SAT-Based Rigorous Explanations for Decision Lists

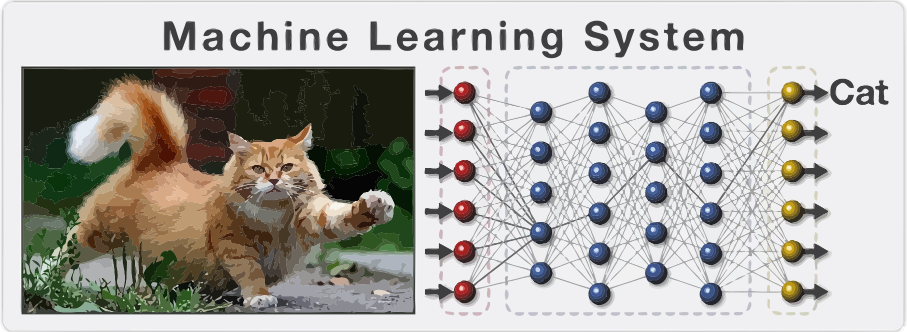
Alexey Ignatiev[1] and Joao Marques-Silva[2]

July 7, 2021 | **SAT**

[1] Monash University, Melbourne, Australia
[2] IRIT, CNRS, Toulouse, France

eXplainable AI

eXplainable AI

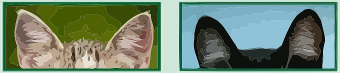


This is a cat.

Current Explanation

This is a cat:

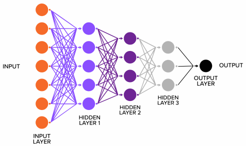
- It has fur, whiskers, and claws.
- It has this feature:



XAI Explanation

©DARPA

Why? Status quo...



	A parrot	Machine learning algorithm
Learns random phrases	✅	✅
Doesn’t understand s**t about what it learns	✅	✅
Occasionally speaks nonsense	✅	✅

©Internet

interpretable ML models

e.g. decision trees, lists, sets

interpretable ML models

e.g. decision trees, lists, sets

posthoc explanation of ML models "on the fly"

rule-based models

rule-based models

⬇

“*transparent*” and easy to interpret

rule-based models

⬇

“*transparent*” and easy to interpret

⬇

come in handy in XAI

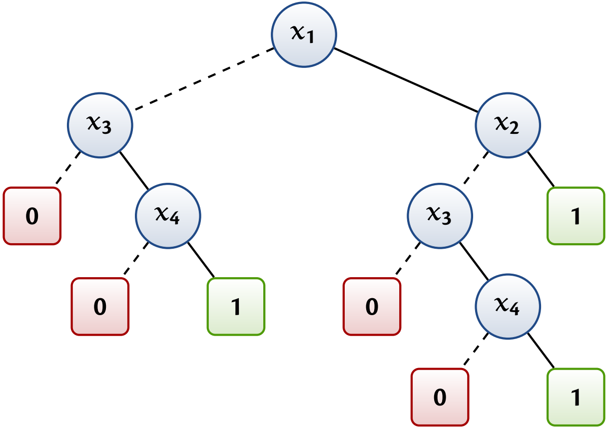
but...

