Logic-Enabled Learning, Verification & Explanation of ML Models

A. Ignatiev, J. Marques-Silva, K. Meel & N. Narodytska

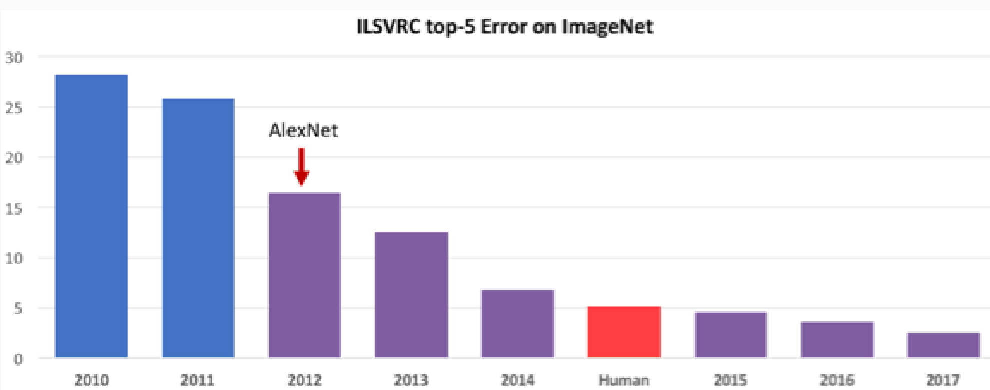
Monash Univ, ANITI/IRIT/CNRS, NU Singapore & VMWare Research

January 08, 2021 | IJCAI Tutorial T22

Many ML successes

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

Image & Speech Recognition

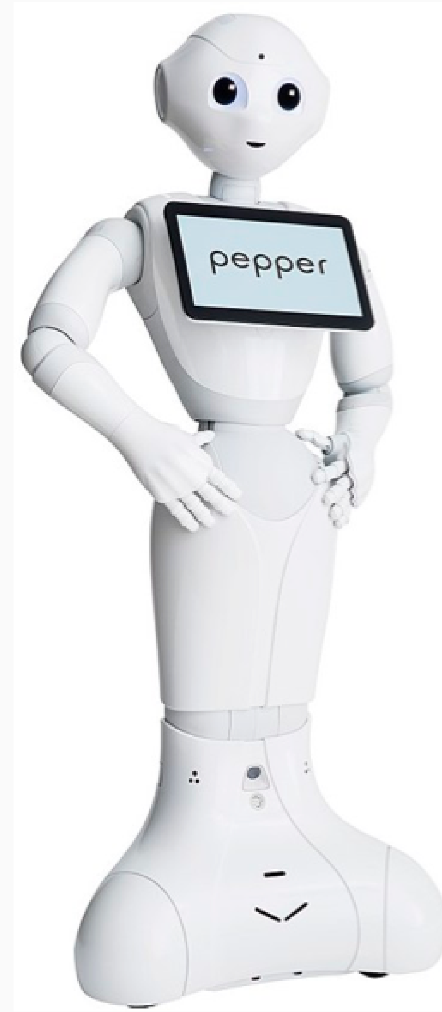


http://gradientscience.org/intro_adversarial/

DeepMind

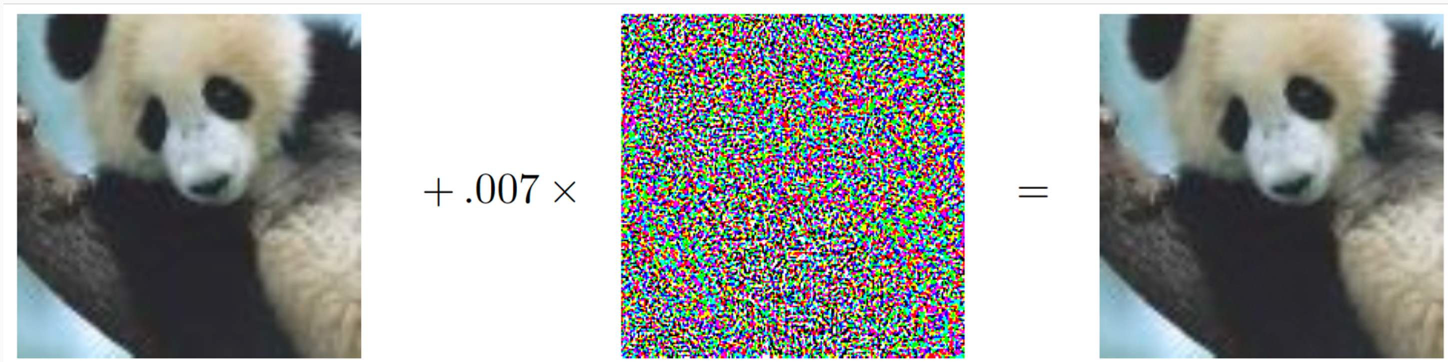
AlphaGo

AlphaGo Zero & **Alpha Zero**



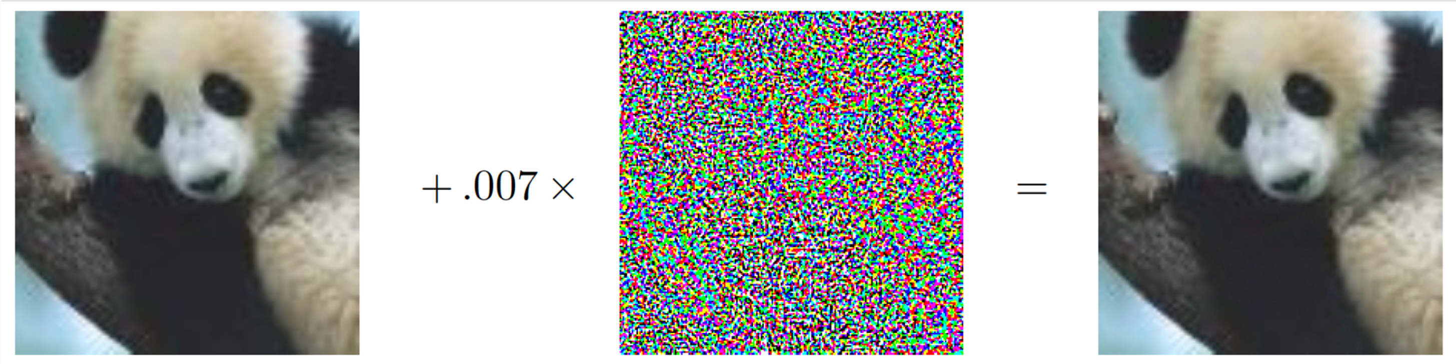
https://fr.wikipedia.org/wiki/Pepper_(robot)

