Logic-Based Explainable Artificial Intelligence

Joao Marques-Silva

ICREA, Univ. Lleida, Catalunya, Spain

ESSAI, Athens, Greece, July 2024

Lecture 03

Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations

Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp

Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp
- Explanations for DTs

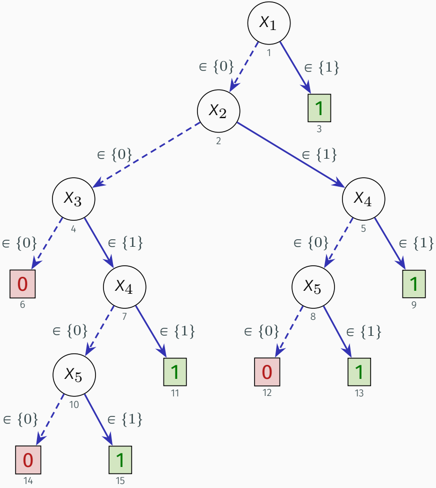
Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp
- Explanations for DTs
- Explanations for XpGs

Recapitulate second lecture

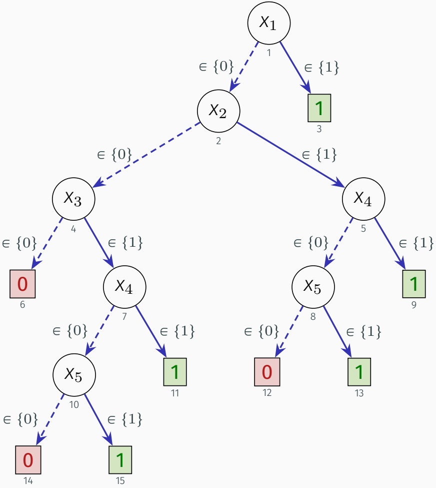
- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp
- Explanations for DTs
- Explanations for XpGs
- Explanations for monotonic classifiers

- Instance: $((0,0,1,0,0),0)$



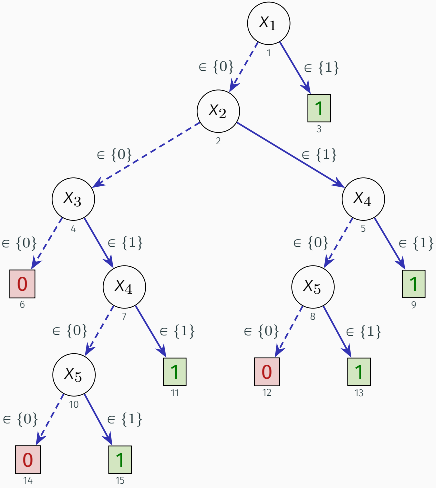
Recap AXps/CXps: DT example

- Instance: $((0, 0, 1, 0, 0), 0)$
- One AXp: $\{1, 4, 5\}$

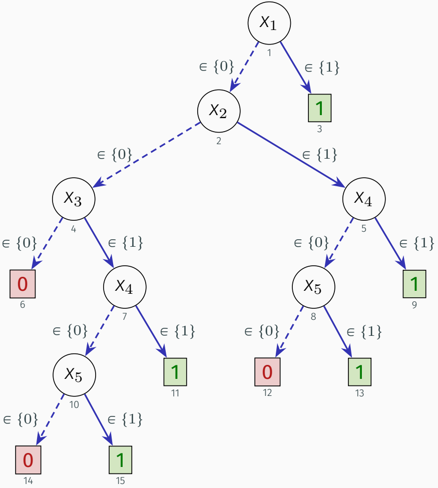


Recap AXps/CXps: DT example

- Instance: $((0,0,1,0,0),0)$
- One AXp: $\{1,4,5\}$
- All CXps:

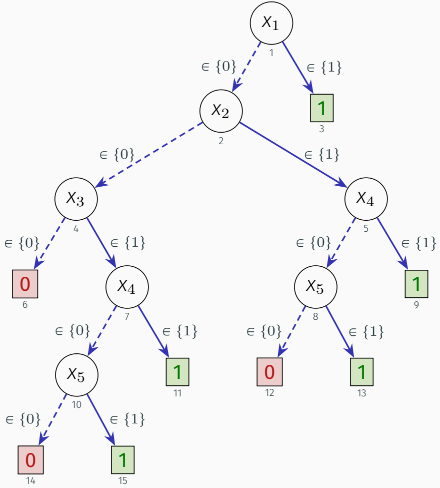


- Instance: $((0,0,1,0,0),0)$
- One AXp: $\{1,4,5\}$
- All CXps:
    - I_1: $\{5\}$

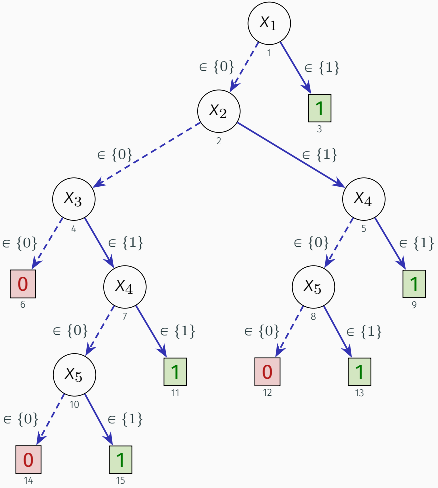


Recap AXps/CXps: DT example

- Instance: $((0, 0, 1, 0, 0), 0)$
- One AXp: $\{1, 4, 5\}$
- All CXps:
  - I_1: $\{5\}$
  - I_2: $\{4\}$

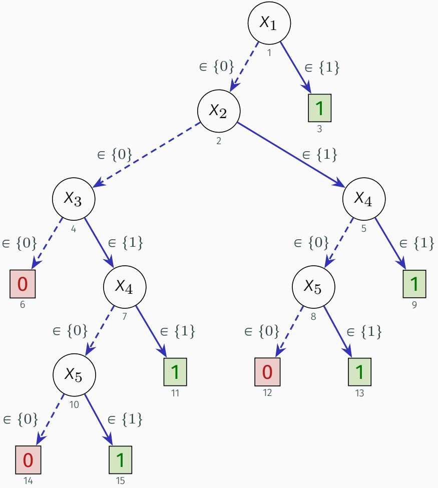


- Instance: $((0, 0, 1, 0, 0), 0)$
- One AXp: $\{1, 4, 5\}$
- All CXps:
  - I_1: $\{5\}$
  - I_2: $\{4\}$
  - I_3: $\{2, 5\}$



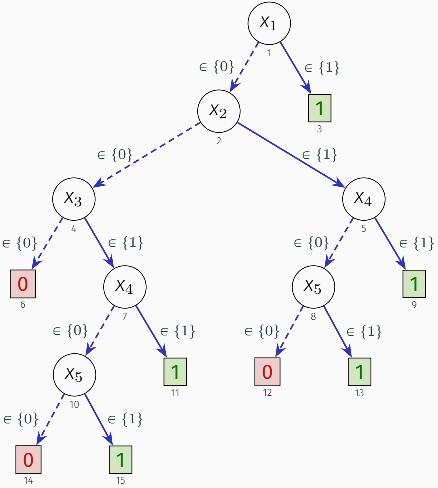
Recap AXps/CXps: DT example

- Instance: $((0,0,1,0,0),0)$
- One AXp: $\{1,4,5\}$
- All CXps:
  - I_1: $\{5\}$
  - I_2: $\{4\}$
  - I_3: $\{2,5\}$
  - I_4: $\{2,4\}$



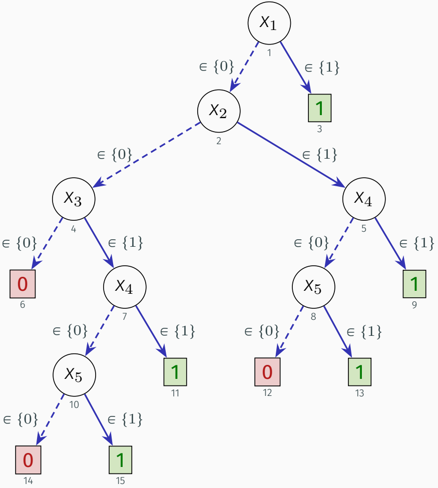
Recap AXps/CXps: DT example

- Instance: $((0,0,1,0,0),0)$
- One AXp: $\{1,4,5\}$
- All CXps:
  - I_1: $\{5\}$
  - I_2: $\{4\}$
  - I_3: $\{2,5\}$
  - I_4: $\{2,4\}$
  - I_5: $\{1\}$



Recap AXps/CXps: DT example

- Instance: $((0, 0, 1, 0, 0), 0)$
- One AXp: $\{1, 4, 5\}$
- All CXps:
  - I_1: $\{5\}$
  - I_2: $\{4\}$
  - I_3: $\{2, 5\}$
  - I_4: $\{2, 4\}$
  - I_5: $\{1\}$
  - $\mathcal{L} = \{\{1\}, \{4\}, \{5\}\}$



Recap AXps/CXps: DL 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
$\mathsf{R}_{\mathsf{DEF}}$:	ELSE		THEN	1

Entry	x_1	x_2	x_3	x_4	Rule	$\kappa_1(\mathbf{x})$
00	0	0	0	0	$\mathsf{R}_{\mathsf{DEF}}$	1
01	0	0	0	1	R_3	0
02	0	0	0	2	$\mathsf{R}_{\mathsf{DEF}}$	1
03	0	0	1	0	$\mathsf{R}_{\mathsf{DEF}}$	1
04	0	0	1	1	R_3	0
05	0	0	1	2	$\mathsf{R}_{\mathsf{DEF}}$	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

Recap AXps/CXps: DL 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

Plan for this course

- Lecture 01 – units:
  - #01: Foundations
- Lecture 02 – units:
  - #02: Principles of symbolic XAI – feature selection
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – units:
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – units:
  - #06: Advanced topics
- Lecture 05 – units:
  - #07: Principles of symbolic XAI – feature attribution (& myth of Shapley values in XAI)
  - #08: Conclusions & research directions

Some comments…

Some necessary comments...

- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**

Some necessary comments...

- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**
  - Most likely answer: **No!**

Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model XYZ?
  - Most likely answer: **No!** But ...

Some necessary comments...

- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?

Some necessary comments...

- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?

Some necessary comments...

- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?

Some necessary comments...

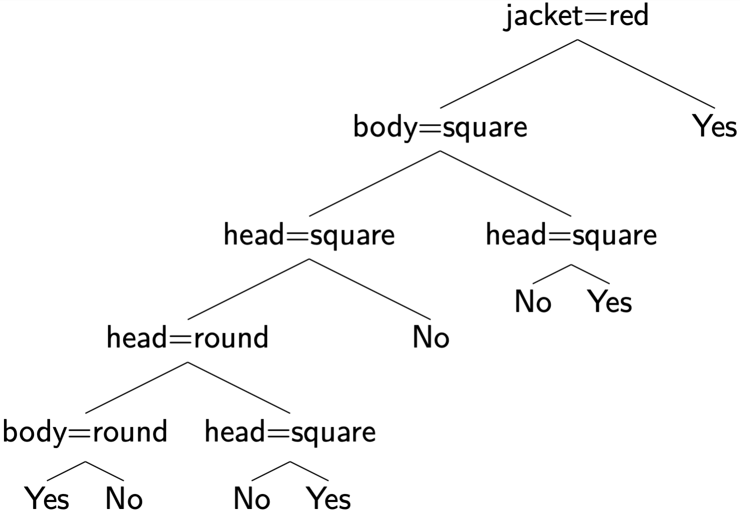
- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?
- For high-risk and safety-critical domains:
  - Would you use an ML model that you cannot explain with rigor, and whose heuristic explanations are likely to be wrong, and so debugging/understanding with rigor is all but impossible?