Decision trees aren't interpretable

$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$

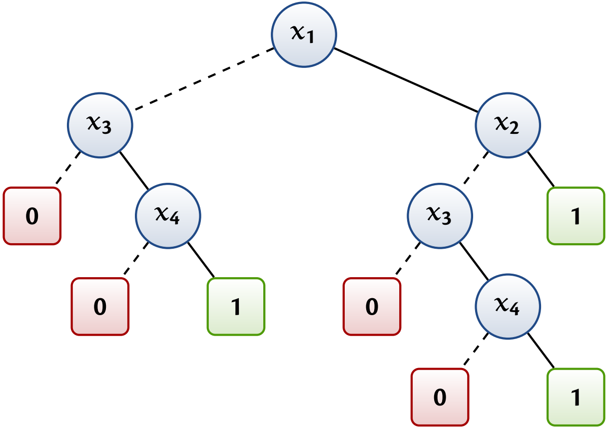
Decision trees **aren't interpretable**

$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$



Decision trees **aren't interpretable**

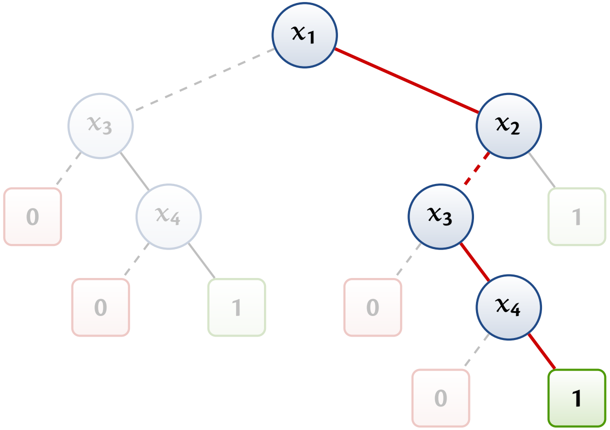
$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$



instance $v = (1, 0, 1, 1)$ — 4 literals in the path

Decision trees **aren't interpretable**

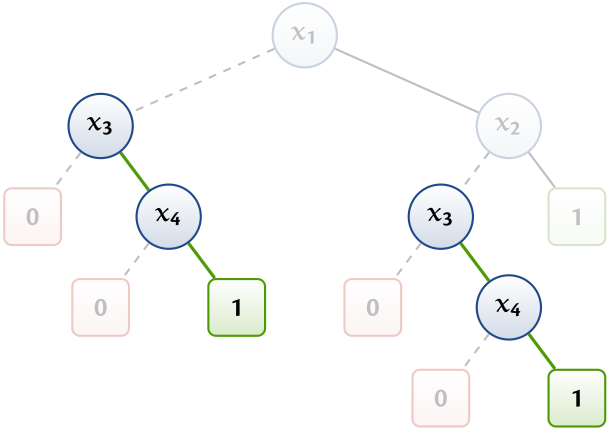
$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$



instance $v = (1, 0, 1, 1)$ — 4 literals in the path

Decision trees aren't interpretable

$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$



instance $v = (1, 0, 1, 1)$ — 4 literals in the path

actual explanation $x_3 = 1 \wedge x_4 = 1$ — 2 literals

DL explainability

classifier $\tau : \mathbb{F} \rightarrow \mathcal{K}$, **instance** $\mathbf{v}$ **s.t.** $\tau(\mathbf{v}) = c$

classifier $\tau : \mathbb{F} \rightarrow \mathcal{K}$, **instance** $\mathbf{v}$ **s.t.** $\tau(\mathbf{v}) = c$

---

abductive explanation $\mathcal{X}$

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

classifier $\tau : \mathbb{F} \rightarrow \mathcal{K}$, **instance** $\mathbf{v}$ **s.t.** $\tau(\mathbf{v}) = \mathbf{c}$

---

abductive explanation $\mathcal{X}$

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

---

contrastive explanation $\mathcal{Y}$

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

DL example and duality

$$\mathbb{F} = \{\mathbf{0}, \mathbf{1}, \mathbf{2}\}^{5} \qquad \mathcal{K} = \{\ominus, \oplus\}$$

DL example and duality

$$\mathbb{F} = \{0, 1, 2\}^5 \qquad \mathcal{K} = \{\ominus, \oplus\}$$

R_0:	IF	$x_1 = 1 \wedge x_2 = 1$	THEN $\ominus$
R_1:	ELSE IF	$x_3 \neq 1$	THEN $\oplus$
R_{DEF}:	ELSE		THEN $\ominus$

DL example and duality

$$\mathbb{F} = \{0, 1, 2\}^5 \qquad \mathcal{K} = \{\ominus, \oplus\}$$

R_0:	IF	$x_1 = 1 \wedge x_2 = 1$	THEN $\ominus$
R_1:	ELSE IF	$x_3 \neq 1$	THEN $\oplus$
R_{DEF}:	ELSE		THEN $\ominus$

observe $\tau(1, 1, 1, 1, 1) = \ominus$

⬇

DL example and duality

$$\mathbb{F} = \{0, 1, 2\}^5 \qquad \mathcal{K} = \{\ominus, \oplus\}$$

R_0:	IF	$x_1 = 1 \wedge x_2 = 1$	THEN $\ominus$
R_1:	ELSE IF	$x_3 \neq 1$	THEN $\oplus$
R_{DEF}:	ELSE		THEN $\ominus$