Problem: ML models are brittle



Goodfellow et al., ICLR'15

Problem: ML models are brittle

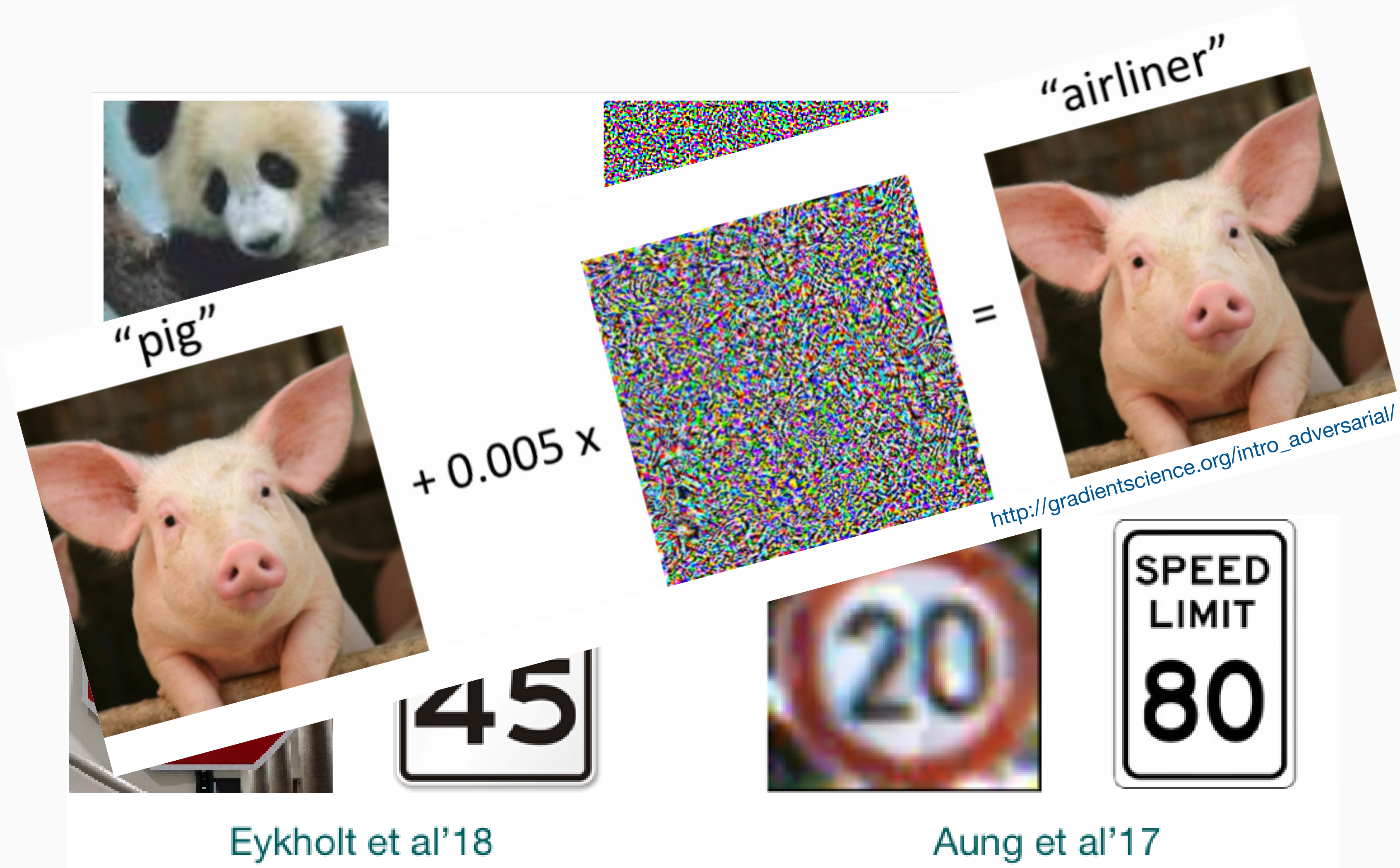


Goodfellow et al., ICLR'15

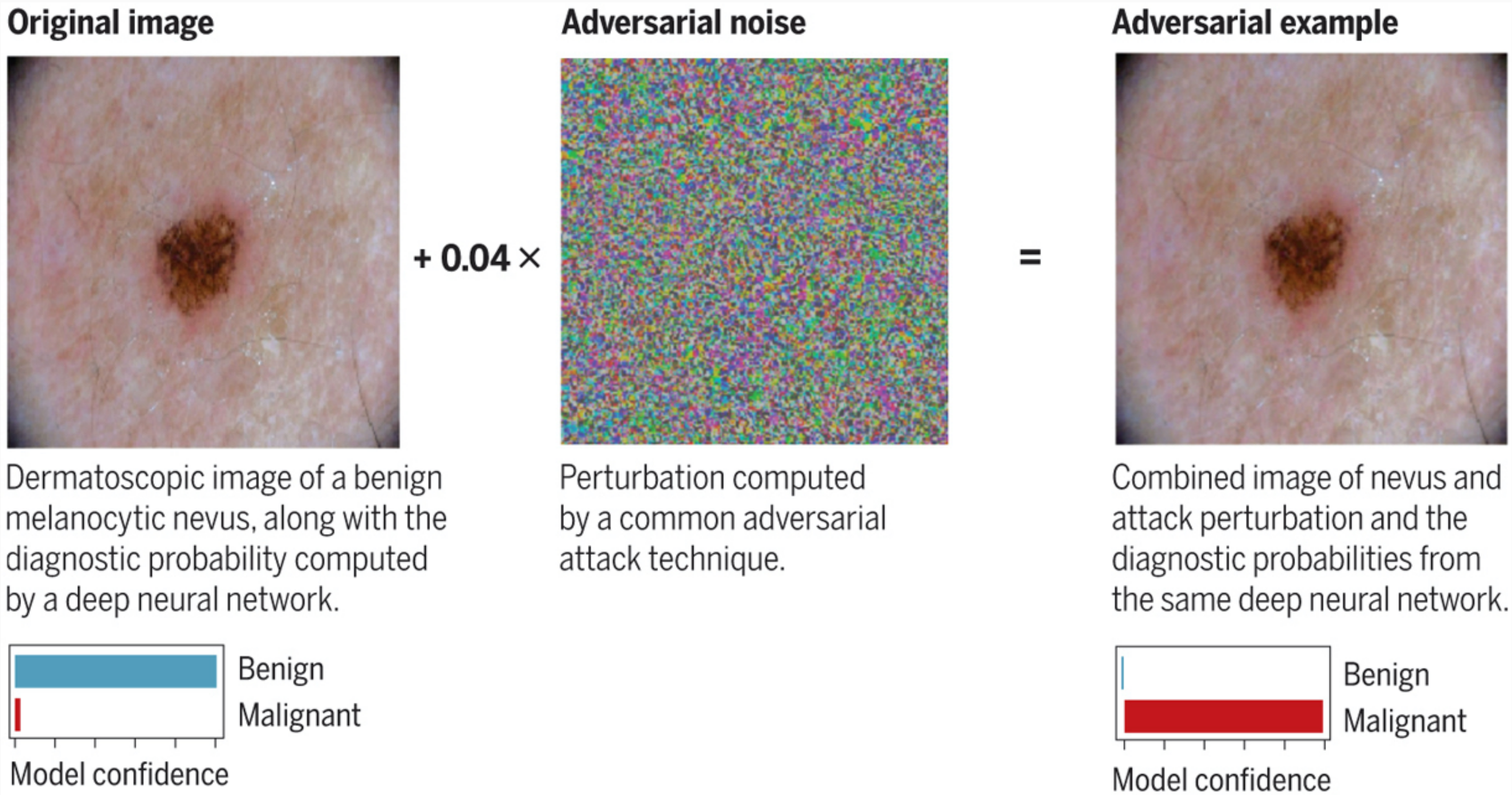
Eykholt et al'18

Aung et al'17

Problem: ML models are brittle

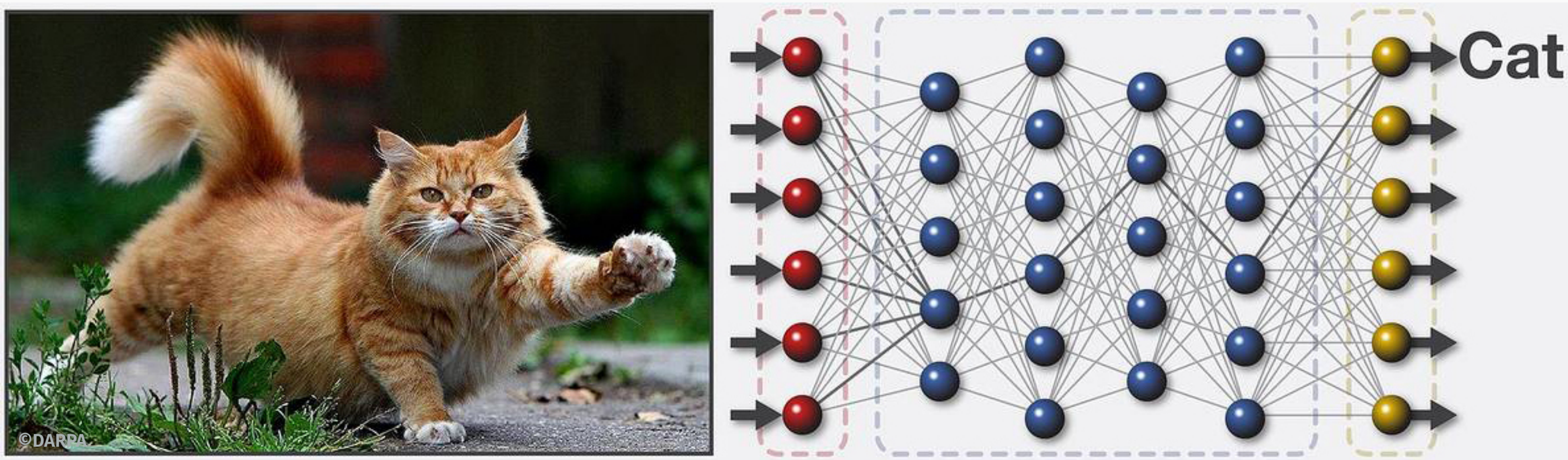


Adversarial examples can be very unsettling

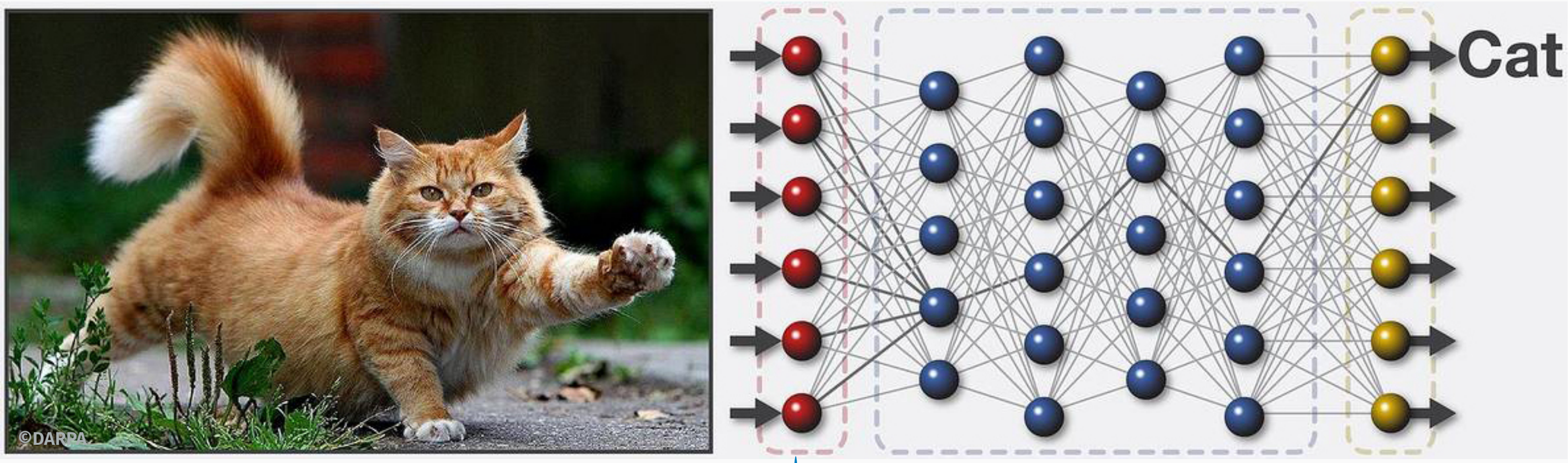


Finlayson et al., Nature 2019

Problem: ML models are opaque

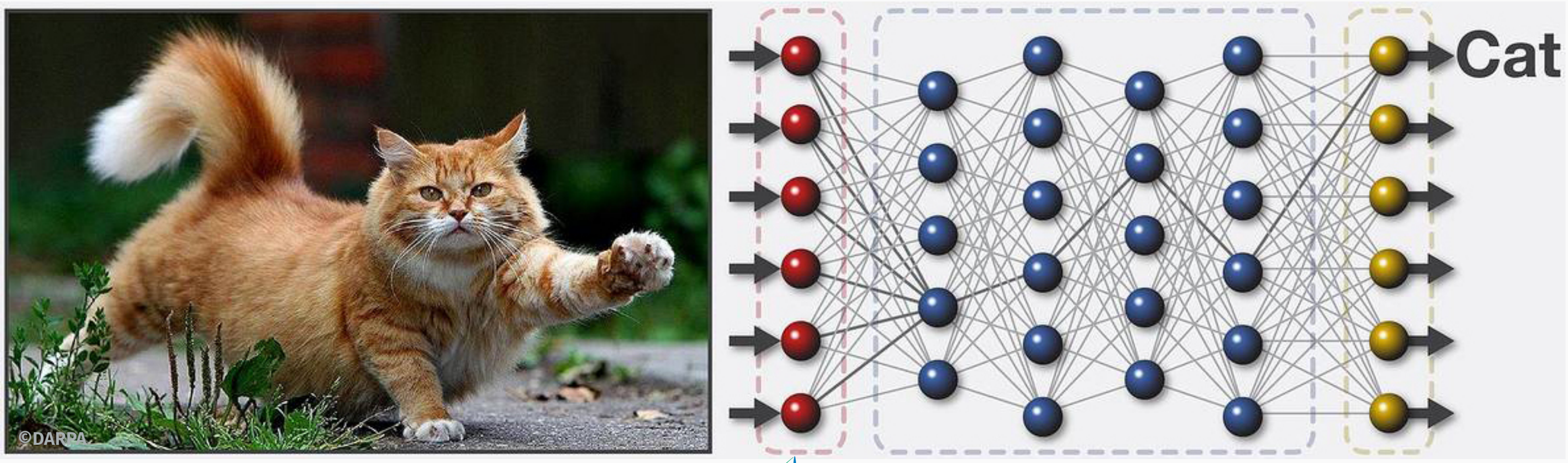


Problem: ML models are opaque



Why does the NN predict a cat?

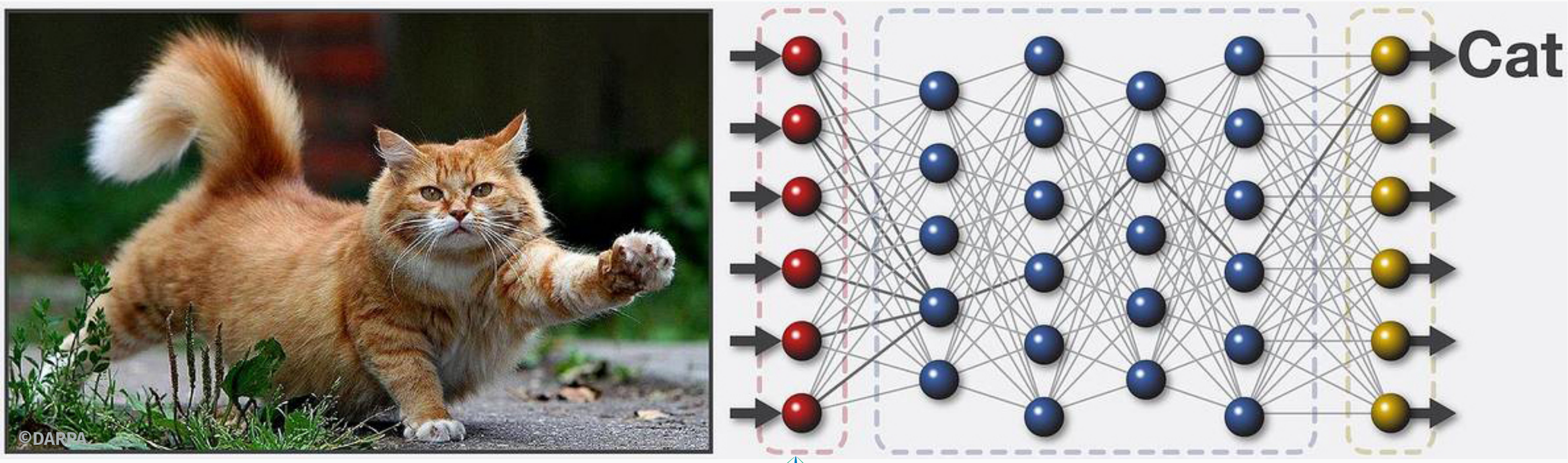
Problem: ML models are opaque



Why does the NN predict a cat?

Which features matter?

Problem: ML models are opaque



Why does the NN predict a cat?

Which features matter?

Are there general explanations??

Why XAI?

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

of 27 April 2016

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

(Text with EEA relevance)

Why XAI?

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

of 27 April 2016

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

(Text with EEA relevance)

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

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

Why XAI?

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

of 27 April 2016

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

(Text with EEA relevance)

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

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

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

Why XAI?

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

of 27 April 2016

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

(Text with EEA relevance)

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

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

POLICY \ US & WORLD \ TECH

TheVerge.com

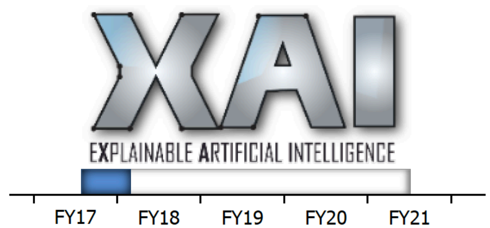
A new bill would force companies to check their algorithms for bias

By Adi Robertson | @thedextriarchy | Apr 10, 2019, 3:52pm EDT

Algorithmic Accountability Act

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

Explainable Artificial Intelligence (XAI)



David Gunning
DARPA/I2O
Program Update November 2017

DARPA

Why XAI?

REGULATION (EU) 2016/67… THE COUNCIL

on the p… …ta and on the free
move… …tion Regulation)

European Union regulation
and a "righ

Bryce Goodma…

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

In order to trust deployed AI systems, we must not only improve their robustness,[5] but also develop ways to make their reasoning intelligible. Intelligibility will help us spot AI that makes mistakes due to distributional drift or incomplete representations of goals and features. Intelligibility will also facilitate control by humans in increasingly common collaborative human/AI teams. Furthermore, intelligibility will help humans learn from AI. Finally, there are legal reasons to want intelligible AI, including the European GDPR and a growing need to assign liability when AI errs.

Weld & Bansal, CACM, Jun'19

TheVerge.com

…mpanies to check their

Algorithmic Accountability Act

…ence (XAI)

…mber 2017

DARPA

©DARPA

XAI & EU guidelines

Search

European Commission > Strategy > Digital Single Market > Reports and studies >

Digital Single Market

REPORT / STUDY | 8 April 2019

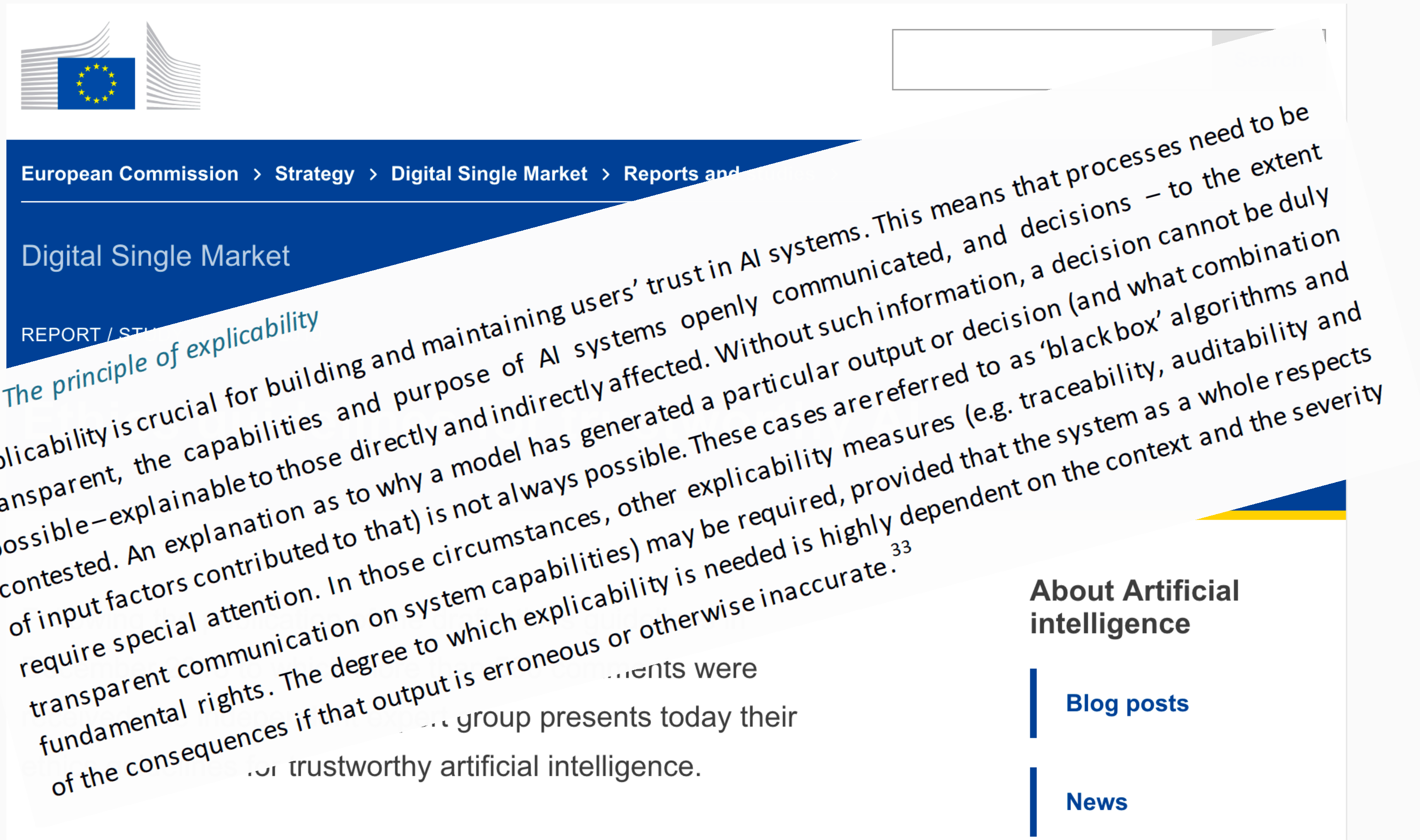
Ethics guidelines for trustworthy AI

Following the publication of the draft ethics guidelines in December 2018 to which more than 500 comments were received, the independent expert group presents today their ethics guidelines for trustworthy artificial intelligence.

