Logic-Based Explainability in Machine Learning

Joao Marques-Silva

IRIT/CNRS & ANITI DeepLever Chair, Toulouse, France

September 2022

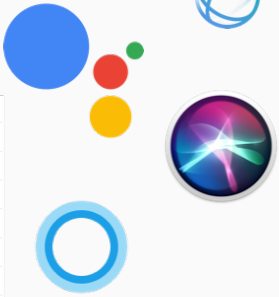
Recent & ongoing ML successes

https://en.wikipedia.org/wiki/Waymo

DeepMind

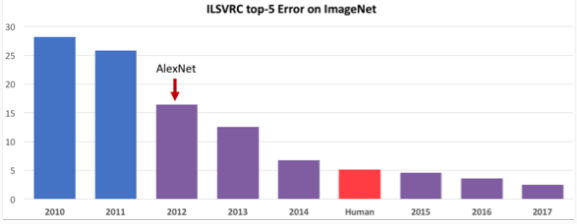
AlphaGo

AlphaGo Zero & **Alpha Zero**



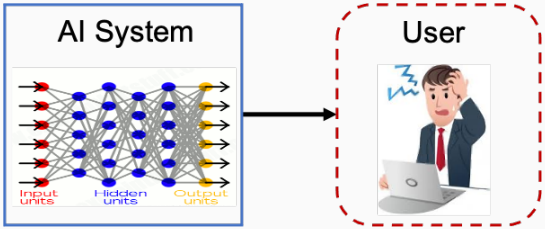
https://fr.wikipedia.org/wiki/Pepper_(robot)

Image & Speech Recognition



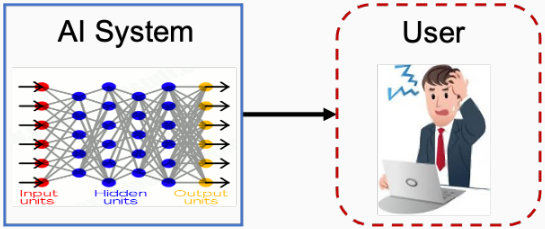
http://gradientscience.org/intro_adversarial/

eXplainable AI (XAI)



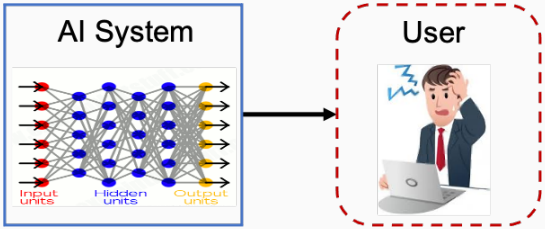
- ML models are most often **opaque**
- Goal of XAI: **to help humans understand ML models**
- Many questions to address:

eXplainable AI (XAI)



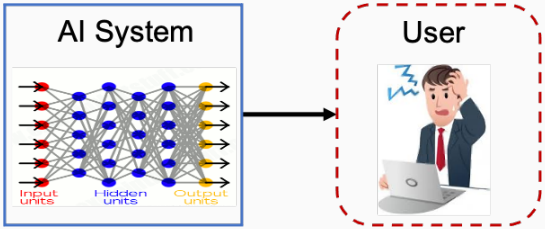
- ML models are most often **opaque**
- Goal of XAI: **to help humans understand ML models**
- Many questions to address:
  - Properties of explanations
    - How to be human understandable?
    - How to answer **Why?** questions? I.e. Why the prediction?
    - How to answer **Why Not?** questions? I.e. Why not some other prediction?
    - Which guarantees of rigor?

eXplainable AI (XAI)



- ML models are most often **opaque**
- Goal of XAI: **to help humans understand ML models**
- Many questions to address:
  - Properties of explanations
    - How to be human understandable?
    - How to answer **Why?** questions? I.e. Why the prediction?
    - How to answer **Why Not?** questions? I.e. Why not some other prediction?
    - Which guarantees of rigor?
  - Other queries: enumeration, membership, preferences, etc.

eXplainable AI (XAI)



- ML models are most often **opaque**
- Goal of XAI: **to help humans understand ML models**
- Many questions to address:
  - Properties of explanations
    - How to be human understandable?
    - How to answer **Why?** questions? I.e. Why the prediction?
    - How to answer **Why Not?** questions? I.e. Why not some other prediction?
    - Which guarantees of rigor?
  - Other queries: enumeration, membership, preferences, etc.
  - Links with robustness, fairness, learning

Importance of XAI

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

European Union regulations on algorithmic decision-making and a "right to explanation"

Bryce Goodman,[1]* Seth Flaxman,[2]

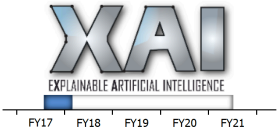
■ *We summarize the potential impact that the European Union's new General Data Protection Regulation will have on the routine use of machine-learning algorithms. Slated to take effect as law across the European Union in 2018, it will place restrictions on automated individual decision making (that is, algorithms that make decisions based on user-level predictors) that "significantly affect" users. When put into practice, the law may also effectively create a right to explanation, whereby a user can ask for an explanation of an algorithmic decision that significantly affects them. We argue that while this law may pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and evaluation frameworks that avoid discrimination and enable explanation.*

Proposal for a

REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

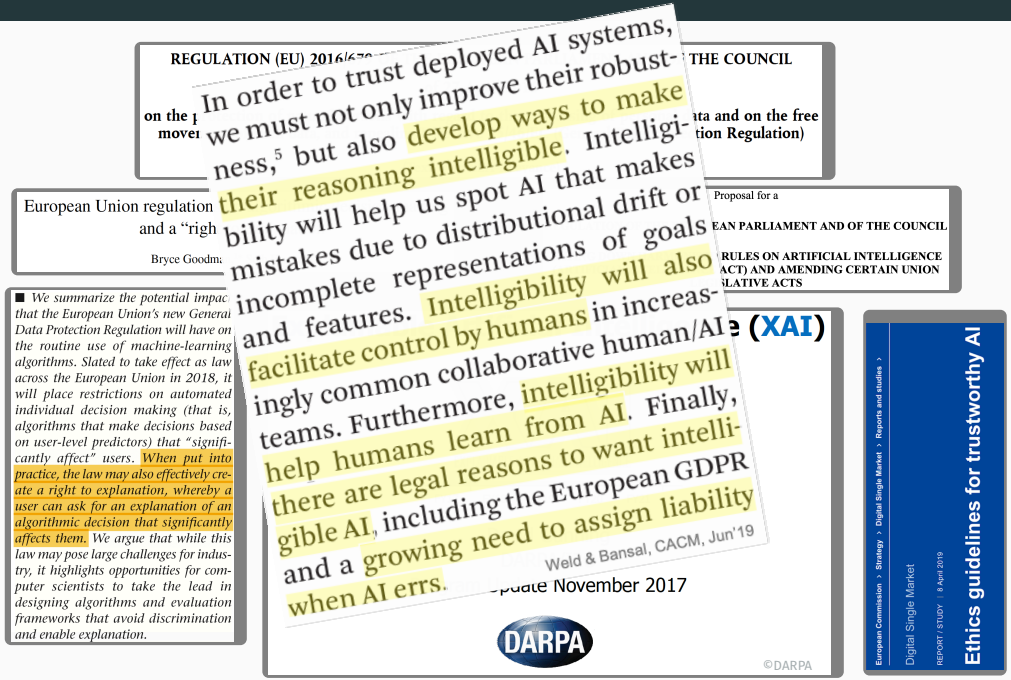
LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS

Explainable Artificial Intelligence (XAI)

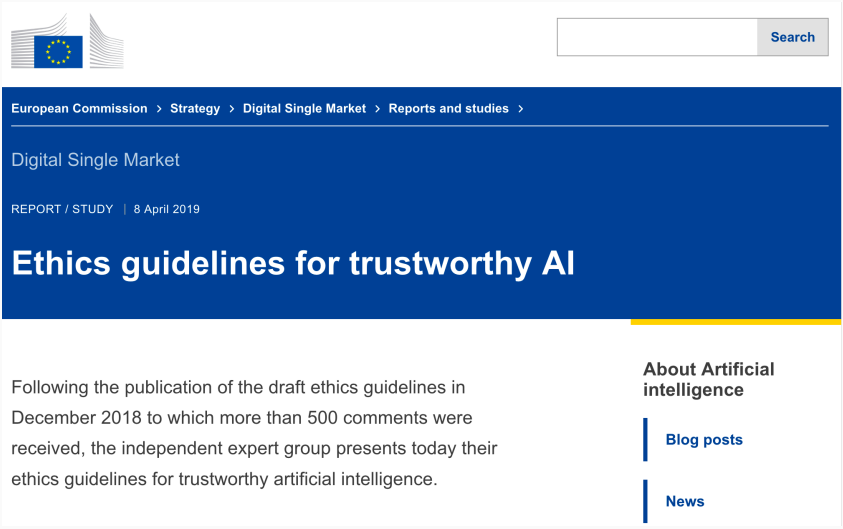


David Gunning
DARPA/I2O
Program Update November 2017

Importance of XAI



XAI & EU guidelines (AI HLEG)



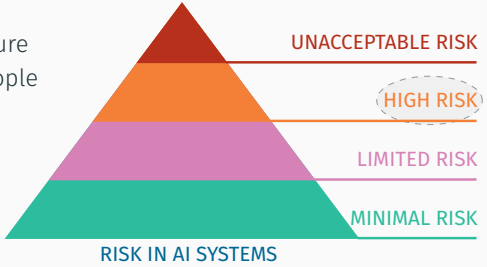
XAI & the principle of explicability

& thousands of recent papers!

XAI for high-risk & safety-critical applications

- High-risk (EU regulations):
  - Law enforcement
  - Management and operation of critical infrastructure
  - Biometric identification and categorization of people
  - ...
- Safety-critical:
  - Self-driving cars
  - Autonomous vehicles
  - Autonomous aereal devices
  - ...

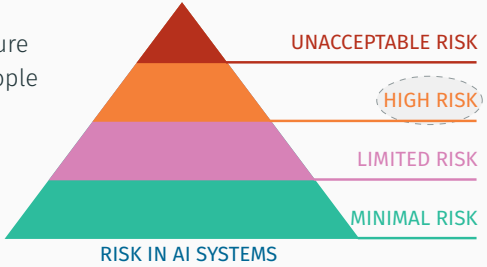
[EU21]



XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):
  - Law enforcement
  - Management and operation of critical infrastructure
  - Biometric identification and categorization of people
  - ...
- **Safety-critical**:
  - Self-driving cars
  - Autonomous vehicles
  - Autonomous aereal devices
  - ...

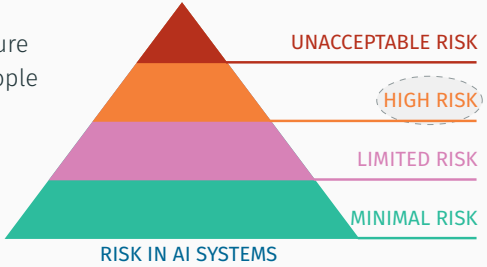
[EU21]



XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):
  - Law enforcement
  - Management and operation of critical infrastructure
  - Biometric identification and categorization of people
  - ...
- **Safety-critical**:
  - Self-driving cars
  - Autonomous vehicles
  - Autonomous aereal devices
  - ...
- **Correctness of explanations is paramount!**
  - To build trust
  - To help debug AI systems
  - To prevent accidents
  - ...

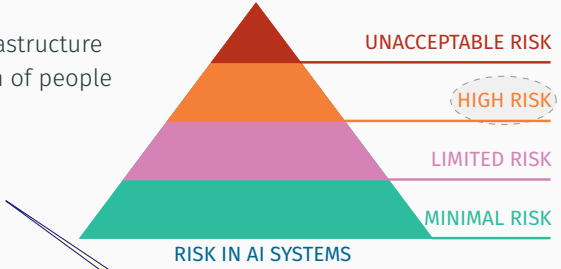
[EU21]



XAI for high-risk & safety-critical applications

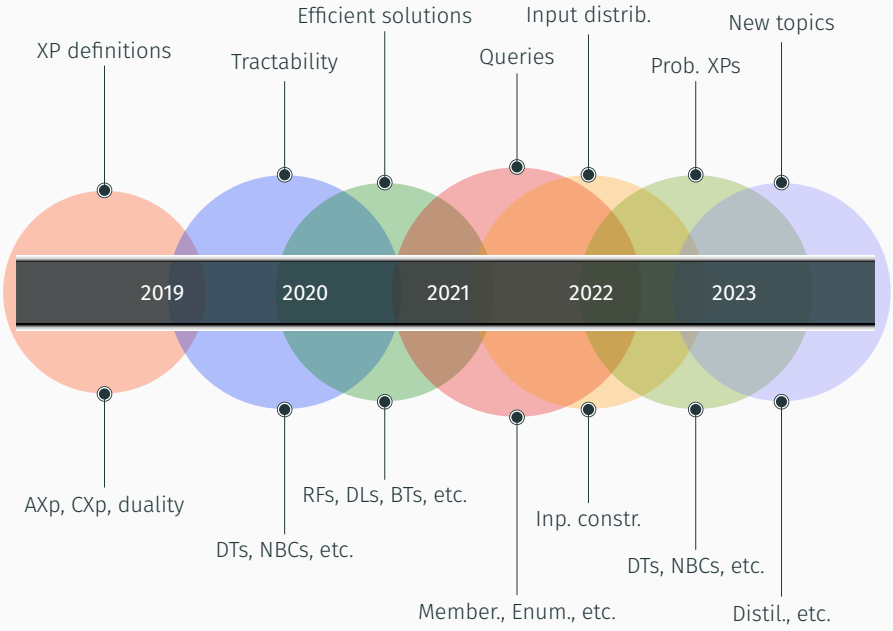
- **High-risk** (EU regulations):
  - Law enforcement
  - Management and operation of critical infrastructure
  - Biometric identification and categorization of people
  - ...
- **Safety-critical**:
  - Self-driving cars
  - Autonomous vehicles
  - Autonomous aereal devices
  - ...
- **Correctness of explanations is paramount!**
  - To build trust
  - To help debug AI systems
  - To prevent accidents
  - ...

[EU21]



Main motivation for our work!

The emergence of formal explainability – timeline



Today's short course – main topics

- Part #0: Motivation
- Part #1: Preliminaries & issues with XAI
- Part #2: Formal explainability
- Part #3: Computing explanations – basics
- Q&A and Break
- Part #4: Computing explanations – encodings
- Part #5: Tractable explanations
- Part #6: Explainability queries
- Part #7: Probabilistic explanations
- Part #8: Additional topics & wrap-up
- Q&A and Conclude

Part 1

Preliminaries & XAI Issues

Definitions – classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, m\}$, each feature i taking values from domain D_i
  - Features can be categorical, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{m} D_i$

- Set of classes $\mathcal{K} = \{c_1, \ldots, c_K\}$

Definitions – classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, m\}$, each feature i taking values from domain D_i
  - Features can be categorical, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{m} D_i$

- Set of classes $\mathcal{K} = \{c_1, \ldots, c_K\}$

- ML model $\mathbb{M}$ computes a (non-constant) classification function $\kappa : \mathbb{F} \to \mathcal{K}$

Definitions – classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, m\}$, each feature i taking values from domain D_i
  - Features can be categorical, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{m} D_i$
- Set of classes $\mathcal{K} = \{c_1, \ldots, c_K\}$
- ML model $\mathbb{M}$ computes a (non-constant) classification function $\kappa : \mathbb{F} \to \mathcal{K}$
- Instance $(\mathbf{v}, c)$ for point $\mathbf{v} = (v_1, \ldots, v_m) \in \mathbb{F}$, with prediction $c = \kappa(\mathbf{v})$, $c \in \mathcal{K}$
  - **Goal:** to compute explanations for $(\mathbf{v}, c)$

Simple examples – (unordered) rules

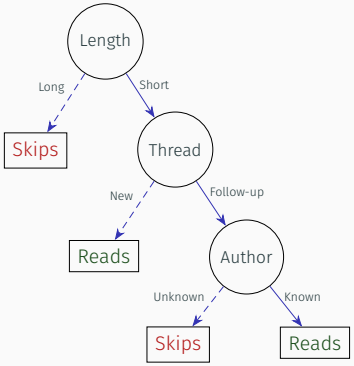
IF $x_1 + x_2 \geq 0$ THEN predict ⊞

IF $x_1 + x_2 < 0$ THEN predict ⊟

$\mathcal{F} = \{1, 2\}; \mathcal{D}_1 = \mathcal{D}_2 = \mathbb{R}; \mathcal{K} = \{⊞, ⊟\}$

Simple examples – (unordered) rules & decision tree

IF $x_1 + x_2 \geq 0$ THEN predict $\boxplus$
IF $x_1 + x_2 < 0$ THEN predict $\boxminus$
$\mathcal{F} = \{1, 2\}; \mathcal{D}_1 = \mathcal{D}_2 = \mathbb{R}; \mathcal{K} = \{\boxplus, \boxminus\}$



Features: Length, Thread, Author
$\mathcal{F} = \{1, 2, 3\}$
$\mathcal{K} = \{\text{Skips}, \text{Reads}\}$
$\mathcal{D}_1 = \{\text{Long}, \text{Short}\}$
$\mathcal{D}_2 = \{\text{New}, \text{Follow-up}\}$
$\mathcal{D}_3 = \{\text{Unknown}, \text{Known}\}$

Outline

Standard tools of the trade

- SAT: decision problem for propositional logic
  - Formulas most often represented in CNF
  - There are optimization variants: MaxSAT, PBO, MinSAT, etc.
  - There are quantified variants: QBF, QMaxSAT, etc.
- SMT: decision problem for (decidable) fragments of first-order logic (FOL)
  - There are optimization variants: MaxSMT, etc.
  - There are quantified variants
- MILP: decision/optimization problems defined on conjunctions of linear inequalities over integer & real-valued variables
- CP: constraint programming
  - There are optimization/quantified variants

Standard tools of the trade

- SAT: decision problem for propositional logic
  - Formulas most often represented in CNF
  - There are optimization variants: MaxSAT, PBO, MinSAT, etc.
  - There are quantified variants: QBF, QMaxSAT, etc.

Basic knowledge on SAT & SMT assumed. See links below.