Some necessary comments...

- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?
- For high-risk and safety-critical domains:
  - Would you use an ML model that you cannot explain with rigor, and whose heuristic explanations are likely to be wrong, and so debugging/understanding with rigor is all but impossible?
- What is the bottom line?
  - For high-risk and safety-critical domains, one **ought** to deploy models that can be explained with rigor
  - If that means using a fairly unexciting NN with up to 100K neurons, that is the cost of trust; **for anything else, one is trying his/her luck, in situations that could become catastrophic!**

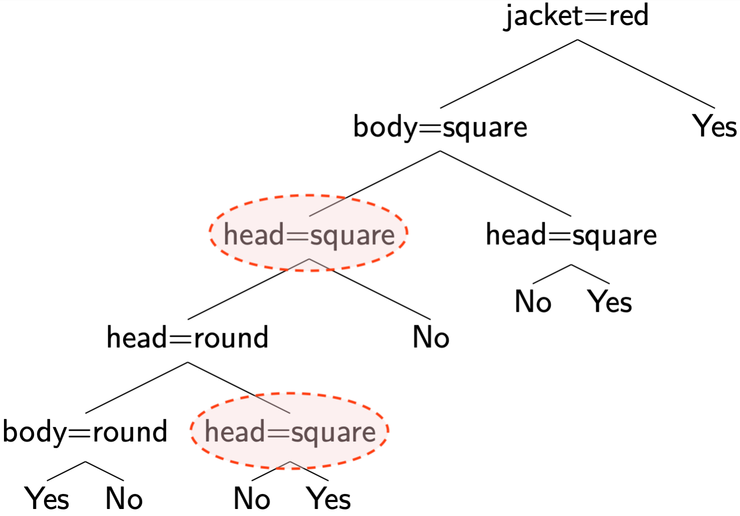
Some necessary comments...

- Std question: **Can we apply symbolic XAI to this highly complex ML model XYZ?**
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?
- For high-risk and safety-critical domains:
  - Would you use an ML model that you cannot explain with rigor, and whose heuristic explanations are likely to be wrong, and so debugging/understanding with rigor is all but impossible?
- What is the bottom line?
  - For high-risk and safety-critical domains, one **ought** to deploy models that can be explained with rigor
  - If that means using a fairly unexciting NN with up to 100K neurons, that is the cost of trust; **for anything else, one is trying his/her luck, in situations that could become catastrophic!**
  - More examples next...



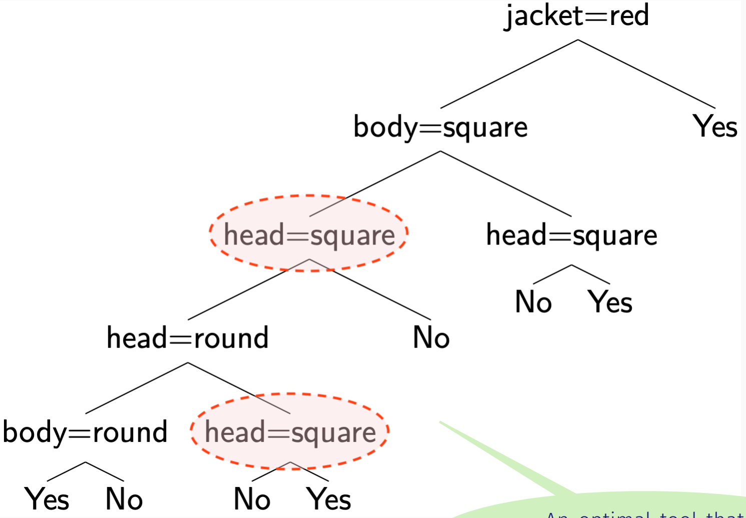
Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer:
Optimal Sparse Decision Trees.
NeurIPS 2019: 7265-7273

Priceless optimal sparse decision trees (OSDT) – & non-optimality!... [HRS19]



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

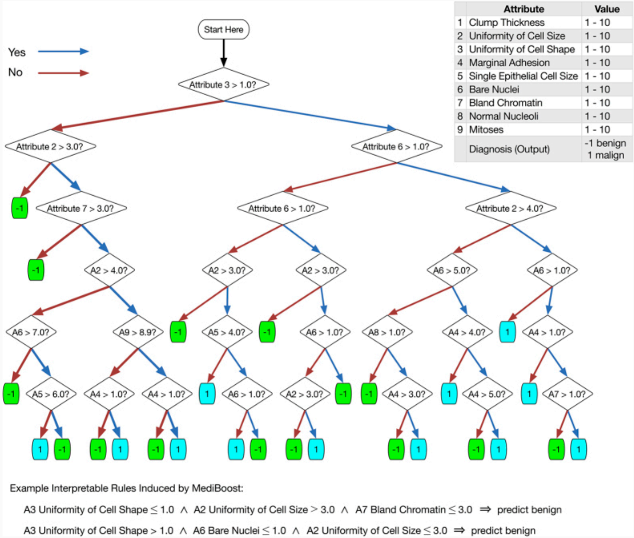
Priceless optimal sparse decision trees (OSDT) – & non-optimality!... [HRS19]



An optimal tool that produces **non-optimal** DTs...!?

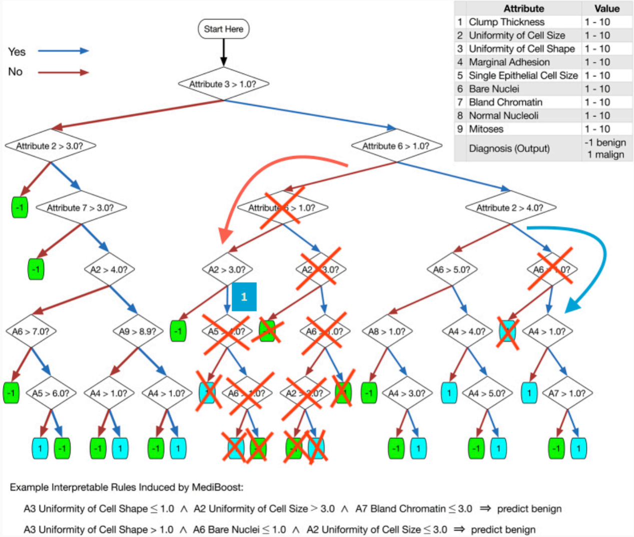
Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer:
Optimal Sparse Decision Trees.
NeurIPS 2019: 7265-7273

BTW, highly problematic decision trees also in precision medicine...



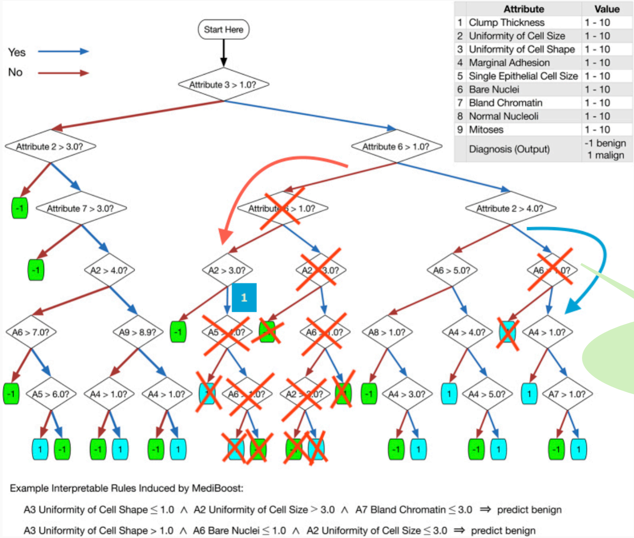
Source: G. Valdes, J.M. Luna, E. Eaton, C.B. Simone, L.H. Ungar, & T.D. Solberg. MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine. Scientific reports, 6(1):1-8, 2016.

BTW, highly problematic decision trees also in precision medicine...



Source: G. Valdes, J.M. Luna, E. Eaton, C.B. Simone, L.H. Ungar, & T.D. Solberg.
MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine.
Scientific reports, 6(1):1-8, 2016.

BTW, highly problematic decision trees also in precision medicine...



And **massive** path redundancy**!**

Source: G. Valdes, J.M. Luna, E. Eaton, C.B. Simone, L.H. Ungar, & T.D. Solberg.
MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine.
Scientific reports, 6(1):1-8, 2016.

And more comments...

- Previous slides: two examples of obviously buggy DTs

And more comments...

- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners

And more comments...

- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners
- Can one really trust the operation of more complex ML models, even those subject to extensive testing?

And more comments...

- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners
- Can one really trust the operation of more complex ML models, even those subject to extensive testing?
- And how to debug complex ML models if heuristic explanations are also incorrect (more later)?

And more comments...

- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners
- Can one really trust the operation of more complex ML models, even those subject to extensive testing?
- And how to debug complex ML models if heuristic explanations are also incorrect (more later)?
- **For trustworthy AI, there exists no alternative to rigorous logic-based explanations!**

Unit #04

(Efficient) Intractability in Symbolic XAI

An encoding for DLs – components

R_1:	IF	(τ_1)	THEN	d_1
R_2:	ELSE IF	(τ_2)	THEN	d_2
		$\ldots$		
R_j:	ELSE IF	(τ_j)	THEN	d_j
		$\ldots$		
R_n:	ELSE IF	(τ_n)	THEN	d_n
$\mathrm{R}_{\mathrm{DEF}}$:	ELSE		THEN	d_{n+1}

R_1:	IF	(τ_1)	THEN	d_1
R_2:	ELSE IF	(τ_2)	THEN	d_2
		$\ldots$		
R_j:	ELSE IF	(τ_j)	THEN	d_j
		$\ldots$		
R_n:	ELSE IF	(τ_n)	THEN	d_n
$\mathrm{R}_{\mathrm{DEF}}$:	ELSE		THEN	d_{n+1}

- 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 $1 \leqslant k < j$, with $e_k = 1$:

$$\left[f_j \leftrightarrow \left(t_j \land \bigwedge_{1 \leqslant k < j, e_k = 1} \neg t_k\right)\right]$$

An encoding for DLs – components

R_1:	IF	(τ_1)	THEN	d_1
R_2:	ELSE IF	(τ_2)	THEN	d_2
		$\ldots$		
R_j:	ELSE IF	(τ_j)	THEN	d_j
		$\ldots$		
R_n:	ELSE IF	(τ_n)	THEN	d_n
$\mathrm{R}_{\mathrm{DEF}}$:	ELSE		THEN	d_{n+1}