About Artificial intelligence

Blog posts

News



The principle of explicability

- Explicability is crucial for building and maintaining users' trust in AI systems. This means that processes need to be transparent, the capabilities and purpose of AI systems openly communicated, and decisions – to the extent possible – explainable to those directly and indirectly affected. Without such information, a decision cannot be duly contested. An explanation as to why a model has generated a particular output or decision (and what combination of input factors contributed to that) is not always possible. These cases are referred to as 'black box' algorithms and require special attention. In those circumstances, other explicability measures (e.g. traceability, auditability and transparent communication on system capabilities) may be required, provided that the system as a whole respects fundamental rights. The degree to which explicability is needed is highly dependent on the context and the severity of the consequences if that output is erroneous or otherwise inaccurate.[33]

Explanations with heuristic approaches unsettling

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

Explanations with heuristic approaches unsettling

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

Similar results for Google's XAI service??

Solutions to problems?

- Assess robustness
- Learn interpretable models
- Explain black-box models
- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?
- Learn interpretable models
- Explain black-box models
- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?

- Learn interpretable models
  - Decision trees; decision sets; decision lists; etc.

- Explain black-box models

- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?
- Learn interpretable models
  - Decision trees; decision sets; decision lists; etc.
- Explain black-box models
  - By using some accepted definition of explanation
- How about heuristic approaches?

Solutions to problems?

- Assess robustness
  - How easy it is to fool and ML model?
- Learn interpretable models
  - Decision trees; decision sets; decision lists; etc.
- Explain black-box models
  - By using some accepted definition of explanation
- How about heuristic approaches?
  - **No** formal guarantees provided

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle
  - NP-hardness does **not** mean impossible to solve!

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle
  - NP-hardness does **not** mean impossible to solve!
- There are efficient reasoners
  - SAT, SMT, CP, ILP, etc.

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle
  - NP-hardness does **not** mean impossible to solve!
- There are efficient reasoners
  - SAT, SMT, CP, ILP, etc.
- Effective problem encodings

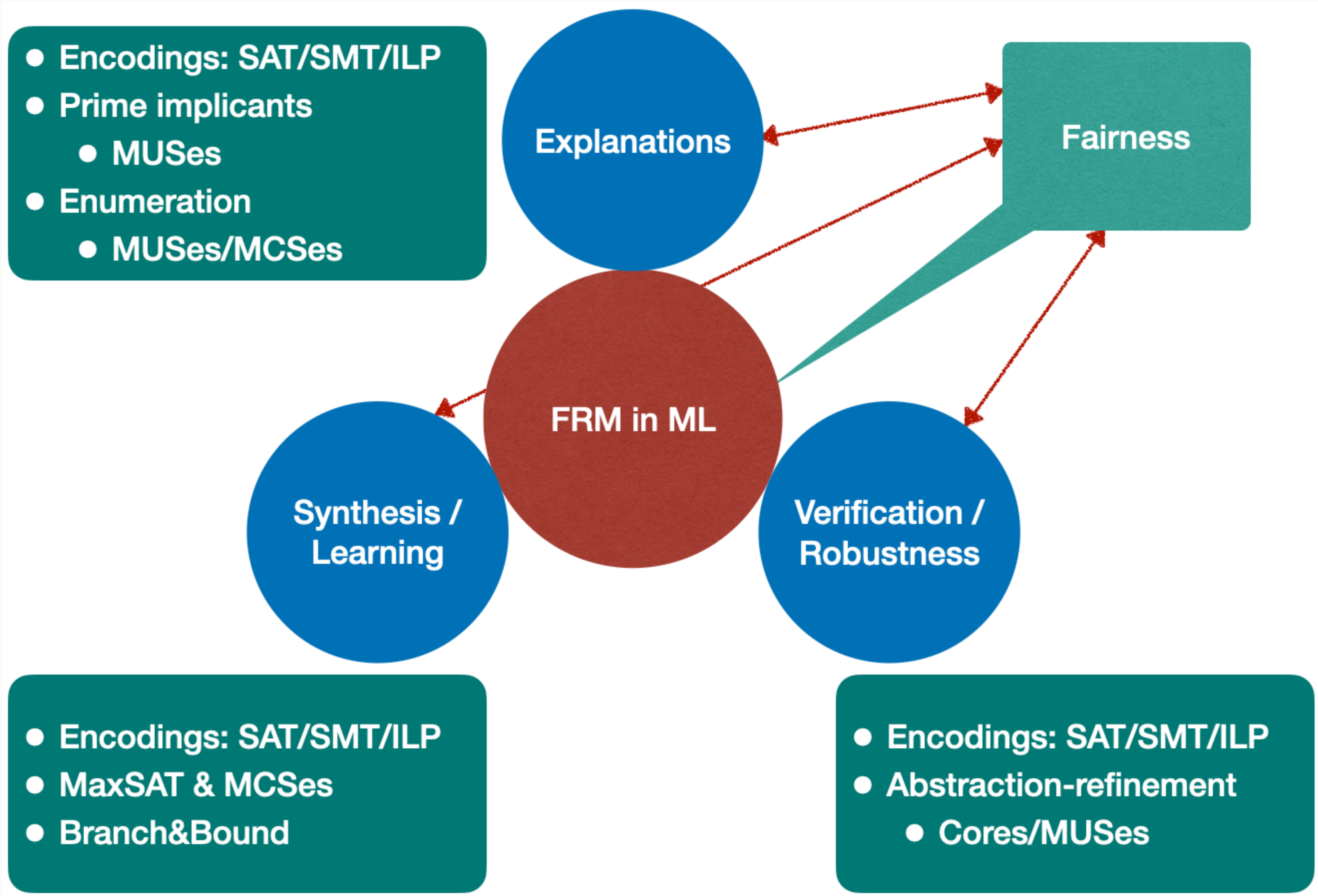
How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle
    - NP-hardness does **not** mean impossible to solve!
- There are efficient reasoners
    - SAT, SMT, CP, ILP, etc.
- Effective problem encodings
- Exploit known solutions
    - Exploit reasoners for efficient problem solving

How/Why to reason about ML models, with formal guarantees?

- Problem complexity **not** necessarily an hopeless obstacle
    - NP-hardness does **not** mean impossible to solve!
- There are efficient reasoners
    - SAT, SMT, CP, ILP, etc.
- Effective problem encodings
- Exploit known solutions
    - Exploit reasoners for efficient problem solving
- **Formal reasoning about ML models is a practically viable option**

Some uses of formal reasoning methods (FRM)



This tutorial – formal reasoning in ML

- Part 01: first contact with formal reasoning tools — Joao
- Part 02: learning interpretable models — Kuldeep
- Part 03: assessing robustness of ML models — Nina
- Part 04: rigorous explanations of ML models — Alexey
- Part 05: recent work on explanations & wrap-up — Joao
  - Duality, tractability & links with fairness

Part 1

Basic Formal Toolbox

Outline

Preliminaries

Logic Encodings of ML Models

Outline

Preliminaries

- Classification Problems in ML
- Logic Overview
- Logic & Optimization
- Reasoning Beyond Propositional Logic
- Additional Concepts

Logic Encodings of ML Models

Classification problems

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

Classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, n\}$, each taking values from a domain D_i
  - Features can be categorical or ordinal, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{n} D_i$
- ML model $\mathbb{M}$ computes classification function $\varphi : \mathbb{F} \to \mathcal{K}$
  - For simplicity, we will use $\mathcal{K} = \{\boxplus, \boxminus\}$

Classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, n\}$, each taking values from a domain D_i
  - Features can be categorical or ordinal, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{n} D_i$
- ML model $\mathbb{M}$ computes classification function $\varphi : \mathbb{F} \to \mathcal{K}$
  - For simplicity, we will use $\mathcal{K} = \{\boxplus, \boxminus\}$
- Instance $\mathbf{v} \in \mathbb{F}$, with prediction $c = \varphi(\mathbf{v})$, $c \in \mathcal{K}$
  - **Obs:** instance $\approx$ example $\approx$ sample $\approx$ point

Classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, n\}$, each taking values from a domain D_i
  - Features can be categorical or ordinal, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{n} D_i$
- ML model $\mathbb{M}$ computes classification function $\varphi : \mathbb{F} \to \mathcal{K}$
  - For simplicity, we will use $\mathcal{K} = \{\boxplus, \boxminus\}$
- Instance $\mathbf{v} \in \mathbb{F}$, with prediction $c = \varphi(\mathbf{v})$, $c \in \mathcal{K}$
  - **Obs:** instance $\approx$ example $\approx$ sample $\approx$ point
- Each $\mathbf{v} \in \mathbb{F}$ is also represented as a set of literals, $\mathcal{C}_{\mathbf{v}} = \{(x_i = v_i) | i \in \mathcal{F}\}$
  - For boolean features, $x_i = 0$ represented by $\neg x_i$ and $x_i = 1$ represented by x_i

Outline

Preliminaries

- Classification Problems in ML
- Logic Overview
- Logic & Optimization
- Reasoning Beyond Propositional Logic
- Additional Concepts

Logic Encodings of ML Models

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: $($, $)$
  - Often restricted to Conjunctive Normal Form (CNF)

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: $($, $)$
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: (,)
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: (,)
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:
    - $\{r, s, a, b, c, d\}$

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: $($, $)$
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:
    - $\{r, s, a, b, c, d\}$
    - $\{r, s, \neg x, y, \neg w, z, \neg a, b, c, d\}$

The SAT problem

- SAT is the decision problem for propositional logic
  - Well-formed propositional formulas, with variables, logical connectives: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$, and parenthesis: (,)
  - Often restricted to Conjunctive Normal Form (CNF)
  - Goal:
    Decide whether formula has a satisfying assignment

- Example:

$$\mathcal{F} \triangleq (r) \wedge (\bar{r} \vee s) \wedge (\neg w \vee a) \wedge (\neg x \vee b) \wedge (\neg y \vee \neg z \vee c) \wedge (\neg b \vee \neg c \vee d)$$

  - Example models:
    - $\{r, s, a, b, c, d\}$
    - $\{r, s, \neg x, y, \neg w, z, \neg a, b, c, d\}$

- SAT is **NP-complete** [Coo71]

The CDCL SAT disruption

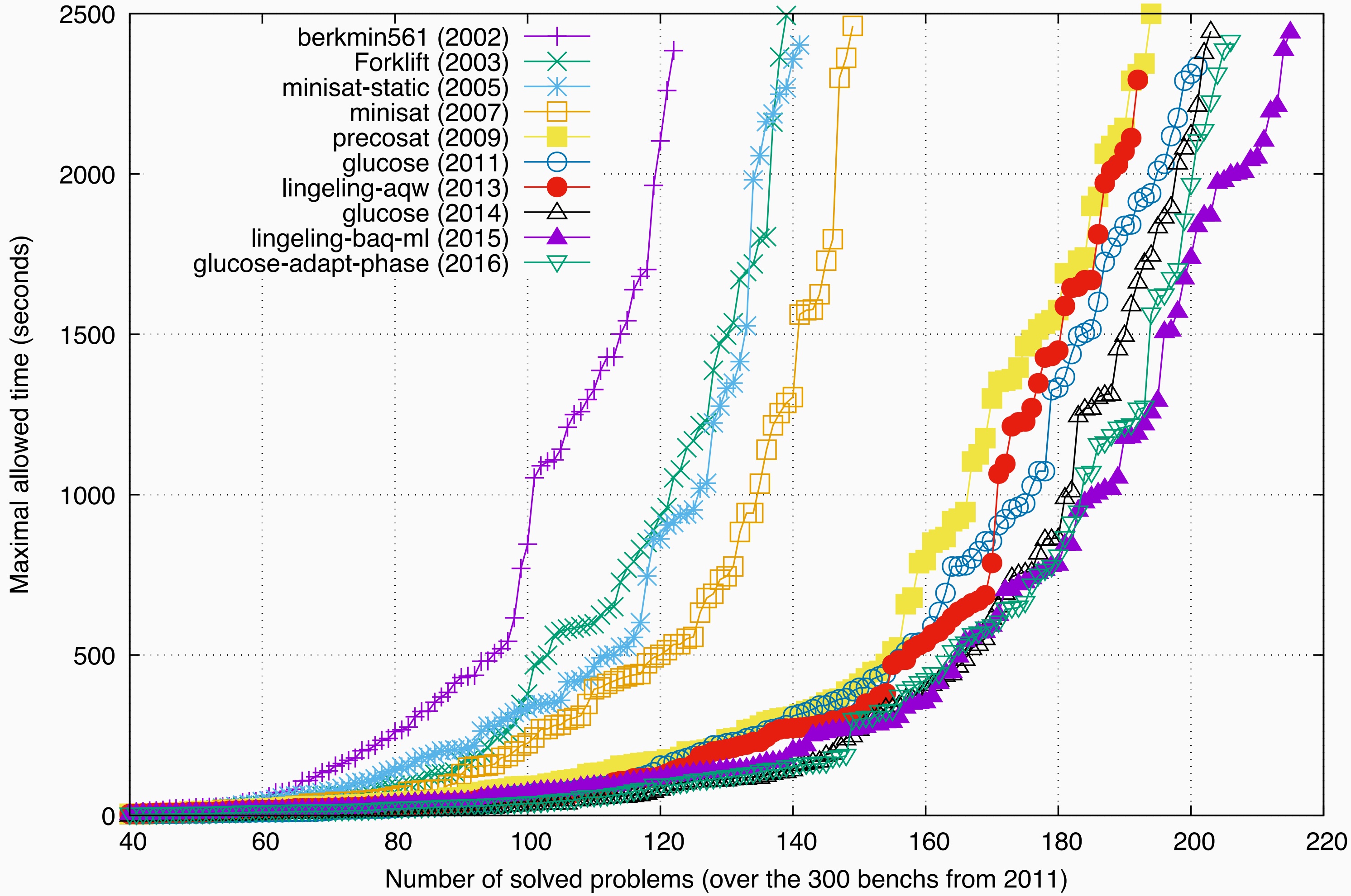
- CDCL SAT solving is a success story of Computer Science