- SMT: decision problem for (decidable) fragments of first-order logic (FOL)
  - There are optimization variants: MaxSMT, etc.
  - There are quantified variants
- MILP: decision/optimization problems defined on conjunctions of linear inequalities over integer & real-valued variables
- CP: constraint programming
  - There are optimization/quantified variants
- Background on SAT/SMT: [BHvMW09]
  - `https://alexeyignatiev.github.io/ssa-school-2019/`
  - `https://alexeyignatiev.github.io/ijcai19tut/`

Basic definitions in propositional logic

- Variables ($\{x, x_1, \ldots\}$) & literals ($x_1, \neg x_1$)
- Well-formed formulas using $\neg$, $\wedge$,$\vee$, $\ldots$
- Clause: disjunction of literals
- Term: conjunction of literals
- Conjunctive normal form (CNF): conjunction of clauses
- Disjunctive normal form (DNF): disjunction of terms
- Simple to generalize to more expressive domains

Entailment

- Let φ represent some formula, defined on feature space $\mathbb{F}$, and representing a function $\varphi : \mathbb{F} \to \{0, 1\}$
- Let τ represent some other formula, also defined on $\mathbb{F}$, and with $\tau : \mathbb{F} \to \{0, 1\}$

Entailment

- Let φ represent some formula, defined on feature space $\mathbb{F}$, and representing a function $\varphi : \mathbb{F} \to \{0, 1\}$
- Let τ represent some other formula, also defined on $\mathbb{F}$, and with $\tau : \mathbb{F} \to \{0, 1\}$
- We say that τ entails φ, written as $\tau \vDash \varphi$, if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \to \varphi(\mathbf{x})]$$

Entailment

- Let φ represent some formula, defined on feature space $\mathbb{F}$, and representing a function $\varphi : \mathbb{F} \to \{0, 1\}$
- Let τ represent some other formula, also defined on $\mathbb{F}$, and with $\tau : \mathbb{F} \to \{0, 1\}$
- We say that τ entails φ, written as $\tau \vDash \varphi$, if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \to \varphi(\mathbf{x})]$$

  - An example:
    - $\mathbb{F} = \{0, 1\}^2$
    - $\varphi(x_1, x_2) = x_1 \vee \neg x_2$
    - Clearly, $x_1 \vDash \varphi$ and $\neg x_2 \vDash \varphi$

Entailment

- Let φ represent some formula, defined on feature space $\mathbb{F}$, and representing a function $\varphi : \mathbb{F} \to \{0, 1\}$
- Let τ represent some other formula, also defined on $\mathbb{F}$, and with $\tau : \mathbb{F} \to \{0, 1\}$
- We say that τ entails φ, written as $\tau \vDash \varphi$, if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \to \varphi(\mathbf{x})]$$

  - An example:
    - $\mathbb{F} = \{0, 1\}^2$
    - $\varphi(x_1, x_2) = x_1 \vee \neg x_2$
    - Clearly, $x_1 \vDash \varphi$ and $\neg x_2 \vDash \varphi$
  - Another example:
    - $\mathbb{F} = \{0, 1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly, $x_1 \wedge x_2 \vDash \varphi$ and $x_1 \wedge x_3 \vDash \varphi$

Entailment

- Let φ represent some formula, defined on feature space $\mathbb{F}$, and representing a function $\varphi : \mathbb{F} \to \{0, 1\}$
- Let τ represent some other formula, also defined on $\mathbb{F}$, and with $\tau : \mathbb{F} \to \{0, 1\}$
- We say that τ entails φ, written as $\tau \vDash \varphi$, if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \to \varphi(\mathbf{x})]$$

  - An example:
    - $\mathbb{F} = \{0, 1\}^2$
    - $\varphi(x_1, x_2) = x_1 \vee \neg x_2$
    - Clearly, $x_1 \vDash \varphi$ and $\neg x_2 \vDash \varphi$
  - Another example:
    - $\mathbb{F} = \{0, 1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly, $x_1 \wedge x_2 \vDash \varphi$ and $x_1 \wedge x_3 \vDash \varphi$

- For non-boolean feature spaces, we let φ_c denote the predicate $\varphi(\mathbf{x}) = c$, i.e. $\varphi_c(\mathbf{x}) \in \{0, 1\}$

Prime implicants & implicates

- A conjunction of literals π (which will be viewed as a set of literals where convenient) is a prime implicant of some function φ if,
  1. $\pi \vDash \varphi$
  2. For any $\pi' \subsetneq \pi$, $\pi' \nvDash \varphi$

Prime implicants & implicates

- A conjunction of literals π (which will be viewed as a set of literals where convenient) is a prime implicant of some function φ if,
  1. $\pi \vDash \varphi$
  2. For any $\pi' \subsetneq \pi$, $\pi' \nvDash \varphi$

  - Example:
    - $\mathbb{F} = \{0,1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly, $x_1 \wedge x_2 \vDash \varphi$
    - Also, $x_1 \nvDash \varphi$ and $x_2 \nvDash \varphi$

Prime implicants & implicates

- A conjunction of literals π (which will be viewed as a set of literals where convenient) is a prime implicant of some function φ if,
  1. $\pi \vDash \varphi$
  2. For any $\pi' \subsetneq \pi$, $\pi' \nvDash \varphi$
  - Example:
    - $\mathbb{F} = \{0,1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly, $x_1 \wedge x_2 \vDash \varphi$
    - Also, $x_1 \nvDash \varphi$ and $x_2 \nvDash \varphi$

- A disjunction of literals ρ (also viewed as a set of literals where convenient) is a prime implicate of some function φ if
  1. $\varphi \vDash \rho$
  2. For any $\rho' \subsetneq \rho$, $\varphi \nvDash \rho'$

Reasoning about inconsistency

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$, $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$

Reasoning about inconsistency

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \vDash \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$, $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \vDash \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$

Reasoning about inconsistency

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$, $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \models \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set $\mathcal{C} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\models \bot$
  - E.g. $\mathcal{C} = \{(\neg x_1)\}$

Reasoning about inconsistency

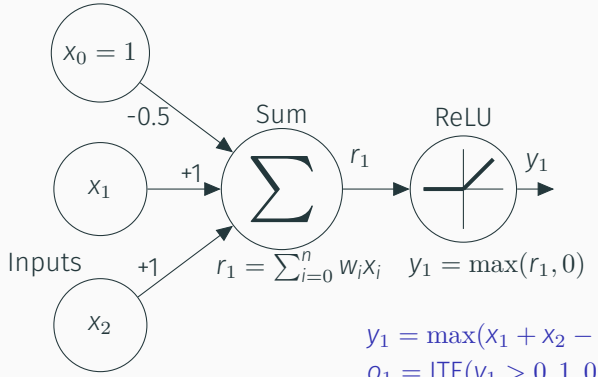
- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$, $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \models \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set $\mathcal{C} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\models \bot$
  - E.g. $\mathcal{C} = \{(\neg x_1)\}$
- Duality:
  - MUSes are minimal-hitting sets (MHSes) of MCSes, and vice-versa [Rei87]

Reasoning about inconsistency

- Formula $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$, with
  - $\mathcal{B}$: background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$: additional (inconsistent) knowledge, i.e. soft constraints
  - And, $\mathcal{T} \models \bot$
  - E.g. $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$, $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set $\mathcal{U} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup \mathcal{U} \models \bot$
  - E.g. $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set $\mathcal{C} \subseteq \mathcal{S}$, s.t. $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\models \bot$
  - E.g. $\mathcal{C} = \{(\neg x_1)\}$
- Duality:
  - MUSes are minimal-hitting sets (MHSes) of MCSes, and vice-versa [Rei87]
- Variants:
  - Smallest(-cost) MCS, i.e. complement of maximum(-cost) satisfiability (MaxSAT)
  - Smallest(-cost) MUS

Outline

Neural networks (NNs) – $(\mathbf{v}, c) = ((1, 0), 1)$



$$y_1 = \max(x_1 + x_2 - 0.5, 0)$$
$$o_1 = \mathsf{ITE}(y_1 > 0, 1, 0)$$

Decision lists (DLs) – ordered rules – $(\mathbf{v}, c) = ((0, 0, 1, 2), 1)$

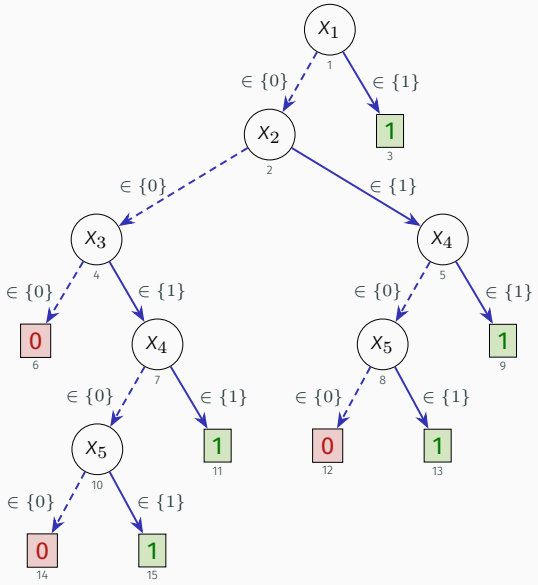
[Fla12]

Item	Definition
$\mathcal{F}_1$	$\{1, 2, 3, 4\}$
$\mathcal{D}_{1i}, i = 1, 2, 3$	$\{0, 1\}$
$\mathcal{D}_{14}$	$\{0, 1, 2\}$
$\mathcal{K}_1$	$\{0, 1\}$
κ_1	see below

R_1:	IF	$(x_1 = 1)$	THEN	0
R_2:	ELSE IF	$(x_2 = 1)$	THEN	1
R_3:	ELSE IF	$(x_4 = 1)$	THEN	0
R_{DEF}:	ELSE		THEN	1

Decision trees (DTs) – $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

[HRS19]



Monotonic classifiers – $(\mathbf{v}, c) = ((10, 10, 5, 0), A)$

[MGC+21]

Variable	Meaning	Range
$\kappa(\cdot) \triangleq M$	Student grade	$\in \{A, B, C, D, E, F\}$
S	Final score	$\in \{0, \ldots, 10\}$

Feat. id	Feat. var.	Feat. name	Domain
1	Q	Quiz	$\{0, \ldots, 10\}$
2	X	Exam	$\{0, \ldots, 10\}$
3	H	Homework	$\{0, \ldots, 10\}$
4	R	Project	$\{0, \ldots, 10\}$

$$
\begin{aligned}
M &= \mathsf{ITE}(S \geq 9, A, \mathsf{ITE}(S \geq 7, B, \mathsf{ITE}(S \geq 5, C, \mathsf{ITE}(S \geq 4, D, \mathsf{ite}(S \geq 2, E, F))))) \\
S &= \max\left[0.3 \times Q + 0.6 \times X + 0.1 \times H, R\right]
\end{aligned}
$$

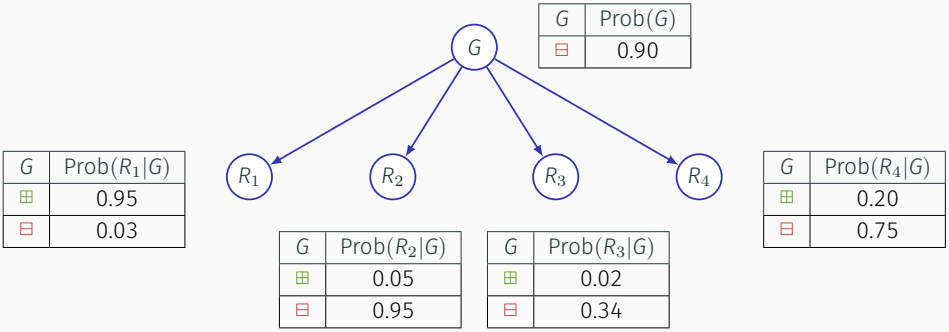
Also, $F \preccurlyeq E \preccurlyeq D \preccurlyeq C \preccurlyeq B \preccurlyeq A$

And, $\kappa(\mathbf{x_1}) \preccurlyeq \kappa(\mathbf{x_2})$ if $\mathbf{x}_1 \leq \mathbf{x}_2$

Other classifiers

- Naive-Bayes Classifier (NBC)

[Fla12, MGC+20]



G	Prob(G)
⊟	0.90

G	Prob(R_1|G)
⊞	0.95
⊟	0.03

G	Prob(R_2|G)
⊞	0.05
⊟	0.95

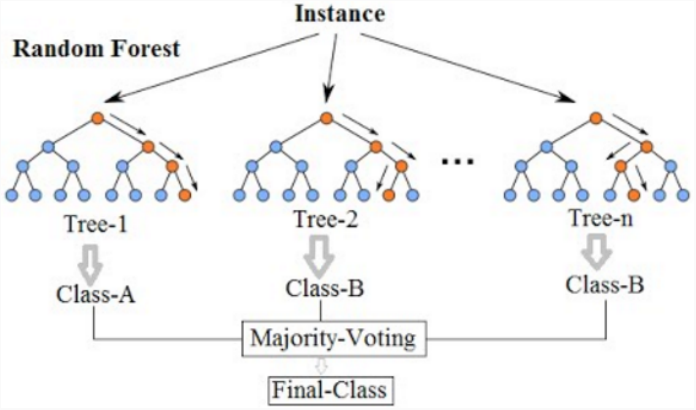
G	Prob(R_3|G)
⊞	0.02
⊟	0.34

G	Prob(R_4|G)
⊞	0.20
⊟	0.75

$$\text{Prob}(G, R_1, R_2, R_3, R_4) = \text{Prob}(G) \times \text{Prob}(R_1|G) \times \text{Prob}(R_2|G) \times \text{Prob}(R_3|G) \times \text{Prob}(R_4|G)$$

Other classifiers

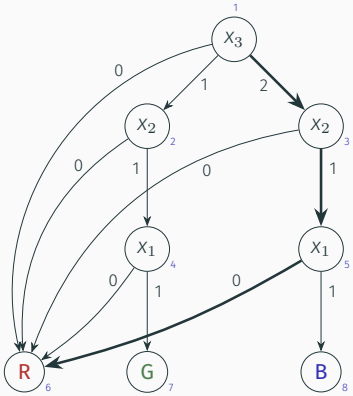
- Naive-Bayes Classifier (NBC)
- Random forests (RFs) & other tree ensembles (TEs), e.g. boosted trees (BTs)



Other classifiers

- Naive-Bayes Classifier (NBC)
- Random forests (RFs) & other tree ensembles (TEs), e.g. boosted trees (BTs)
- Decision graphs, decision diagrams, etc.

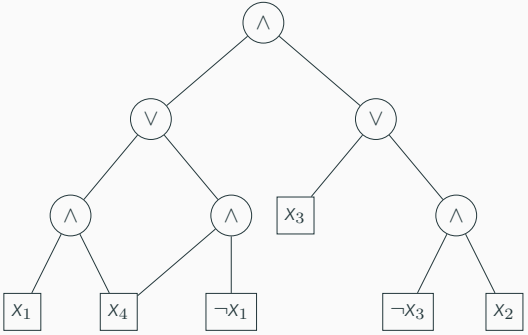
[HIIM21]



Other classifiers

- Naive-Bayes Classifier (NBC)
- Random forests (RFs) & other tree ensembles (TEs), e.g. boosted trees (BTs)
- Decision graphs, decision diagrams, etc.
- Propositional languages

[HII+22]



Outline

Non-formal XAI approaches – among best known

- Model-agnostic (MA) explainers:
  - LIME & SHAP [RSG16, LL17]
    - Goal: learn a simple interpretable ML model, e.g. linear classifier, decision tree, etc.
    - Approach: train classifier, sample-based vs. game theory
  - Anchor: [RSG18]
    - Goal: learn features deemed more relevant for prediction
    - Anchor is sample-based
  - No formal guarantees of rigor in computed explanations
- Intrinsic interpretability (II) [Rud19, Mol20]
  - (Interpretable) model is the explanation
  - E.g., DTs, DLs, DSs, etc.

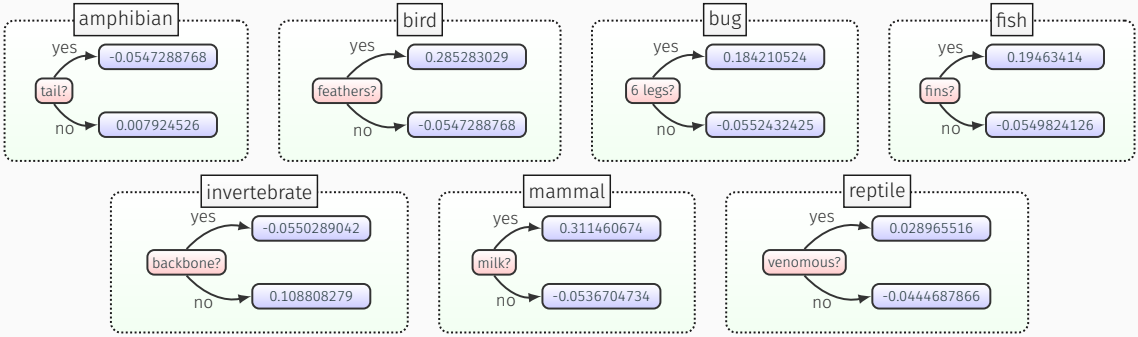
Non-formal XAI approaches – among best known

- Model-agnostic (MA) explainers:
  - LIME & SHAP [RSG16, LL17]
    - Goal: learn a simple interpretable ML model, e.g. linear classifier, decision tree, etc.
    - Approach: train classifier, sample-based vs. game theory
  - Anchor: [RSG18]
    - Goal: learn features deemed more relevant for prediction
    - Anchor is sample-based
  - No formal guarantees of rigor in computed explanations
- Intrinsic interpretability (II) [Rud19, Mol20]
  - (Interpretable) model is the explanation
  - E.g., DTs?, DLs?, DSs?, etc.?

Can MA explainers be trusted?
Is II indeed *interpretable*?