observe $\tau(1, 1, 1, 1, 1) = \ominus$

AXps $\mathbb{X} = \{\{1, 2\}, \{3\}\}$

DL example and duality

$$\mathbb{F} = \{0, 1, 2\}^5 \qquad \mathcal{K} = \{\ominus, \oplus\}$$

R_0:	IF	$x_1 = 1 \wedge x_2 = 1$	THEN $\ominus$
R_1:	ELSE IF	$x_3 \neq 1$	THEN $\oplus$
R_{DEF}:	ELSE		THEN $\ominus$

observe $\tau(1, 1, 1, 1, 1) = \ominus$

AXps $\mathbb{X} = \{\{1, 2\}, \{3\}\}$

CXps $\mathbb{Y} = \{\{1, 3\}, \{2, 3\}\}$

DL example and duality

$\mathbb{F} = \{0, 1, 2\}^5 \qquad \mathcal{K} = \{\ominus, \oplus\}$

R_0:	IF	$x_1 = 1 \wedge x_2 = 1$	THEN $\ominus$
R_1:	ELSE IF	$x_3 \neq 1$	THEN $\oplus$
R_{DEF}:	ELSE		THEN $\ominus$

observe $\tau(1, 1, 1, 1, 1) = \ominus$

AXps $\mathbb{X} = \{\{1, 2\}, \{3\}\}$

CXps $\mathbb{Y} = \{\{1, 3\}, \{2, 3\}\}$

minimal hitting set duality!

Interpretability issue – just like with DTs

$$f(x_1,\ldots,x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$

R_0:	IF	$x_1 = 0 \wedge x_3 = 0$	THEN $f = 0$
R_1:	ELSE IF	$x_1 = 0 \wedge x_3 = 1 \wedge x_4 = 0$	THEN $f = 0$
R_2:	ELSE IF	$x_1 = 0 \wedge x_3 = 1 \wedge x_4 = 1$	THEN $f = 1$
R_3:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 0$	THEN $f = 0$
R_4:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 1 \wedge x_4 = 0$	THEN $f = 0$
R_5:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 1 \wedge x_4 = 1$	THEN $f = 1$
R_6:	ELSE IF	$x_1 = 1 \wedge x_2 = 1$	THEN $f = 1$
R_{DEF}:	ELSE		THEN $f = 1$

Interpretability issue – just like with DTs

$$f(x_1, \ldots, x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$

R_0:	IF	$x_1 = 0 \wedge x_3 = 0$	THEN $f = 0$
R_1:	ELSE IF	$x_1 = 0 \wedge x_3 = 1 \wedge x_4 = 0$	THEN $f = 0$
R_2:	ELSE IF	$x_1 = 0 \wedge x_3 = 1 \wedge x_4 = 1$	THEN $f = 1$
R_3:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 0$	THEN $f = 0$
R_4:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 1 \wedge x_4 = 0$	THEN $f = 0$
R_5:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 1 \wedge x_4 = 1$	THEN $f = 1$
R_6:	ELSE IF	$x_1 = 1 \wedge x_2 = 1$	THEN $f = 1$
R_{DEF}:	ELSE		THEN $f = 1$

instance $v = (1, 0, 1, 1)$ — rule R_5 fires the prediction

Interpretability issue – just like with DTs

$$f(x_1,\ldots,x_n) = \bigvee_{i=1}^{n/2} x_{2i-1} \wedge x_{2i}, \text{ with } n = 4$$

R_0:	IF	$x_1 = 0 \wedge x_3 = 0$	THEN $f = 0$
R_1:	ELSE IF	$x_1 = 0 \wedge x_3 = 1 \wedge x_4 = 0$	THEN $f = 0$
R_2:	ELSE IF	$x_1 = 0 \wedge x_3 = 1 \wedge x_4 = 1$	THEN $f = 1$
R_3:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 0$	THEN $f = 0$
R_4:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 1 \wedge x_4 = 0$	THEN $f = 0$
R_5:	ELSE IF	$x_1 = 1 \wedge x_2 = 0 \wedge x_3 = 1 \wedge x_4 = 1$	THEN $f = 1$
R_6:	ELSE IF	$x_1 = 1 \wedge x_2 = 1$	THEN $f = 1$
R_{DEF}:	ELSE		THEN $f = 1$

instance $\mathbf{v} = (1, 0, 1, 1)$ — rule R_5 fires the prediction

actual AXp — $x_3 = 1 \wedge x_4 = 1$ — 2 literals

Are DLs hard to explain?