The CDCL SAT disruption

- CDCL SAT solving is a success story of Computer Science
    - Conflict-Driven Clause Learning (CDCL)
    - (CDCL) SAT has impacted many different fields
    - Hundreds (thousands?) of practical applications

CDCL SAT solver (continued) improvement

[Source: Simon 2015]



How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem

- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - *N* agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem
- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents

```
*** tracker: a pathfinding tool ***

Initialization ... CPU Time: 0.004711
Number of variables: 113315
Tentative makespan 1
Number of variables: 226630
Number of assumptions: 1
c Running SAT solver ... CPU Time: 0.718112
c Done running SAT solver ... CPU Time: 0.830099
No solution for makespan 1
Elapsed CPU Time: 0.830112
Tentative makespan 2
Number of variables: 339945
Number of assumptions: 1
c Running SAT solver ... CPU Time: 1.27113
c Done running SAT solver ... CPU Time: 1.27114
No solution for makespan 2
Elapsed CPU Time: 1.27114
...
...
Tentative makespan 24
Number of variables: 2832875
Number of assumptions: 1
c Running SAT solver ... CPU Time: 11.8653
c Done running SAT solver ... CPU Time: 11.8653
No solution for makespan 24
Elapsed CPU Time: 11.8653
Tentative makespan 25
Number of variables: 2946190
Number of assumptions: 1
c Running SAT solver ... CPU Time: 12.3491
c Done running SAT solver ... CPU Time: 16.6882
Solution found for makespan 25
Elapsed CPU Time: 16.6995
```

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem
- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents
  - Formula w/ 2832875 variables!
  - Formula w/ 2946190 variables!

```
*** tracker: a pathfinding tool ***

Initialization ... CPU Time: 0.004711
Number of variables: 113315
Tentative makespan 1
Number of variables: 226630
Number of assumptions: 1
c Running SAT solver ... CPU Time: 0.718112
c Done running SAT solver ... CPU Time: 0.830099
No solution for makespan 1
Elapsed CPU Time: 0.830112
Tentative makespan 2
Number of variables: 339945
Number of assumptions: 1
c Running SAT solver ... CPU Time: 1.27113
c Done running SAT solver ... CPU Time: 1.27114
No solution for makespan 2
Elapsed CPU Time: 1.27114
...
...
Tentative makespan 24
Number of variables: 2832875
Number of assumptions: 1
c Running SAT solver ... CPU Time: 11.8653
c Done running SAT solver ... CPU Time: 11.8653
No solution for makespan 24
Elapsed CPU Time: 11.8653
Tentative makespan 25
Number of variables: 2946190
Number of assumptions: 1
c Running SAT solver ... CPU Time: 12.3491
c Done running SAT solver ... CPU Time: 16.6882
Solution found for makespan 25
Elapsed CPU Time: 16.6995
```

How good are SAT solvers? – an example

- Cooperative pathfinding (CPF)
  - N agents on some grid/graph
  - Start positions
  - Goal positions
  - Minimize makespan
  - Restricted planning problem
- Concrete example
  - Gaming grid
  - 1039 vertices
  - 1928 edges
  - 100 agents
  - Formula w/ 2832875 variables!
  - Formula w/ 2946190 variables!
- **Note:** In the early 90s, SAT solvers could solve formulas with a few hundred variables!

```
*** tracker: a pathfinding tool ***

Initialization ... CPU Time: 0.004711
Number of variables: 113315
Tentative makespan 1
Number of variables: 226630
Number of assumptions: 1
c Running SAT solver ... CPU Time: 0.718112
c Done running SAT solver ... CPU Time: 0.830099
No solution for makespan 1
Elapsed CPU Time: 0.830112
Tentative makespan 2
Number of variables: 339945
Number of assumptions: 1
c Running SAT solver ... CPU Time: 1.27113
c Done running SAT solver ... CPU Time: 1.27114
No solution for makespan 2
Elapsed CPU Time: 1.27114
...
...
Tentative makespan 24
Number of variables: 2832875
Number of assumptions: 1
c Running SAT solver ... CPU Time: 11.8653
c Done running SAT solver ... CPU Time: 11.8653
No solution for makespan 24
Elapsed CPU Time: 11.8653
Tentative makespan 25
Number of variables: 2946190
Number of assumptions: 1
c Running SAT solver ... CPU Time: 12.3491
c Done running SAT solver ... CPU Time: 16.6882
Solution found for makespan 25
Elapsed CPU Time: 16.6995
```

Grasping the search space ...

- Number of seconds since the Big Bang: $\approx 10^{17}$

Grasping the search space ...

- Number of seconds since the Big Bang: $\approx 10^{17}$

- Number of fundamental particles in observable universe: $\approx 10^{80}$ (or $\approx 10^{85}$)

Grasping the search space …

- Number of seconds since the Big Bang: $\approx 10^{17}$
- Number of fundamental particles in observable universe: $\approx 10^{80}$ (or $\approx 10^{85}$)
- Search space with 2832875 propositional variables (worst case):

Grasping the search space ...

- Number of seconds since the Big Bang: $\approx 10^{17}$

- Number of fundamental particles in observable universe: $\approx 10^{80}$ (or $\approx 10^{85}$)

- Search space with 2832875 propositional variables (worst case):
  - # of assignments to $> 2.8 \times 10^6$ variables: $\gg 10^{840000}$ ‼
  - **Obs:** SAT solvers at present (but formula dependent)

Outline

Preliminaries

- Classification Problems in ML
- Logic Overview
- Logic & Optimization
- Reasoning Beyond Propositional Logic
- Additional Concepts

Logic Encodings of ML Models

Optimization with maximum satisfiability (MaxSAT)

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Unsatisfiable formula

Optimization with maximum satisfiability (MaxSAT)

$\boxed{x_6 \lor x_2}$	$\neg x_6 \lor x_2$	$\neg x_2 \lor x_1$	$\neg x_1$
$\neg x_6 \lor x_8$	$x_6 \lor \neg x_8$	$x_2 \lor x_4$	$\boxed{\neg x_4 \lor x_5}$
$\boxed{x_7 \lor x_5}$	$\neg x_7 \lor x_5$	$\neg x_5 \lor x_3$	$\neg x_3$

- Unsatisfiable formula
- Find largest subset of clauses that is satisfiable

Optimization with maximum satisfiability (MaxSAT)

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\boxed{\neg x_1}$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$\boxed{x_7 \vee x_5}$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Unsatisfiable formula
- Find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula

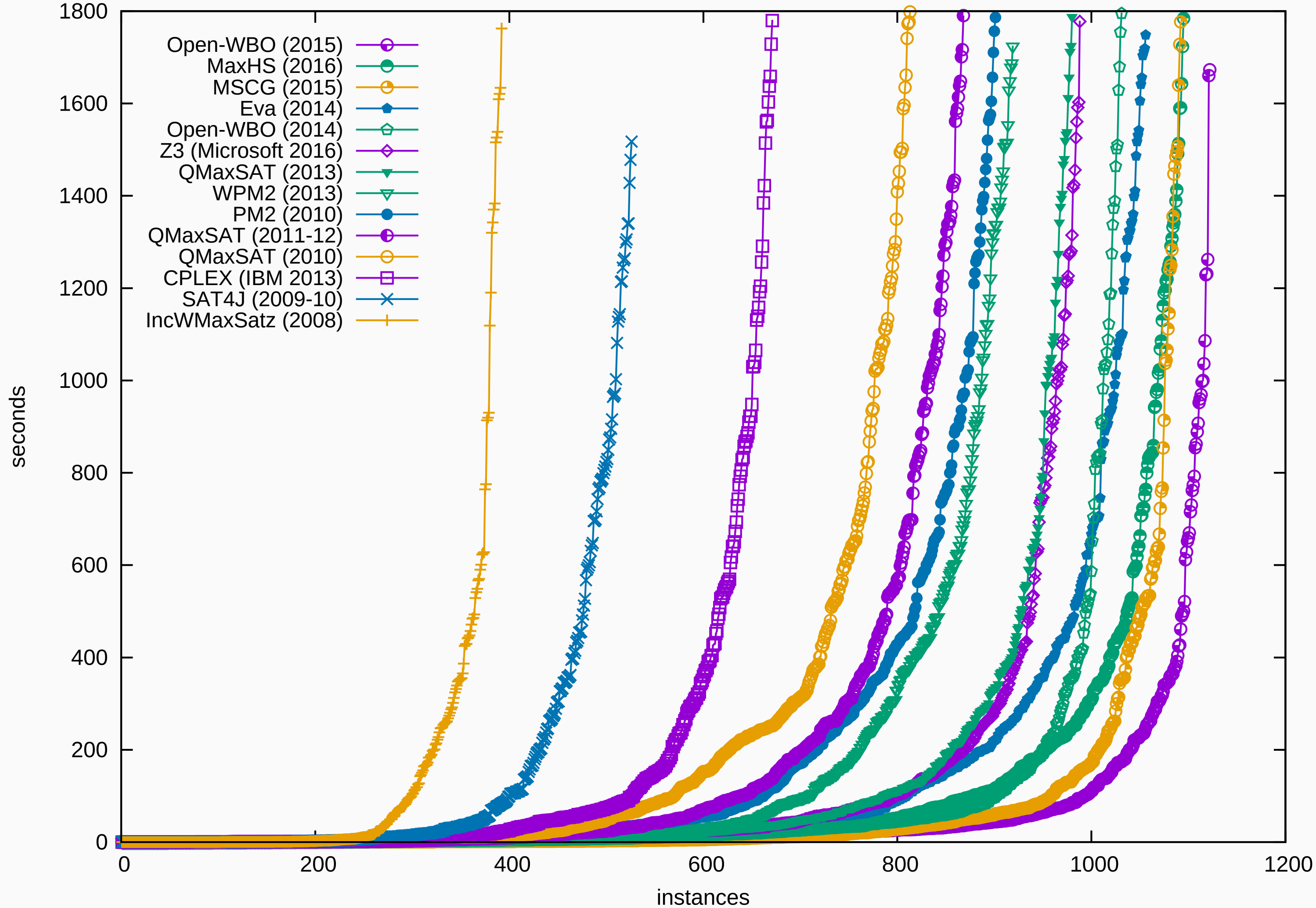
Optimization with maximum satisfiability (MaxSAT)

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\boxed{\neg x_1}$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$\boxed{x_7 \vee x_5}$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Unsatisfiable formula
- Find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula
- The MaxSAT solution is one of the smallest (cost) MCSes

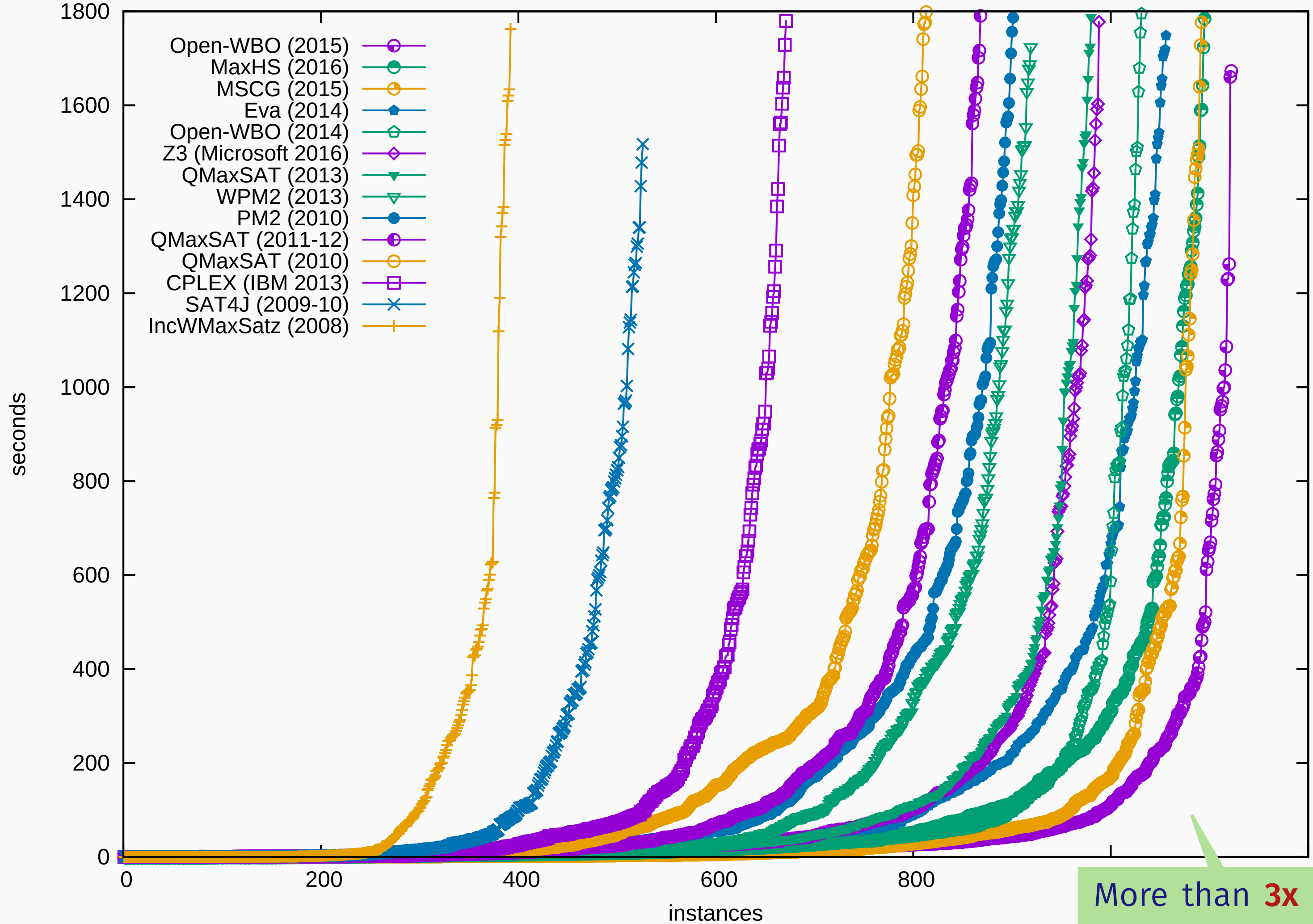
The MaxSAT (r)evolution

The MaxSAT (r)evolution – partial MaxSAT



Source: [2018 MaxSAT Eval. organizers]