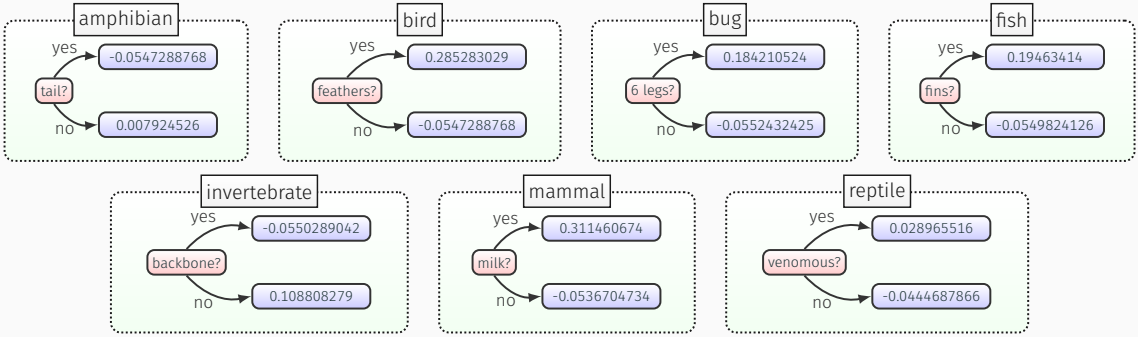
An example – BT for zoo dataset

[INM19c, Ign20]



An example – BT for zoo dataset

[INM19c, Ign20]



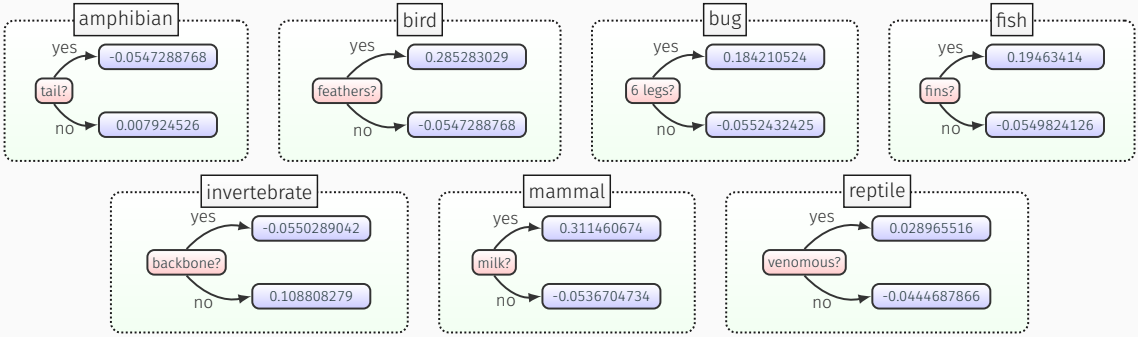
- Example instance:

IF (animal_name = pitviper) ∧ ¬hair ∧ ¬feathers ∧ eggs ∧ ¬milk ∧ ¬airborne ∧ ¬aquatic ∧ predator ∧ ¬toothed ∧ backbone ∧ breathes ∧ venomous ∧ ¬fins ∧ (legs = 0) ∧ tail ∧ ¬domestic ∧ ¬catsize

THEN (class = reptile)

An example – BT for zoo dataset & Anchor

[INM19c, Ign20]



- Example instance (& Anchor picks):

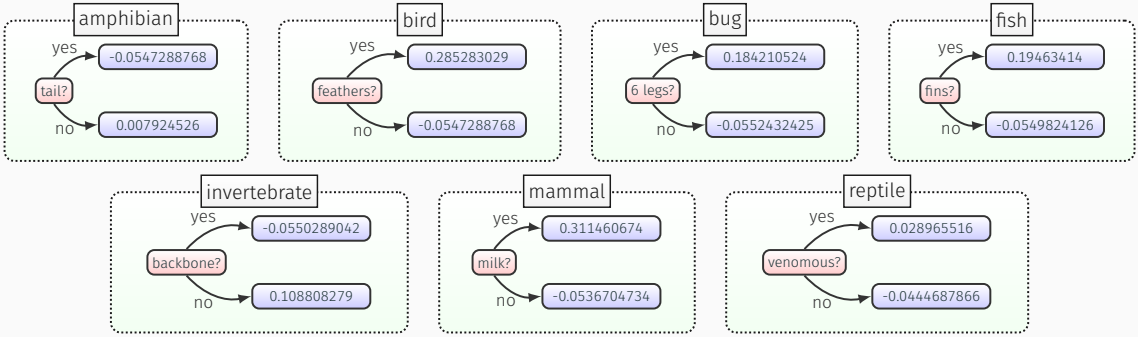
[RSG18]

IF (animal_name = pitviper) ∧ ¬***hair*** ∧ ¬feathers ∧ eggs ∧ ¬***milk*** ∧ ¬airborne ∧ ¬aquatic ∧ predator ∧ ¬***toothed*** ∧ backbone ∧ breathes ∧ venomous ∧ ¬***fins*** ∧ (legs = 0) ∧ tail ∧ ¬domestic ∧ ¬catsize

THEN (class = reptile)

An example – BT for zoo dataset & Anchor

[INM19c, Ign20]

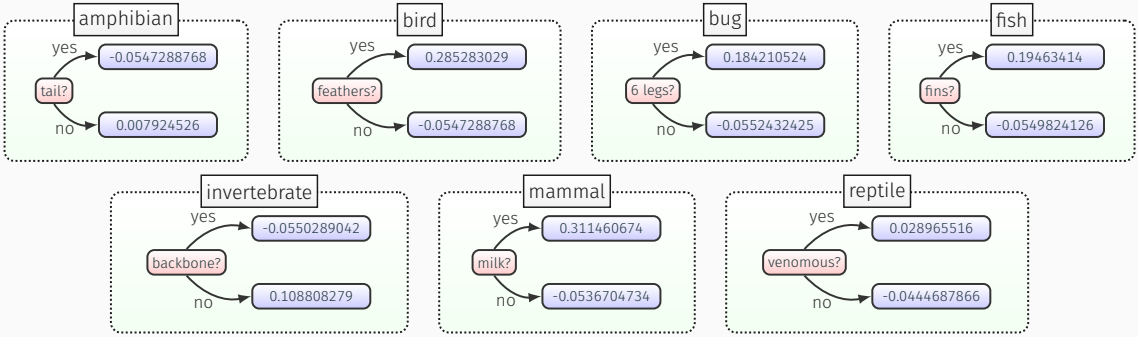


- Explanation obtained with Anchor: [RSG18]

IF	$\neg$*hair* $\wedge$ $\neg$*milk* $\wedge$ $\neg$*toothed* $\wedge$ $\neg$*fins*
THEN	(class = reptile)

An example – BT for zoo dataset & Anchor

[INM19c, Ign20]



- **But**, explanation **incorrectly "explains"** another instance (from training data!)

IF (animal_name = toad) ∧ ¬***hair*** ∧ ¬feathers ∧ eggs ∧ ¬***milk*** ∧
¬airborne ∧ ¬aquatic ∧ ¬predator ∧ ¬***toothed*** ∧ backbone ∧ breathes ∧
¬venomous ∧ ¬***fins*** ∧ (legs = 4) ∧ ¬tail ∧ ¬domestic ∧ ¬catsize

THEN (class = amphibian)

Model-agnostic explainers cannot be trusted

[INM19c]

Incorrect explanations:

Classifier for deciding bank loans

Model-agnostic explainers cannot be trusted

[INM19c]

Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie :– (v_1, Y) and Clive :– (v_2, N)

Model-agnostic explainers cannot be trusted

[INM19c]

Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie $:-(v_1, \mathsf{Y})$ and Clive $:-(v_2, \mathsf{N})$

Explanation X: age $= 45$, salary $= 50\text{K}$

Model-agnostic explainers cannot be trusted

[INM19c]

Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie $:-(v_1, \mathsf{Y})$ and Clive $:-(v_2, \mathsf{N})$

Explanation X: age $= 45$, salary $= 50\text{K}$

And,

- X is consistent with Bessie $:-(\mathbf{v}_1, \mathsf{Y})$
- X is consistent with Clive $:-(\mathbf{v}_2, \mathsf{N})$

Model-agnostic explainers cannot be trusted

[INM19c]

Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie $:-(v_1, \mathsf{Y})$ and Clive $:-(v_2, \mathsf{N})$

Explanation X: age $= 45$, salary $= 50\text{K}$

And,

X is consistent with Bessie $:-(\mathbf{v}_1, \mathsf{Y})$

X is consistent with Clive $:-(\mathbf{v}_2, \mathsf{N})$

$\therefore$ different outcomes & same explanation !?

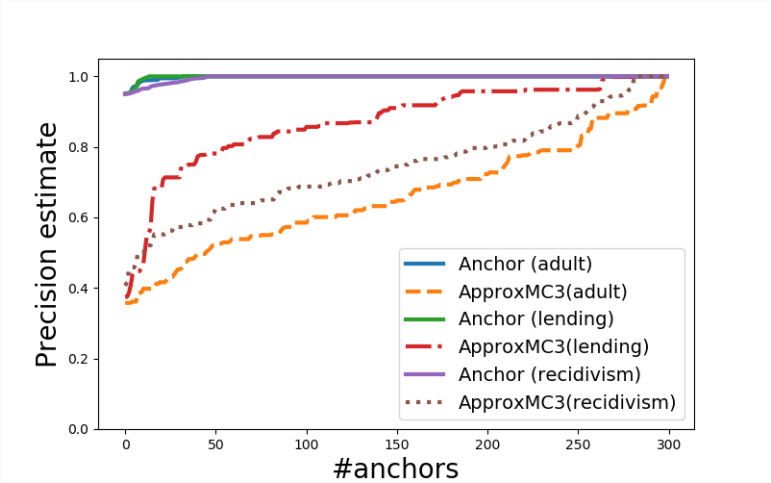
Incorrect explanations are ubiquitous...

[INM19c, Ign20]

Dataset	(# unique)	Explanations								
		incorrect			redundant			correct		
		LIME	Anchor	SHAP	LIME	Anchor	SHAP	LIME	Anchor	SHAP
adult	(5579)	61.3%	80.5%	70.7%	7.9%	1.6%	10.2%	30.8%	17.9%	19.1 %
lending	(4414)	24.0%	3.0%	17.0%	0.4%	0.0%	2.5%	75.6%	97.0%	80.5%
rcdv	(3696)	94.1%	99.4%	85.9%	4.6%	0.4%	7.9%	1.3%	0.2%	6.2%
compas	(778)	71.9%	84.4%	60.4%	20.6%	1.7%	27.8%	7.5%	13.9%	11.8%
german	(1000)	85.3%	99.7%	63.0%	14.6%	0.2%	37.0%	0.1 %	0.1 %	0.0%

Incorrect explanations are ubiquitous & likely...

[NSM+19]



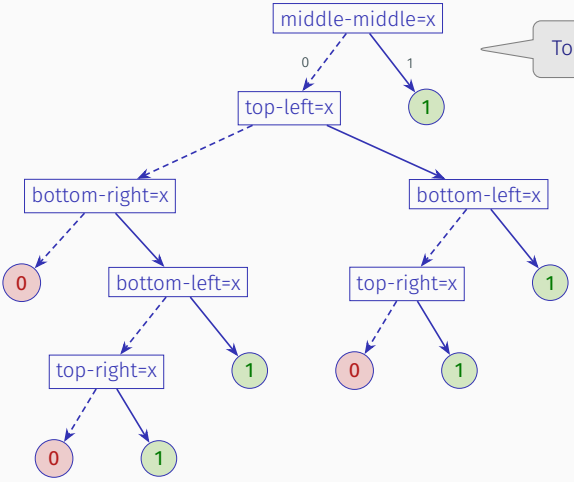
Decision trees are interpretable...

[Bre01, Rud19, Mol20]

"On interpretability, trees rate an A+." [Bre01]

General agreement on interpretability of decision trees [Rud19, Mol20, ANS20, Int21]

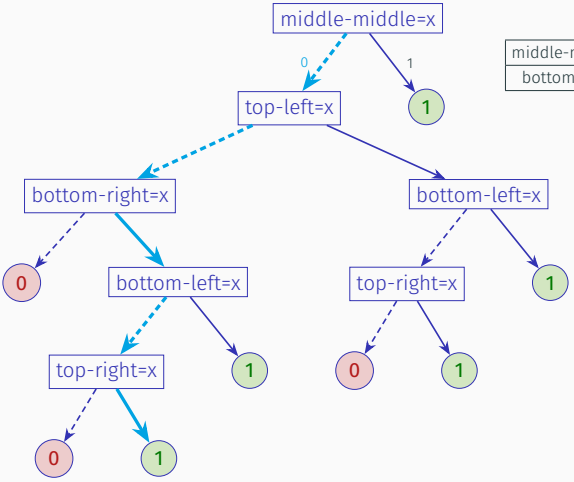
Decision tree explanations can be redundant



Tool: OSDT; train accuracy: 82.881%

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer: **Optimal Sparse Decision Trees**. **NeurIPS 2019**: 7265-7273

Decision tree explanations can be redundant



middle-middle $=$ x iff MM $=1$	top-left $=$ x iff TL $=1$	
bottom-right $=$ x iff BR $=1$	bottom-left $=$ x iff BL $=1$	top-right $=$ x iff TR $=1$

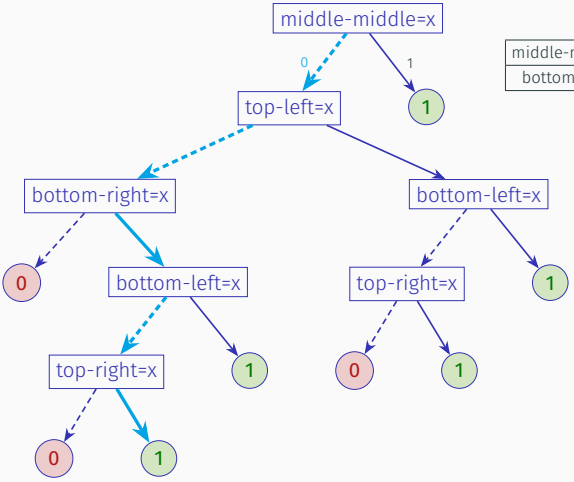
Q: What is explanation for

$(\text{MM} = 0) \wedge (\text{TL} = 0) \wedge (\text{BR} = 1) \wedge$

$(\text{BL} = 0) \wedge (\text{TR} = 1)$?

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer: **Optimal Sparse Decision Trees**. **NeurIPS 2019**: 7265-7273

Decision tree explanations can be redundant



middle-middle $=$ x iff MM $=1$	top-left $=$ x iff TL $=1$	
bottom-right $=$ x iff BR $=1$	bottom-left $=$ x iff BL $=1$	top-right $=$ x iff TR $=1$

Q: What is explanation for

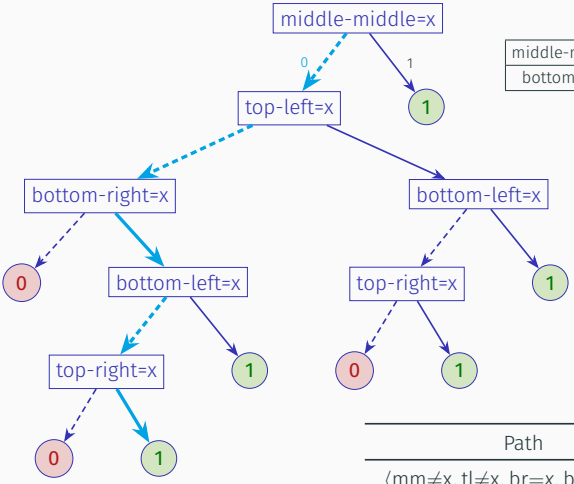
$(\text{MM}=0) \wedge (\text{TL}=0) \wedge (\text{BR}=1) \wedge$

$(\text{BL}=0) \wedge (\text{TR}=1)$?

IF $(\neg\text{MM} \wedge \neg\text{TL} \wedge \text{BR} \wedge \neg\text{BL} \wedge \text{TR})$ THEN $(\kappa(\cdot)=\mathbf{1})$?

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer: **Optimal Sparse Decision Trees**. **NeurIPS 2019**: 7265-7273

Decision tree explanations can be redundant



middle-middle = x iff MM = 1	top-left = x iff TL = 1	
bottom-right = x iff BR = 1	bottom-left = x iff BL = 1	top-right = x iff TR = 1

Q: What is explanation for

$(\text{MM} = 0) \land (\text{TL} = 0) \land (\text{BR} = 1) \land$

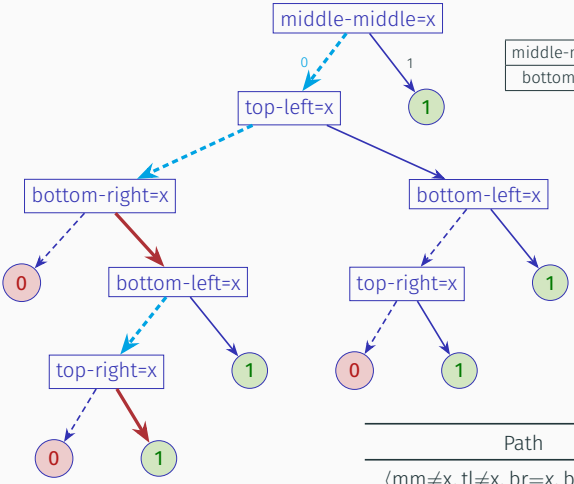
$(\text{BL} = 0) \land (\text{TR} = 1)$?

IF $(\neg\text{MM} \land \neg\text{TL} \land \text{BR} \land \neg\text{BL} \land \text{TR})$ THEN $(\kappa(\cdot) = \mathbf{1})$?

Path	Reduced XP	Dropped
⟨mm≠x, tl≠x, br=x, bl≠x, tr=x⟩		

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer: **Optimal Sparse Decision Trees**. **NeurIPS 2019**: 7265-7273

Decision tree explanations can be arbitrarily redundant



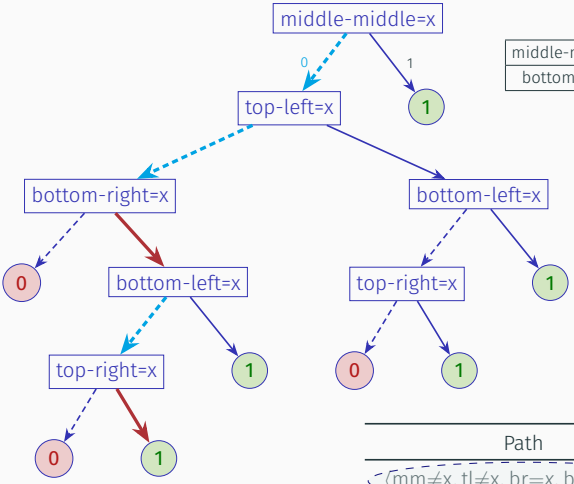
middle-middle = x iff MM = 1	top-left = x iff TL = 1	
bottom-right = x iff BR = 1	bottom-left = x iff BL = 1	top-right = x iff TR = 1

BR = 1, TR = 1			
MM	BL	TL	$\kappa(\cdot)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Path	Reduced XP	Dropped
$\langle$mm≠x, tl≠x, br=x, bl≠x, tr=x$\rangle$	$\langle$br=x, tr=x$\rangle$	{mm≠x, tl≠x, bl≠x}

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer: **Optimal Sparse Decision Trees**. **NeurIPS 2019**: 7265-7273

Decision tree explanations can be arbitrarily redundant



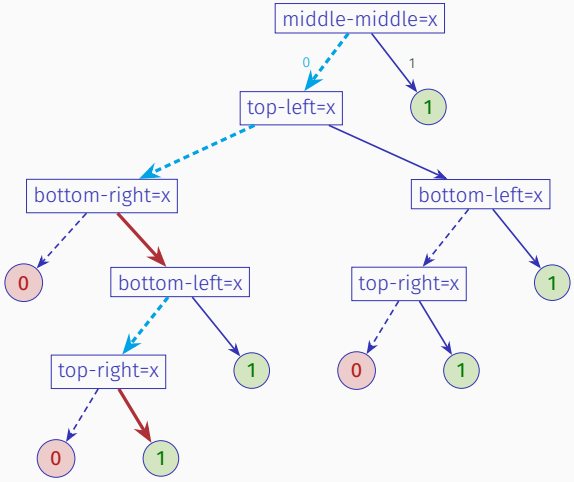
middle-middle = x iff MM = 1	top-left = x iff TL = 1	
bottom-right = x iff BR = 1	bottom-left = x iff BL = 1	top-right = x iff TR = 1

BR = 1, TR = 1			
MM	BL	TL	$\kappa(\cdot)$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Path	Reduced XP	Dropped
⟨mm≠x, tl≠x, br=x, bl≠x, tr=x⟩	⟨br=x, tr=x⟩	{mm≠x, tl≠x, bl≠x}

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer: Optimal Sparse Decision Trees. NeurIPS 2019: 7265-7273

Decision tree explanations can be arbitrarily redundant



# tree paths	8
# red. paths	5
% red. paths	62.5%

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer: Optimal Sparse Decision Trees. NeurIPS 2019: 7265-7273

Decision trees are interpretable...

[Bre01, Rud19, Mol20]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

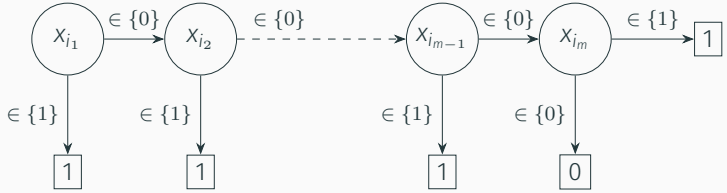
Decision trees are interpretable...

[Bre01, Rud19, Mol20]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Decision tree (DT), by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:



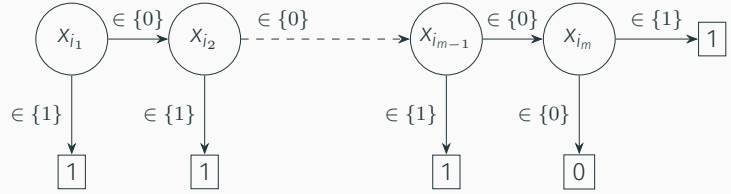
Decision trees are interpretable...

