)
$\langle NTPP \rangle$ (non-tangential proper part)
$\langle NTPPi \rangle$ (non-tangential proper part inverse)

$$\langle G \rangle \left(\neg p \wedge \langle PO \rangle q \wedge \langle TPP \rangle r\right)$$

What's next?



- Minimization of logical formulas;
- Neuro-symbolic hybrids;
- Rule extraction from ensembles of trees;
- Generalization to fuzzy logics.

Modal Symbolic Learning: A Tutorial

Giovanni Pagliarini[1,2] giovanni.pagliarini@unife.it
Guido Sciavicco[1] guido.sciavicco@unife.it

December 6-7-8$^{\text{th}}$, 2023

[1]Applied Computational Logic and Artificial Intelligence (ACLAI) Laboratory,
Department of Mathematics and Computer Science, University of Ferrara, Italy

[2]Department of Mathematical, Physical, and Computer Sciences, University of Parma, Italy

Thanks to

Mauro Milella	Federico Manzella
Michele Ghiotti	Eduard I. Stan
Lorenzo Balboni	Patrik Cavina