- 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 $1 \leqslant j \leqslant n$, to be consistent (i.e. some rule up to j with prediction other than c to fire):

$$\left(\bigvee_{1\leqslant j\leqslant n, e_j=0} f_j\right)$$

An encoding for DLs – components

R_1:	IF	(τ_1)	THEN	d_1
R_2:	ELSE IF	(τ_2)	THEN	d_2
		$\ldots$		
R_j:	ELSE IF	(τ_j)	THEN	d_j
		$\ldots$		
R_n:	ELSE IF	(τ_n)	THEN	d_n
$\mathrm{R}_{\mathrm{DEF}}$:	ELSE		THEN	d_{n+1}

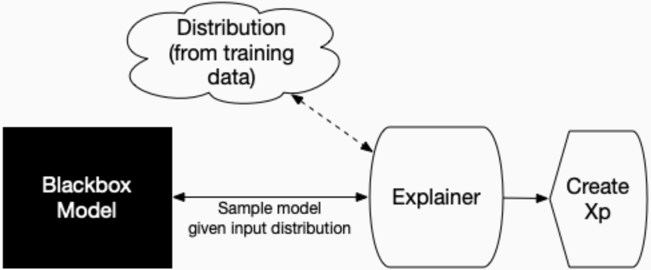
- The set of soft clauses is given by: $\mathcal{S} \triangleq \{(l_i), i = 1, \ldots, m\}$
- The set of hard clauses is given by:

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

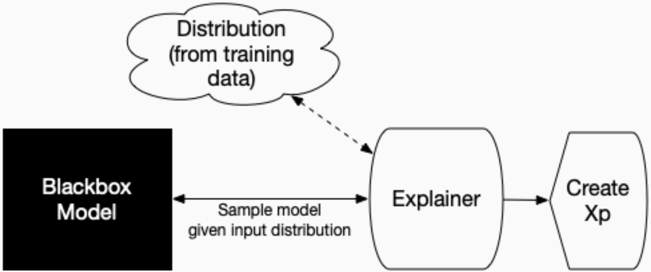
- $\mathcal{B} \cup \mathcal{S} \vDash \bot$
  - MUSes are AXp's & MCSes are CXp's

Outline – Unit #04

What is model-agnostic explainability?

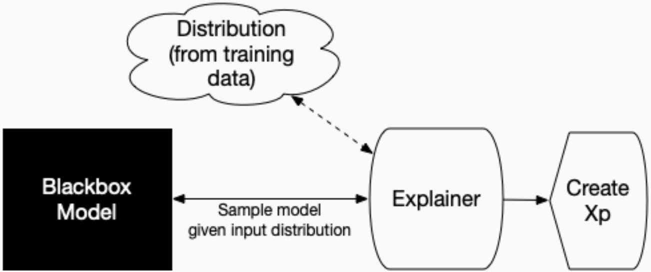


What is model-agnostic explainability?



- Wildly popular XAI approach [RSG16, LL17, RSG18]
  - **Feature attribution:** LIME, SHAP, ... [RSG16, LL17]
  - **Feature selection:** Anchors, ... [RSG18]

What is model-agnostic explainability?

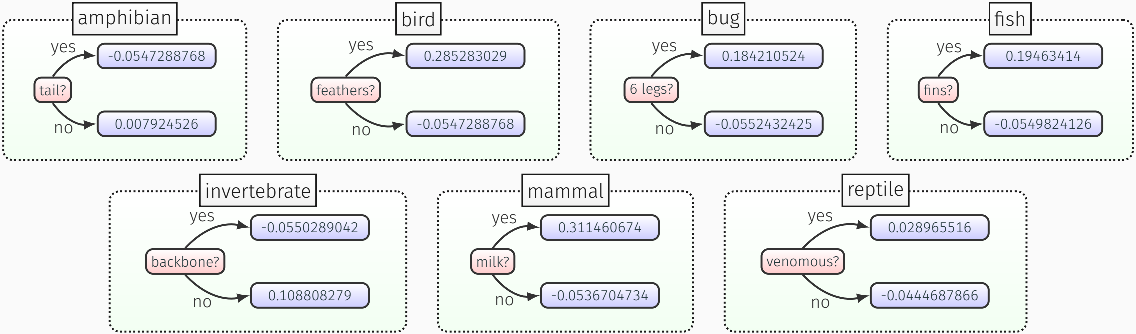


- Wildly popular XAI approach [RSG16, LL17, RSG18]
  - **Feature attribution:** LIME, SHAP, ... [RSG16, LL17]
  - **Feature selection:** Anchors, ... [RSG18]

- **Q:** Are model-agnostic explanations rigorous?

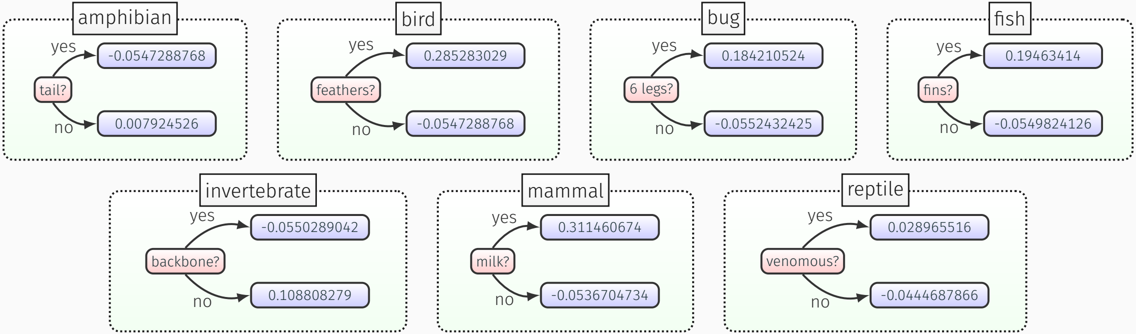
Easy to spot problems – BT for zoo dataset

[INM19b, Ign20]



Easy to spot problems – BT for zoo dataset

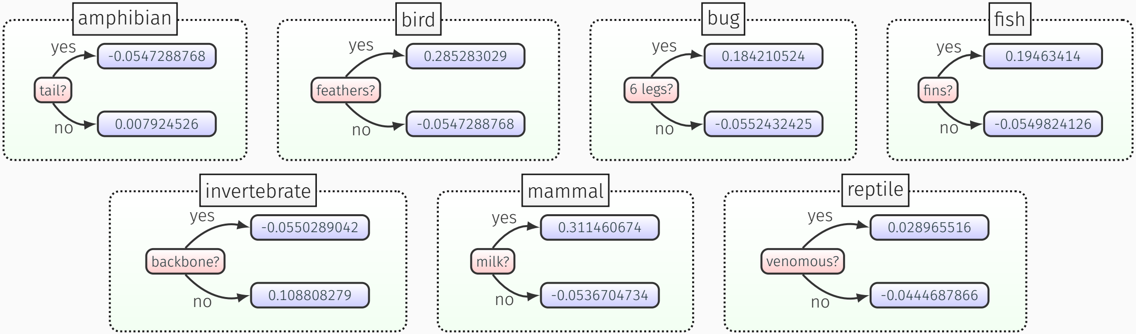
[INM19b, Ign20]



- Example instance:

IF $(\text{animal\_name} = \text{pitviper}) \land \neg\text{hair} \land \neg\text{feathers} \land \text{eggs} \land \neg\text{milk} \land \neg\text{airborne} \land \neg\text{aquatic} \land \text{predator} \land \neg\text{toothed} \land \text{backbone} \land \text{breathes} \land \text{venomous} \land \neg\text{fins} \land (\text{legs} = 0) \land \text{tail} \land \neg\text{domestic} \land \neg\text{catsize}$

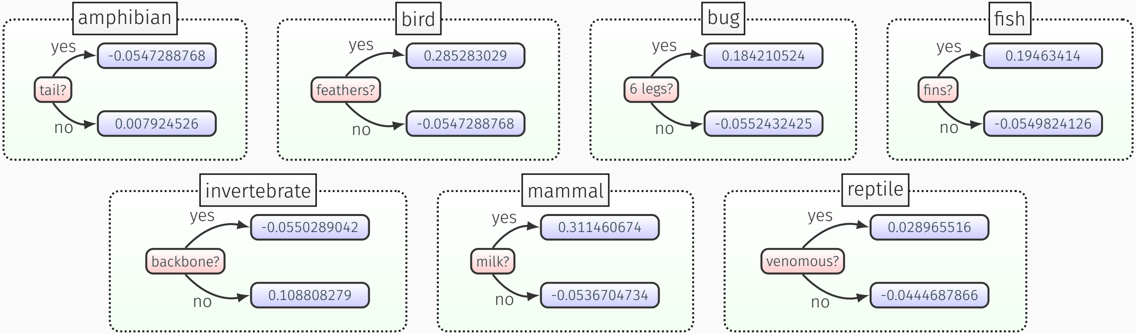
THEN $(\text{class} = \text{reptile})$



- Example instance (& Anchor picks): [RSG18]

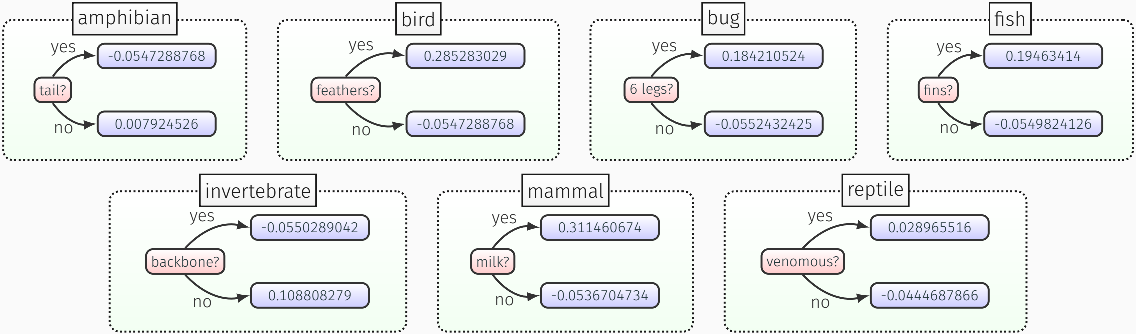
IF (animal_name = pitviper) $\wedge$ ***¬hair*** $\wedge$ ¬feathers $\wedge$ eggs $\wedge$ ***¬milk*** $\wedge$ ¬airborne $\wedge$ ¬aquatic $\wedge$ predator $\wedge$ ***¬toothed*** $\wedge$ backbone $\wedge$ breathes $\wedge$ venomous $\wedge$ ***¬fins*** $\wedge$ (legs = 0) $\wedge$ tail $\wedge$ ¬domestic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

THEN (class = reptile)



- Explanation obtained with Anchor: [RSG18]

IF	$\neg \textit{hair} \wedge \neg \textit{milk} \wedge \neg \textit{toothed} \wedge \neg \textit{fins}$
THEN	$(\text{class} = \text{reptile})$