[Bre01, Rud19, Mol20]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Decision tree (DT), by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:



- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1

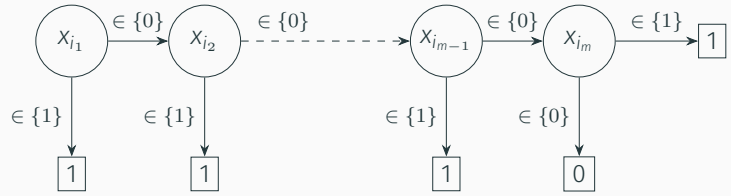
Decision trees are interpretable...

[Bre01, Rud19, Mol20]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Decision tree (DT), by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:



- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1
- Explanation using path in DT: $\{i_1, i_2, \ldots, i_m\}$, i.e.

$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \ldots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \ldots, x_m)$$

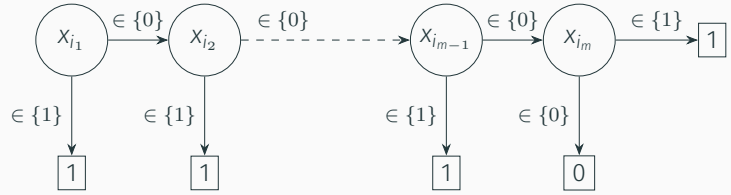
Decision trees are interpretable...

[Bre01, Rud19, Mol20]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Decision tree (DT), by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:



- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1
- Explanation using path in DT: $\{i_1, i_2, \ldots, i_m\}$, i.e.

$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \ldots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \ldots, x_m)$$

- **But $\{i_m\}$ suffices for prediction, i.e.** $\forall(\mathbf{x} \in \{0, 1\}^m).(x_{i_m}) \rightarrow \kappa(\mathbf{x})$

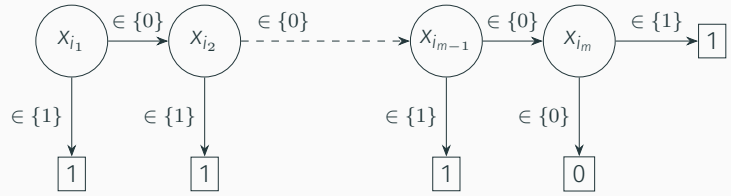
Minimal decision trees are not interpretable!

[IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Decision tree (DT), by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:



- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1
- Explanation using path in DT: $\{i_1, i_2, \ldots, i_m\}$, i.e.

$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \ldots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \ldots, x_m)$$

- **But $\{i_m\}$ suffices for prediction, i.e.** $\forall(\mathbf{x} \in \{0, 1\}^m).(x_{i_m}) \rightarrow \kappa(\mathbf{x})$

  True explanations arbitrarily smaller than paths in DT; $\therefore$ **DTs are not interpretable***

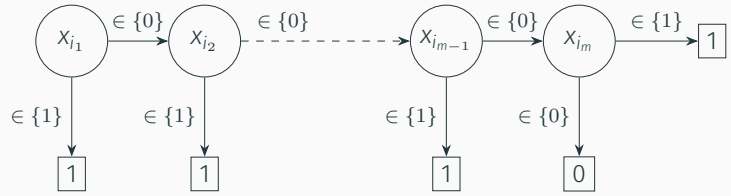
Minimal decision trees are not interpretable!

[IIM20, HIIM21, IIM22]

- Classifier, with $x_1, \ldots, x_m \in \{0, 1\}$:

$$\kappa(x_1, x_2, \ldots, x_{m-1}, x_m) = \bigvee_{i=1}^{m} x_i$$

- Decision tree (DT), by picking variables in order $\langle i_1, i_2, \ldots, i_m \rangle$, permutation of $\langle 1, 2, \ldots, m \rangle$:



- Point: $(x_{i_1}, x_{i_2}, \ldots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \ldots, 0, 1)$, and prediction 1
- Explanation using path in DT: $\{i_1, i_2, \ldots, i_m\}$, i.e.

$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \ldots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \ldots, x_m)$$

- **But $\{i_m\}$ suffices for prediction, i.e. $\forall(\mathbf{x} \in \{0, 1\}^m).(x_{i_m}) \rightarrow \kappa(\mathbf{x})$**

True explanations arbitrarily smaller than paths in DT; **$\therefore$ DTs are not interpretable***

*Pick any definition of interpretability that correlates with succinctness ...

[Lip18]

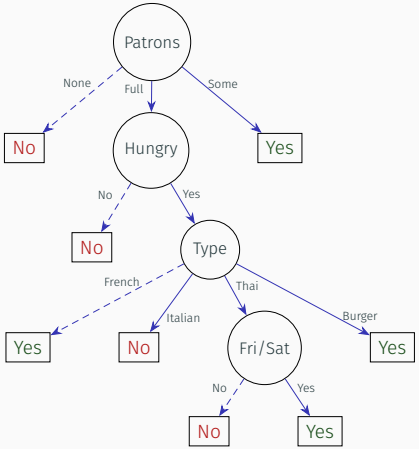
Explanation redundancy in DTs is ubiquitous – published DT examples

[IIM22]

DT Ref	D	#N	#P	%R	%C	%m	%M	%avg
[Alp14, Ch. 09, Fig. 9.1]	2	5	3	33	25	50	50	50
[Alp16, Ch. 03, Fig. 3.2]	2	5	3	33	25	50	50	50
[Bra20, Ch. 01, Fig. 1.3]	4	9	5	60	25	25	50	36
[BA97, Figure 1]	3	12	7	14	8	33	33	33
[BBHK10, Ch. 08, Fig. 8.2]	3	7	4	25	12	50	50	50
[BFOS84, Ch. 01, Fig. 1.1]	3	7	4	50	25	33	33	33
[DL01, Ch. 01, Fig. 1.2a]	2	5	3	33	25	33	33	33
[DL01, Ch. 01, Fig. 1.2b]	2	5	3	33	25	33	33	33
[KMND20, Ch. 04, Fig. 4.14]	3	7	4	25	12	50	50	50
[KMND20, Sec. 4.7, Ex. 4]	2	5	3	33	25	50	50	50
[Qui93, Ch. 01, Fig. 1.3]	3	12	7	28	17	33	50	41
[RM08, Ch. 01, Fig. 1.5]	3	9	5	20	12	33	33	33
[RM08, Ch. 01, Fig. 1.4]	3	7	4	50	25	33	33	33
[WFHP17, Ch. 01, Fig. 1.2]	3	7	4	25	12	50	50	50
[VLE$^+$16, Figure 4]	6	39	20	65	63	20	40	33
[Fla12, Ch. 02, Fig. 2.1(right)]	2	5	3	33	25	50	50	50
[Kot13, Figure 1]	3	10	6	33	11	33	33	33
[Mor82, Figure 1]	3	9	5	80	75	33	50	41
[PM17, Ch. 07, Fig. 7.4]	3	7	4	50	25	33	33	33
[RN10, Ch. 18, Fig. 18.6]	4	12	8	25	6	25	33	29
[SB14, Ch. 18, Page 212]	2	5	3	33	25	50	50	50
[Zho12, Ch. 01, Fig. 1.3]	2	5	3	33	25	33	33	33
[BHO09, Figure 1b]	4	13	7	71	50	33	50	36
[Zho21, Ch. 04, Fig. 4.3]	4	14	9	11	2	25	25	25

Many DTs have paths that are not minimal XPs – Russell&Norvig's book

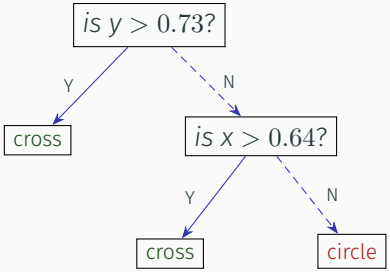
[RN10]



- Explanation for $(P, H, T, W) = (\text{Full}, \text{Yes}, \text{Thai}, \text{No})$?

Many DTs have paths that are not minimal XPs – Zhou's book

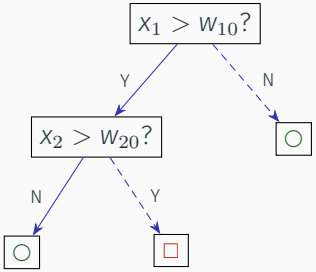
[Zho12]



- Explanation for $(x, y) = (1.25, -1.13)$?

Obs: True explanations can be computed for categorical, integer or real-valued features !

Many DTs have paths that are not minimal XPs – Alpaydin's book



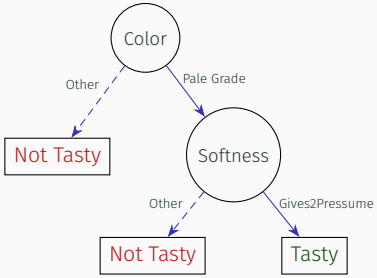
[Alp14]

- Explanation for $(x_1, x_2) = (\alpha, \beta)$, with $\alpha > w_{10}$ and $\beta \leq w_{20}$?

Obs: True explanations can be computed for categorical, integer or real-valued features !

Many DTs have paths that are not minimal XPs – S.-S.&B.-D.'s book

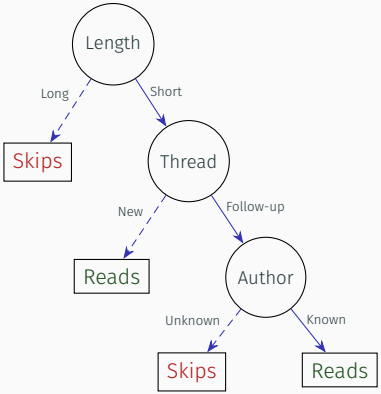
[SB14]



- Explanation for $(\text{color}, \text{softness}) = (\text{Pale Grade}, \text{Other})$?

Many DTs have paths that are not minimal XPs – Poole&Mackworth's book

[PM17]



- Explanation for $(L, T, A) = (\text{Short}, \text{Follow-Up}, \text{Unknown})$?
- Explanation for $(L, T, A) = (\text{Short}, \text{Follow-Up}, \text{Known})$?

Explanation redundancy in DTs is ubiquitous – DTs from datasets

[IIM20, HIIM21, IIM22]

Dataset	(#F	#S)	IAI									ITI								
			D	#N	%A	#P	%R	%C	%m	%M	%avg	D	#N	%A	#P	%R	%C	%m	%M	%avg
adult	(12	6061)	6	83	78	42	33	25	20	40	25	17	509	73	255	75	91	10	66	22
anneal	(38	886)	6	29	99	15	26	16	16	33	21	9	31	100	16	25	4	12	20	16
backache	(32	180)	4	17	72	9	33	39	25	33	30	3	9	91	5	80	87	50	66	54
bank	(19	36 293)	6	113	88	57	5	12	16	20	18	19	1467	86	734	69	64	7	63	27
biodegradation	(41	1052)	5	19	65	10	30	1	25	50	33	8	71	76	36	50	8	14	40	21
cancer	(9	449)	6	37	87	19	36	9	20	25	21	5	21	84	11	54	10	25	50	37
car	(6	1728)	6	43	96	22	86	89	20	80	45	11	57	98	29	65	41	16	50	30
colic	(22	357)	6	55	81	28	46	6	16	33	20	4	17	80	9	33	27	25	25	25
compas	(11	1155)	6	77	34	39	17	8	16	20	17	15	183	37	92	66	43	12	60	27
contraceptive	(9	1425)	6	99	49	50	8	2	20	60	37	17	385	48	193	27	32	12	66	21
dermatology	(34	366)	6	33	90	17	23	3	16	33	21	7	17	95	9	22	0	14	20	17
divorce	(54	150)	5	15	90	8	50	19	20	33	24	2	5	96	3	33	16	50	50	50
german	(21	1000)	6	25	61	13	38	10	20	40	29	10	99	72	50	46	13	12	40	22
heart-c	(13	302)	6	43	65	22	36	18	20	33	22	4	15	75	8	87	81	25	50	34
heart-h	(13	293)	6	37	59	19	31	4	20	40	24	8	25	77	13	61	60	20	50	32
kr-vs-kp	(36	3196)	6	49	96	25	80	75	16	60	33	13	67	99	34	79	43	7	70	35
lending	(9	5082)	6	45	73	23	73	80	16	50	25	14	507	65	254	69	80	12	75	25
letter	(16	18 668)	6	127	58	64	1	0	20	20	20	46	4857	68	2429	6	7	6	25	9
lymphography	(18	148)	6	61	76	31	35	25	16	33	21	6	21	86	11	9	0	16	16	16
mortality	(118	13 442)	6	111	74	56	8	14	16	20	17	26	865	76	433	61	61	7	54	19
mushroom	(22	8124)	6	39	100	20	80	44	16	33	24	5	23	100	12	50	31	20	40	25
pendigits	(16	10 992)	6	121	88	61	0	0	—	—	—	38	937	85	469	25	86	6	25	11
promoters	(58	106)	1	3	90	2	0	0	—	—	—	3	9	81	5	20	14	33	33	33
recidivism	(15	3998)	6	105	61	53	28	22	16	33	18	15	611	51	306	53	38	9	44	16
seismic_bumps	(18	2578)	6	37	89	19	42	19	20	33	24	8	39	93	20	60	79	20	60	42
shuttle	(9	58 000)	6	63	99	32	28	7	20	33	23	23	159	99	80	33	9	14	50	30
soybean	(35	623)	6	63	88	32	9	5	25	25	25	16	71	89	36	22	1	9	12	10
spambase	(57	4210)	6	63	75	32	37	12	16	33	19	15	143	91	72	76	98	7	58	25
spect	(22	228)	6	45	82	23	60	51	20	50	35	6	15	86	8	87	98	50	83	65
splice	(2	3178)	3	7	50	4	0	0	—	—	—	88	177	55	89	0	0	—	—	—

Outline

Rules with ordinal features

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1, 2\}$ (integer)

  Rules:

  IF $2x_1 + x_2 > 0$ THEN predict ⊞

  IF $2x_1 - x_2 \leq 0$ THEN predict ⊟

Rules with ordinal features

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1, 2\}$ (integer)

  Rules:

  IF $2x_1 + x_2 > 0$ THEN predict ⊞

  IF $2x_1 - x_2 \leq 0$ THEN predict ⊟

- **Q:** Can the model predict both ⊞ and ⊟ for some instance?

Rules with ordinal features

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1, 2\}$ (integer)

  Rules:

  IF $2x_1 + x_2 > 0$ THEN predict ⊞

  IF $2x_1 - x_2 \leq 0$ THEN predict ⊟

- **Q:** Can the model predict both ⊞ and ⊟ for some instance?
  - Yes, of course: pick $x_1 = 0$ and $x_2 = 1$

Rules with ordinal features

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1, 2\}$ (integer)

  Rules:

  IF $2x_1 + x_2 > 0$ THEN predict ⊞

  IF $2x_1 - x_2 \leq 0$ THEN predict ⊟

- **Q:** Can the model predict both ⊞ and ⊟ for some instance?
  - Yes, of course: pick $x_1 = 0$ and $x_2 = 1$
  - A formalization:

$$y_p \leftrightarrow (2x_1 + x_2 > 0) \ \wedge \ y_n \leftrightarrow (2x_1 - x_2 \leq 0) \ \wedge \ (y_p) \ \wedge \ (y_n)$$

... and solve with SMT solver

$\therefore$ There exists a model iff there exists a point in feature space yielding both predictions

Decision sets

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1\}$ (boolean)

  Rules:

  IF $x_1 \wedge \neg x_2 \wedge x_3$ THEN predict ⊞

  IF $x_1 \wedge \neg x_3 \wedge x_4$ THEN predict ⊟

  IF $x_3 \wedge x_4$ THEN predict ⊟

Decision sets

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1\}$ (boolean)

  Rules:

  IF $x_1 \wedge \neg x_2 \wedge x_3$ THEN predict ⊞

  IF $x_1 \wedge \neg x_3 \wedge x_4$ THEN predict ⊟

  IF $x_3 \wedge x_4$ THEN predict ⊟

- **Q:** Can the model predict both ⊞ and ⊟ for some instance?

Decision sets

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1\}$ (boolean)

  Rules:

  IF $x_1 \wedge \neg x_2 \wedge x_3$ THEN predict ⊞

  IF $x_1 \wedge \neg x_3 \wedge x_4$ THEN predict ⊟

  IF $x_3 \wedge x_4$ THEN predict ⊟

- **Q:** Can the model predict both ⊞ and ⊟ for some instance?
  - Yes, certainly: pick $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$

Decision sets

- Example ML model:

  Features: $x_1, x_2 \in \{0, 1\}$ (boolean)

  Rules:

  $$\begin{array}{llll} \text{IF} & x_1 \wedge \neg x_2 \wedge x_3 & \text{THEN} & \text{predict } \boxplus \\ \text{IF} & x_1 \wedge \neg x_3 \wedge x_4 & \text{THEN} & \text{predict } \boxminus \\ \text{IF} & x_3 \wedge x_4 & \text{THEN} & \text{predict } \boxminus \end{array}$$

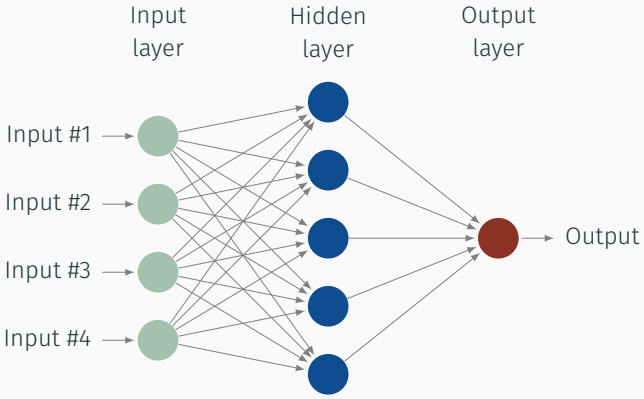
- **Q:** Can the model predict both $\boxplus$ and $\boxminus$ for some instance?
  - Yes, certainly: pick $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
  - A formalization:

$$\begin{aligned} & y_{p,1} \leftrightarrow (x_1 \wedge \neg x_2 \wedge x_3) \wedge \\ & y_{n,1} \leftrightarrow (x_1 \wedge \neg x_3 \wedge x_4) \wedge \\ & y_{n,2} \leftrightarrow (x_3 \wedge x_4) \ \wedge \ (y_p \leftrightarrow y_{p,1}) \wedge \\ & (y_n \leftrightarrow (y_{n,1} \vee y_{n,2})) \wedge (y_p) \ \wedge \ (y_n) \end{aligned}$$

... and solve with SAT solver (after clausification)

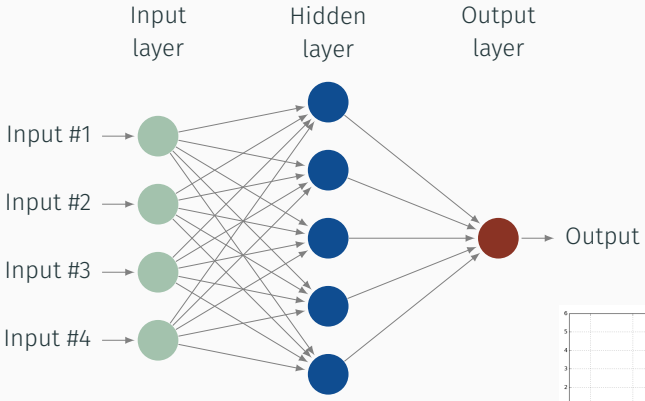
$\therefore$ There exists a model iff there exists a point in feature space yielding both predictions

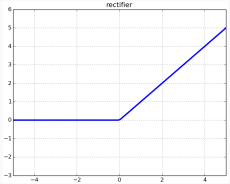
Neural networks



- Each layer (except first) viewed as a **block**, and
  - Compute $\mathbf{x}'$ given input $\mathbf{x}$, weights matrix $\mathbf{A}$, and bias vector $\mathbf{b}$
  - Compute output $\mathbf{y}$ given $\mathbf{x}'$ and activation function

Neural networks





- Each layer (except first) viewed as a **block**, and
  - Compute $\mathbf{x}'$ given input $\mathbf{x}$, weights matrix $\mathbf{A}$, and bias vector $\mathbf{b}$
  - Compute output $\mathbf{y}$ given $\mathbf{x}'$ and activation function
- Each unit uses a ReLU activation function

[NH10]

Encoding NNs using MILP

Computation for a NN ReLU **block**, in two steps:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$
$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

Encoding NNs using MILP

Computation for a NN ReLU **block**, in two steps:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$
$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

Encoding each **block**: [FJ18]

$$\sum_{j=1}^{n} a_{i,j} x_j + b_i = y_i - s_i$$
$$z_i = 1 \rightarrow y_i \leq 0$$
$$z_i = 0 \rightarrow s_i \leq 0$$
$$y_i \geq 0, s_i \geq 0, z_i \in \{0, 1\}$$

Simpler encodings exist, but **not** as effective [KBD+17]

Encoding NNs using MILP

Computation for a NN ReLU **block**, in two steps:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$
$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

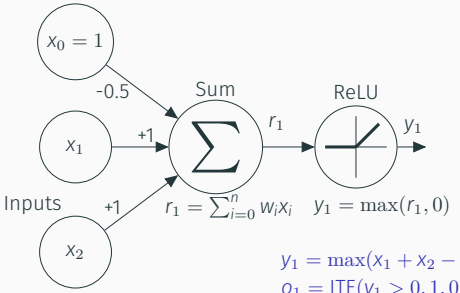
Modeling ML models with logic is not only possible but also simple !