SAT query:

SAT query:

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

SAT query:

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

IM query:

SAT query:

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

IM query:

$$\forall(\mathbf{x} \in \mathbb{F}).\ \ \rho(\mathbf{x}) \rightarrow \tau(\mathbf{x}) = c$$

1. DLSAT is $\mathcal{NP}$-complete

Are DLs hard to explain? Results.

1. DLSAT is $\mathcal{NP}$-complete

2. No polytime algorithm for DLIM unless $\mathcal{P} = \mathcal{NP}$

Are DLs hard to explain? Results.

1. DLSAT is $\mathcal{NP}$-complete

2. No polytime algorithm for DLIM unless $\mathcal{P} = \mathcal{NP}$

see paper for details!

decision lists:

finding an AXp is not polytime unless $\mathrm{P} = \mathrm{NP}$

decision lists:

finding an AXp is not polytime unless $P = NP$

decision sets:

finding an AXp is D^P-complete

decision lists:

finding an AXp is not polytime unless $\mathrm{P} = \mathrm{NP}$

decision sets:

finding an AXp is D^{P}-complete

in contrast to decision trees!

Propositional encoding

Propositional encoding

(see paper for notation and details)

(see paper for notation and details)

---

rule j $\in \mathfrak{R}$ fires:

(see paper for notation and details)

---

rule $j \in \mathfrak{R}$ fires:

$$\varphi(j) \triangleq \left(\bigwedge_{k \in \mathfrak{R},\ \mathfrak{o}(k) < \mathfrak{o}(j)} \neg \mathfrak{l}(k) \right) \wedge \mathfrak{l}(j)$$

(see paper for notation and details)

rule $j \in \mathfrak{R}$ fires:

$$\varphi(j) \triangleq \left(\bigwedge_{k \in \mathfrak{R},\ \mathfrak{o}(k) < \mathfrak{o}(j)} \neg \mathfrak{l}(k) \right) \wedge \mathfrak{l}(j)$$

unsatisfiable $\mathcal{S} \wedge \mathcal{H}$ **s.t.**

(see paper for notation and details)

rule $j \in \mathfrak{R}$ fires:

$$\varphi(j) \triangleq \left(\bigwedge_{k \in \mathfrak{R},\ \mathfrak{o}(k) < \mathfrak{o}(j)} \neg \mathfrak{l}(k) \right) \wedge \mathfrak{l}(j)$$

unsatisfiable $\mathcal{S} \wedge \mathcal{H}$ s.t.

$$\mathcal{S} \triangleq I_v$$

(see paper for notation and details)

rule $j \in \mathfrak{R}$ fires:

$$\varphi(j) \triangleq \left(\bigwedge_{k \in \mathfrak{R},\ \mathfrak{o}(k) < \mathfrak{o}(j)} \neg \mathfrak{l}(k) \right) \wedge \mathfrak{l}(j)$$

unsatisfiable $\mathcal{S} \wedge \mathcal{H}$ s.t.

$$\mathcal{S} \triangleq I_v \qquad \mathcal{H} \triangleq \bigvee_{j \in \mathfrak{R},\ \mathfrak{c}(j) = \mathfrak{c}(i)} \varphi(j)$$

(see paper for notation and details)

rule $j \in \mathfrak{R}$ fires:

$$\varphi(j) \triangleq \left(\bigwedge_{k \in \mathfrak{R},\, \mathfrak{o}(k) < \mathfrak{o}(j)} \neg \mathfrak{l}(k) \right) \wedge \mathfrak{l}(j)$$

unsatisfiable $\mathcal{S} \wedge \mathcal{H}$ s.t.

$$\mathcal{S} \triangleq I_v \qquad \mathcal{H} \triangleq \bigvee_{j \in \mathfrak{R},\, \mathfrak{c}(j) = \mathfrak{c}(i)} \varphi(j)$$

instance v, prediction $\mathfrak{c}(i)$:

(see paper for notation and details)

rule $j \in \mathfrak{R}$ fires:

$$\varphi(j) \triangleq \left(\bigwedge_{k \in \mathfrak{R},\, \mathfrak{o}(k) < \mathfrak{o}(j)} \neg \mathfrak{l}(k) \right) \wedge \mathfrak{l}(j)$$

unsatisfiable $\mathcal{S} \wedge \mathcal{H}$ s.t.

$$\mathcal{S} \triangleq I_v \qquad \mathcal{H} \triangleq \bigvee_{j \in \mathfrak{R},\, \mathfrak{c}(j) = \mathfrak{c}(i)} \varphi(j)$$

instance v, prediction $\mathfrak{c}(i)$:

AXps are MUSes

Propositional encoding

(see paper for notation and details)

---

rule $j \in \mathfrak{R}$ fires:

$$\varphi(j) \triangleq \left(\bigwedge_{k \in \mathfrak{R},\, \mathfrak{o}(k) < \mathfrak{o}(j)} \neg \mathfrak{l}(k) \right) \wedge \mathfrak{l}(j)$$

---

unsatisfiable $\mathcal{S} \wedge \mathcal{H}$ s.t.

$$\mathcal{S} \triangleq I_v \qquad \mathcal{H} \triangleq \bigvee_{j \in \mathfrak{R},\, \mathfrak{c}(j) = \mathfrak{c}(i)} \varphi(j)$$

---

instance v, prediction $\mathfrak{c}(i)$:

AXps are MUSes **CXps are MCSes**

Experimental results

Experimental setup

- **machine configuration:**
  - Quad-Core Intel Core i5-8259U 2.30GHz, with 16GByte RAM

Experimental setup

- **machine configuration:**
  - Quad-Core Intel Core i5-8259U 2.30GHz, with 16GByte RAM
  - running macOS Big Sur 11.2.3

Experimental setup

- **machine configuration:**
  - Quad-Core Intel Core i5-8259U 2.30GHz, with 16GByte RAM
  - running macOS Big Sur 11.2.3
  - 1800s timeout + 4GB memout

Experimental setup

- **machine configuration:**
    - Quad-Core Intel Core i5-8259U 2.30GHz, with 16GByte RAM
    - running macOS Big Sur 11.2.3
    - 1800s timeout + 4GB memout
- **UCI MLR + PMLB + ML explainability and fairness**

Experimental setup

- **machine configuration:**
  - Quad-Core Intel Core i5-8259U 2.30GHz, with 16GByte RAM
  - running macOS Big Sur 11.2.3
  - 1800s timeout + 4GB memout
- **UCI MLR + PMLB + ML explainability and fairness**
  - **360 benchmarks** in total (72 datasets $\times$ 5-cross validation)

Experimental setup

- **machine configuration:**
  - Quad-Core Intel Core i5-8259U 2.30GHz, with 16GByte RAM
  - running macOS Big Sur 11.2.3
  - 1800s timeout + 4GB memout
- **UCI MLR + PMLB + ML explainability and fairness**
  - **360 benchmarks** in total (72 datasets × 5-cross validation)
  - **CN2 decision lists:**
    - `https://orangedatamining.com/`
    - **6–2055** rules
    - **6–6754** literals (total)

Experimental setup

- **machine configuration:**
  - Quad-Core Intel Core i5-8259U 2.30GHz, with 16GByte RAM
  - running macOS Big Sur 11.2.3
  - 1800s timeout + 4GB memout
- **UCI MLR + PMLB + ML explainability and fairness**
  - **360 benchmarks** in total (72 datasets × 5-cross validation)
  - **CN2 decision lists:**
    - `https://orangedatamining.com/`
    - **6–2055** rules
    - **6–6754** literals (total)
  - **SAT encoding:**
    - **7–15340** variables
    - **9–3932987** clauses

Experimental setup

- **Python + PySAT:**
  - Glucose3 SAT solver
  - incremental oracle calls

Experimental setup

- **Python + PySAT:**
  - Glucose3 SAT solver
  - incremental oracle calls
  - `https://github.com/alexeyignatiev/xdl-tool`

Experimental setup

- **Python + PySAT:**
  - Glucose3 SAT solver
  - incremental oracle calls
  - `https://github.com/alexeyignatiev/xdl-tool`
- **direct CXp enumeration:**
  - LBX-like MCS enumeration
  - “Clause D” heuristic