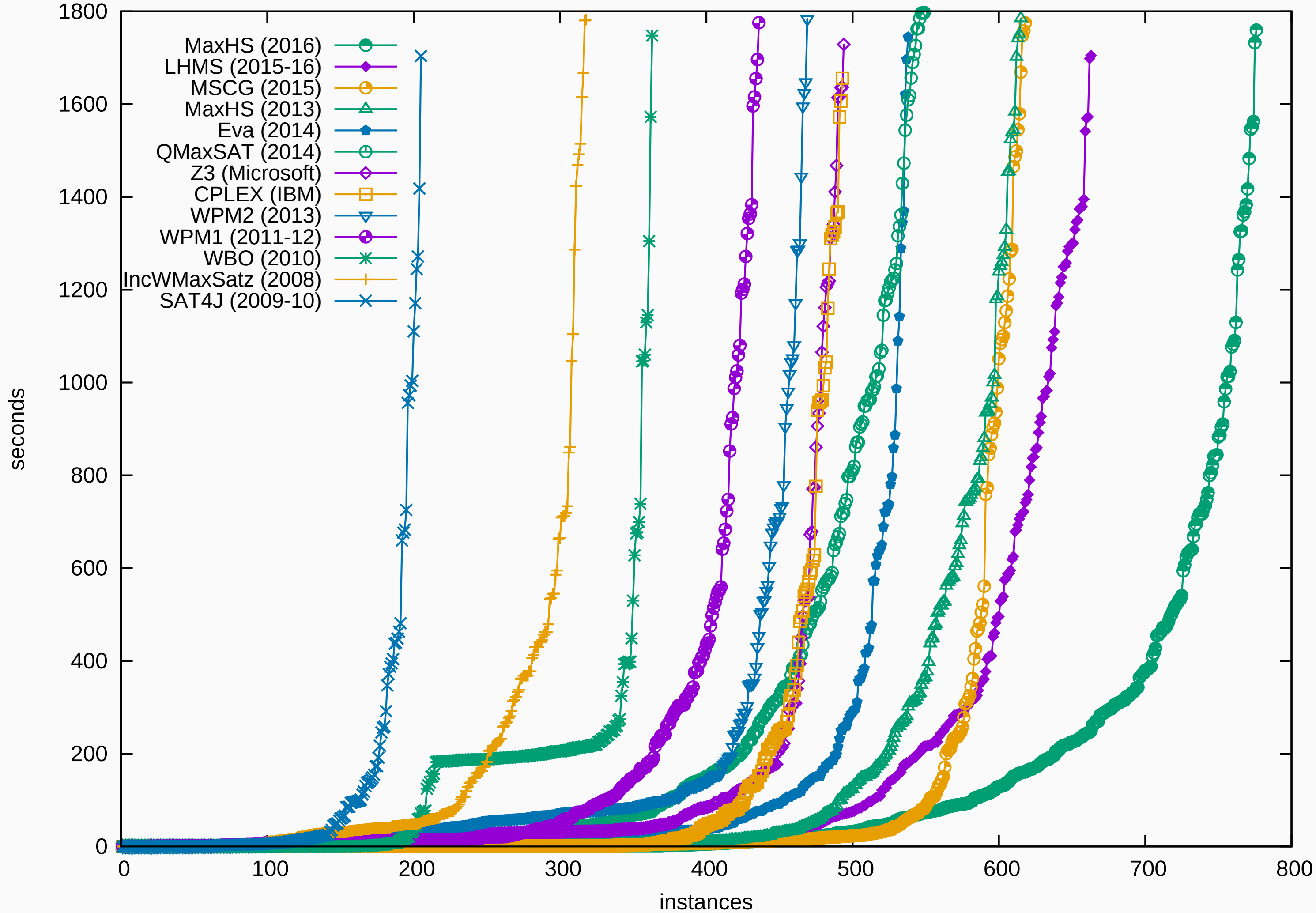
The MaxSAT (r)evolution – partial MaxSAT



More than **3x** more instances solved ‼

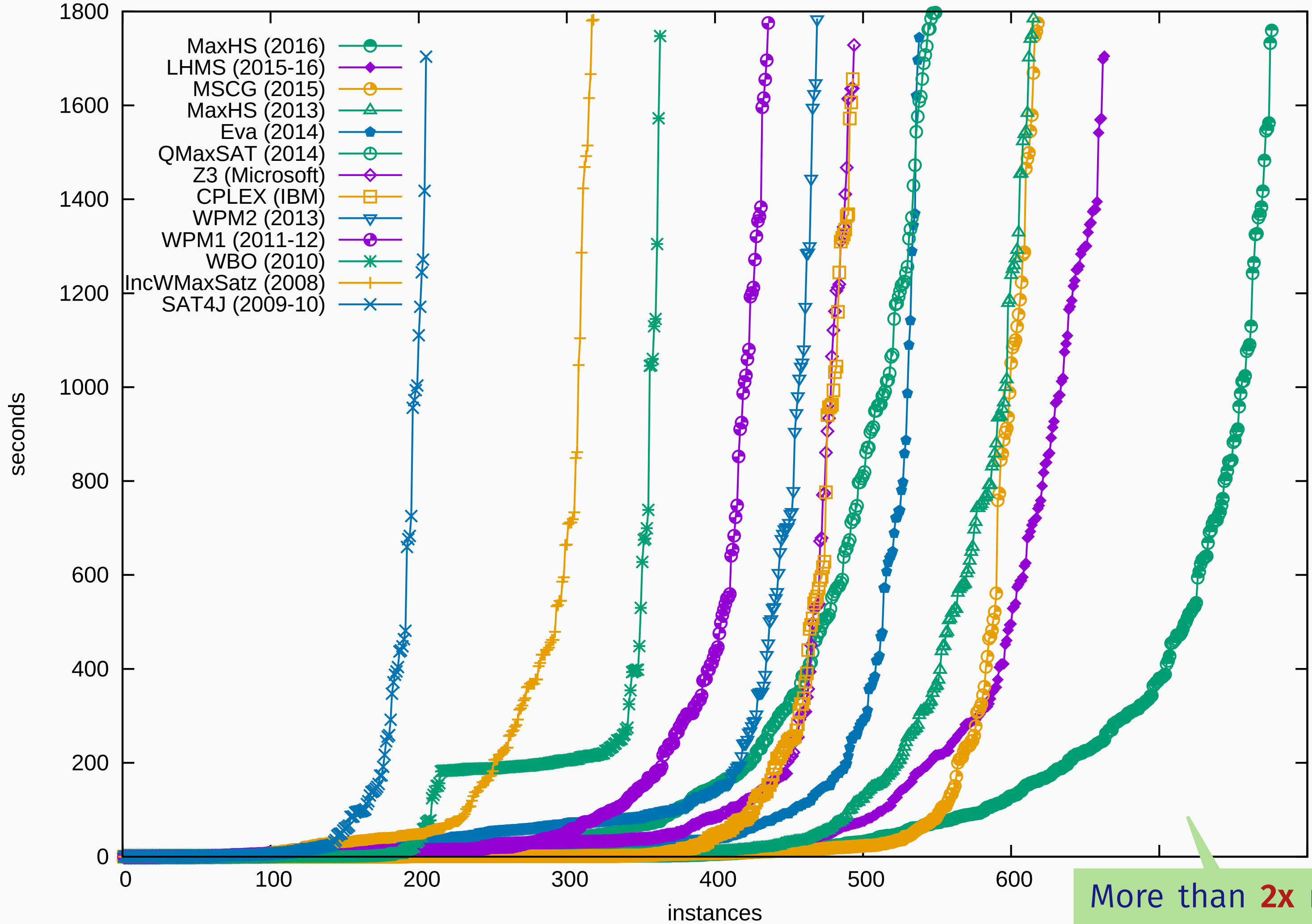
Source: [2018 MaxSAT Eval. organizers]

The MaxSAT (r)evolution – weighted MaxSAT



Source: [2018 MaxSAT Eval. organizers]

The MaxSAT (r)evolution – weighted MaxSAT



More than **2x** more instances solved **!!**

Source: [2018 MaxSAT Eval. organizers]

Outline

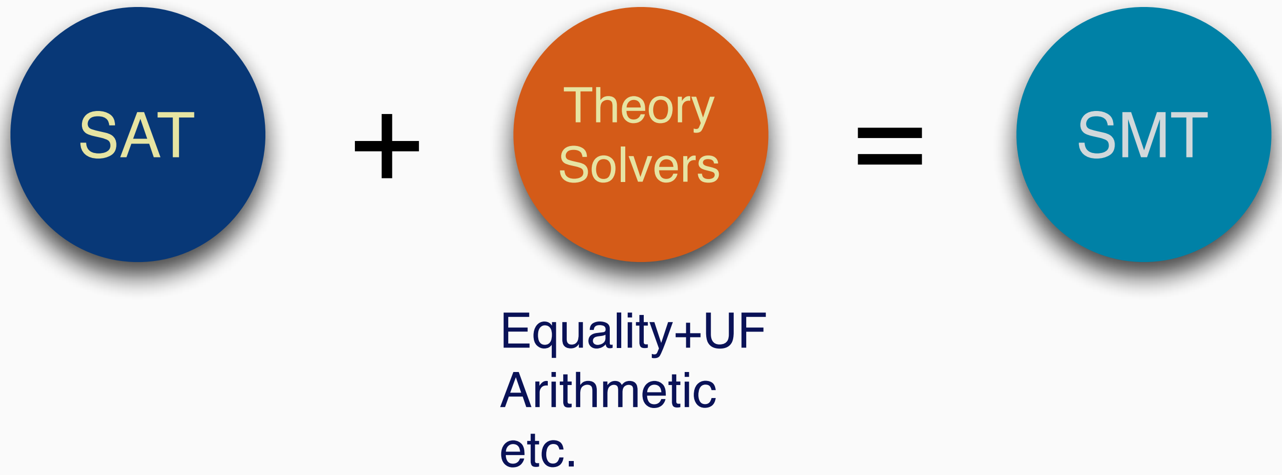
Preliminaries

- Classification Problems in ML
- Logic Overview
- Logic & Optimization
- Reasoning Beyond Propositional Logic
- Additional Concepts

Logic Encodings of ML Models

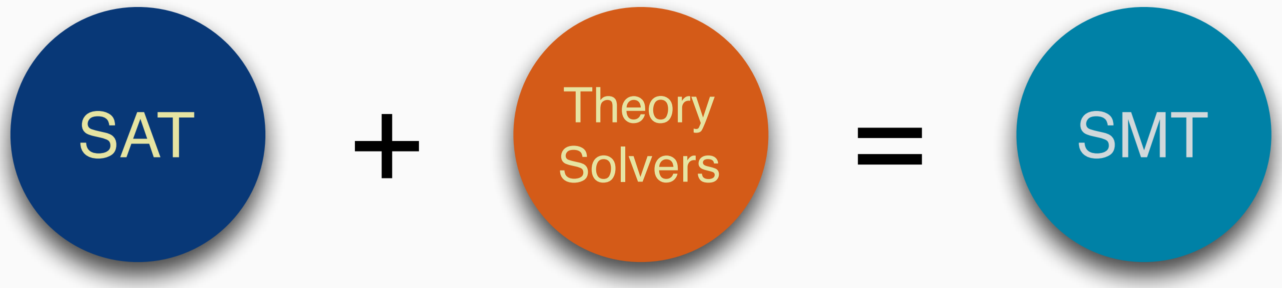
Satisfiability Modulo Theories (SMT)

- Automate reasoning in (fragments of) first-order logic (FOL)



Satisfiability Modulo Theories (SMT)

- Automate reasoning in (fragments of) first-order logic (FOL)

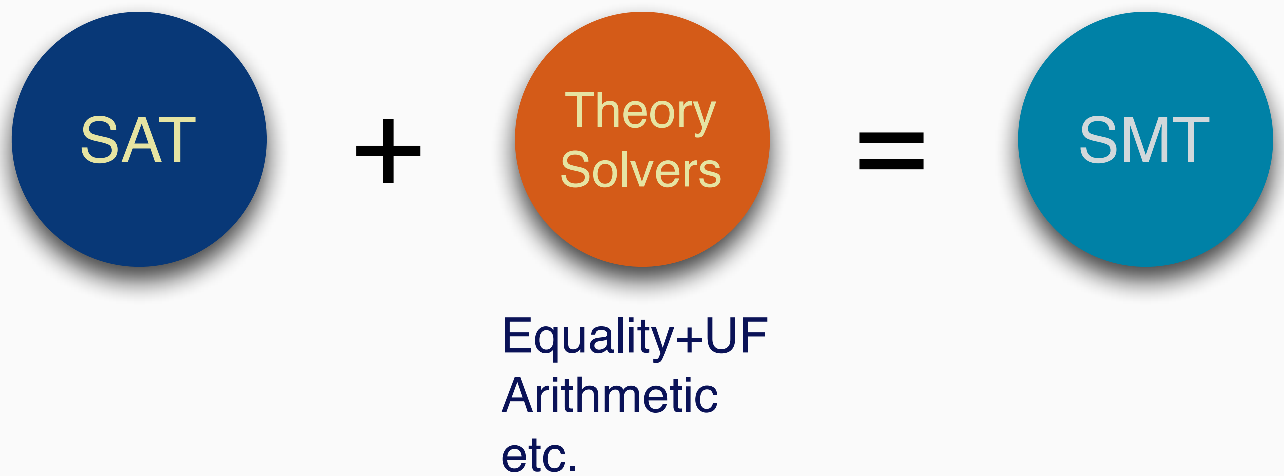


Equality+UF
Arithmetic
etc.