Encoding each **block**: [FJ18]

$$\sum_{j=1}^{n} a_{i,j} x_j + b_i = y_i - s_i$$
$$z_i = 1 \rightarrow y_i \leq 0$$
$$z_i = 0 \rightarrow s_i \leq 0$$
$$y_i \geq 0, s_i \geq 0, z_i \in \{0, 1\}$$

Simpler encodings exist, but not as effective [KBD+17]

Encoding a simple NN in MILP



$y_1 = \max(x_1 + x_2 - 0.5, 0)$
$o_1 = \mathsf{ITE}(y_1 > 0, 1, 0)$

x_1	x_2	r_1	y_1	o_1
0	0	-0.5	0	0
1	0	0.5	0.5	1
0	1	0.5	0.5	1
1	1	1.5	1.5	1

MILP encoding:

$x_1 + x_2 - 0.5 = y_1 - s_1$
$z_1 = 1 \rightarrow y_1 \leq 0$
$z_1 = 0 \rightarrow s_1 \leq 0$
$o_1 = (y_1 > 0)$
$x_1, x_2, z_1, o_1 \in \{0, 1\}$
$y_1, s_1 \geq 0$

Instance: $(\mathbf{x}, c) = ((1, 0), 1)$

$1 + 0 - 0.5 = 0.5 - 0$
$1 \vee 0.5 \leq 0$
$0 \vee 0 \leq 0$
$1 = (0.5 > 0)$
$x_1 = 1, x_2 = 0, z_1 = 0, o_1 = 1$
$y_1 = 0.5, s_1 = 0$

Checking: $\mathbf{x} = (0, 0)$

$0 + 0 - 0.5 = 0 - 0.5$
$0 \vee 0 \leq 0$
$1 \vee 0.5 \leq 0$
$0 = (0 > 0)$
$x_1 = 0, x_2 = 0, z_1 = 1, o_1 = 0$
$y_1 = 0, s_1 = 0.5$

Part 2

Formal Explainability

Outline

Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$

Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\text{WeakAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

- Defining AXp:

$$\text{AXp}(\mathcal{X}) := \text{WeakAXp}(\mathcal{X}) \land \forall(\mathcal{X}' \subsetneq \mathcal{X}). \neg\text{WeakAXp}(\mathcal{X}')$$

Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\text{WeakAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

- Defining AXp:

$$\text{AXp}(\mathcal{X}) := \text{WeakAXp}(\mathcal{X}) \land \forall(\mathcal{X}' \subsetneq \mathcal{X}). \neg\text{WeakAXp}(\mathcal{X}')$$

- But, WeakAXp is **monotone**; hence,

$$\text{AXp}(\mathcal{X}) := \text{WeakAXp}(\mathcal{X}) \land \forall(t \in \mathcal{X}). \neg\text{WeakAXp}(\mathcal{X} \setminus \{t\})$$

Abductive explanations – answering Why? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- **Abductive explanation** (AXp, PI-explanation): [SCD18, INM19a]
  - **Subset-minimal** set of features $\mathcal{X} \subseteq \mathcal{F}$ sufficient for **ensuring** prediction

$$\text{WeakAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

- Defining AXp:

$$\text{AXp}(\mathcal{X}) := \text{WeakAXp}(\mathcal{X}) \land \forall(\mathcal{X}' \subsetneq \mathcal{X}). \neg\text{WeakAXp}(\mathcal{X}')$$

- But, WeakAXp is **monotone**; hence,

$$\text{AXp}(\mathcal{X}) := \text{WeakAXp}(\mathcal{X}) \land \forall(t \in \mathcal{X}). \neg\text{WeakAXp}(\mathcal{X} \setminus \{t\})$$

- Finding one AXp (example algorithm; many more exist):
  - Let $\mathcal{X} = \mathcal{F}$
  - Invariant: $\text{WeakAXp}(\mathcal{X})$ must hold. **Why?**
  - Analyze features in any order, one feature i at a time
    - If $\text{WeakAXp}(\mathcal{X} \setminus \{i\})$ holds, then remove i from $\mathcal{X}$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$?

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No
- AXp $\mathcal{X} = \{4\}$

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0, 1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No
- AXp $\mathcal{X} = \{4\}$

- In general, validity/consistency checked with SAT/SMT/MILP/CP reasoners

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$. AXp?
- Define $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).\neg x_2 \land \neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).\neg x_3 \land x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\forall(\mathbf{x} \in \{0,1\}^4).\top \rightarrow \kappa(x_1, x_2, x_3, x_4)$? No
- AXp $\mathcal{X} = \{4\}$

- In general, validity/consistency checked with SAT/SMT/MILP/CP reasoners
  - **Obs:** for some classes of classifiers, poly-time algorithms exist

Recap weak AXp: $\forall(\mathbf{x} \in \mathbb{F}).\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$

Outline

Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$

Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\exists (\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$$

Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\text{WeakCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$$

- Defining CXp:

$$\text{CXp}(\mathcal{Y}) := \text{WeakCXp}(\mathcal{Y}) \land \forall(\mathcal{Y}' \subsetneq \mathcal{Y}). \neg\text{WeakCXp}(\mathcal{Y}')$$

Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\mathsf{WeakCXp}(\mathcal{Y}) \quad \coloneqq \quad \exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$$

- Defining CXp:

$$\mathsf{CXp}(\mathcal{Y}) \coloneqq \mathsf{WeakCXp}(\mathcal{Y}) \land \forall(\mathcal{Y}' \subsetneq \mathcal{Y}).\neg\mathsf{WeakCXp}(\mathcal{Y}')$$

- But, WeakCXp is also monotone; hence,

$$\mathsf{CXp}(\mathcal{Y}) \coloneqq \mathsf{WeakCXp}(\mathcal{Y}) \land \forall(t \in \mathcal{Y}).\neg\mathsf{WeakCXp}(\mathcal{Y} \setminus \{t\})$$

Constrastive explanations – answering Why not? questions

- Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp): [Mil19, INAM20]
  - Subset-minimal set of features $\mathcal{Y} \subseteq \mathcal{F}$ sufficient for changing prediction

$$\text{WeakCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}).\bigwedge\nolimits_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$$

- Defining CXp:

$$\text{CXp}(\mathcal{Y}) := \text{WeakCXp}(\mathcal{Y}) \land \forall(\mathcal{Y}' \subsetneq \mathcal{Y}).\neg\text{WeakCXp}(\mathcal{Y}')$$

- But, WeakCXp is also monotone; hence,

$$\text{CXp}(\mathcal{Y}) := \text{WeakCXp}(\mathcal{Y}) \land \forall(t \in \mathcal{Y}).\neg\text{WeakCXp}(\mathcal{Y} \setminus \{t\})$$

- Finding one CXp:
  - Let $\mathcal{Y} = \mathcal{F}$
  - Invariant: $\text{WeakCXp}(\mathcal{Y})$ must hold. Why?
  - Analyze features in any order, one feature i at a time
    - If $\text{WeakCXp}(\mathcal{Y} \setminus \{i\})$ holds, then remove i from $\mathcal{Y}$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land x_4 \land \neg\kappa(x_1, x_2, x_3, x_4)$?

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land x_4 \land \neg\kappa(x_1, x_2, x_3, x_4)$? No

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^{4} x_i$$

- Point $\mathbf{v} = (0, 0, 0, 1)$ with prediction $\kappa(\mathbf{v}) = 1$
- Define $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 2 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 3 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land \neg\kappa(x_1, x_2, x_3, x_4)$? Yes
- Can feature 4 be removed, i.e. $\exists(\mathbf{x} \in \{0, 1\}^4).\neg x_1 \land \neg x_2 \land \neg x_3 \land x_4 \land \neg\kappa(x_1, x_2, x_3, x_4)$? No
- CXp $\mathcal{Y} = \{4\}$
- **Obs:** AXp is MHS of CXp and vice-versa...

Recap weak CXp: $\exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge_{j \notin \mathcal{Y}}(x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$

Outline

More on AXp's and CXp's

- Duality: [INAM20, INM19b]
  - Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
  - AXp's are MHSes of CXp's and vice-versa
    - Result builds on work of R. Reiter [Rei87]

More on AXp's and CXp's

- Duality: [INAM20, INM19b]
  - Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
  - AXp's are MHSes of CXp's and vice-versa
    - Result builds on work of R. Reiter [Rei87]

- AXp's/CXp's are model-based, i.e. explanations are sound given the model

More on AXp's and CXp's

- Duality: [INAM20, INM19b]
  - Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
  - AXp's are MHSes of CXp's and vice-versa
    - Result builds on work of R. Reiter [Rei87]

- AXp's/CXp's are model-based, i.e. explanations are sound given the model

- AXp's/CXp's are localized explanations (i.e. wrt $\mathbf{v}$) that hold globally

More on AXp's and CXp's

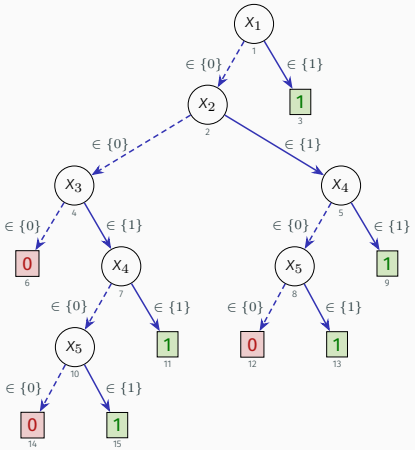
- Duality: [INAM20, INM19b]
  - Instance $(\mathbf{v}, c)$, i.e. $c = \kappa(\mathbf{v})$
  - AXp's are MHSes of CXp's and vice-versa
    - Result builds on work of R. Reiter [Rei87]

- AXp's/CXp's are model-based, i.e. explanations are sound given the model

- AXp's/CXp's are localized explanations (i.e. wrt $\mathbf{v}$) that hold globally

- Global AXp's defined wrt class [INM19b]
  - Counterexamples (CEx's): break (i.e. inconsistent with) prediction of class
  - MHS duality between global AXp's and CEx's
    - Also, connection with adversarial examples

Case study: DT running example



- Instance: $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	1

- AXp's: $\{3, 5\}$
  - Other features irrelevant for ensuring the prediction
- CXp's:
  - $\{3\}$, by flipping the value of feature 3
  - $\{5\}$, by flipping the value of feature 5
  - But also, $\{\{3\}, \{5\}\}$ by MHS duality

Case study: DL running example

R_1:	IF	$(x_1 = 1)$	THEN	0
R_2:	ELSE IF	$(x_2 = 1)$	THEN	1
R_3:	ELSE IF	$(x_4 = 1)$	THEN	0
R_{DEF}:	ELSE		THEN	1

Entry	x_1	x_2	x_3	x_4	Rule	$\kappa_1(\mathbf{x})$
00	0	0	0	0	R_{DEF}	1
01	0	0	0	1	R_3	0
02	0	0	0	2	R_{DEF}	1
03	0	0	1	0	R_{DEF}	1
04	0	0	1	1	R_3	0
05	0	0	1	2	R_{DEF}	1
06	0	1	0	0	R_2	1
07	0	1	0	1	R_2	1
08	0	1	0	2	R_2	1
09	0	1	1	0	R_2	1
10	0	1	1	1	R_2	1
11	0	1	1	2	R_2	1
12	1	0	0	0	R_1	0
13	1	0	0	1	R_1	0
14	1	0	0	2	R_1	0
15	1	0	1	0	R_1	0
16	1	0	1	1	R_1	0
17	1	0	1	2	R_1	0
18	1	1	0	0	R_1	0
19	1	1	0	1	R_1	0
20	1	1	0	2	R_1	0
21	1	1	1	0	R_1	0
22	1	1	1	1	R_1	0
23	1	1	1	2	R_1	0

Case study: DL running example

R_1: IF $(x_1 = 1)$ THEN 0

R_2: ELSE IF $(x_2 = 1)$ THEN 1

R_3: ELSE IF $(x_4 = 1)$ THEN 0

R_{DEF}: ELSE THEN 1

- Instance: $(\mathbf{v}, c) = ((0, 0, 1, 2), 1)$
- AXp's: $\{1, 4\}$ (prediction unchanged)
- CXp's:
  - $\{1\}$, by flipping the value of feature 1
  - $\{4\}$, by flipping the value of feature 4
  - But also, $\{\{1\}, \{4\}\}$ by MHS duality

Entry	x_1	x_2	x_3	x_4	Rule	$\kappa_1(\mathbf{x})$
00	0	0	0	0	R_{DEF}	1
01	0	0	0	1	R_3	0
02	0	0	0	2	R_{DEF}	1
03	0	0	1	0	R_{DEF}	1
04	0	0	1	1	R_3	0
05	0	0	1	2	R_{DEF}	1
06	0	1	0	0	R_2	1
07	0	1	0	1	R_2	1
08	0	1	0	2	R_2	1
09	0	1	1	0	R_2	1
10	0	1	1	1	R_2	1
11	0	1	1	2	R_2	1
12	1	0	0	0	R_1	0
13	1	0	0	1	R_1	0
14	1	0	0	2	R_1	0
15	1	0	1	0	R_1	0
16	1	0	1	1	R_1	0
17	1	0	1	2	R_1	0
18	1	1	0	0	R_1	0
19	1	1	0	1	R_1	0
20	1	1	0	2	R_1	0
21	1	1	1	0	R_1	0
22	1	1	1	1	R_1	0
23	1	1	1	2	R_1	0

Part 3

Computing Explanations – Basics

Outline

How to compute XPs in general?

- Recall:
  - For (weak) AXp:

$$\mathsf{WeakAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

  - For (weak) CXp:

$$\mathsf{WeakCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}).\ \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$$

How to compute XPs in general?

- Recall:
  - For (weak) AXp:

$$\text{WeakAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

  - For (weak) CXp:

$$\text{WeakCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$$

- Encode predicates for computing AXp/CXp
  - For AXp (double negate logic statement):

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg \text{co}\left(\left[\!\left[\left(\bigwedge\nolimits_{i \in \mathcal{S}} (x_i = v_i)\right) \land (\kappa(\mathbf{x}) \neq c)\right]\!\right]\right)$$

  - For CXp (use logic statement):

$$\mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \text{co}\left(\left[\!\left[\left(\bigwedge\nolimits_{i \in \mathcal{F} \setminus \mathcal{S}} (x_i = v_i)\right) \land (\kappa(\mathbf{x}) \neq c)\right]\!\right]\right)$$

How to compute XPs in general?

- Recall:
  - For (weak) AXp:

$$\text{WeakAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

  - For (weak) CXp:

$$\text{WeakCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}). \bigwedge\nolimits_{j \notin \mathcal{Y}} (x_j = v_j) \land (\kappa(\mathbf{x}) \neq c)$$

- Encode predicates for computing AXp/CXp
  - For AXp (double negate logic statement):

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg \text{co}\left(\left[\!\left[\left(\bigwedge\nolimits_{i \in \mathcal{S}} (x_i = v_i)\right) \land (\kappa(\mathbf{x}) \neq c)\right]\!\right]\right)$$

  - For CXp (use logic statement):

$$\mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \text{co}\left(\left[\!\left[\left(\bigwedge\nolimits_{i \in \mathcal{F} \setminus \mathcal{S}} (x_i = v_i)\right) \land (\kappa(\mathbf{x}) \neq c)\right]\!\right]\right)$$

- And, find subset or cardinality minimal set

Computing one AXp/CXp

[MM20]

Input: Predicate $\mathbb{P}$, parameterized by $\mathcal{T}$, $\mathcal{F}$, κ, $\mathbf{v}$
Output: One XP $\mathcal{S}$

1: **procedure** oneXP($\mathbb{P}$)
2: $\quad\mathcal{S} \leftarrow \mathcal{F}$ ▷ Initialization: $\mathbb{P}(\mathcal{S})$ holds
3: $\quad$**for** $i \in \mathcal{F}$ **do** ▷ Loop invariant: $\mathbb{P}(\mathcal{S})$ holds
4: $\quad\quad$**if** $\mathbb{P}(\mathcal{S} \setminus \{i\})$ **then**
5: $\quad\quad\quad\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ ▷ Update $\mathcal{S}$ only if $\mathbb{P}(\mathcal{S} \setminus \{i\})$ holds
6: $\quad$**return** $\mathcal{S}$ ▷ Returned set $\mathcal{S}$: $\mathbb{P}(\mathcal{S})$ holds

Computing smallest AXp/CXp – with hitting set dualization

[MM20]

Input: Predicate $\mathbb{P}_{\text{axp}}$, parameterized by $\mathcal{T}$, $\mathcal{F}$, κ, $\mathbf{v}$
Output: Smallest XP $\mathcal{M}$

1: **procedure** minXP($\mathbb{P}$)
2: $\quad\mathcal{H} \leftarrow \emptyset$
3: $\quad$**while true do**
4: $\quad\quad\mathcal{M} \leftarrow$ MinimumHS($\mathcal{H}$)
5: $\quad\quad$**if** $\mathbb{P}_{\text{axp}}(\mathcal{M})$ **then**
6: $\quad\quad\quad$**return** $\mathcal{M}$
7: $\quad\quad$**else**
8: $\quad\quad\quad\mu \leftarrow$ GetAssignment()
9: $\quad\quad\quad\mathcal{L} \leftarrow$ PickFalseLits($\mathcal{H} \setminus \mathcal{M}, \mu$)
10: $\quad\quad\quad\mathcal{H} \leftarrow \mathcal{H} \cup \mathcal{L}$

Computing smallest AXp/CXp – with hitting set dualization

[MM20]

Input: Predicate $\mathbb{P}_{\text{axp}}$, parameterized by $\mathcal{T}$, $\mathcal{F}$, κ, $\mathbf{v}$
Output: Smallest XP $\mathcal{M}$

```
1:  procedure minXP(ℙ)
2:      ℋ ← ∅
3:      while true do
4:          ℳ ← MinimumHS(ℋ)
5:          if ℙ_axp(ℳ) then
6:              return ℳ
7:          else
8:              μ ← GetAssignment()
9:              ℒ ← PickFalseLits(ℋ \ ℳ, μ)
10:             ℋ ← ℋ ∪ ℒ
```

- Obs: computing smallest CXp akin to solving MaxSAT [INM19a]

Outline

Initial results for NNs (with SMT/MILP)

[INM19a]

Dataset			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	m	1	0.03	0.05	–	–	–
		a	8.79	1.38	0.33	–	–	–
		M	14	17.00	1.43	–	–	–
backache	(32)	m	13	0.13	0.14	–	–	–
		a	19.28	5.08	0.85	–	–	–
		M	26	22.21	2.75	–	–	–
breast-cancer	(9)	m	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	m	4	0.05	0.07	4	–	0.07
		a	8.62	3.32	0.32	7.89	–	5.14
		M	13	60.74	0.60	13	–	39.06
hepatitis	(19)	m	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	m	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	m	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Initial results for NNs (with SMT/MILP)

[INM19a]

First rigorous approach for **explaining** NNs !