- **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

Incorrect explanations:

Classifier for deciding bank loans

Model-agnostic explainers cannot be trusted

[INM19b]

Incorrect explanations:

Classifier for deciding bank loans

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

Model-agnostic explainers cannot be trusted

[INM19b]

Incorrect explanations:

Classifier for deciding bank loans

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

Explanation X: age $= 45$, salary $=$ 50K

Model-agnostic explainers cannot be trusted

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 $=$ 50K

And,

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

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

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 $=$ 50K

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 **!?**

How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*

How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*
- Let $\mathcal{X}$ be the features reported by model-agnostic tool

How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*
- Let $\mathcal{X}$ be the features reported by model-agnostic tool
- Check whether $\mathcal{X}$ is a (rigorous) (W)AXp:
  1. $\mathcal{X}$ is sufficient for prediction:

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

  2. And, $\mathcal{X}$ is subset-minimal:

$$\forall(t \in \mathcal{X}).\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in (\mathcal{X} \setminus \{t\})} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) \neq c)$$

  Depending on logic encoding used for classifier, different automated reasoners can be employed

How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*
- Let $\mathcal{X}$ be the features reported by model-agnostic tool
- Check whether $\mathcal{X}$ is a (rigorous) (W)AXp:
  1. $\mathcal{X}$ is sufficient for prediction:

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

  2. And, $\mathcal{X}$ is subset-minimal:

$$\forall(t \in \mathcal{X}).\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j\in(\mathcal{X}\setminus\{t\})}(x_j = v_j) \rightarrow (\kappa(\mathbf{x}) \neq c)$$

  Depending on logic encoding used for classifier, different automated reasoners can be employed

- Approach is bounded by scalability of rigorous explanations...

How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019 [INM19b, Ign20, YIS+23]

How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019 [INM19b, Ign20, YIS+23]
- Results for boosted trees, due to non-scalability with NNs [CG16]

How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019 [INM19b, Ign20, YIS+23]
- Results for boosted trees, due to non-scalability with NNs [CG16]
- Some results for Anchors [RSG18]

Dataset	% Incorrect	% Redundant	% Correct
adult	80.5%	1.6%	17.9%
lending	3.0%	0.0%	97.0%
rcdv	99.4%	0.4%	0.2%
compas	84.4%	1.7%	13.9%
german	99.7%	0.2%	0.1%

How serious is the lack of rigor of model-agnostic explanations?

- **Obs: Lack of rigor of model-agnostic explanations known since 2019** [INM19b, Ign20, YIS+23]
- Results for boosted trees, due to non-scalability with NNs [CG16]
- Some results for Anchors [RSG18]

Dataset	% Incorrect	% Redundant	% Correct
adult	80.5%	1.6%	17.9%
lending	3.0%	0.0%	97.0%
rcdv	99.4%	0.4%	0.2%
compas	84.4%	1.7%	13.9%
german	99.7%	0.2%	0.1%

- **Obs:** Results are not positive even if we count how often prediction changes [NSM+19]
  - In this case, BNNs were used, to allow for model counting...

How serious is the lack of rigor of model-agnostic explanations?

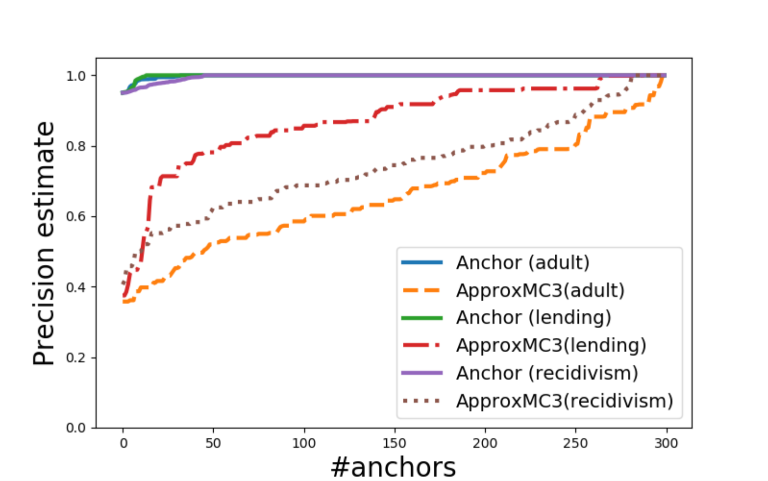
- **Obs: Lack of rigor of model-agnostic explanations known since 2019** [INM19b, Ign20, YIS+23]
- Results for boosted trees, due to non-scalability with NNs [CG16]
- Some results for Anchors [RSG18]

Dataset	% Incorrect	% Redundant	% Correct
adult	80.5%	1.6%	17.9%
lending	3.0%	0.0%	97.0%
rcdv	99.4%	0.4%	0.2%
compas	84.4%	1.7%	13.9%
german	99.7%	0.2%	0.1%

- **Obs:** Results are not positive even if we count how often prediction changes [NSM+19]
  - In this case, BNNs were used, to allow for model counting...
- For feature attribution we proposed different ways of assessing rigor [INM19b, NSM+19, Ign20, YIS+23]

Incorrect explanations are ubiquitous & likely...

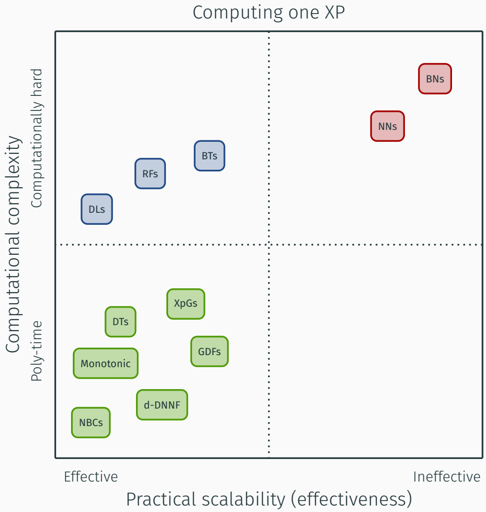
[NSM+19]



Outline – Unit #04

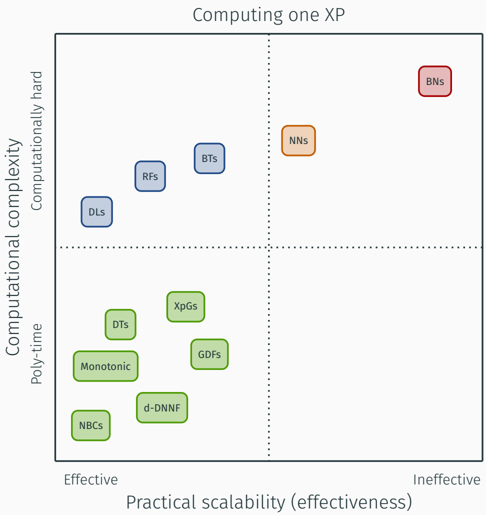
Efficacy map – progress until 2022

[MI22, Mar22, MS23]



[INM19b, 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) [INM19b, Ign20, IISMS22]
  - Comp. hard, and ineffective (hard in practice):
    - Neural networks (NNs) [INM19a]
    - Bayesian networks (BNs) [SCD18]



[INM19b, 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) [INM19b, Ign20, IISMS22]
  - Comp. hard, but some practical scalability:
    - Neural networks (NNs) [HM23]
  - Comp. hard, and ineffective (hard in practice):
    - Bayesian networks (BNs) [SCD18]

Results for RFs in 2021 (with SAT)

[IMS21]

Dataset	(#F	#C	#I)	RF			CNF		SAT oracle				AXp (**RFxpl**)				Anchor	
				D	#N	%A	#var	#cl	MxS	MxU	#S	#U	Mx	m	avg	%w	avg	%w
ann-thyroid	(21	3	718)	4	2192	98	17854	29230	0.12	0.15	2	18	0.36	0.05	0.13	96	0.32	4
appendicitis	(7	2	43)	6	1920	90	5181	10085	0.02	0.02	4	3	0.05	0.01	0.03	100	0.48	0
banknote	(4	2	138)	5	2772	97	8068	16776	0.01	0.01	2	2	0.03	0.02	0.02	100	0.19	0
biodegradation	(41	2	106	5	4420	88	11007	23842	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	11963	0.04	0.02	6	7	0.07	0.01	0.04	100	0.85	0
ionosphere	(34	2	71)	5	2096	87	7174	14406	0.02	0.02	22	11	0.11	0.02	0.03	100	12.43	0
karhunen	(64	10	200)	5	6198	91	36708	70224	1.06	1.41	35	29	14.64	0.65	2.78	100	28.15	0
letter	(16	26	398	8	44304	82	28991	68148	1.97	3.31	8	8	6.91	0.24	1.61	70	2.48	30
magic	(10	2	381)	6	9840	84	29530	66776	0.51	1.84	6	4	2.13	0.07	0.14	99	0.91	1
new-thyroid	(5	3	43)	5	1766	100	17443	28134	0.03	0.01	3	2	0.08	0.03	0.05	100	0.36	0
pendigits	(16	10	220)	6	12004	95	30522	59922	2.40	1.32	10	6	4.11	0.14	0.94	96	3.68	4
ring	(20	2	740	6	6188	89	19114	42362	0.27	0.44	11	9	1.25	0.05	0.25	92	7.25	8
segmentation	(19	7	42)	4	1966	90	21288	35381	0.11	0.17	8	10	0.53	0.11	0.31	100	4.13	0
shuttle	(9	7	116	3	1460	99	18669	29478	0.11	0.08	2	7	0.34	0.05	0.14	99	0.42	1
sonar	(60	2	42)	5	2614	88	9938	20537	0.04	0.06	36	24	0.43	0.04	0.09	100	23.02	0
spectf	(44	2	54)	5	2306	88	6707	13449	0.07	0.06	20	24	0.34	0.02	0.07	100	8.12	0
texture	(40	11	550)	5	5724	87	34293	64187	0.79	0.63	23	17	3.24	0.19	0.93	100	28.13	0
twonorm	(20	2	740	5	6266	94	21198	46901	0.08	0.08	12	8	0.28	0.06	0.10	100	5.73	0
vowel	(13	11	198)	6	10176	90	44523	88696	1.66	2.11	8	5	4.52	0.15	1.15	66	1.67	34
waveform-40	(40	3	500	5	6232	83	30438	58380	0.50	0.86	15	25	7.07	0.11	0.88	100	11.93	0
wpbc	(33	2	78)	5	2432	76	9078	18675	1.00	1.53	20	13	5.33	0.03	0.65	79	3.91	21

Results for NNs in 2019 (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

Results for NNs in 2019 (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

Results for NNs in 2019 (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	10.72

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

Results for NNs in 2023 (using Marabou [KHI$^+$19])