- Problem representation in propositional logic (PL):
  - Positive: Efficient (in practice) SAT algorithms
  - Negative: Expressiveness via CNF encodings

Satisfiability Modulo Theories (SMT)

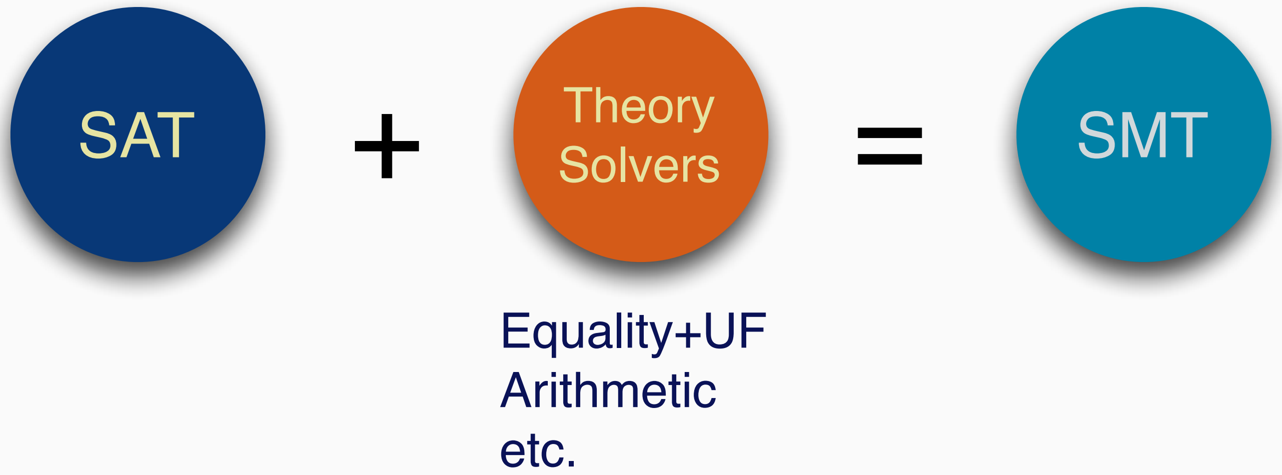
- Automate reasoning in (fragments of) first-order logic (FOL)



- Problem representation in propositional logic (PL):
    - Positive: Efficient (in practice) SAT algorithms
    - Negative: Expressiveness via CNF encodings
- PL + domain-specific reasoning
    - Positive: Improved expressiveness
    - Negative: Can be (far) less efficient than SAT

Satisfiability Modulo Theories (SMT)

- Automate reasoning in (fragments of) first-order logic (FOL)



- Problem representation in propositional logic (PL):
  - Positive: Efficient (in practice) SAT algorithms
  - Negative: Expressiveness via CNF encodings
- PL + domain-specific reasoning
  - Positive: Improved expressiveness
  - Negative: Can be (far) less efficient than SAT
- **Note:** Standard definitions of FOL apply

An example

- All x_i variables integer

An example

- All x_i variables integer
- Solve:

$$((x_4 - x_2 \leq 3) \vee (x_4 - x_3 \geq 5)) \wedge (x_4 - x_3 \leq 6) \wedge$$
$$(x_1 - x_2 \leq -1) \wedge (x_1 - x_3 \leq -2) \wedge (x_1 - x_4 \leq -1) \wedge (x_2 - x_1 \leq 2) \wedge$$
$$(x_3 - x_2 \leq -1) \wedge ((x_3 - x_4 \leq -2) \vee (x_4 - x_3 \geq 2))$$

An example

- All x_i variables integer
- Solve:

$$((x_4 - x_2 \leq 3) \vee (x_4 - x_3 \geq 5)) \wedge (x_4 - x_3 \leq 6) \wedge$$
$$(x_1 - x_2 \leq -1) \wedge (x_1 - x_3 \leq -2) \wedge (x_1 - x_4 \leq -1) \wedge (x_2 - x_1 \leq 2) \wedge$$
$$(x_3 - x_2 \leq -1) \wedge ((x_3 - x_4 \leq -2) \vee (x_4 - x_3 \geq 2))$$

- Integer difference logic (with Boolean structure)

An example

- All x_i variables integer
- Solve:

$$
\begin{aligned}
&((x_4 - x_2 \leq 3) \vee (x_4 - x_3 \geq 5)) \wedge (x_4 - x_3 \leq 6) \wedge \\
&(x_1 - x_2 \leq -1) \wedge (x_1 - x_3 \leq -2) \wedge (x_1 - x_4 \leq -1) \wedge (x_2 - x_1 \leq 2) \wedge \\
&(x_3 - x_2 \leq -1) \wedge ((x_3 - x_4 \leq -2) \vee (x_4 - x_3 \geq 2))
\end{aligned}
$$

- Integer difference logic (with Boolean structure)
- Unsatisfiable (**Why?**)

Another example

- All $t_{i,j}$ variables integer

Another example

- All $t_{i,j}$ variables integer
- Solve:

$$
\begin{aligned}
&(t_{1,1} \geq 0) \wedge (t_{1,2} \geq t_{1,1} + 2) \wedge (t_{1,2} + 1 \leq 8) \wedge \\
&(t_{2,1} \geq 0) \wedge (t_{2,2} \geq t_{1,1} + 3) \wedge (t_{2,2} + 1 \leq 8) \wedge \\
&(t_{3,1} \geq 0) \wedge (t_{3,2} \geq t_{1,1} + 2) \wedge (t_{3,2} + 3 \leq 8) \wedge \\
&((t_{1,1} \geq t_{2,1} + 3) \vee (t_{2,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{1,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{2,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{2,1} + 3)) \wedge \\
&((t_{1,2} \geq t_{2,2} + 1) \vee (t_{2,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{1,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{2,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{2,2} + 1))
\end{aligned}
$$

Another example

- All $t_{i,j}$ variables integer
- Solve:

$$\begin{array}{l}(t_{1,1} \geq 0) \land (t_{1,2} \geq t_{1,1} + 2) \land (t_{1,2} + 1 \leq 8) \land \\ (t_{2,1} \geq 0) \land (t_{2,2} \geq t_{1,1} + 3) \land (t_{2,2} + 1 \leq 8) \land \\ (t_{3,1} \geq 0) \land (t_{3,2} \geq t_{1,1} + 2) \land (t_{3,2} + 3 \leq 8) \land \\ ((t_{1,1} \geq t_{2,1} + 3) \lor (t_{2,1} \geq t_{1,1} + 2)) \land \\ ((t_{1,1} \geq t_{3,1} + 2) \lor (t_{3,1} \geq t_{1,1} + 2)) \land \\ ((t_{2,1} \geq t_{3,1} + 2) \lor (t_{3,1} \geq t_{2,1} + 3)) \land \\ ((t_{1,2} \geq t_{2,2} + 1) \lor (t_{2,2} \geq t_{1,2} + 1)) \land \\ ((t_{1,2} \geq t_{3,2} + 3) \lor (t_{3,2} \geq t_{1,2} + 1)) \land \\ ((t_{2,2} \geq t_{3,2} + 3) \lor (t_{3,2} \geq t_{2,2} + 1))\end{array}$$

- Another example of integer difference logic (with Boolean structure)

Another example

- All $t_{i,j}$ variables integer
- Solve:

$$
\begin{aligned}
&(t_{1,1} \geq 0) \wedge (t_{1,2} \geq t_{1,1} + 2) \wedge (t_{1,2} + 1 \leq 8) \wedge \\
&(t_{2,1} \geq 0) \wedge (t_{2,2} \geq t_{1,1} + 3) \wedge (t_{2,2} + 1 \leq 8) \wedge \\
&(t_{3,1} \geq 0) \wedge (t_{3,2} \geq t_{1,1} + 2) \wedge (t_{3,2} + 3 \leq 8) \wedge \\
&((t_{1,1} \geq t_{2,1} + 3) \vee (t_{2,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{1,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{1,1} + 2)) \wedge \\
&((t_{2,1} \geq t_{3,1} + 2) \vee (t_{3,1} \geq t_{2,1} + 3)) \wedge \\
&((t_{1,2} \geq t_{2,2} + 1) \vee (t_{2,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{1,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{1,2} + 1)) \wedge \\
&((t_{2,2} \geq t_{3,2} + 3) \vee (t_{3,2} \geq t_{2,2} + 1))
\end{aligned}
$$

- Another example of integer difference logic (with Boolean structure)
- Satisfiable, with model: $t_{1,1} = 5; t_{1,2} = 7; t_{2,1} = 2; t_{2,2} = 6; t_{3,1} = 0; t_{3,2} = 7;$

Outline

Preliminaries

- Classification Problems in ML
- Logic Overview
- Logic & Optimization
- Reasoning Beyond Propositional Logic
- Additional Concepts

Logic Encodings of ML Models

Entailment

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

Entailment

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

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

Entailment

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

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

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

Entailment

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

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

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

Entailment

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

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

  - An example:
    - $\mathbb{F} = \{0, 1\}^2$
    - $\varphi(x_1, x_2) = x_1 \vee \neg x_2$
    - Clearly, $x_1 \vDash \varphi$ and $\neg x_2 \vDash \varphi$
  - Another example:
    - $\mathbb{F} = \{0, 1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly, $x_1 \wedge x_2 \vDash \varphi$ and $x_1 \wedge x_3 \vDash \varphi$
- For non-boolean feature spaces, we let φ_c denote the predicate $\varphi(\mathbf{x}) = c$, i.e. $\varphi_c(\mathbf{x}) \in \{0, 1\}$

Prime implicants & implicates

- A conjunction of literals π (which will be viewed as a set of literals where convenient) is a **prime implicant** of some function φ if,
    1. $\pi \models \varphi$
    2. For any $\pi' \subsetneq \pi$, $\pi' \not\models \varphi$

Prime implicants & implicates

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

Prime implicants & implicates

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

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

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

Recap tools of the trade

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

Recap tools of the trade

- SAT: decision problem for propositional logic
  - Formulas most often represented in CNF
  - There are optimization variants: MaxSAT, PBO, MinSAT, etc.
  - There are quantified variants: QBF, QMaxSAT, etc.
- SMT: decision problem for (decidable) fragments of first-order logic (FOL)
  - There are optimization variants: MaxSMT, etc.
  - There are quantified variants
- MILP: decision/optimization problems defined on conjunctions of linear inequalities over integer & real-valued variables
- CP: constraint programming
  - There are optimization/quantified variants
- Background on SAT/SMT:
  - `https://alexeyignatiev.github.io/ssa-school-2019/`
  - `https://alexeyignatiev.github.io/ijcai19tut/`

Outline

Preliminaries

Logic Encodings of ML Models

Rules with ordinal features

- Example ML model:

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

  Rules:

  IF $2x_1 + x_2 > 0$ THEN predict $\boxplus$

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

Rules with ordinal features

- Example ML model:

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

  Rules:

$$
\begin{array}{llll}
\text{IF} & 2x_1 + x_2 > 0 & \text{THEN} & \text{predict } \boxplus \\
\text{IF} & 2x_1 - x_2 \leq 0 & \text{THEN} & \text{predict } \boxminus
\end{array}
$$

- **Q:** Can the model predict both $\boxplus$ and $\boxminus$ for some instance?

Rules with ordinal features

- Example ML model:

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

  Rules:

  $$\text{IF} \quad 2x_1 + x_2 > 0 \quad \text{THEN} \quad \text{predict } \boxplus$$
  $$\text{IF} \quad 2x_1 - x_2 \leq 0 \quad \text{THEN} \quad \text{predict } \boxminus$$

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

Rules with ordinal features

- Example ML model:

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

  Rules:

$$\begin{array}{llll} \text{IF} & 2x_1 + x_2 > 0 & \text{THEN} & \text{predict } \boxplus \\ \text{IF} & 2x_1 - x_2 \leq 0 & \text{THEN} & \text{predict } \boxminus \end{array}$$

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

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

… and solve with SMT solver

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

Decision sets

- Example ML model:

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

  Rules:

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

Decision sets

- Example ML model:

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

  Rules:

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

- **Q:** Can the model predict both $\boxplus$ and $\boxminus$ for some instance?