			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	m	1	0.03	0.05	–	–	–
		a	8.79	1.38	0.33	–	–	–
		M	14	17.00	1.43	–	–	–
backache	(32)	m	13	0.13	0.14	–	–	–
		a	19.28	5.08	0.85	–	–	–
		M	26	22.21	2.75	–	–	–
breast-cancer	(9)	m	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	m	4	0.05	0.07	4	–	0.07
		a	8.62	3.32	0.32	7.89	–	5.14
		M	13	60.74	0.60	13	–	39.06
hepatitis	(19)	m	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	m	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	m	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Initial results for NNs (with SMT/MILP)

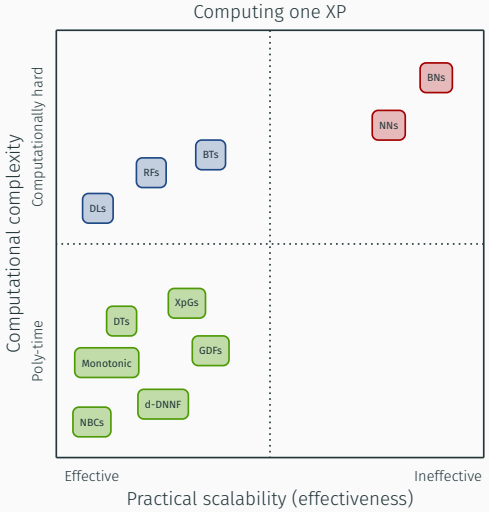
[INM19a]

First rigorous approach for **explaining** NNs !

			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	m	1	0.03	0.05	–	–	–
		a	8.79	1.38	0.33	–	–	–
		M	14	17.00	1.43	–	–	–
backache	(32)	m	13	0.13	0.14	–	–	–
		a	19.28	5.08	0.85	–	–	–
		M	26	22.21	2.75	–	–	–
breast-cancer	(9)	m	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	m	4	0.05	0.07	4	–	0.07
		a	8.62	3.32	0.32	7.89	–	5.14
		M	13	60.74	0.60	13	–	39.06
hepatitis	(19)	m	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	m	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	m	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

Scales to (a few) tens of neurons...

Efficacy map – current status



[INM19c, Ign20, IIM20, MGC+20, MGC+21, HIIM21, IMS21, IM21, CM21, HII+22, IISMS22]

- **Formal explanations efficient for several families of classifiers**
  - Polynomial-time:
    - Naive-Bayes classifiers (NBCs) [MGC+20]
    - Decision trees (DTs) [IIM20, HIIM21]
    - XpG's: DTs, OBDDs, OMDDs, etc. [HIIM21]
    - Monotonic classifiers [MGC+21]
    - Propositional languages (e.g. d-DNNF, ...) [HII+22]
    - Additional results [CM21, HII+22]
  - Comp. hard, but effective (efficient in practice):
    - Random forests (RFs) [IMS21]
    - Decision lists (DLs) [IM21]
    - Boosted trees (BTs) [INM19c, Ign20, IISMS22]
  - Comp. hard, and ineffective (hard in practice):
    - Neural networks (NNs) [INM19a]
    - Bayesian networks (BNs) [SCD18]

Results for RFs (using SAT oracles)

[IMS21]

Dataset	(#F	#C	#I)	RF			CNF		SAT oracle				PI-expl (**RFxp1**)				Anchor	
				D	#N	%A	#var	#cl	MxS	MxU	#S	#U	Mx	m	avg	%w	avg	%w
ann-thyroid	(21	3	718)	4	2192	98	17 854	29 230	0.12	0.15	2	18	0.36	0.05	0.13	96	0.32	4
appendicitis	(7	2	43)	6	1920	90	5181	10 085	0.02	0.02	4	3	0.05	0.01	0.03	100	0.48	0
banknote	(4	2	138)	5	2772	97	8068	16 776	0.01	0.01	2	2	0.03	0.02	0.02	100	0.19	0
biodegradation	(41	2	106)	5	4420	88	11 007	23 842	0.31	1.05	17	22	2.27	0.04	0.29	97	4.07	3
heart-c	(13	2	61)	5	3910	85	5594	11 963	0.04	0.02	6	7	0.07	0.01	0.04	100	0.85	0
ionosphere	(34	2	71)	5	2096	87	7174	14 406	0.02	0.02	22	11	0.11	0.02	0.03	100	12.43	0
karhunen	(64	10	200)	5	6198	91	36 708	70 224	1.06	1.41	35	29	14.64	0.65	2.78	100	28.15	0
letter	(16	26	398)	8	44 304	82	28 991	68 148	1.97	3.31	8	8	6.91	0.24	1.61	70	2.48	30
magic	(10	2	381)	6	9840	84	29 530	66 776	0.51	1.84	6	4	2.13	0.07	0.14	99	0.91	1
new-thyroid	(5	3	43)	5	1766	100	17 443	28 134	0.03	0.01	3	2	0.08	0.03	0.05	100	0.36	0
pendigits	(16	10	220)	6	12 004	95	30 522	59 922	2.40	1.32	10	6	4.11	0.14	0.94	96	3.68	4
ring	(20	2	740)	6	6188	89	19 114	42 362	0.27	0.44	11	9	1.25	0.05	0.25	92	7.25	8
segmentation	(19	7	42)	4	1966	90	21 288	35 381	0.11	0.17	8	10	0.53	0.11	0.31	100	4.13	0
shuttle	(9	7	1160)	3	1460	99	18 669	29 478	0.11	0.08	2	7	0.34	0.05	0.14	99	0.42	1
sonar	(60	2	42)	5	2614	88	9938	20 537	0.04	0.06	36	24	0.43	0.04	0.09	100	23.02	0
spectf	(44	2	54)	5	2306	88	6707	13 449	0.07	0.06	20	24	0.34	0.02	0.07	100	8.12	0
texture	(40	11	550)	5	5724	87	34 293	64 187	0.79	0.63	23	17	3.24	0.19	0.93	100	28.13	0
twonorm	(20	2	740)	5	6266	94	21 198	46 901	0.08	0.08	12	8	0.28	0.06	0.10	100	5.73	0
vowel	(13	11	198)	6	10 176	90	44 523	88 696	1.66	2.11	8	5	4.52	0.15	1.15	66	1.67	34
waveform-40	(40	3	500)	5	6232	83	30 438	58 380	0.50	0.86	15	25	7.07	0.11	0.88	100	11.93	0
wpbc	(33	2	78)	5	2432	76	9078	18 675	1.00	1.53	20	13	5.33	0.03	0.65	79	3.91	21

Part 4

Computing Explanations – Encodings

Outline

Computing XPs of DLs

An encoding for DLs – components

$$
\begin{array}{lllll}
\mathsf{R}_1\text{:} & \text{IF} & (\tau_1) & \text{THEN} & d_1 \\
\mathsf{R}_2\text{:} & \text{ELSE IF} & (\tau_2) & \text{THEN} & d_2 \\
 & & \cdots & & \\
\mathsf{R}_j\text{:} & \text{ELSE IF} & (\tau_j) & \text{THEN} & d_j \\
 & & \cdots & & \\
\mathsf{R}_n\text{:} & \text{ELSE IF} & (\tau_n) & \text{THEN} & d_n \\
\mathsf{R}_{\text{DEF}}\text{:} & \text{ELSE} & & \text{THEN} & d_{n+1}
\end{array}
$$

- Clauses for encoding ϕ: $\mathfrak{E}_\phi(z_1, \ldots)$, such that $z_1 = 1$ iff $\phi = 1$
- For τ_j: $\mathfrak{E}_{\tau_j}(t_j, \ldots)$
- For $x_i = v_i$: $\mathfrak{E}_{x_i = v_i}(l_i, \ldots)$
- Let $e_j = 1$ iff d_j matches c
- Prediction change with rule up to R_j (with $d_j \neq c$), if $\tau_j \nvDash \bot$ and $\tau_k \vDash \bot$, for $1 \leq k < j$, with $e_k = 1$:

$$\left[f_j \leftrightarrow \left(t_j \wedge \bigwedge\nolimits_{1 \leq k < j, e_k = 1} \neg t_k \right) \right]$$

An encoding for DLs – components

$$
\begin{array}{llllll}
\mathsf{R}_1\text{:} & \mathsf{IF} & (\tau_1) & \mathsf{THEN} & d_1 \\
\mathsf{R}_2\text{:} & \mathsf{ELSE\ IF} & (\tau_2) & \mathsf{THEN} & d_2 \\
 & & \cdots & & \\
\mathsf{R}_j\text{:} & \mathsf{ELSE\ IF} & (\tau_j) & \mathsf{THEN} & d_j \\
 & & \cdots & & \\
\mathsf{R}_n\text{:} & \mathsf{ELSE\ IF} & (\tau_n) & \mathsf{THEN} & d_n \\
\mathsf{R}_{\mathsf{DEF}}\text{:} & \mathsf{ELSE} & & \mathsf{THEN} & d_{n+1}
\end{array}
$$

- Clauses for encoding ϕ: $\mathfrak{E}_\phi(z_1, \ldots)$, such that $z_1 = 1$ iff $\phi = 1$
- For τ_j: $\mathfrak{E}_{\tau_j}(t_j, \ldots)$
- For $x_i = v_i$: $\mathfrak{E}_{x_i = v_i}(l_i, \ldots)$
- Let $e_j = 1$ iff d_j matches c
- Require that at least one f_j, with $e_j = 0$ and $1 \leq j \leq n$, to be consistent (i.e. some rule up to j with prediction other than c to fire):

$$\left(\bigvee\nolimits_{1 \leq j \leq n, e_j = 0} f_j \right)$$

An encoding for DLs – components

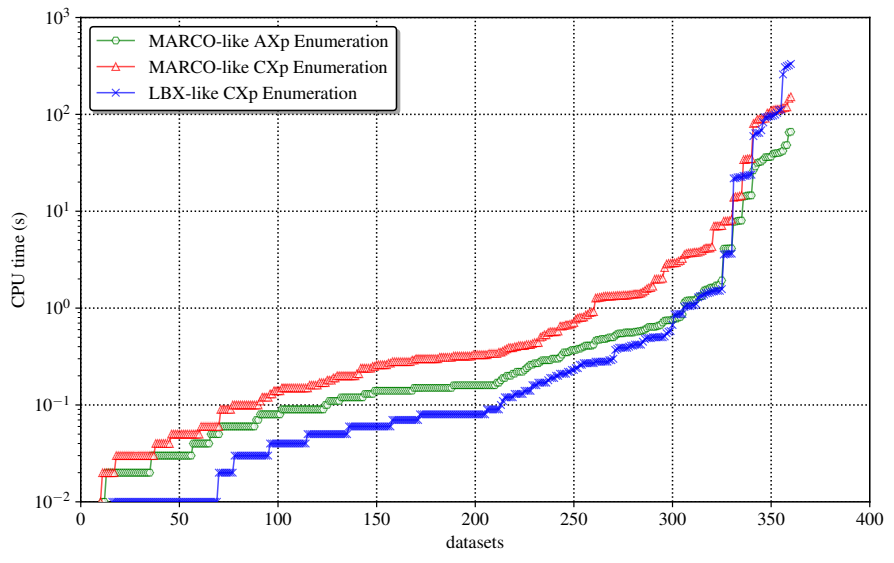
$$
\begin{array}{lllll}
\mathsf{R}_1\colon & \text{IF} & (\tau_1) & \text{THEN} & d_1 \\
\mathsf{R}_2\colon & \text{ELSE IF} & (\tau_2) & \text{THEN} & d_2 \\
 & & \cdots & & \\
\mathsf{R}_j\colon & \text{ELSE IF} & (\tau_j) & \text{THEN} & d_j \\
 & & \cdots & & \\
\mathsf{R}_n\colon & \text{ELSE IF} & (\tau_n) & \text{THEN} & d_n \\
\mathsf{R}_{\mathsf{DEF}}\colon & \text{ELSE} & & \text{THEN} & d_{n+1}
\end{array}
$$

- The set of soft clauses is given by: $\mathcal{S} \triangleq \{(l_i), i = 1, \ldots, m\}$
- The set of hard clauses is given by:

$$
\mathcal{B} \triangleq \bigwedge\nolimits_{1 \leq i \leq m} \mathfrak{E}_{x_i = v_i}(l_i, \ldots) \wedge \bigwedge\nolimits_{1 \leq j \leq n} \mathfrak{E}_{\tau_j}(t_j, \ldots) \wedge
$$
$$
\bigwedge\nolimits_{1 \leq j \leq n, e_j = 0} \left(f_j \leftrightarrow \left(t_j \wedge \bigwedge\nolimits_{1 \leq k < j, e_k = 1} \neg t_k \right) \right) \wedge \left(\bigvee\nolimits_{1 \leq j \leq n, e_j = 0} f_j \right)
$$

- $\mathcal{B} \cup \mathcal{S} \models \bot$
  - MUSes are AXp's & MCSes are CXp's

Empirical Evidence



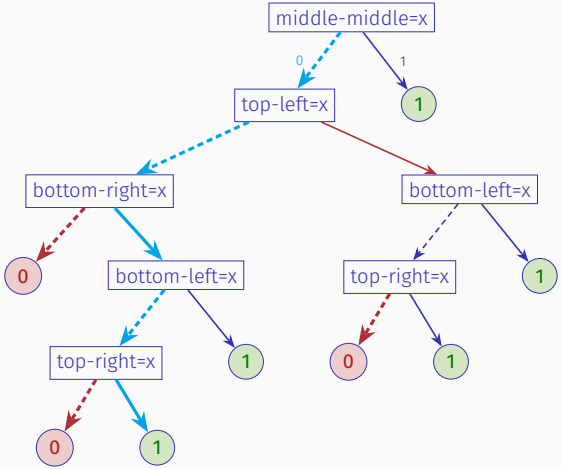
Part 5

Tractable Explanations

Outline

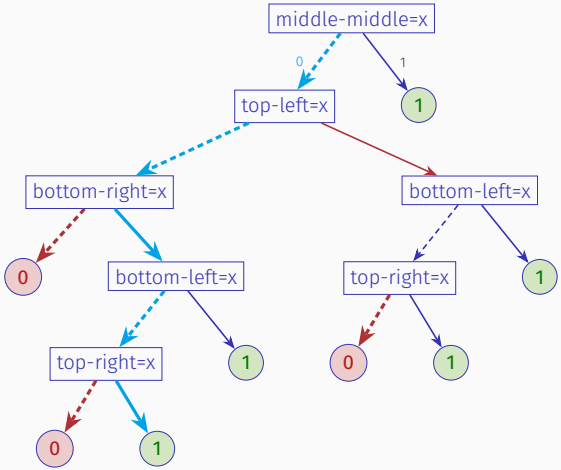
DT explanations

[IIM20, IIM22]



DT explanations

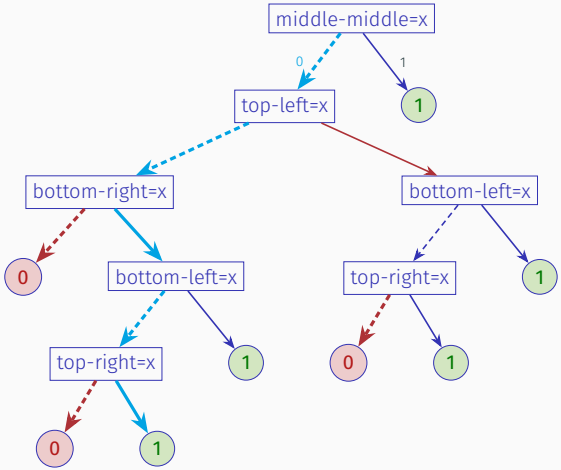
[IIM20, IIM22]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time

DT explanations

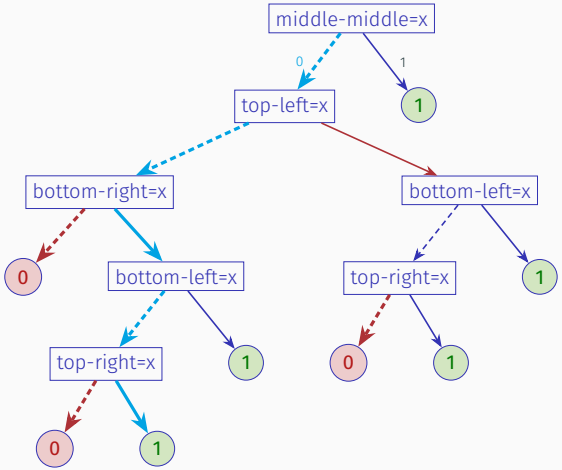
[IIM20, IIM22]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- For prediction 1, it suffices to ensure all paths with prediction 0 remain inconsistent

DT explanations in polynomial time

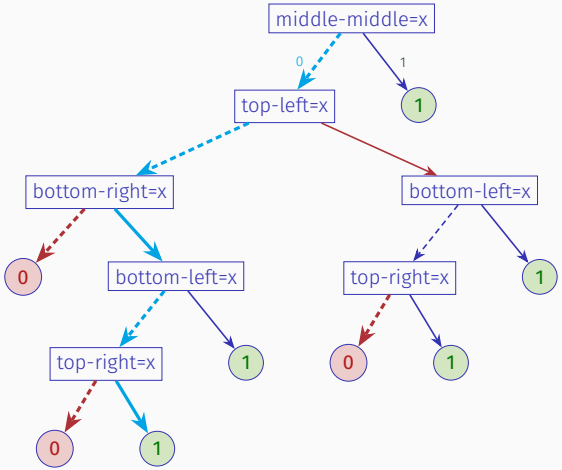
[IIM20, IIM22]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- For prediction 1, it suffices to ensure all paths with prediction 0 remain inconsistent
  - I.e. find a subset-minimal hitting set of all 0 paths; these are the features to keep
    - E.g. BR and TR suffice for prediction
  - Well-known to be solvable in polynomial time [EG95]