DNN	points	|AXp|	#Calls	Time	#TO	|AXp|	#Calls	Time	#TO
			$\epsilon = 0.1$				$\epsilon = 0.05$		
ACASXU_1_5	#1	3	5	185.9	0	2	5	113.8	0
	#2	2	5	273.8	0	1	5	33.2	0
	#3	0	5	714.2	0	0	5	4.3	0
ACASXU_3_1	#1	0	5	2219.3	0	0	5	14.2	0
	#2	2	5	4263.5	1	0	5	1853.1	0
	#3	1	5	581.8	0	0	5	355.9	0
ACASXU_3_2	#1	3	5	13739.3	2	1	5	6890.1	1
	#2	3	5	226.4	0	2	5	125.1	0
	#3	2	5	1740.6	0	2	5	173.6	0
ACASXU_3_5	#1	4	5	43.6	0	2	5	59.4	0
	#2	3	5	5039.4	0	2	5	4303.8	1
	#3	2	5	5574.9	1	2	5	2660.3	0
ACASXU_3_6	#1	1	5	6225.0	1	0	5	51.0	0
	#2	3	5	4957.2	1	2	5	1897.3	0
	#3	1	5	196.1	0	1	5	919.2	0
ACASXU_3_7	#1	3	5	6256.2	0	4	5	26.9	0
	#2	4	5	311.3	0	1	5	6958.6	1
	#3	2	5	7756.5	1	1	5	7807.6	1
ACASXU_4_1	#1	2	5	12413.0	2	1	5	5090.5	1
	#2	1	5	5035.1	1	0	5	2335.6	0
	#3	4	5	1237.3	0	4	5	1143.4	0
ACASXU_4_2	#1	4	5	15.9	0	4	5	12.1	0
	#2	3	5	1507.6	0	1	5	111.3	0
	#3	2	5	5641.6	2	0	5	1639.1	0

Results for NNs in 2023 (using Marabou [KHI+19])

[HM23]

DNN	points	|AXp|	#Calls	Time	#TO	|AXp|	#Calls	Time	#TO
			$\epsilon = 0.1$				$\epsilon = 0.05$		
ACASXU_1_5	#1	3	5	185.9	0	2	5	113.8	0
	#2	2	5	273.8	0	1	5	33.2	0
	#3	0	5	714.2	0	0	5	4.3	0
ACASXU_3_1	#1	0	5	2219.3	0	0	5	14.2	0
	#2	2	5	4263.5	1	0	5	1853.1	0
	#3	1	5	581.8	0	0	5	355.9	0
ACASXU_3_2	#1	3	5	13739.3	2	1	5	6890.1	1
	#2	3	5	226.4	0	2	5	125.1	0
	#3	2	5	1740.6	0	2	5	173.6	0
ACASXU_3_5	#1	4	5	43.6	0	2	5	59.4	0
	#2	3	5	5039.4	0	2	5	4303.8	1
	#3	2	5	5574.9	1	2	5	2660.3	0
ACASXU_3_6	#1	1	5	6225.0	1	0	5	51.0	0
	#2	3	5	4957.2	1	2	5	1897.3	0
	#3	1	5	196.1	0	1	5	919.2	0
ACASXU_3_7	#1	3	5	6256.2	0	4	5	26.9	0
	#2	4	5	311.3	0	1	5	6958.6	1
	#3	2	5	7756.5	1	1	5	7807.6	1
ACASXU_4_1	#1	2	5	12413.0	2	1	5	5090.5	1
	#2	1	5	5035.1	1	0	5	2335.6	0
	#3	4	5	1237.3	0	4	5	1143.4	0
ACASXU_4_2	#1	4	5	15.9	0	4	5	12.1	0
	#2	3	5	1507.6	0	1	5	111.3	0
	#3	2	5	5641.6	2	0	5	1639.1	0

Scales to a few **hundred** neurons

More recent results (from 2024)...

[IHM+24]

Model	Deletion							SwiftXplain						
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8

More recent results (from 2024)... [IHM+24]

Model	Deletion							SwiftXplain						
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8

Scales to **tens of thousands** of neurons!

More recent results (from 2024)...

[IHM+24]

Model	Deletion							SwiftXplain						
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8

Scales to **tens of thousands** of neurons!

Largest for MNIST: **10142** neurons
Largest for GSTRB: **94308** neurons

Unit #05

Queries in Symbolic XAI

Outline – Unit #05

Enumeration of Explanations

Feature Necessity & Relevancy

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]
  - Recall: 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]
  - Recall: for DTs, enumeration of CXp's is in P [HIIM21, IIM22]
- There are algorithms for direct enumeration of 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]
  - Recall: for DTs, enumeration of CXp's is in P [HIIM21, IIM22]
- There are algorithms for direct enumeration of 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]
  - Recall: for DTs, enumeration of CXp's is in P [HIIM21, IIM22]
- There are algorithms for direct enumeration of 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]
  - Recall: for DTs, enumeration of CXp's is in P [HIIM21, IIM22]
- There are algorithms for direct enumeration of 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

Recall computing one AXp/CXp – oneXP

Input: Predicate $\mathbb{P}$, parameterized by $\mathcal{T}$, $\mathcal{M}$

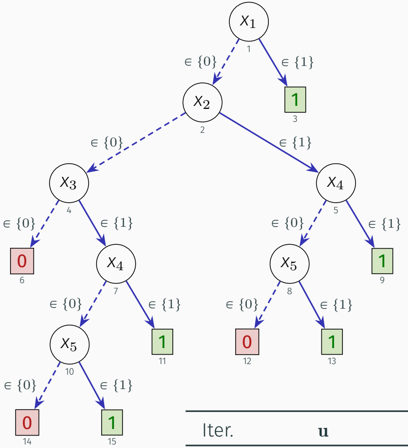
Output: One XP $\mathcal{S}$

procedure oneXP($\mathbb{P}$)
    $\mathcal{S} \leftarrow \mathcal{F}$ ▷ Initialization: $\mathbb{P}(\mathcal{S})$ holds
    for $i \in \mathcal{F}$ **do** ▷ Loop invariant: $\mathbb{P}(\mathcal{S})$ holds
        if $\mathbb{P}(\mathcal{S}\backslash\{i\})$ **then**
            $\mathcal{S} \leftarrow \mathcal{S}\backslash\{i\}$ ▷ Update $\mathcal{S}$ only if $\mathbb{P}(\mathcal{S}\backslash\{i\})$ holds
    return $\mathcal{S}$ ▷ Returned set $\mathcal{S}$: $\mathbb{P}(\mathcal{S})$ holds

Generic oracle-based enumeration 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 $U = \{u_1, \ldots, u_m\}$; initially no constraints
repeat
  $(\text{outc}, \mathbf{u}) \leftarrow \text{SAT}(\mathcal{H})$ ▷ Use SAT oracle to pick assignment s.t. known constraints in $\mathcal{H}$
  if $\text{outc} = \text{true}$ **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 $\mathbb{P}_{\text{cxp}}(\mathcal{U}; \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$ **then** ▷ $\mathcal{U} = \mathcal{F} \setminus \mathcal{S} \supseteq$ some CXp
      $\mathcal{P} \leftarrow \text{oneXP}(\mathcal{U}; \mathbb{P}_{\text{cxp}}, \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$
      $\text{reportCXp}(\mathcal{P})$
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\vee_{i \in \mathcal{P}} \neg u_i)\}$ ▷ $\mathcal{P} \subseteq \mathcal{U}$: one 1-value variable must be 0 in future iterations
    else ▷ $\mathcal{S} \supseteq$ some AXp
      $\mathcal{P} \leftarrow \text{oneXP}(\mathcal{S}; \mathbb{P}_{\text{axp}}, \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$
      $\text{reportAXp}(\mathcal{P})$
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\vee_{i \in \mathcal{P}} u_i)\}$ ▷ $\mathcal{P} \subseteq \mathcal{S}$: one 0-value variable must be 1 in future iterations
until $\text{outc} = \text{false}$



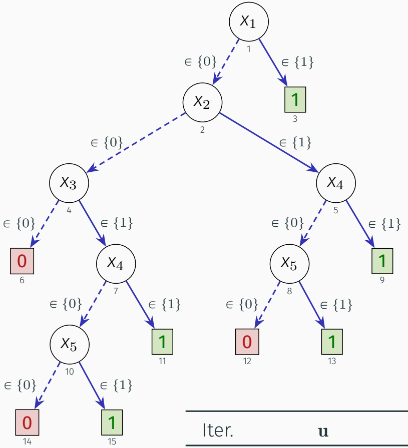
- Instance: $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

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	Resulting $\mathcal{H}$
1	$(1,1,1,1,1)$	$\emptyset$	1	–	$\{3\}$	$(\neg u_3)$	$\{(\neg u_3)\}$
2	$(1,1,0,1,1)$	$\{3\}$	1	–	$\{5\}$	$(\neg u_5)$	$\{(\neg u_3),(\neg u_5)\}$
3	$(1,1,0,1,0)$	$\{3,5\}$	0	$\{3,5\}$	–	$(u_3 \vee u_5)$	$\{(\neg u_3),(\neg u_5),(u_3 \vee u_5)\}$
5	[outc = **false**]	–	–	–	–	–	$\{(\neg u_3),(\neg u_5),(u_3 \vee u_5)\}$

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	Resulting $\mathcal{H}$
1	$(0,0,0,0,0)$	$\{1,2,3,4,5\}$	0	$\{3,5\}$	–	$(u_3 \lor u_5)$	$\{(u_3 \lor u_5)\}$
2	$(0,0,1,0,0)$	$\{1,2,4,5\}$	1	–	$\{3\}$	$(\neg u_3)$	$\{(u_3 \lor u_5), (\neg u_3)\}$
3	$(0,0,1,0,1)$	$\{1,2,4\}$	1	–	$\{5\}$	$(\neg u_5)$	$\{(u_3 \lor u_5), (\neg u_3), (\neg u_5)\}$
5	[outc = **false**]	–	–	–	–	–	$\{(u_3 \lor u_5), (\neg u_3), (\neg u_5)\}$

DTs admit more efficient algorithms

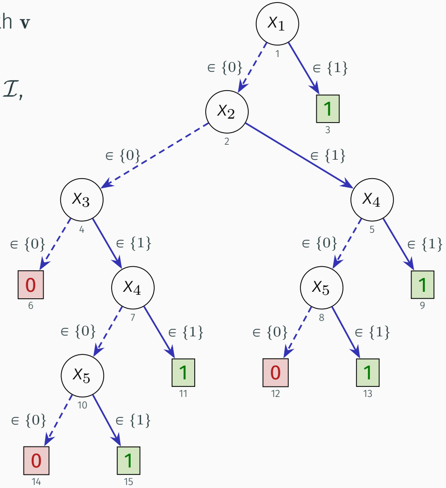
- Recall:
  - Given instance $(\mathbf{v}, c)$, create set $\mathcal{I}$
  - For each path P_k with prediction $d \neq c$:
    - Let I_k denote the features with literals inconsistent with $\mathbf{v}$
    - Add I_k to $\mathcal{I}$
  - Remove from $\mathcal{I}$ the sets that have a proper subset in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

DTs admit more efficient algorithms

- Recall:
  - 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 $\mathcal{I}$, and duplicates
- $\mathcal{I}$ 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