Decision sets

- Example ML model:

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

  Rules:

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

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

Decision sets

- Example ML model:

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

  Rules:

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

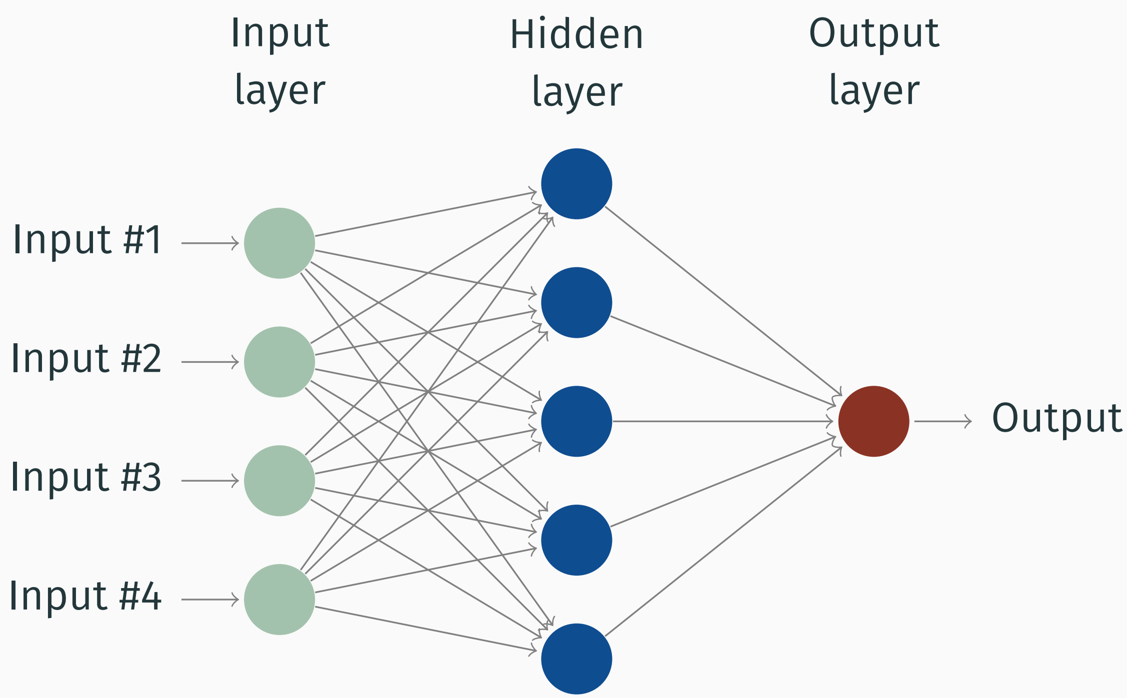
- **Q:** Can the model predict both $\boxplus$ and $\boxminus$ for some instance?
  - Yes, certainly: pick $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
  - A formalization:

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

… and solve with SAT solver (after clausification)

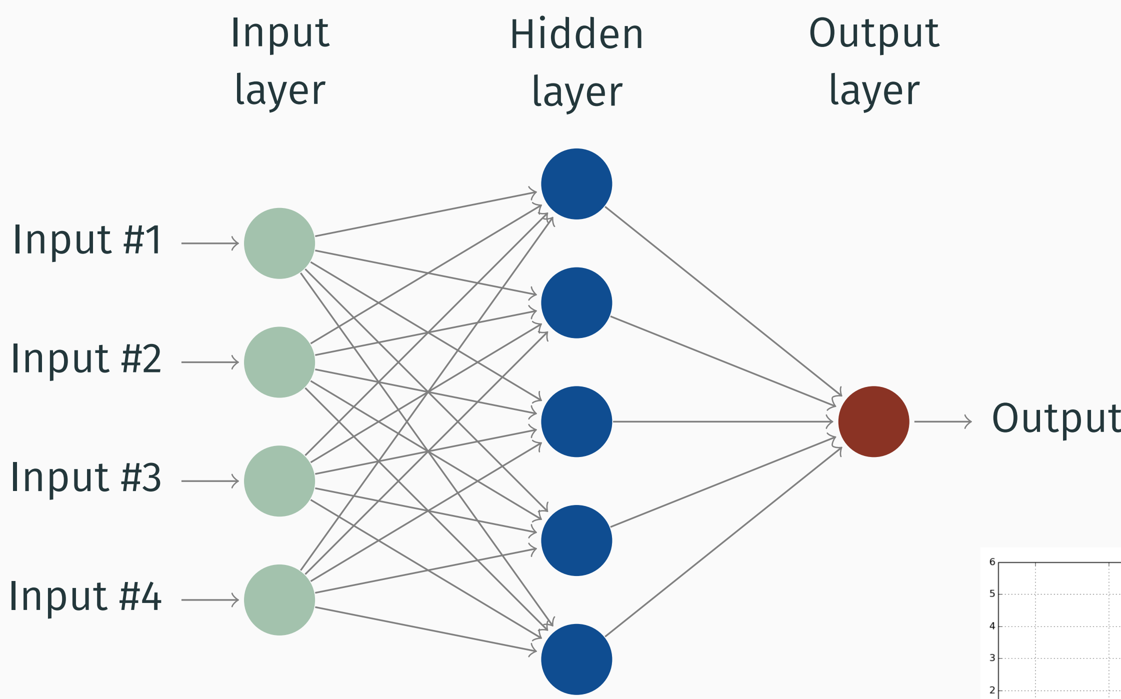
$\therefore$ There exists a model iff there exists a point in feature space yielding both predictions

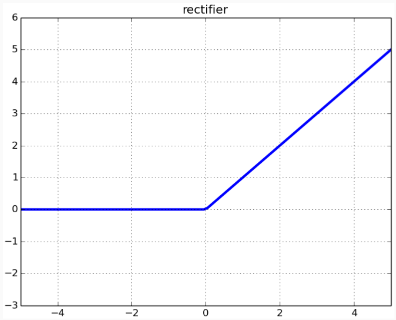
Neural networks



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

Neural networks





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

[NH10]

Encoding NNs using MILP

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

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

Encoding NNs using MILP

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

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

Encoding each **block**: [FJ18]

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

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

Encoding NNs using MILP

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

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

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

Encoding each **block**: [FJ18]

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

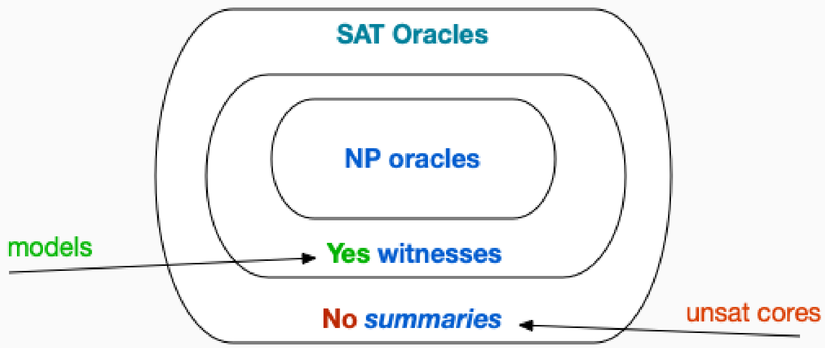
Simpler encodings exist, but **not** as effective [KBD+17]

Oracle-based problem solving

- Many problems are **not** decision problems

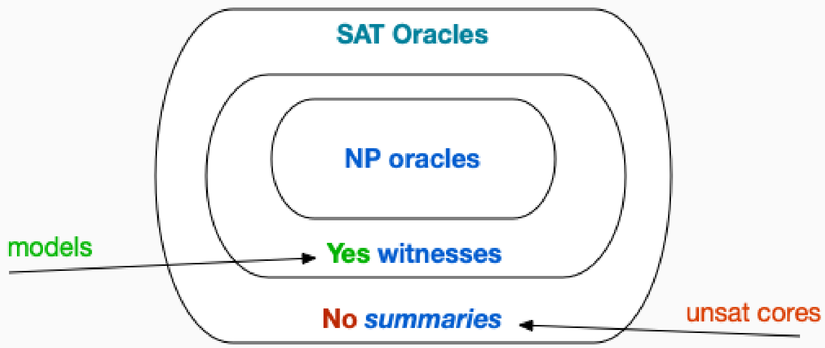
Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
  - Find one minimal set
  - Enumerate minimal/optimal solutions
  - Other problems



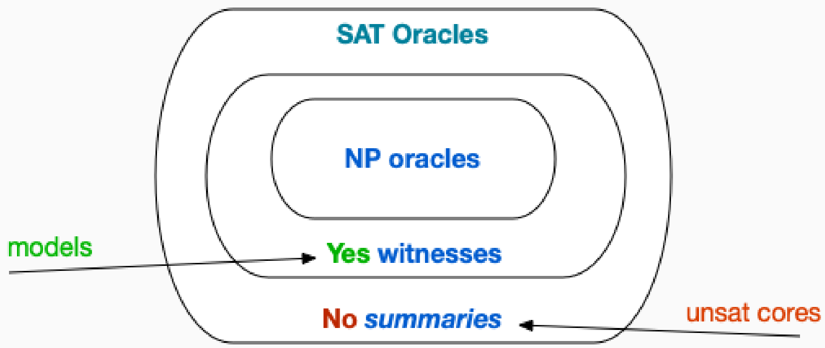
Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
    - Maximum satisfiability (MaxSAT), pseudo-boolean optimization (PBO)
    - But also MaxSMT, etc.
  - Find one minimal set
  - Enumerate minimal/optimal solutions
  - Other problems



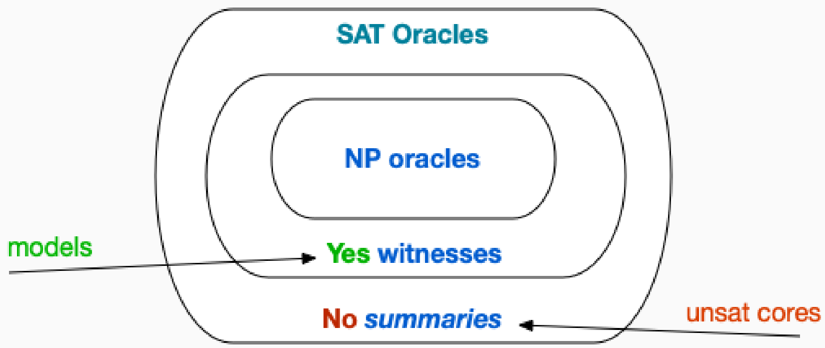
Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
    - Maximum satisfiability (MaxSAT), pseudo-boolean optimization (PBO)
    - But also MaxSMT, etc.
  - Find one minimal set
    - Reason about inconsistency: MUSes/MCSes
    - Compile knowledge: prime implicants/implicates
  - Enumerate minimal/optimal solutions
  - Other problems



Oracle-based problem solving

- Many problems are **not** decision problems
- Use decision procedures as **oracles** for
  - Optimize some cost function
    - Maximum satisfiability (MaxSAT), pseudo-boolean optimization (PBO)
    - But also MaxSMT, etc.
  - Find one minimal set
    - Reason about inconsistency: MUSes/MCSes
    - Compile knowledge: prime implicants/implicates
  - Enumerate minimal/optimal solutions
    - Enumerate MaxSAT solutions
    - Enumerate primes, MUSes, MCSes
  - Other problems
    - Propositional abduction
    - Etc.



Questions?

References i

[Coo71] Stephen A. Cook.
The complexity of theorem-proving procedures.
In *STOC*, pages 151–158. ACM, 1971.

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

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

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

Interpretable Classification

Given training data, learn function that correctly classifies that data, performs suitably well on unseen data, and offers human-interpretable functions for the predictions made

Interpretable Classification

Given training data, learn function that correctly classifies that data, performs suitably well on unseen data, and offers human-interpretable functions for the predictions made

Given training data, learn **decision sets/decision trees** that correctly classify that data, perform suitably well on unseen data, and offer human-interpretable functions for the predictions made

Recipe

Step 1 Discretization of the training and test dataset

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Recipe

- Step 1 Discretization of the training and test dataset
- Step 2 Define the grammar of the classifier
- Step 3 Hard Constraints to capture structure of the rules

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Step 3 Hard Constraints to capture structure of the rules

Step 4 Hard Constraints to capture evaluation of rules: A rule must

- return True on positive example and False on negative example

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Step 3 Hard Constraints to capture structure of the rules

Step 4 Hard Constraints to capture evaluation of rules: A rule must

- return True on positive example and False on negative example

Step 5 Soft Constraints

- Minimize the size of rules

Recipe

Step 1 Discretization of the training and test dataset

Step 2 Define the grammar of the classifier

Step 3 Hard Constraints to capture structure of the rules

Step 4 Hard Constraints to capture evaluation of rules: A rule must

- return True on positive example and False on negative example

Step 5 Soft Constraints

- Minimize the size of rules

Step 6 Rely on progress in SAT and MaxSAT solving over the past decade

Outline

Discretization

Classification via Decision Sets

Decision Sets via MaxSAT

Incremental learning

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

- Suppose Height can range between 50 and 250 cm and weight ranges between 100 and 300.
- Do we need variable for every value of H and W?

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

- Suppose Height can range between 50 and 250 cm and weight ranges between 100 and 300.
- Do we need variable for every value of H and W?
- One-hot encoding: Only introduce variables to differentiate two distinct data points.
  - Variables corresponding to $H \geq 170$, $H \geq 165$, $H \geq 172$, $H \geq 175$ suffice
  - Variables corresponding to $W \geq 200$ and $W \geq 180$

Discretization

Ex.	Height (H)	Weight (W)	Risk (R)
e_1	160	210	0
e_2	175	210	0
e_3	170	190	1
e_4	166	190	0
e_5	172	170	1

Ex.	$H \geq 170$	$H \geq 165$	$H \geq 172$	$H \geq 175$	$W > 200$	$W > 180$	Risk (R)
e_1	0	0	0	0	1	0	0
e_2	1	0	1	1	1	0	0
e_3	1	1	0	0	0	1	1
e_4	0	1	0	0	0	1	0
e_5	1	1	1	0	0	0	1

Outline

Discretization

Classification via Decision Sets

Decision Sets via MaxSAT

Incremental learning

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- Binary **features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - $f_1 \triangleq$ V, $f_2 \triangleq$ C, $f_3 \triangleq$ M, and $f_4 \triangleq$ E
  - Literals: f_r and $\neg f_r$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- Binary **features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - $f_1 \triangleq$ V, $f_2 \triangleq$ C, $f_3 \triangleq$ M, and $f_4 \triangleq$ E
  - Literals: f_r and $\neg f_r$
- **Feature space**: $\mathcal{U} \triangleq \prod_{r=1}^{K}\{f_r, \neg f_r\}$

Classification problems

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Training data (or **examples**): $\mathcal{E} = \{e_1, \ldots, e_M\}$
- Binary **features**: $\mathcal{F} = \{f_1, \ldots, f_K\}$
  - $f_1 \triangleq$ V, $f_2 \triangleq$ C, $f_3 \triangleq$ M, and $f_4 \triangleq$ E
  - Literals: f_r and $\neg f_r$
- **Feature space**: $\mathcal{U} \triangleq \prod_{r=1}^{K} \{f_r, \neg f_r\}$
- Binary classification: $\mathcal{C} = \{c_0 = 0, c_1 = 1\}$
  - $\mathcal{E}$ partitioned into $\mathcal{E}^-$ and $\mathcal{E}^+$

Example

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Binary features: $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$
  - $f_1 \triangleq \mathrm{V}$, $f_2 \triangleq \mathrm{C}$, $f_3 \triangleq \mathrm{M}$, and $f_4 \triangleq \mathrm{E}$
- e_1 is represented by the 2-tuple (π_1, ς_1),
  - $\pi_1 = (\neg\mathrm{V}, \neg\mathrm{C}, \mathrm{M}, \neg\mathrm{E})$
  - $\varsigma_1 = 0$
- $\mathcal{U} = \{\mathrm{V}, \neg\mathrm{V}\} \times \{\mathrm{C}, \neg\mathrm{C}\} \times \{\mathrm{M}, \neg\mathrm{M}\} \times \{\mathrm{E}, \neg\mathrm{E}\}$