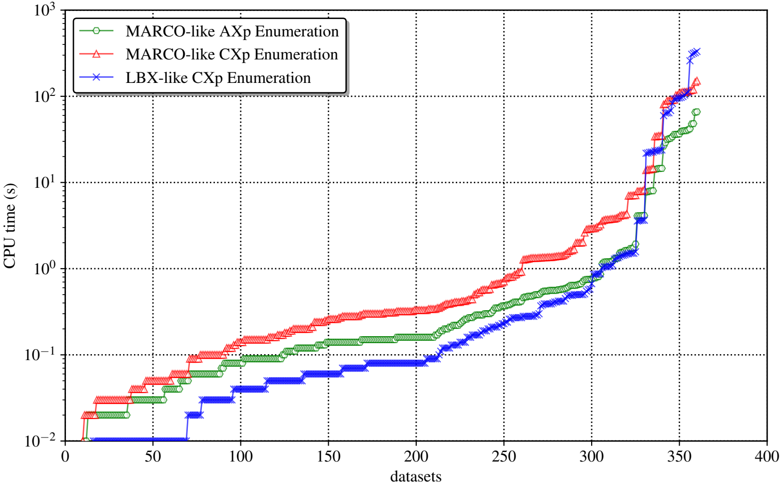
Experimental setup

- **Python + PySAT:**
  - Glucose3 SAT solver
  - incremental oracle calls
  - `https://github.com/alexeyignatiev/xdl-tool`
- **direct CXp enumeration:**
  - LBX-like MCS enumeration
  - "Clause D" heuristic
- **MARCO-like XP enumeration:**
  - *targets* either AXps or CXps
  - *computes* both AXps and CXps

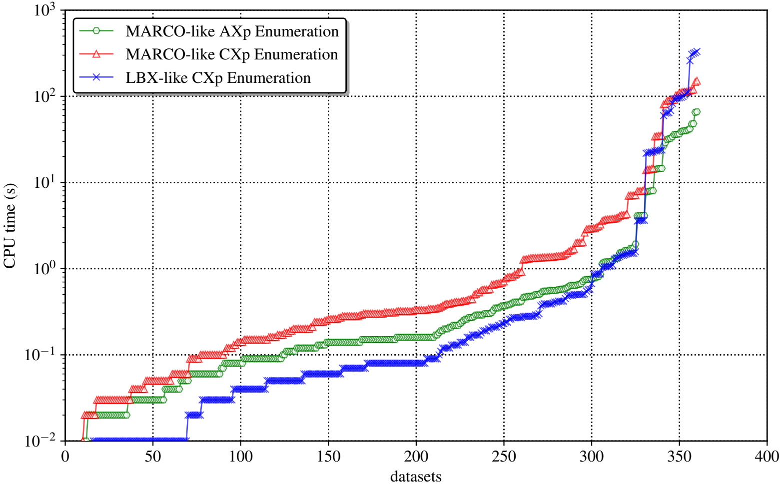
Experimental setup

- **Python + PySAT:**
  - Glucose3 SAT solver
  - incremental oracle calls
  - `https://github.com/alexeyignatiev/xdl-tool`
- **direct CXp enumeration:**
  - LBX-like MCS enumeration
  - "Clause D" heuristic
- **MARCO-like XP enumeration:**
  - *targets* either AXps or CXps
  - *computes* both AXps and CXps
  - **minimum hitting sets** — RC2 MaxSAT
  - **XP reduction** — deletion-based linear search

Results – raw performance

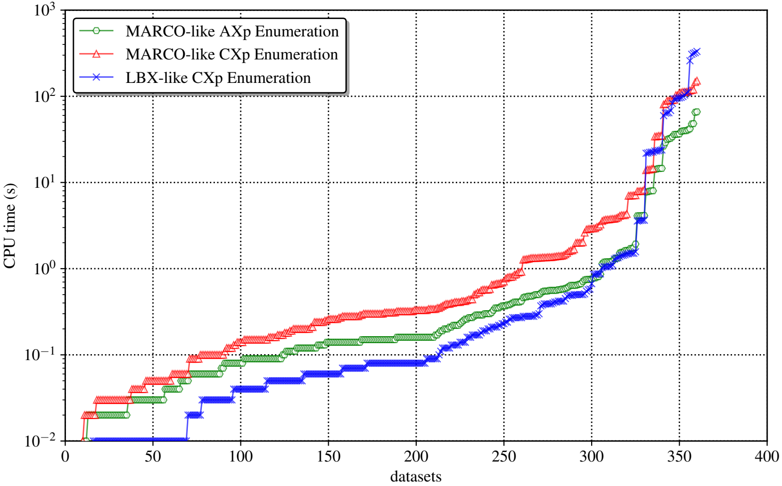


Results – raw performance



all approaches finish complete XP enumeration within <1000 sec.