- Recall:
  - 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 $\mathcal{I}$, and duplicates
- $\mathcal{I}$ 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
- Example:
  - $I_1 = \{3\}$
  - $I_2 = \{5\}$
  - $I_3 = \{2, 5\}$
  - $\therefore$ keep I_1 an I_2
  - AXp's: MHSes yield $\{\{3, 5\}\}$



Outline – Unit #05

Enumeration of Explanations

Feature Necessity & Relevancy

(Conditioned) Classifier Decision Problem ((C)CDP)

[HCM+23]

(Conditioned) Classifier Decision Problem ((C)CDP)

[HCM+23]

- Given $c \in \mathcal{K}$, CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

(Conditioned) Classifier Decision Problem ((C)CDP)

[HCM+23]

- Given $c \in \mathcal{K}$, CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

- Given $\mathcal{S} \subseteq \mathcal{F}$, instance $(\mathbf{v}, c)$, CCDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{i \in \mathcal{S}}(x_i = v_i) \wedge (\kappa(\mathbf{x}) = c)$$

(Conditioned) Classifier Decision Problem ((C)CDP)

[HCM+23]

- Given $c \in \mathcal{K}$, CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

- Given $\mathcal{S} \subseteq \mathcal{F}$, instance $(\mathbf{v}, c)$, CCDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{i \in \mathcal{S}} (x_i = v_i) \wedge (\kappa(\mathbf{x}) = c)$$

- **Claim:** (C)CDP is in polynomial-time for DTs, decision graphs, monotonic classifiers, among others

(Conditioned) Classifier Decision Problem ((C)CDP)

[HCM+23]

- Given $c \in \mathcal{K}$, CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

- Given $\mathcal{S} \subseteq \mathcal{F}$, instance $(\mathbf{v}, c)$, CCDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).\bigwedge_{i \in \mathcal{S}}(x_i = v_i) \land (\kappa(\mathbf{x}) = c)$$

- **Claim:** (C)CDP is in polynomial-time for DTs, decision graphs, monotonic classifiers, among others
- **Claim:** (C)CDP is in NP-complete for DLs, RFs, BTs, boolean NNs and BNNs

Feature necessity

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

$\mathbb{A}$: encodes the set of all **irreducible rules** for prediction c given $\mathbf{v}$

Feature necessity

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} \coloneqq \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} \coloneqq \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

$\mathbb{A}$: encodes the set of all **irreducible rules** for prediction c given $\mathbf{v}$

- Features common to all AXps in $\mathbb{A}$ and all CXps in $\mathbb{C}$:

$$N_{\mathbb{A}} \coloneqq \bigcap_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$N_{\mathbb{C}} \coloneqq \bigcap_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

Feature necessity

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} \coloneqq \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$
$$\mathbb{C} \coloneqq \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

$\mathbb{A}$: encodes the set of all **irreducible rules** for prediction c given $\mathbf{v}$

- Features common to all AXps in $\mathbb{A}$ and all CXps in $\mathbb{C}$:

$$N_{\mathbb{A}} \coloneqq \bigcap_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$
$$N_{\mathbb{C}} \coloneqq \bigcap_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- $N_{\mathbb{A}}$ and $N_{\mathbb{C}}$ need not be equal
  - $\mathbb{A} = \{\{1\}, \{2, 3\}\}$

Feature necessity

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

$\mathbb{A}$: encodes the set of all **irreducible rules** for prediction c given $\mathbf{v}$

- Features common to all AXps in $\mathbb{A}$ and all CXps in $\mathbb{C}$:

$$N_{\mathbb{A}} := \bigcap_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$N_{\mathbb{C}} := \bigcap_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- $N_{\mathbb{A}}$ and $N_{\mathbb{C}}$ need not be equal
  - $\mathbb{A} = \{\{1\}, \{2, 3\}\}$
- A feature i is necessary for abductive explanations (AXp-necessary) if $i \in N_{\mathbb{A}}$

Feature necessity

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} \coloneqq \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} \coloneqq \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

$\mathbb{A}$: encodes the set of all **irreducible rules** for prediction c given $\mathbf{v}$

- Features common to all AXps in $\mathbb{A}$ and all CXps in $\mathbb{C}$:

$$N_{\mathbb{A}} \coloneqq \bigcap\nolimits_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$N_{\mathbb{C}} \coloneqq \bigcap\nolimits_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- $N_{\mathbb{A}}$ and $N_{\mathbb{C}}$ need not be equal
  - $\mathbb{A} = \{\{1\}, \{2, 3\}\}$
- A feature i is necessary for abductive explanations (AXp-necessary) if $i \in N_{\mathbb{A}}$
- A feature i is necessary for contrastive explanations (CXp-necessary) if $i \in N_{\mathbb{C}}$

More on feature necessity

[HCM+23]

More on feature necessity

[HCM+23]

- **Claim #01:** $t \in \mathcal{F}$ is AXp-necessary iff $\{t\}$ is a CXp

More on feature necessity

[HCM+23]

- **Claim #01:** $t \in \mathcal{F}$ is AXp-necessary iff $\{t\}$ is a CXp
- **Claim #02:** $t \in \mathcal{F}$ is CXp-necessary iff $\{t\}$ is a AXp

More on feature necessity

[HCM+23]

- **Claim #01:** $t \in \mathcal{F}$ is AXp-necessary iff $\{t\}$ is a CXp
- **Claim #02:** $t \in \mathcal{F}$ is CXp-necessary iff $\{t\}$ is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others

More on feature necessity

[HCM+23]

- **Claim #01:** $t \in \mathcal{F}$ is AXp-necessary iff $\{t\}$ is a CXp
- **Claim #02:** $t \in \mathcal{F}$ is CXp-necessary iff $\{t\}$ is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others
- **Claim #04:** AXp-necessity of $t \in \mathcal{F}$ is in P if t has a domain size which is polynomially-bounded on instance size

More on feature necessity

[HCM+23]

- **Claim #01:** $t \in \mathcal{F}$ is AXp-necessary iff $\{t\}$ is a CXp
- **Claim #02:** $t \in \mathcal{F}$ is CXp-necessary iff $\{t\}$ is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others
- **Claim #04:** AXp-necessity of $t \in \mathcal{F}$ is in P if t has a domain size which is polynomially-bounded on instance size
  - This holds for any classifier!

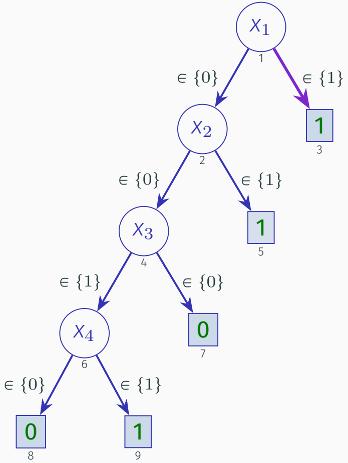
More on feature necessity

[HCM+23]

- **Claim #01:** $t \in \mathcal{F}$ is AXp-necessary iff $\{t\}$ is a CXp
- **Claim #02:** $t \in \mathcal{F}$ is CXp-necessary iff $\{t\}$ is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others
- **Claim #04:** AXp-necessity of $t \in \mathcal{F}$ is in P if t has a domain size which is polynomially-bounded on instance size
  - This holds for any classifier!
  - Let $\mathbf{u}$ be obtained from $\mathbf{v}$ by replacing the constant v_t by some variable $u_t \in \mathcal{D}_t$
  - Feature t is AXp-necessary if $\kappa(\mathbf{u}) \neq \kappa(\mathbf{v})$ for some value $u_t \in \mathcal{D}_t$

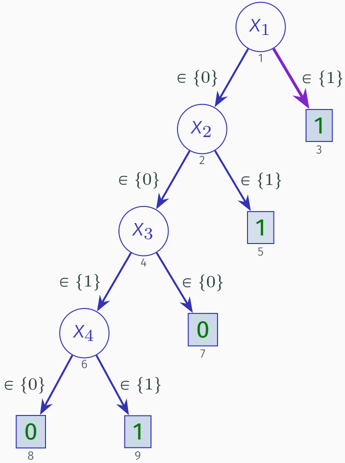
An example

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



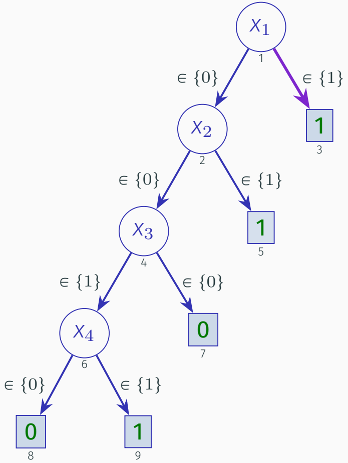
An example

- Instance $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?



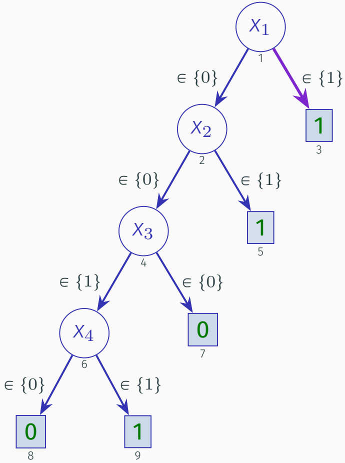
An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$?



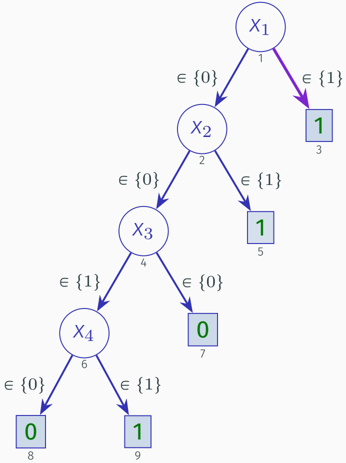
An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0),0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1,0,0,0) \neq \kappa(0,0,0,0)$?
  - Yes! Thus, feature 1 is AXp-necessary



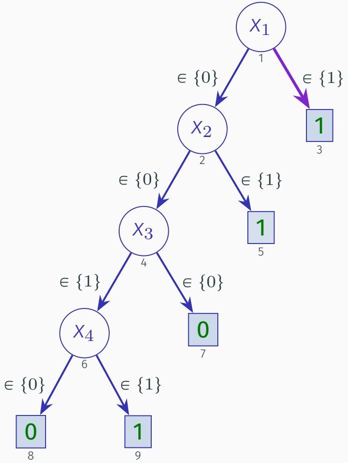
An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1,0,0,0) \neq \kappa(0,0,0,0)$?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?



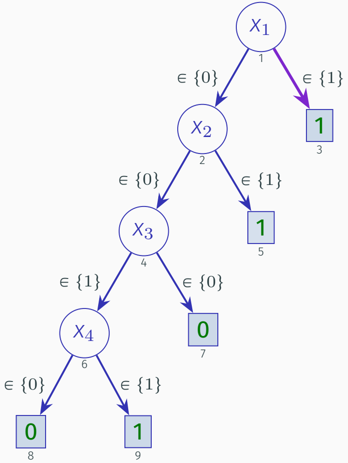
An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0),0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1,0,0,0) \neq \kappa(0,0,0,0)$?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0,0,u_3,0) \neq \kappa(0,0,0,0)$?