DT explanations in polynomial time

[IIM20, IIM22]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- For prediction 1, it suffices to ensure all paths with prediction 0 remain inconsistent
  - I.e. find a subset-minimal hitting set of all 0 paths; these are the features to keep
    - E.g. BR and TR suffice for prediction
  - Well-known to be solvable in polynomial time [EG95]
- Alternative: use Horn encoding (see notes)

Preliminary results for DTs

[IIM20, HIIM21, IIM22]

Dataset	(#F	#S)	IAI									ITI								
			D	#N	%A	#P	%R	%C	%m	%M	%avg	D	#N	%A	#P	%R	%C	%m	%M	%avg
adult	(12	6061)	6	83	78	42	33	25	20	40	25	17	509	73	255	75	91	10	66	22
anneal	(38	886)	6	29	99	15	26	16	16	33	21	9	31	100	16	25	4	12	20	16
backache	(32	180)	4	17	72	9	33	39	25	33	30	3	9	91	5	80	87	50	66	54
bank	(19	36 293)	6	113	88	57	5	12	16	20	18	19	1467	86	734	69	64	7	63	27
biodegradation	(41	1052)	5	19	65	10	30	1	25	50	33	8	71	76	36	50	8	14	40	21
cancer	(9	449)	6	37	87	19	36	9	20	25	21	5	21	84	11	54	10	25	50	37
car	(6	1728)	6	43	96	22	86	89	20	80	45	11	57	98	29	65	41	16	50	30
colic	(22	357)	6	55	81	28	46	6	16	33	20	4	17	80	9	33	27	25	25	25
compas	(11	1155)	6	77	34	39	17	8	16	20	17	15	183	37	92	66	43	12	60	27
contraceptive	(9	1425)	6	99	49	50	8	2	20	60	37	17	385	48	193	27	32	12	66	21
dermatology	(34	366)	6	33	90	17	23	3	16	33	21	7	17	95	9	22	0	14	20	17
divorce	(54	150)	5	15	90	8	50	19	20	33	24	2	5	96	3	33	16	50	50	50
german	(21	1000)	6	25	61	13	38	10	20	40	29	10	99	72	50	46	13	12	40	22
heart-c	(13	302)	6	43	65	22	36	18	20	33	22	4	15	75	8	87	81	25	50	34
heart-h	(13	293)	6	37	59	19	31	4	20	40	24	8	25	77	13	61	60	20	50	32
kr-vs-kp	(36	3196)	6	49	96	25	80	75	16	60	33	13	67	99	34	79	43	7	70	35
lending	(9	5082)	6	45	73	23	73	80	16	50	25	14	507	65	254	69	80	12	75	25
letter	(16	18 668)	6	127	58	64	1	0	20	20	20	46	4857	68	2429	6	7	6	25	9
lymphography	(18	148)	6	61	76	31	35	25	16	33	21	6	21	86	11	9	0	16	16	16
mortality	(118	13 442)	6	111	74	56	8	14	16	20	17	26	865	76	433	61	61	7	54	19
mushroom	(22	8124)	6	39	100	20	80	44	16	33	24	5	23	100	12	50	31	20	40	25
pendigits	(16	10 992)	6	121	88	61	0	0	—	—	—	38	937	85	469	25	86	6	25	11
promoters	(58	106)	1	3	90	2	0	0	—	—	—	3	9	81	5	20	14	33	33	33
recidivism	(15	3998)	6	105	61	53	28	22	16	33	18	15	611	51	306	53	38	9	44	16
seismic_bumps	(18	2578)	6	37	89	19	42	19	20	33	24	8	39	93	20	60	79	20	60	42
shuttle	(9	58 000)	6	63	99	32	28	7	20	33	23	23	159	99	80	33	9	14	50	30
soybean	(35	623)	6	63	88	32	9	5	25	25	25	16	71	89	36	22	1	9	12	10
spambase	(57	4210)	6	63	75	32	37	12	16	33	19	15	143	91	72	76	98	7	58	25
spect	(22	228)	6	45	82	23	60	51	20	50	35	6	15	86	8	87	98	50	83	65
splice	(2	3178)	3	7	50	4	0	0	—	—	—	88	177	55	89	0	0	—	—	—

Outline

Explaining monotonic classifiers

- Instance $(\mathbf{v}, c)$
- Domain for $i \in \mathcal{F}$: $\lambda(i) \leq x_i \leq \mu(i)$
- Idea: refine lower and upper bounds on the prediction
  - $\mathbf{v}_\mathsf{L}$ and $\mathbf{v}_\mathsf{U}$
- Utilities:
  - FixAttr(i):

    $\mathbf{v}_\mathsf{L} \leftarrow (v_{\mathsf{L}_1}, \ldots, v_i, \ldots, v_{\mathsf{L}_N})$
    $\mathbf{v}_\mathsf{U} \leftarrow (v_{\mathsf{U}_1}, \ldots, v_i, \ldots, v_{\mathsf{U}_N})$
    $(\mathcal{A}, \mathcal{B}) \leftarrow (\mathcal{A} \setminus \{i\}, \mathcal{B} \cup \{i\})$
    return $(\mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{A}, \mathcal{B})$
  - FreeAttr(i):

    $\mathbf{v}_\mathsf{L} \leftarrow (v_{\mathsf{L}_1}, \ldots, \lambda(i), \ldots, v_{\mathsf{L}_N})$
    $\mathbf{v}_\mathsf{U} \leftarrow (v_{\mathsf{U}_1}, \ldots, \mu(i), \ldots, v_{\mathsf{U}_N})$
    $(\mathcal{A}, \mathcal{B}) \leftarrow (\mathcal{A} \setminus \{i\}, \mathcal{B} \cup \{i\})$
    return $(\mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{A}, \mathcal{B})$

Computing one AXp

1: $\mathbf{v}_\mathsf{L} \leftarrow (v_1, \ldots, v_N)$
2: $\mathbf{v}_\mathsf{U} \leftarrow (v_1, \ldots, v_N)$ ▷ Ensures: $\kappa(\mathbf{v}_\mathsf{L}) = \kappa(\mathbf{v}_\mathsf{U})$
3: $(\mathcal{C}, \mathcal{D}, \mathcal{P}) \leftarrow (\mathcal{F}, \emptyset, \emptyset)$ ▷ $\mathcal{S}$: Some possible seed
4: **for all** $i \in \mathcal{S}$ **do**
5: $(\mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{C}, \mathcal{D}) \leftarrow \text{FreeAttr}(i, \mathbf{v}, \mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{C}, \mathcal{D})$ ▷ Require: $\kappa(\mathbf{v}_\mathsf{L}) = \kappa(\mathbf{v}_\mathsf{U})$, given $\mathcal{S}$
6: **for all** $i \in \mathcal{F} \setminus \mathcal{S}$ **do** ▷ Loop inv.: $\kappa(\mathbf{v}_\mathsf{L}) = \kappa(\mathbf{v}_\mathsf{U})$
7: $(\mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{C}, \mathcal{D}) \leftarrow \text{FreeAttr}(i, \mathbf{v}, \mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{C}, \mathcal{D})$
8: **if** $\kappa(\mathbf{v}_\mathsf{L}) \neq \kappa(\mathbf{v}_\mathsf{U})$ **then** ▷ If invariant broken, fix it
9: $(\mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{D}, \mathcal{P}) \leftarrow \text{FixAttr}(i, \mathbf{v}, \mathbf{v}_\mathsf{L}, \mathbf{v}_\mathsf{U}, \mathcal{D}, \mathcal{P})$
10: **return** $\mathcal{P}$

Computing one AXp – example

- $\lambda(i) = 0$ and $\mu(i) = 10$
- $\mathbf{v} = (10, 10, 5, 0)$, with $\kappa(\mathbf{v}) = A$
- Q: find one AXp (CXp is similar)

Feat.	Initial values		Changed values		Predictions		Dec.	Resulting values	
	$\mathbf{v}_L$	$\mathbf{v}_U$	$\mathbf{v}_L$	$\mathbf{v}_U$	$\kappa(\mathbf{v}_L)$	$\kappa(\mathbf{v}_U)$		$\mathbf{v}_L$	$\mathbf{v}_U$
1	(10,10,5,0)	(10,10,5,0)	(0,10,5,0)	(10,10,5,0)	C	A	✓	(10,10,5,0)	(10,10,5,0)
2	(10,10,5,0)	(10,10,5,0)	(10,0,5,0)	(10,10,5,0)	E	A	✓	(10,10,5,0)	(10,10,5,0)
3	(10,10,5,0)	(10,10,5,0)	(10,10,0,0)	(10,10,10,0)	A	A	✗	(10,10,0,0)	(10,10,10,0)
4	(10,10,0,0)	(10,10,10,0)	(10,10,0,0)	(10,10,10,10)	A	A	✗	(10,10,0,0)	(10,10,10,10)

Part 6

Explainability Queries

Outline

How to navigate the space of XPs?

- Goal: iteratively list yet unlisted XPs (either AXp's or CXp's)

How to navigate the space of XPs?

- Goal: iteratively list yet unlisted XPs (either AXp's or CXp's)

- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC+20]
  - For monotonic classifiers: enumeration is computationally hard [MGC+21]
  - For DTs, enumeration of CXp's is in P [HIIM21, IIM22]

How to navigate the space of XPs?

- Goal: iteratively list yet unlisted XPs (either AXp's or CXp's)
- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC+20]
  - For monotonic classifiers: enumeration is computationally hard [MGC+21]
  - For DTs, enumeration of CXp's is in P [HIIM21, IIM22]
- There are algorithms for enumerating CXp's
  - Akin to enumerating MCSes

How to navigate the space of XPs?

- Goal: iteratively list yet unlisted XPs (either AXp's or CXp's)

- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC+20]
  - For monotonic classifiers: enumeration is computationally hard [MGC+21]
  - For DTs, enumeration of CXp's is in P [HIIM21, IIM22]

- There are algorithms for enumerating CXp's
  - Akin to enumerating MCSes
- No known algorithms for direct enumeration of AXp's [MM20]
  - Akin to enumerating MUSes

How to navigate the space of XPs?

- Goal: iteratively list yet unlisted XPs (either AXp's or CXp's)

- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC+20]
  - For monotonic classifiers: enumeration is computationally hard [MGC+21]
  - For DTs, enumeration of CXp's is in P [HIIM21, IIM22]

- There are algorithms for enumerating CXp's
  - Akin to enumerating MCSes
- No known algorithms for direct enumeration of AXp's [MM20]
  - Akin to enumerating MUSes
- Enumeration of MCSes + dualization often not realistic [LS08, FK96]
  - There can be too many CXp's...

How to navigate the space of XPs?

- Goal: iteratively list yet unlisted XPs (either AXp's or CXp's)

- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC+20]
  - For monotonic classifiers: enumeration is computationally hard [MGC+21]
  - For DTs, enumeration of CXp's is in P [HIIM21, IIM22]

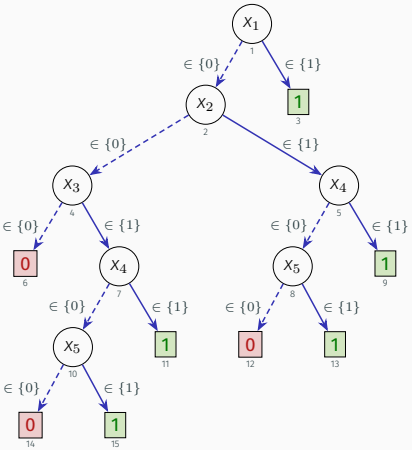
- There are algorithms for enumerating CXp's
  - Akin to enumerating MCSes
- No known algorithms for direct enumeration of AXp's [MM20]
  - Akin to enumerating MUSes
- Enumeration of MCSes + dualization often not realistic [LS08, FK96]
  - There can be too many CXp's...
- Best solution is a MARCO-like algorithm (for enumerating MUSes) [LPMM16]
  - On-demand enumeration of AXp's/CXp's

Generic oracle-based algorithm

Input: Parameters $\mathbb{P}_{\text{axp}}$, $\mathbb{P}_{\text{cxp}}$, $\mathcal{T}$, $\mathcal{F}$, κ, $\mathbf{v}$

$\mathcal{H} \leftarrow \emptyset$ ▷ $\mathcal{H}$ defined on set $U = \{u_1, \ldots, u_m\}$
repeat
  $(\text{outc}, \mathbf{u}) \leftarrow \text{SAT}(\mathcal{H})$
  if outc $=$ **true then**
    $\mathcal{S} \leftarrow \{i \in \mathcal{F} \mid u_i = 0\}$ ▷ $\mathcal{S}$: *fixed* features
    $\mathcal{U} \leftarrow \{i \in \mathcal{F} \mid u_i = 1\}$ ▷ $\mathcal{U}$: *universal* features; $\mathcal{F} = \mathcal{S} \cup \mathcal{U}$
    if $\mathbb{P}_{\text{cxp}}(\mathcal{S}; \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$ **then** ▷ $\mathcal{F} \setminus \mathcal{S} \supseteq$ some CXp
      $\mathcal{P} \leftarrow \text{oneXP}(\mathcal{S}; \mathbb{P}_{\text{cxp}}, \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$
      reportCXp($\mathcal{P}$)
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\vee_{i \in \mathcal{P}} \neg u_i)\}$
    else ▷ $\mathcal{S} \supseteq$ some AXp
      $\mathcal{P} \leftarrow \text{oneXP}(\mathcal{S}; \mathbb{P}_{\text{axp}}, \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$
      reportAXp($\mathcal{P}$)
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\vee_{i \in \mathcal{P}} u_i)\}$
until outc $=$ **false**

DT classifier – example run of enumerator

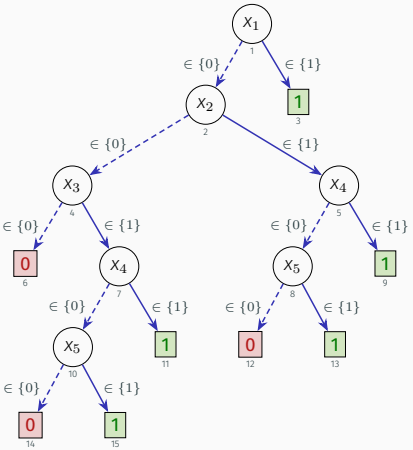


- Instance: $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	1

DT classifier – example run of enumerator



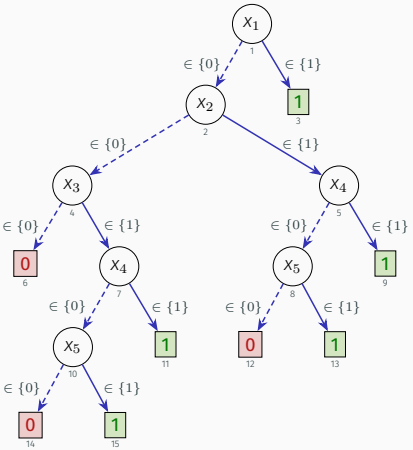
- Instance: $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	1

Iter.	$\mathbf{u}$	$\mathcal{S}$	$\mathbb{P}_{\text{cxp}}(\cdot)$	AXp	CXp	Clause
1	$(1, 1, 1, 1, 1)$	$\emptyset$	1	–	$\{3\}$	$(\neg u_3)$
2	$(1, 1, 0, 1, 1)$	$\{3\}$	1	–	$\{5\}$	$(\neg u_5)$
3	$(1, 1, 0, 1, 0)$	$\{3, 5\}$	0	$\{3, 5\}$	–	$(u_3 \vee u_5)$
5	$[\text{outc} = \textbf{false}]$	–	–	–	–	–

DT classifier – another example run of enumerator



- Instance: $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

x_3	x_5	x_1	x_2	x_4	$\kappa_2(\mathbf{x})$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	1

Iter.	$\mathbf{u}$	$\mathcal{S}$	$\mathbb{P}_{\text{cxp}}(\cdot)$	AXp	CXp	Clause
1	$(0, 0, 0, 0, 0)$	$\{1, 2, 3, 4, 5\}$	0	$\{3, 5\}$	–	$(u_3 \vee u_5)$
2	$(0, 0, 1, 0, 0)$	$\{1, 2, 4, 5\}$	1	–	$\{3\}$	$(\neg u_3)$
3	$(0, 0, 1, 0, 1)$	$\{1, 2, 4\}$	1	–	$\{5\}$	$(\neg u_5)$
5	[outc = **false**]	–	–	–	–	–

DTs admit more efficient algorithms

- Given instance $(\mathbf{v}, c)$, create set $\mathcal{I}$
  - For each path P_k with prediction $d \neq c$:
    - Let I_k denote the features with literals inconsistent with $\mathbf{v}$
    - Add I_k to $\mathcal{I}$
  - Remove from $\mathcal{I}$ the sets that have a proper subset in $\mathcal{I}$
- $\mathcal{I}$ is the set of CXp's – algorithm runs in poly-time

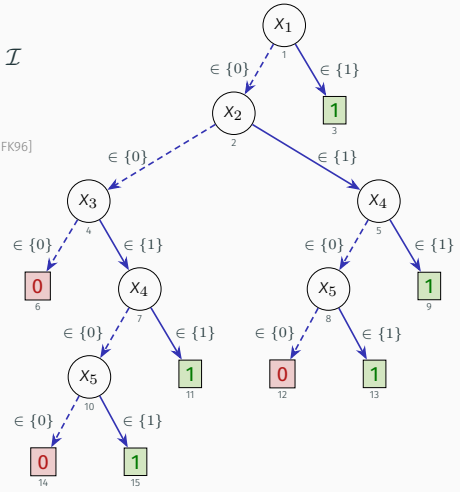
DTs admit more efficient algorithms

- Given instance $(\mathbf{v}, c)$, create set $\mathcal{I}$
  - For each path P_k with prediction $d \neq c$:
    - Let I_k denote the features with literals inconsistent with $\mathbf{v}$
    - Add I_k to $\mathcal{I}$
  - Remove from $\mathcal{I}$ the sets that have a proper subset in $\mathcal{I}$
- $\mathcal{I}$ is the set of CXp's – algorithm runs in poly-time
- For AXp's: run std dualization algorithm [FK96]
  - Obs: starting hypergraph is poly-size!
  - And each MHS is an AXp

DTs admit more efficient algorithms

- Given instance $(\mathbf{v}, c)$, create set $\mathcal{I}$
  - For each path P_k with prediction $d \neq c$:
    - Let I_k denote the features with literals inconsistent with $\mathbf{v}$
    - Add I_k to $\mathcal{I}$
  - Remove from $\mathcal{I}$ the sets that have a proper subset in $\mathcal{I}$
- $\mathcal{I}$ is the set of CXp's – algorithm runs in poly-time
- For AXp's: run std dualization algorithm
  - Obs: starting hypergraph is poly-size!
  - And each MHS is an AXp
- Example:
  - $I_1 = \{3\}$
  - $I_2 = \{5\}$
  - $I_3 = \{2, 5\}$
  - $\therefore$ keep I_1 an I_2
  - AXp's: MHSes yield $\{\{3, 5\}\}$

[FK96]



Outline

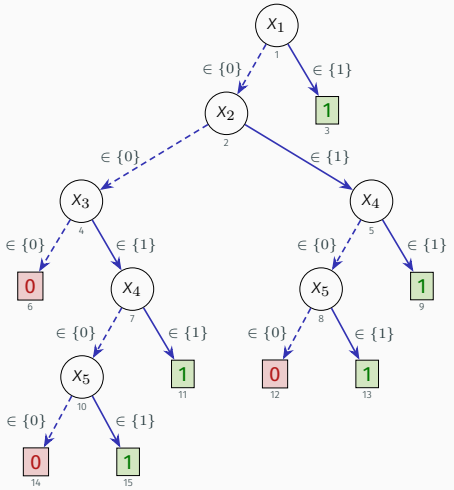
What is feature membership?