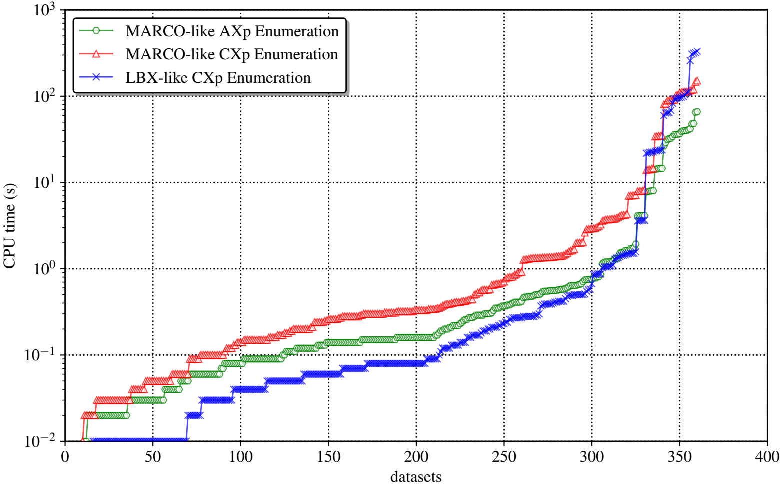
Results – raw performance



all approaches finish complete XP enumeration within <1000 sec.

MARCO-like setup — targeting AXps may pay off

Results – raw performance

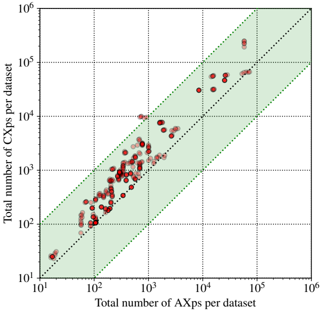


all approaches finish complete XP enumeration within <1000 sec.

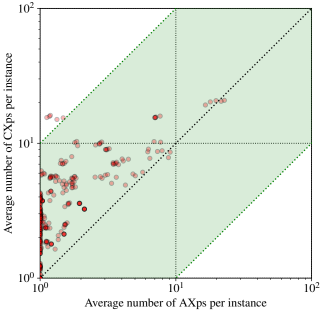
MARCO-like setup — targeting AXps may pay off

direct CXp enumeration is slower *(too many XPs?)*

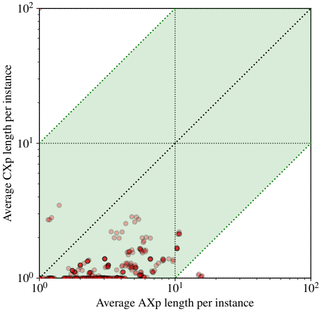
Results – AXps vs. CXps



(a) total number of AXps and CXps

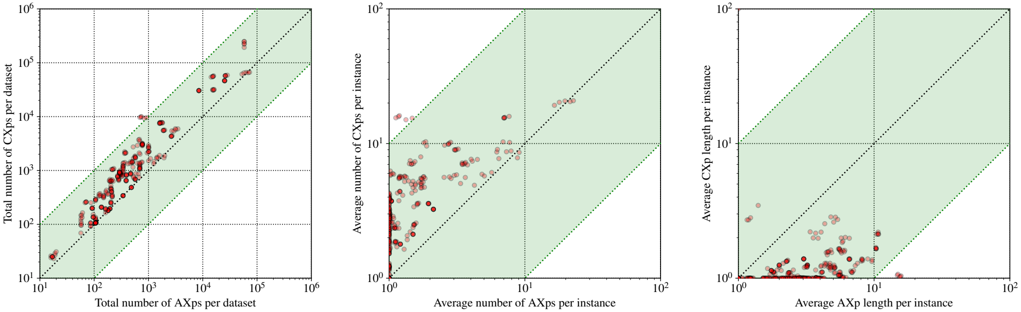


(b) avg. number of AXps and CXps



(c) avg. explanation size

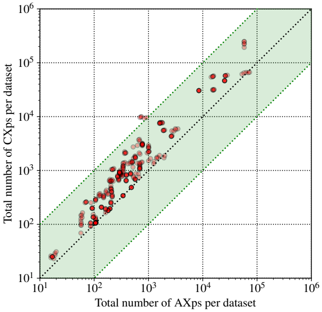
Results – AXps vs. CXps



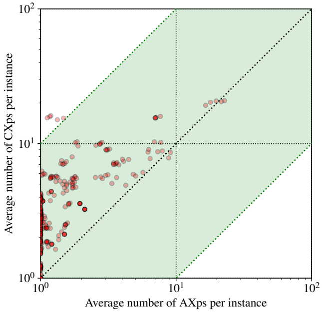
(a) total number of AXps and CXps **(b)** avg. number of AXps and CXps **(c)** avg. explanation size