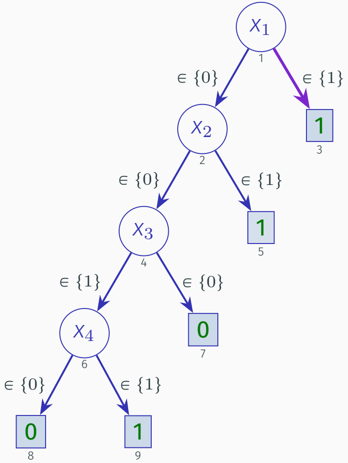
An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0),0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1,0,0,0) \neq \kappa(0,0,0,0)$?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0,0,u_3,0) \neq \kappa(0,0,0,0)$?
  - No! Thus, feature 3 is not AXp-necessary



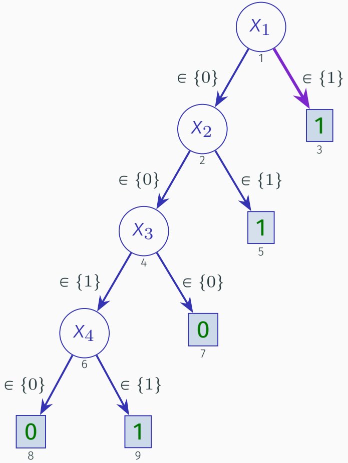
An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1, 0, 0, 0) \neq \kappa(0,0,0,0)$?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0, 0, u_3, 0) \neq \kappa(0,0,0,0)$?
  - No! Thus, feature 3 is not AXp-necessary
- Confirmation:
  - CXps:
  - AXps:



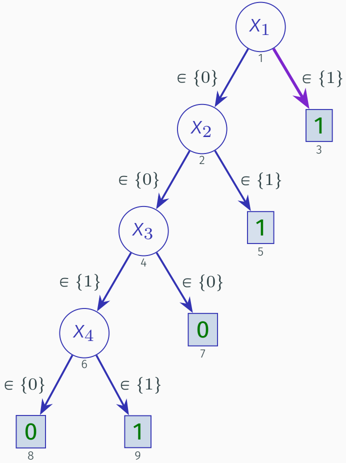
An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0),0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1,0,0,0) \neq \kappa(0,0,0,0)$?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0,0,u_3,0) \neq \kappa(0,0,0,0)$?
  - No! Thus, feature 3 is not AXp-necessary
- Confirmation:
  - CXps: $\{\{1\},\{2\},\{3,4\}\}$
  - AXps:



An example

- Instance $(\mathbf{v}, c) = ((0,0,0,0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$?
  - No! Thus, feature 3 is not AXp-necessary
- Confirmation:
  - CXps: $\{\{1\}, \{2\}, \{3, 4\}\}$
  - AXps: $\{\{1, 2, 3\}, \{1, 2, 4\}\}$



Feature relevancy

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

Feature relevancy

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in $\mathbb{A}$ and in some CXp in $\mathbb{C}$:

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

Feature relevancy

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in $\mathbb{A}$ and in some CXp in $\mathbb{C}$:

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- **Claim**: $F_{\mathbb{A}} = F_{\mathbb{C}}$
  - I.e. a feature exists in some AXp iff it exists in some CXp

Feature relevancy

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in $\mathbb{A}$ and in some CXp in $\mathbb{C}$:

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- **Claim**: $F_{\mathbb{A}} = F_{\mathbb{C}}$
  - I.e. a feature exists in some AXp iff it exists in some CXp
- A feature $i \in \mathcal{F}$ is **relevant** if $i \in F_{\mathbb{A}}$ (and so, if $i \in F_{\mathbb{C}}$)
  - A feature is relevant if it is included in some AXp (or CXp)

Feature relevancy

- Consider instance $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \,|\, \mathsf{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in $\mathbb{A}$ and in some CXp in $\mathbb{C}$:

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- **Claim**: $F_{\mathbb{A}} = F_{\mathbb{C}}$
  - I.e. a feature exists in some AXp iff it exists in some CXp
- A feature $i \in \mathcal{F}$ is **relevant** if $i \in F_{\mathbb{A}}$ (and so, if $i \in F_{\mathbb{C}}$)
  - A feature is relevant if it is included in some AXp (or CXp)
- A feature $i \in \mathcal{F}$ is **irrelevant** if $i \notin F_{\mathbb{A}}$ (and so, if $i \notin F_{\mathbb{C}}$)
  - A feature is irrelevant if it is not included in any AXp (or CXp)

An example

- Consider the classifier:

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

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

An example

- Consider the classifier:

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

    - $(\mathbf{v}, c) = ((0, 0, 0, 1), 1)$
- $\mathbb{A} = \{\{4\}\} = \mathbb{C}$
    - Why?
        - If 4 fixed, then prediction *must* be 1
        - If 4 is allowed to change, then prediction changes
        - Values of 1, 2, 3 not used to fix/change the prediction

An example

- Consider the classifier:

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

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

- $\mathbb{A} = \{\{4\}\} = \mathbb{C}$
  - Why?
    - If 4 fixed, then prediction *must* be 1
    - If 4 is allowed to change, then prediction changes
    - Values of 1, 2, 3 not used to fix/change the prediction

- Feature 4 is relevant, since it is included in one (and the only) AXp/CXp

An example

- Consider the classifier:

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

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

- $\mathbb{A} = \{\{4\}\} = \mathbb{C}$
  - Why?
    - If 4 fixed, then prediction *must* be 1
    - If 4 is allowed to change, then prediction changes
    - Values of 1, 2, 3 not used to fix/change the prediction

- Feature 4 is relevant, since it is included in one (and the only) AXp/CXp

- Features 1, 2, 3 are irrelevant, since there are not included in any AXp/CXp
  - Obs: **irrelevant features are absolutely unimportant!**

    We could propose some other explanation by adding features 1, 2 or 3 to AXp $\{4\}$, but prediction would remain unchanged for **any** value assigned to those features
    - And we aim for **irreducibility** (**Occam's razor is a mainstay of AI/ML**)

Deciding feature relevancy

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,
    - $\mathcal{P}$ is a WAXp (i.e. universal quantification)

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,
    - $\mathcal{P}$ is a WAXp (i.e. universal quantification)
    - $\mathcal{P}\setminus\{t\}$ is a not a WAXp (i.e. universal quantification again)

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,
    - $\mathcal{P}$ is a WAXp (i.e. universal quantification)
    - $\mathcal{P}\backslash\{t\}$ is a not a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with $\exists\forall$ alternation

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,
    - $\mathcal{P}$ is a WAXp (i.e. universal quantification)
    - $\mathcal{P}\setminus\{t\}$ is a not a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with $\exists\forall$ alternation
- For DTs, deciding feature relevancy is in P; Why?

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,
    - $\mathcal{P}$ is a WAXp (i.e. universal quantification)
    - $\mathcal{P}\backslash\{t\}$ is a not a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with $\exists\forall$ alternation
- For DTs, deciding feature relevancy is in P; Why?
  - **Obs:** We know that $F_{\mathbb{A}} = F_{\mathbb{C}}$; thus

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^{P} – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,
    - $\mathcal{P}$ is a WAXp (i.e. universal quantification)
    - $\mathcal{P}\backslash\{t\}$ is a not a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with $\exists\forall$ alternation
- For DTs, deciding feature relevancy is in P; Why?
  - **Obs:** We know that $F_{\mathbb{A}} = F_{\mathbb{C}}$; thus
    - Computing all CXps in polynomial-time decides feature relevancy

Deciding feature relevancy

- Deciding feature relevancy is in Σ_2^P – intuition:
  - Pick a set of features $\mathcal{P}$ containing t (i.e. existential quantification), such that,
    - $\mathcal{P}$ is a WAXp (i.e. universal quantification)
    - $\mathcal{P}\backslash\{t\}$ is a not a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with $\exists\forall$ alternation
- For DTs, deciding feature relevancy is in P; Why?
  - **Obs:** We know that $F_{\mathbb{A}} = F_{\mathbb{C}}$; thus
    - Computing all CXps in polynomial-time decides feature relevancy
- General case: best solution is to exploit **abstraction refinement**

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If $\mathrm{WAXp}(\mathcal{X})$ holds and $\mathrm{WAXp}(\mathcal{X} \setminus \{t\})$ does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If WAXp($\mathcal{X}$) holds and WAXp($\mathcal{X} \setminus \{t\}$) does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

Proof:

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If WAXp($\mathcal{X}$) holds and WAXp($\mathcal{X} \backslash \{t\}$) does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

Proof:

- Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If $\text{WAXp}(\mathcal{X})$ holds and $\text{WAXp}(\mathcal{X} \setminus \{t\})$ does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

Proof:

- Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.
- Then $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$.

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If WAXp($\mathcal{X}$) holds and WAXp($\mathcal{X} \setminus \{t\}$) does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

Proof:

- Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.
- Then $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$.
- But then, by monotonicity, WAXp($\mathcal{X} \setminus \{t\}$) must hold (i.e. any superset of $\mathcal{Z}$ is a weak AXp); hence a contradiction.

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If WAXp($\mathcal{X}$) holds and WAXp($\mathcal{X} \setminus \{t\}$) does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

  Proof:
  - Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.
  - Then $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$.
  - But then, by monotonicity, WAXp($\mathcal{X} \setminus \{t\}$) must hold (i.e. any superset of $\mathcal{Z}$ is a weak AXp); hence a contradiction.

- Approach:

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If WAXp($\mathcal{X}$) holds and WAXp($\mathcal{X}\setminus\{t\}$) does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

  Proof:

  - Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.
  - Then $\mathcal{Z} \subseteq \mathcal{X}\setminus\{t\}$.
  - But then, by monotonicity, WAXp($\mathcal{X}\setminus\{t\}$) must hold (i.e. any superset of $\mathcal{Z}$ is a weak AXp); hence a contradiction.

- Approach:
  - Repeatedly guess weak WAXp candidates $\mathcal{X}$, with $t \in \mathcal{X}$ [e.g. use SAT oracle]

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If $\mathrm{WAXp}(\mathcal{X})$ holds and $\mathrm{WAXp}(\mathcal{X} \setminus \{t\})$ does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

  Proof:

  - Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.
  - Then $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$.
  - But then, by monotonicity, $\mathrm{WAXp}(\mathcal{X} \setminus \{t\})$ must hold (i.e. any superset of $\mathcal{Z}$ is a weak AXp); hence a contradiction.

- Approach:
  - Repeatedly guess weak WAXp candidates $\mathcal{X}$, with $t \in \mathcal{X}$ [e.g. use SAT oracle]
  - Check that WAXp condition holds for $\mathcal{X}$: $\mathrm{WAXp}(\mathcal{X})$; and [e.g. use WAXp oracle]

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If $\text{WAXp}(\mathcal{X})$ holds and $\text{WAXp}(\mathcal{X} \setminus \{t\})$ does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

Proof:

- Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.
- Then $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$.
- But then, by monotonicity, $\text{WAXp}(\mathcal{X} \setminus \{t\})$ must hold (i.e. any superset of $\mathcal{Z}$ is a weak AXp); hence a contradiction.