- The feature membership problem (FMP):
  - Given an instance $(\mathbf{v}, c)$ and a (sensitive) target feature $t \in \mathcal{F}$
  - Q: Does there exist an AXp/CXp that includes t?

What is feature membership?

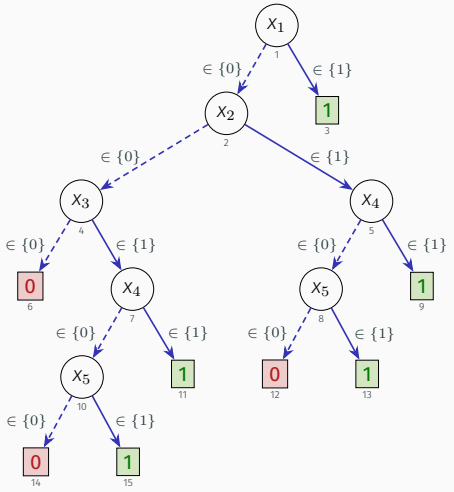
- The feature membership problem (FMP):
  - Given an instance $(\mathbf{v}, c)$ and a (sensitive) target feature $t \in \mathcal{F}$
  - Q: Does there exist an AXp/CXp that includes t?

- For example DT and instance $((0, 0, 1, 0, 1), 1)$
  - For 3 and 5, the answer to FMP is **true**
  - For 1, 2, 4, the answer to FMP is **false**



What is feature membership?

- The feature membership problem (FMP):
  - Given an instance $(\mathbf{v}, c)$ and a (sensitive) target feature $t \in \mathcal{F}$
  - Q: Does there exist an AXp/CXp that includes t?
- For example DT and instance $((0,0,1,0,1),1)$
  - For 3 and 5, the answer to FMP is **true**
  - For 1, 2, 4, the answer to FMP is **false**
- Some results:
  - Σ_2^{P}-hard for DNF classifiers [HIIM21]
    - Hence, Σ_2^{P}-hard for NNs, RFs, BTs, etc.
  - In P for decision trees [HIIM21]
  - In NP when computing AXp/CXp is in P [HM22]



Additional observations

- FMP in P for DTs:
  - Compute all CXp's in poly-time
  - Check whether target feature included in some CXp

Additional observations

- FMP in P for DTs:
  - Compute all CXp's in poly-time
  - Check whether target feature included in some CXp

- FMP in NP if computing one AXp is in P:
  - Guess a set $\mathcal{X}$, that contains t
  - Check (in poly-time) that $\mathcal{X}$ is weak AXp
  - For each feature $j \in \mathcal{X}$, check (in poly-time) that $\mathcal{X} \setminus \{j\}$ is not a weak AXp
  - Overall algorithm runs in poly-time

Outline

Additional explicability queries

- Class queries [AKM20, ABB+21]
  - Mandatory/forbidden features for class
  - Necessary features for class
  - ...
- Explanation queries:
  - Smallest AXp's
  - Finding one AXp
  - Finding on CXp
- Complexity: [ABB+21]
  - Poly-time for DTs
  - Computationally hard for other classifiers: DLs, RFs, BTs, boolean NNs, BNNs

Part 7

Probabilistic Explanations

Outline

Defining Probabilistic Explanations

Why probabilistic XPs?

- Explanation size can exceed cognitive limits of human decision makers
- Smallest explanations are rarely significantly smaller than irreducible explanations
- Goal of probabilistic explanations is to trade off smaller size for less rigor
  - I.e. find set $\mathcal{X}$ such that

$$\Pr_{\mathbf{x}}(\kappa(\mathbf{x}) = c \,|\, \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geq \delta$$

  Obs: for $\delta = 1$, set $\mathcal{X}$ is a (weak) AXp
- Weak probabilistic abductive explanation:

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) \quad := \quad \Pr_{\mathbf{x}}(\kappa(\mathbf{x}) = c \,|\, \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geq \delta$$

- Probabilistic abductive explanation:
  - Subset-minimal (PAXp) vs. cardinality minimal (minPAXp)

Overview of main results

- For boolean circuit classifiers, computing one minPAXp is NP^{PP}-hard [WMHK21]
- For DTs, finding one minPAXp/PAXp is NP-hard [IIN+22, ABRS22]
  - Approximate Weak PAXp's can be computed in linear time (see notes)
- For NBCs, dynamic programming approach [IM22]
- Recent hardness results [ABRS22]
- General approach?

Part 8

Additional Topics & Conclusions

Outline

Additional Topics

Beyond ML Explanations

Links with fairness, robustness & model learning

- Robustness: [INM19b]
  - Link with adversarial examples
- Model learning: [NIPM18]
  - Efforts on learning *interpretable* models
- Fairness: [ICS+20]
  - Criterion: Fairness Through Unawareness (FTU)
  - Assess bias on dataset/classifier
  - Bias on classifier related with XPs

Some additional topics

- Input constraints [GR22, YIS+22]
- Surrogate models [BAMT21]
- Generalized explanation literals [IIM22]
- Localized explanations
  - Require sufficiency close to reference point
- Certification of explainability
  - Certify results generated with implemented algorithms

Outline

Explainability beyond ML explanations

- Model-based diagnosis & abduction
- AI planning
- Optimization
- Problem solving
- ...
- XPs relevant for any sort of algorithmic decision making

Conclusions

- Critical limitations of widely used XAI approaches:
  - Model-agnostic methods can compute **incorrect** explanations
    - Similar limitations for methods based on saliency maps for NNs
  - Intrinsic interpretability explanations can be (very) redundant

Conclusions

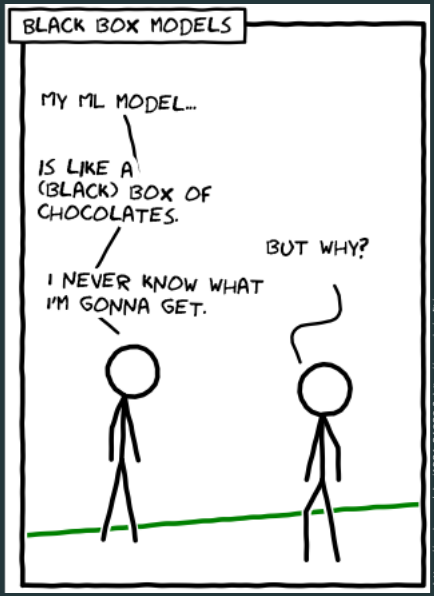
- Critical limitations of widely used XAI approaches:
  - Model-agnostic methods can compute **incorrect** explanations
    - Similar limitations for methods based on saliency maps for NNs
  - Intrinsic interpretability explanations can be (very) redundant
- **Formal explainability in AI** (FXAI):
  - Logic-based, rigorous definitions of explanations
    - Initial theoretical insights, e.g. duality between AXp's and CXp's (and so between "Why?" and "Why not?" explanations)
  - Other problems of crucial importance: enumeration, membership, etc.

Conclusions

- Critical limitations of widely used XAI approaches:
  - Model-agnostic methods can compute incorrect explanations
    - Similar limitations for methods based on saliency maps for NNs
  - Intrinsic interpretability explanations can be (very) redundant
- Formal explainability in AI (FXAI):
  - Logic-based, rigorous definitions of explanations
    - Initial theoretical insights, e.g. duality between AXp's and CXp's (and so between "Why?" and "Why not?" explanations)
  - Other problems of crucial importance: enumeration, membership, etc.
- Ongoing research related with:
  - Computation of AXp's & CXp's
  - Enumeration of explanations
  - Deciding membership of features in explanations
  - Probabilistic rigorous explanations
  - Surrogate models, input constraints, etc.
  - Also, complexity of explainability

Q & A

Acknowledgment: joint work with A. Ignatiev, Y. Izza, X. Huang, M. Cooper, N. Asher, N. Narodytska, E. Hebrard, M. Siala, et al.


BLACK BOX MODELS
MY ML MODEL...
IS LIKE A (BLACK) BOX OF CHOCOLATES.
I NEVER KNOW WHAT I'M GONNA GET.
BUT WHY?
https://arxiv.org/abs/1901.01686 & http://cmx.io/edit/

References i

[ABB+21] Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marquis.
On the computational intelligibility of boolean classifiers.
In *KR*, pages 74–86, 2021.

[ABRS22] Marcelo Arenas, Pablo Barceló, Miguel Romero, and Bernardo Subercaseaux.
On computing probabilistic explanations for decision trees.
CoRR, abs/2207.12213, 2022.

[AKM20] Gilles Audemard, Frédéric Koriche, and Pierre Marquis.
On tractable XAI queries based on compiled representations.
In *KR*, pages 838–849, 2020.

[Alp14] Ethem Alpaydin.
Introduction to machine learning.
MIT press, 2014.

[Alp16] Ethem Alpaydin.
Machine Learning: The New AI.
MIT Press, 2016.

[ANS20] Gaël Aglin, Siegfried Nijssen, and Pierre Schaus.
PyDL8.5: a library for learning optimal decision trees.
In *IJCAI*, pages 5222–5224, 2020.

References ii

[BA97] Leonard A. Breslow and David W. Aha.
Simplifying decision trees: A survey.
Knowledge Eng. Review, 12(1):1–40, 1997.

[BAMT21] Ryma Boumazouza, Fahima Cheikh Alili, Bertrand Mazure, and Karim Tabia.
ASTERYX: A model-agnostic sat-based approach for symbolic and score-based explanations.
In *CIKM*, pages 120–129, 2021.

[BBHK10] Michael R. Berthold, Christian Borgelt, Frank Höppner, and Frank Klawonn.
***Guide to Intelligent Data Analysis - How to Intelligently Make Sense of Real Data*, volume 42 of *Texts in Computer Science*.**
Springer, 2010.

[BFOS84] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone.
***Classification and Regression Trees*.**
Wadsworth, 1984.

[BHO09] Christian Bessiere, Emmanuel Hebrard, and Barry O'Sullivan.
Minimising decision tree size as combinatorial optimisation.
In *CP*, pages 173–187, 2009.

References iii

[BHvMW09] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors.
Handbook of Satisfiability, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

[Bra20] Max Bramer.
Principles of Data Mining, 4th Edition.
Undergraduate Topics in Computer Science. Springer, 2020.

[Bre01] Leo Breiman.
Statistical modeling: The two cultures.
Statistical science, 16(3):199–231, 2001.

[CM21] Martin C. Cooper and Joao Marques-Silva.
On the tractability of explaining decisions of classifiers.
In *CP*, October 2021.

[DL01] Sašo Džeroski and Nada Lavrač, editors.
Relational data mining.
Springer, 2001.

[EG95] Thomas Eiter and Georg Gottlob.
Identifying the minimal transversals of a hypergraph and related problems.
SIAM J. Comput., 24(6):1278–1304, 1995.

References iv

[EU21] EU.
European Artificial Intelligence Act – Proposal.
https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1623335154975&uri=CELEX%3A52021PC0206, 2021.

[FJ18] Matteo Fischetti and Jason Jo.
Deep neural networks and mixed integer linear optimization.
Constraints, 23(3):296–309, 2018.

[FK96] Michael L. Fredman and Leonid Khachiyan.
On the complexity of dualization of monotone disjunctive normal forms.
J. Algorithms, 21(3):618–628, 1996.

[Fla12] Peter A. Flach.
Machine Learning - The Art and Science of Algorithms that Make Sense of Data.
Cambridge University Press, 2012.

[GR22] Niku Gorji and Sasha Rubin.
Sufficient reasons for classifier decisions in the presence of domain constraints.
In *AAAI*, February 2022.

References v

[HII+22] Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, Martin Cooper, Nicholas Asher, and Joao Marques-Silva.
Tractable explanations for d-DNNF classifiers.
In *AAAI*, February 2022.

[HIIM21] Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva.
On efficiently explaining graph-based classifiers.
In *KR*, November 2021.
Preprint available from `https://arxiv.org/abs/2106.01350`.

[HM22] Xuanxiang Huang and João Marques-Silva.
On deciding feature membership in explanations of SDD & related classifiers.
CoRR, abs/2202.07553, 2022.

[HRS19] Xiyang Hu, Cynthia Rudin, and Margo Seltzer.
Optimal sparse decision trees.
In *NeurIPS*, pages 7265–7273, 2019.

[ICS+20] Alexey Ignatiev, Martin C. Cooper, Mohamed Siala, Emmanuel Hebrard, and Joao Marques-Silva.
Towards formal fairness in machine learning.
In *CP*, pages 846–867, 2020.

References vi

[Ign20] Alexey Ignatiev.
Towards trustable explainable AI.
In *IJCAI*, pages 5154–5158, 2020.

[IIM20] Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva.
On explaining decision trees.
CoRR, abs/2010.11034, 2020.

[IIM22] Yacine Izza, Alexey Ignatiev, and João Marques-Silva.
On tackling explanation redundancy in decision trees.
J. Artif. Intell. Res., 2022.
In Press. Preprint available from `https://doi.org/10.48550/arXiv.2205.09971`.

[IIN+22] Yacine Izza, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.
Provably precise, succinct and efficient explanations for decision trees.
CoRR, abs/2205.09569, 2022.

[IISMS22] Alexey Ignatiev, Yacine Izza, Peter J. Stuckey, and Joao Marques-Silva.
Using MaxSAT for efficient explanations of tree ensembles.
In *AAAI*, February 2022.

References vii

[IM21] Alexey Ignatiev and Joao Marques-Silva.
SAT-based rigorous explanations for decision lists.
In *SAT*, pages 251–269, July 2021.

[IM22] Yacine Izza and João Marques-Silva.
On computing relevant features for explaining NBCs.
CoRR, abs/2207.04748, 2022.

[IMS21] Yacine Izza and Joao Marques-Silva.
On explaining random forests with SAT.
In *IJCAI*, pages 2584–2591, July 2021.

[INAM20] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva.
From contrastive to abductive explanations and back again.
In *AIxIA*, pages 335–355, 2020.

[INM19a] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
Abduction-based explanations for machine learning models.
In *AAAI*, pages 1511–1519, 2019.

[INM19b] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
On relating explanations and adversarial examples.
In *NeurIPS*, pages 15857–15867, 2019.

References viii

[INM19c] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
On validating, repairing and refining heuristic ML explanations.
CoRR, abs/1907.02509, 2019.

[Int21] Interpretable AI.
`https://www.interpretable.ai/`, 21.

[KBD+17] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer.
Reluplex: An efficient SMT solver for verifying deep neural networks.
In *CAV*, pages 97–117, 2017.

[KMND20] John D Kelleher, Brian Mac Namee, and Aoife D'arcy.
Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.
MIT Press, 2020.

[Kot13] Sotiris B. Kotsiantis.
Decision trees: a recent overview.
Artif. Intell. Rev., 39(4):261–283, 2013.

[Lip18] Zachary C. Lipton.
The mythos of model interpretability.
Commun. ACM, 61(10):36–43, 2018.

References ix

[LL17] Scott M. Lundberg and Su-In Lee.
A unified approach to interpreting model predictions.
In *NIPS*, pages 4765–4774, 2017.

[LPMM16] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva.
Fast, flexible MUS enumeration.
Constraints, 21(2):223–250, 2016.

[LS08] Mark H. Liffiton and Karem A. Sakallah.
Algorithms for computing minimal unsatisfiable subsets of constraints.
J. Autom. Reasoning, 40(1):1–33, 2008.

[MGC+20] Joao Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska.
Explaining naive bayes and other linear classifiers with polynomial time and delay.
In *NeurIPS*, 2020.

[MGC+21] Joao Marques-Silva, Thomas Gerspacher, Martinc C. Cooper, Alexey Ignatiev, and Nina Narodytska.
Explanations for monotonic classifiers.
In *ICML*, pages 7469–7479, July 2021.

[Mil19] Tim Miller.
Explanation in artificial intelligence: Insights from the social sciences.
Artif. Intell., 267:1–38, 2019.

References x

[MM20] João Marques-Silva and Carlos Mencía.
Reasoning about inconsistent formulas.
In *IJCAI*, pages 4899–4906, 2020.

[Mol20] Christoph Molnar.
Interpretable machine learning.
Lulu.com, 2020.
`https://christophm.github.io/interpretable-ml-book/`.

[Mor82] Bernard M. E. Moret.
Decision trees and diagrams.
ACM Comput. Surv., 14(4):593–623, 1982.

[NH10] Vinod Nair and Geoffrey E. Hinton.
Rectified linear units improve restricted boltzmann machines.
In *ICML*, pages 807–814, 2010.

[NIPM18] Nina Narodytska, Alexey Ignatiev, Filipe Pereira, and Joao Marques-Silva.
Learning optimal decision trees with SAT.
In *IJCAI*, pages 1362–1368, 2018.

References xi

[NSM+19] Nina Narodytska, Aditya A. Shrotri, Kuldeep S. Meel, Alexey Ignatiev, and Joao Marques-Silva.
Assessing heuristic machine learning explanations with model counting.
In *SAT*, pages 267–278, 2019.

[PM17] David Poole and Alan K. Mackworth.
Artificial Intelligence - Foundations of Computational Agents.
CUP, 2017.

[Qui93] J Ross Quinlan.
C4.5: programs for machine learning.
Morgan-Kaufmann, 1993.

[Rei87] Raymond Reiter.
A theory of diagnosis from first principles.
Artif. Intell., 32(1):57–95, 1987.

[RM08] Lior Rokach and Oded Z Maimon.
Data mining with decision trees: theory and applications.
World scientific, 2008.

[RN10] Stuart J. Russell and Peter Norvig.
Artificial Intelligence - A Modern Approach.
Pearson Education, 2010.

References xii

[RSG16] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.
"why should I trust you?": Explaining the predictions of any classifier.
In *KDD*, pages 1135–1144, 2016.

[RSG18] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.
Anchors: High-precision model-agnostic explanations.
In *AAAI*, pages 1527–1535. AAAI Press, 2018.

[Rud19] Cynthia Rudin.
Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.
Nature Machine Intelligence, 1(5):206–215, 2019.

[SB14] Shai Shalev-Shwartz and Shai Ben-David.
Understanding Machine Learning - From Theory to Algorithms.
Cambridge University Press, 2014.

[SCD18] Andy Shih, Arthur Choi, and Adnan Darwiche.
A symbolic approach to explaining bayesian network classifiers.
In *IJCAI*, pages 5103–5111, 2018.

References xiii

[VLE+16] Gilmer Valdes, José Marcio Luna, Eric Eaton, Charles B Simone, Lyle H Ungar, and Timothy D Solberg.
MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine.
Scientific reports, 6(1):1–8, 2016.

[WFHP17] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal.
***Data Mining*.**
Morgan Kaufmann, 2017.

[WMHK21] Stephan Wäldchen, Jan MacDonald, Sascha Hauch, and Gitta Kutyniok.
The computational complexity of understanding binary classifier decisions.
J. Artif. Intell. Res., 70:351–387, 2021.

[YIS+22] Jinqiang Yu, Alexey Ignatiev, Peter J. Stuckey, Nina Narodytska, and João Marques-Silva.
Eliminating the impossible, whatever remains must be true.
CoRR, abs/2206.09551, 2022.

[Zho12] Zhi-Hua Zhou.
***Ensemble methods: foundations and algorithms*.**
CRC press, 2012.

References xiv

[Zho21] Zhi-Hua Zhou.
Machine Learning.
Springer, 2021.