16–72838 AXps **vs.** **23–248825 CXps** *per dataset*

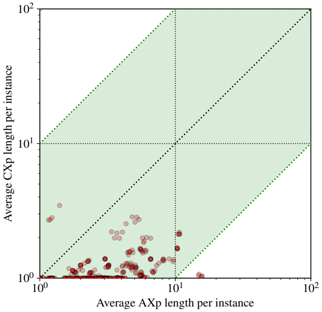
Results – AXps vs. CXps



(a) total number of AXps and CXps



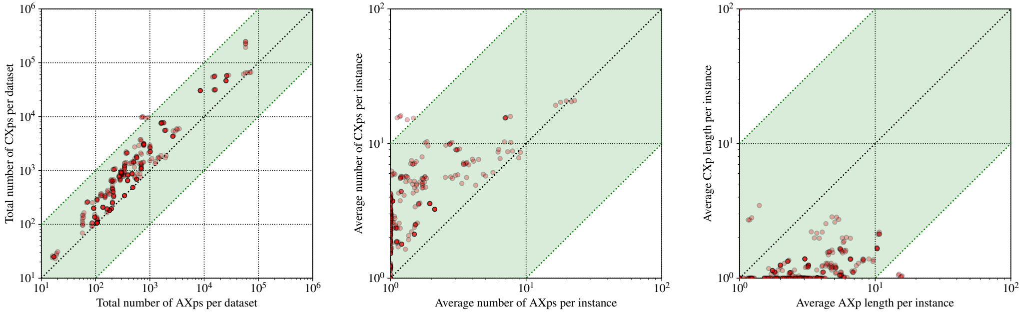
(b) avg. number of AXps and CXps



(c) avg. explanation size

16–72838 AXps	**vs.**	**23–248825 CXps**	*per dataset*
1–22.7 AXps	**vs.**	**1–20.8 CXps**	*per instance*

Results – AXps vs. CXps



(a) total number of AXps and CXps **(b)** avg. number of AXps and CXps **(c)** avg. explanation size

16–72838 AXps	**vs.**	**23–248825 CXps**	*per dataset*
1–22.7 AXps	**vs.**	**1–20.8 CXps**	*per instance*
1–15.8 lits per AXp	**vs.**	**$\leq$2.8 lits per CXp**	

Summary

Summary and future work

- **rigorous explanations for decision lists:**

Summary and future work

- **rigorous explanations for decision lists:**
  - DLs **may be uninterpretable**
    - just like decision trees!

Summary and future work

- **rigorous explanations for decision lists:**
  - **DLs may be uninterpretable**
    - just like decision trees!
  - **finding one explanation is not polytime, unless $\mathrm{P} = \mathrm{NP}$**
    - same for decision sets!
    - *and in contrast to decision trees!*

Summary and future work

- **rigorous explanations for decision lists:**
  - **DLs may be uninterpretable**
    - just like decision trees!
  - **finding one explanation is not polytime, unless $\mathrm{P} = \mathrm{NP}$**
    - same for decision sets!
    - *and in contrast to decision trees!*
  - **encoding to propositional logic**
    - **use of SAT oracles**
    - finding one AXp or CXp
    - efficient MARCO-like ***enumeration***!

Summary and future work

- **rigorous explanations for decision lists:**
  - **DLs may be uninterpretable**
    - just like decision trees!
  - **finding one explanation is not polytime, unless $P = NP$**
    - same for decision sets!
    - *and in contrast to decision trees!*
  - **encoding to propositional logic**
    - **use of SAT oracles**
    - finding one AXp or CXp
    - efficient MARCO-like ***enumeration***!
- **future work**
  - explain ***other ML models*** with SAT?

Summary and future work

- **rigorous explanations for decision lists:**
  - **DLs may be uninterpretable**
    - just like decision trees!
  - **finding one explanation is not polytime, unless $P = NP$**
    - same for decision sets!
    - *and in contrast to decision trees!*
  - **encoding to propositional logic**
    - **use of SAT oracles**
    - finding one AXp or CXp
    - efficient MARCO-like ***enumeration***!
- **future work**
  - explain ***other ML models*** with SAT?
  - ***efficiently?***

Questions?