- Approach:
  - Repeatedly guess weak WAXp candidates $\mathcal{X}$, with $t \in \mathcal{X}$ [e.g. use SAT oracle]
  - Check that WAXp condition holds for $\mathcal{X}$: $\text{WAXp}(\mathcal{X})$; and [e.g. use WAXp oracle]
  - Check that WAXp condition fails for $\mathcal{X} \setminus \{t\}$: $\neg\text{WAXp}(\mathcal{X} \setminus \{t\})$ [e.g. use WAXp oracle]

Abstraction refinement for feature relevancy

- **Claim**: $\mathcal{X} \subseteq \mathcal{F}$ and $t \in \mathcal{X}$. If $\text{WAXp}(\mathcal{X})$ holds and $\text{WAXp}(\mathcal{X} \setminus \{t\})$ does not hold, then **any** AXp $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ must contain feature t.

  Proof:

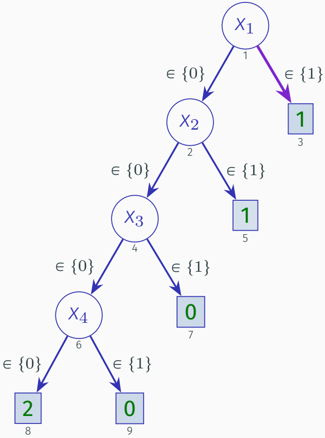
  - Let $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$ be an AXp such that $t \notin \mathcal{Z}$.
  - Then $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$.
  - But then, by monotonicity, $\text{WAXp}(\mathcal{X} \setminus \{t\})$ must hold (i.e. any superset of $\mathcal{Z}$ is a weak AXp); hence a contradiction.

- Approach:
  - Repeatedly guess weak WAXp candidates $\mathcal{X}$, with $t \in \mathcal{X}$ [e.g. use SAT oracle]
  - Check that WAXp condition holds for $\mathcal{X}$: $\text{WAXp}(\mathcal{X})$; and [e.g. use WAXp oracle]
  - Check that WAXp condition fails for $\mathcal{X} \setminus \{t\}$: $\neg\text{WAXp}(\mathcal{X} \setminus \{t\})$ [e.g. use WAXp oracle]
  - Block counterexamples in both cases

A general abstraction refinement algorithm

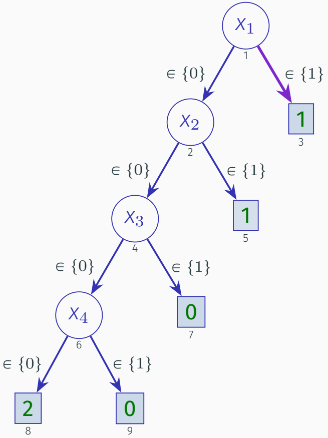
Input: Instance $\mathbf{v}$, Target Feature t; Feature Set $\mathcal{F}$, Classifier κ

function FRPCGR($\mathbf{v}, t; \mathcal{F}, \kappa$)
  $\mathcal{H} \leftarrow \emptyset$ $\rhd$ $\mathcal{H}$ overapproximates the subsets of $\mathcal{F}$ that do not contain an AXp containing t
  repeat
    $(\text{outc}, \mathbf{s}) \leftarrow \text{SAT}(\mathcal{H}, s_t)$ $\rhd$ **Use SAT oracle to pick candidate WAXp containing** t
    if outc $=$ **true then**
      $\mathcal{P} \leftarrow \{i \in \mathcal{F} \mid s_i = 1\}$ $\rhd$ Set $\mathcal{P}$ is the candidate WAXp, and $t \in \mathcal{P}$
      $\mathcal{D} \leftarrow \{i \in \mathcal{F} \mid s_i = 0\}$ $\rhd$ Set $\mathcal{D}$ contains the features not included in $\mathcal{P}$
      if $\neg\text{WAXp}(\mathcal{P})$ **then** $\rhd$ Is $\mathcal{P}$ not a WAXp?
        $\mathcal{H} \leftarrow \mathcal{H} \cup \text{newPosCl}(\mathcal{D}; t, \kappa)$ $\rhd$ $\mathcal{P}$ is *not* a WAXp; must pick some non-picked feature
      else $\rhd$ $\mathcal{P}$ is a WAXp
        if $\neg\text{WAXp}(\mathcal{P} \setminus \{t\})$ **then** $\rhd$ $\mathcal{P}$ without t not a WAXp?
          reportWeakAXp($\mathcal{P}$) $\rhd$ Feature t is included in any AXp $\mathcal{X} \subseteq \mathcal{P}$
          return true
        $\mathcal{H} \leftarrow \mathcal{H} \cup \text{newNegCl}(\mathcal{P}; t, \kappa)$ $\rhd$ $\text{WAXp}(\mathcal{P} \setminus \{t\})$ holds; some feature in $\mathcal{P}$ must not be picked
  until outc $=$ **false**
  return false $\rhd$ If $\mathcal{H}$ becomes inconsistent, then there is no AXp that contains t



- Instance: $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Is $t = 1$ relevant?

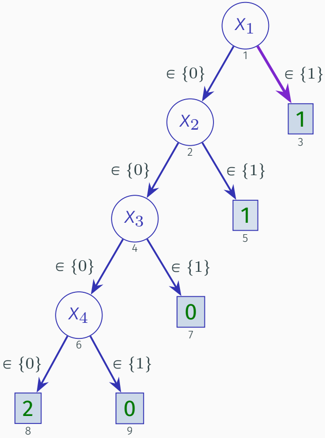
An example: feature relevancy for DT, using abstraction refinement



- Instance: $(\mathbf{v}, c) = ((1,1,1,1),1)$
- Is $t=1$ relevant?

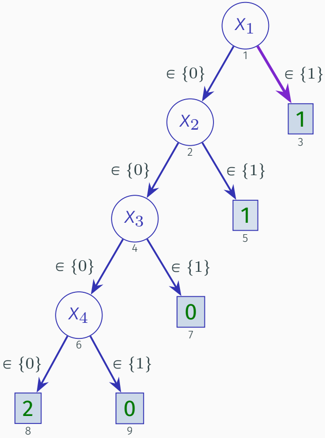
		$t=1$			
$\mathbf{s}$	$\mathcal{P}$	WAXp($\mathcal{P}$)	WAXp($\mathcal{P}\setminus\{t\}$)	Return?	Clause
$(1,1,1,1)$	$\{1,2,3,4\}$	✓	✓	——	$(\neg u_2 \vee \neg u_3 \vee \neg u_4)$
$(1,1,0,1)$	$\{1,2,4\}$	✓	✓	——	$(\neg u_2 \vee \neg u_4)$
$(1,1,0,0)$	$\{1,2\}$	✓	✓	——	$(\neg u_2)$
$(1,0,0,0)$	$\{1\}$	✓	✗	true	——

Another example



- Instance: $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Is $t = 4$ relevant?

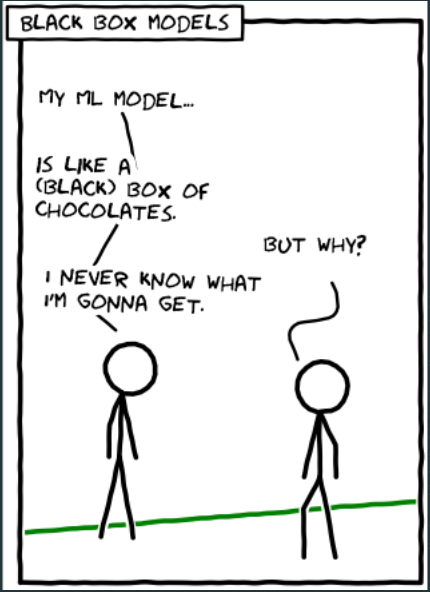
Another example



- Instance: $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Is $t = 4$ relevant?

		$t = 4$			
$\mathbf{s}$	$\mathcal{P}$	WAXp($\mathcal{P}$)	WAXp($\mathcal{P} \setminus \{t\}$)	Return?	Clause
$(1, 1, 1, 1)$	$\{1, 2, 3, 4\}$	✓	✓	——	$(\neg u_1 \vee \neg u_2 \vee \neg u_3)$
$(1, 1, 0, 1)$	$\{1, 2, 4\}$	✓	✓	——	$(\neg u_1 \vee \neg u_2)$
$(1, 0, 0, 1)$	$\{1, 4\}$	✓	✓	——	$(\neg u_1)$
$(0, 1, 0, 1)$	$\{2, 4\}$	✓	✓	——	$(\neg u_2)$
$(0, 0, 0, 1)$	$\{4\}$	✗	—	——	$(u_1 \vee u_2 \vee u_3)$
$(0, 0, 1, 1)$	$\{3, 4\}$	✗	—	——	$(u_1 \vee u_2)$
[outc = **false**]	——	—	—	**false**	——

Questions?



https://arxiv.org/abs/1901.01686 & http://cmx.io/edit/

References i

[CG16] Tianqi Chen and Carlos Guestrin.
XGBoost: A scalable tree boosting system.
In *KDD*, pages 785–794, 2016.

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

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

[HCM+23] Xuanxiang Huang, Martin C. Cooper, António Morgado, Jordi Planes, and João Marques-Silva.
Feature necessity & relevancy in ML classifier explanations.
In *TACAS*, pages 167–186, 2023.

[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`.

[HM23] Xuanxiang Huang and João Marques-Silva.
From robustness to explainability and back again.
CoRR, abs/2306.03048, 2023.

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

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

[IHM+24] Yacine Izza, Xuanxiang Huang, Antonio Morgado, Jordi Planes, Alexey Ignatiev, and Joao Marques-Silva.
Distance-restricted explanations: Theoretical underpinnings & efficient implementation.
CoRR, abs/2405.08297, 2024.

[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., 75:261–321, 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.

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

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

[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 validating, repairing and refining heuristic ML explanations.
CoRR, abs/1907.02509, 2019.

[KHI+19] Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljic, David L. Dill, Mykel J. Kochenderfer, and Clark W. Barrett.
The marabou framework for verification and analysis of deep neural networks.
In *CAV*, pages 443–452, 2019.

[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.

[Mar22] João Marques-Silva.
Logic-based explainability in machine learning.
In *Reasoning Web*, pages 24–104, 2022.

[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.

[MI22] João Marques-Silva and Alexey Ignatiev.
Delivering trustworthy AI through formal XAI.
In *AAAI*, pages 12342–12350, 2022.

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

[MS23] Joao Marques-Silva.
Disproving XAI myths with formal methods – initial results.
In *ICECCS*, 2023.

[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.

[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.

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

[YIS+23] Jinqiang Yu, Alexey Ignatiev, Peter J. Stuckey, Nina Narodytska, and Joao Marques-Silva.
Eliminating the impossible, whatever remains must be true: On extracting and applying background knowledge in the context of formal explanations.
In *AAAI*, 2023.