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K} \{f_r, \neg f_r\}$

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K}\{f_r, \neg f_r\}$
- A **rule** is a 2-tuple (π, c), with itemset $\pi \in \mathcal{I}$, and class $c \in \mathcal{C}$
  Rule (π, c) interpreted as:

  IF all specified literals in π are true, **THEN** pick class c

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K}\{f_r, \neg f_r\}$
- A **rule** is a 2-tuple (π, c), with itemset $\pi \in \mathcal{I}$, and class $c \in \mathcal{C}$
  Rule (π, c) interpreted as:

  IF all specified literals in π are true, **THEN** pick class c

- A **decision set** $\mathbb{S}$ is a finite set of rules – **unordered**

Itemsets & decision sets

- Given $\mathcal{F}$, an **itemset** π is an element of $\mathcal{I} \triangleq \prod_{r=1}^{K}\{f_r, \neg f_r\}$
- A **rule** is a 2-tuple (π, c), with itemset $\pi \in \mathcal{I}$, and class $c \in \mathcal{C}$
  Rule (π, c) interpreted as:

  IF all specified literals in π are true, **THEN** pick class c

- A **decision set** $\mathbb{S}$ is a finite set of rules – **unordered**
- A rule of the form $\mathfrak{D} \triangleq (\emptyset, c)$ denotes the **default rule** of a decision set $\mathbb{S}$
  - Default rule is **optional** and used **only** when other rules do not apply on some feature space point
  - In this talk, we will seek to learn

Example

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Rule 1: $((\neg \mathrm{M}, \neg \mathrm{E}), c_1)$
  - Meaning: **if** ¬Meeting and ¬Expo **then** Hike
- Rule 2: $((\mathrm{V}, \neg \mathrm{C}), c_1)$
  - Meaning: **if** Vacation and ¬Concert **then** Hike
- Rule 3: $((\neg \mathrm{V}, \mathrm{M}), c_0)$
  - Meaning: **if** ¬Vacation and Meeting **then** ¬Hike

Example

Ex.	Vacation (V)	Concert (C)	Meeting (M)	Expo (E)	Hike (H)
e_1	0	0	1	0	0
e_2	1	0	0	0	1
e_3	0	0	1	1	0
e_4	1	0	0	1	1
e_5	0	1	1	0	0
e_6	0	1	1	1	0
e_7	1	1	0	1	1

- Rule 1: $((\neg \text{M}, \neg \text{E}), c_1)$
  - Meaning: **if** ¬Meeting and ¬Expo **then** Hike
- Rule 2: $((\text{V}, \neg \text{C}), c_1)$
  - Meaning: **if** Vacation and ¬Concert **then** Hike
- Rule 3: $((\neg \text{V}, \text{M}), c_0)$
  - Meaning: **if** ¬Vacation and Meeting **then** ¬Hike
- Default rule: $(\emptyset, c_0)$
  - Meaning: if all other rules do not apply, then pick ¬Hike

Succinct explanations

- If a rule fires, the set of literals represents the **explanation** for the predicted class
  - Explanation is **succinct** : **only** the literals in the rule used; independent of example
- For the default class, **must** pick one **falsified** literal in **every** rule that predicts a different class
  - Explanation is **not succinct** : explanation depends on **each** example
- **Obs:** **Uninteresting** to predict c_1 as **negation** of c_0 (and vice-versa)
  - Explanations also **not** succinct

Stating our goals

- Assumptions:
  - Also, let $\mathcal{E}^- \wedge \mathcal{E}^+ \models \bot$

Stating our goals

- Assumptions:
  - Also, let $\mathcal{E}^- \wedge \mathcal{E}^+ \vDash \perp$

- DNF functions to compute:
  - F^0 for predicting c_0, while **ensuring** $\mathcal{E}^- \vDash F^0$
  - F^1 for predicting c_1, while **ensuring** $\mathcal{E}^+ \vDash F^1$

Different Possibilities

- MinDS_0:

  Find the **smallest** DNF formulas F^0 and F^1 such that:

  1. $\mathcal{E}^- \models F^0$
  2. $\mathcal{E}^+ \models F^1$
  3. $F^1 \leftrightarrow F^0 \models \bot$

  – **Obs:** MinDS_0 ensures **succinct** explanations

    - Computes F^0 and F^1 (i.e. **no** negation) and **no** default rule

Different Possibilities

- MinDS_0:

  Find the **smallest** DNF formulas F^0 and F^1 such that:

  1. $\mathcal{E}^- \vDash F^0$
  2. $\mathcal{E}^+ \vDash F^1$
  3. $F^1 \leftrightarrow F^0 \vDash \perp$

  - **Obs:** MinDS_0 ensures **succinct** explanations
    - Computes F^0 and F^1 (i.e. **no** negation) and **no** default rule
- MinDS_3: Minimize F^1 such that
  1. $\mathcal{E}^+ \vDash F^1$
  2. $F^1 \wedge \mathcal{E}^- \vDash \perp$

  - No succinct explanations for F^0
- MinDS_4: Minimize F^0 such that
  1. $\mathcal{E}^- \vDash F^0$
  2. $F^0 \wedge \mathcal{E}^+ \vDash \perp$

  - No succinct explanations for F^1

Outline

Discretization

Classification via Decision Sets

Decision Sets via MaxSAT

- Handling Noise
- Addressing Scalability Challenge
- Experimental Results

Incremental learning

Boolean Formulation of MinDS$_3$

- DNF representation for F^1
- Consider N terms
  - $F^1 := F_1^1 \vee F_2^1 \cdots F_N^1$, where

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge((b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

    - If $b_{i,1}$ is true, then f_1 is in F_i^1.
    - If $c_{i,1}$ is true, then $\neg f_1$ is in F_i^1.
    - If $d_{i,1}$ is true, then f_1 and $\neg f_1$ do not appear in F_i^1
  - F_i^1 is a DNF term if exactly one of $\{b_{i,r}, c_{i,r}, d_{i,r}\}$ is true for each r.

Boolean Formulation of MinDS$_3$

- DNF representation for F^1

- Consider N terms
  - $F^1 := F_1^1 \vee F_2^1 \cdots F_N^1$, where

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge((b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

    - If $b_{i,1}$ is true, then f_1 is in F_i^1.
    - If $c_{i,1}$ is true, then $\neg f_1$ is in F_i^1.
    - If $d_{i,1}$ is true, then f_1 and $\neg f_1$ do not appear in F_i^1
  - F_i^1 is a DNF term if exactly one of $\{b_{i,r}, c_{i,r}, d_{i,r}\}$ is true for each r.

- Goal: Find values of $\{b_{i,j}, c_{i,j}, d_{i,j}\}$

MaxSAT Formulation

- Recall
  - $\sigma(r, q)$: value of feature f_r for e_q

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge ((b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Structural Constraints: $\bigwedge_{i,r} \mathit{ExactlyOne}(b_{i,r}, c_{i,r}, d_{i,r})$
- $\mathcal{E}^+ \models F^1$: For $e_q \in \mathcal{E}^+$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 1$ (Hard)
- $F^1 \wedge \mathcal{E}^- \models \bot$: For $e_q \in \mathcal{E}^-$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 0$ (Hard)
- Soft Constraints: $\mathcal{S}_{i,r} := (\neg b_{i,r}) c_{i,r}$; $\qquad W(\mathcal{S}_{i,r}) = 1$
  - Minimize the size of each term
  - Can have different objective functions

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F_1^1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F_2^1 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. For e_1, we have $F^1[\bigwedge_r f_r \mapsto \sigma(r, q)] =$
   $((c_{1,1} \vee d_{1,1}) \wedge (b_{1,2} \vee d_{1,2}) \wedge (c_{1,3} \vee d_{1,3})) \vee$

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F_1^1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F_2^1 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. For e_1, we have $F^1[\bigwedge_r f_r \mapsto \sigma(r, q)] =$
   $((c_{1,1} \vee d_{1,1}) \wedge (b_{1,2} \vee d_{1,2}) \wedge (c_{1,3} \vee d_{1,3})) \vee$
   $((c_{2,1} \vee d_{2,1}) \wedge (b_{2,2} \vee d_{2,2}) \wedge (c_{2,3} \vee d_{2,3}))$

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F_1^1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F_2^1 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. Suppose, MaxSAT solver returns $b_{1,1} = c_{1,2} = d_{1,3} = d_{2,1} = d_{2,3} = b_{2,3} = 1$; then the rule is

Example

Ex.	Vacation (V) f_1	Meeting (M) f_2	Expo (E) f_3	Hike (H) Label
e_1	0	1	0	1
e_2	1	0	0	0
e_3	0	1	1	1

Suppose, we want to learn F^1 of one term ,i.e., $N = 1$. Remember,
$F_1^1 = (b_{1,1} \cdot f_1 \vee c_{1,1} \cdot \neg f_1 \vee d_{1,1}) \vee (b_{1,2} \cdot f_2 \vee c_{1,2} \cdot \neg f_2 \vee d_{1,2}) \wedge (b_{1,3} \cdot f_3 \vee c_{1,3} \cdot \neg f_3 \vee d_{1,3})$
$F_2^1 = (b_{2,1} \cdot f_1 \vee c_{2,1} \cdot \neg f_1 \vee d_{2,1}) \vee (b_{2,2} \cdot f_2 \vee c_{2,2} \cdot \neg f_2 \vee d_{2,2}) \vee (b_{2,3} \cdot f_3 \vee c_{2,3} \cdot \neg f_3 \vee d_{2,3})$

1. Suppose, MaxSAT solver returns $b_{1,1} = c_{1,2} = d_{1,3} = d_{2,1} = d_{2,3} = b_{2,3} = 1$; then the rule is $F^1 = (f_1 \wedge \neg f_2) \vee (f_2)$

Tools

- The MaxSAT formulation is NP-hard
- Use Local search based approaches [LBS, KDD-16]
  - Local search-based:
    git clone `git@github.com:jirifilip/pyIDS.git`
- Use MaxSAT solvers [IPNM, IJCAR-18]
  - Significant progress in MaxSAT solving over the past decade
  - Usage of symmetry breaking predicates
  - MaxSAT-based Decision sets
    git clone `https://github.com/alexeyignatiev/minds`

Tools

- The MaxSAT formulation is NP-hard
- Use Local search based approaches [LBS, KDD-16]
  - Local search-based:
    git clone `git@github.com:jirifilip/pyIDS.git`
- Use MaxSAT solvers [IPNM, IJCAR-18]
  - Significant progress in MaxSAT solving over the past decade
  - Usage of symmetry breaking predicates
  - MaxSAT-based Decision sets
    git clone `https://github.com/alexeyignatiev/minds`
- Results: Over a set of 49 instances, local-search based approach can handle only 2 instances while MaxSAT based approach can optimal decision sets of 42 instances [IPNM, IJCAR-18]

Looking Beyond: Handling Noise

- Noisy data sets: collection of data, non-existence of perfect rules
  - The optimal decision sets are too large.

Looking Beyond: Handling Noise

- Noisy data sets: collection of data, non-existence of perfect rules
  - The optimal decision sets are too large.
- MinDS$_3$: Minimize F^1 and such that
  1. $\mathcal{E}^+ \models F^1$
  2. $F^1 \wedge \mathcal{E}^- \models \perp$
  - No succinct explanations for F^0
- Noisy MinDS$_3$: Minimize F^1, such that
  1. $\mathbb{1}_q = 1$ if $e_q \not\models F^1$ for $e_q \in \mathcal{E}^+$ or $e_q \models F^1$ for $e_q \in \mathcal{E}^+$
  2. Minimize $|F| + \lambda \sum_q \mathbb{1}_q$

MaxSAT Formulation for Noisy Setting

[MM, CP-18]

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge (b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Notations
  - Variables: $\{b_{i,r}, c_{i,r}, d_{i,r}, \eta_q\}$
  - e_q: example q
  - $\sigma(r, q)$: sign of feature f_r for e_q

MaxSAT Formulation for Noisy Setting

[MM, CP-18]

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge (b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Notations
  - Variables: $\{b_{i,r}, c_{i,r}, d_{i,r}, \eta_q\}$
  - e_q: example q
  - $\sigma(r, q)$: sign of feature f_r for e_q
- Hard Constraints:
  - Structural Constraints: $\bigwedge_{i,r} ExactlyOne(b_{i,r}, c_{i,r}, d_{i,r})$
  - $\mathcal{E}^+ \models F^1$: For $e_q \in \mathcal{E}^+$, $F^1[\bigwedge_r f_r \mapsto \sigma(r, q)] = 1 \oplus \eta_q$ (Hard)
  - $F^1 \wedge \mathcal{E}^- \models \bot$: For $e_q \in \mathcal{E}^-$, $F^1[\bigwedge_r f_r \mapsto \sigma(r, q)] = 0 \oplus \eta_q$ (Hard)

MaxSAT Formulation for Noisy Setting

[MM, CP-18]

$$F_i^1 = ((b_{i,1} \cdot f_1 \vee c_{i,1} \cdot \neg f_1 \vee d_{i,1}) \cdots \wedge (b_{i,r} \cdot f_r \vee c_{i,r} \cdot \neg f_r \vee d_{i,r}) \cdots \wedge (b_{i,K} \cdot f_K \vee c_{i,K} \cdot \neg f_K \vee d_{i,K}))$$

- Notations
  - Variables: $\{b_{i,r}, c_{i,r}, d_{i,r}, \eta_q\}$
  - e_q: example q
  - $\sigma(r,q)$: sign of feature f_r for e_q
- Hard Constraints:
  - Structural Constraints: $\bigwedge_{i,r} ExactlyOne(b_{i,r}, c_{i,r}, d_{i,r})$
  - $\mathcal{E}^+ \models F^1$: For $e_q \in \mathcal{E}^+$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 1 \oplus \eta_q$ (Hard)
  - $F^1 \wedge \mathcal{E}^- \models \bot$: For $e_q \in \mathcal{E}^-$, $F^1[\bigwedge_r f_r \mapsto \sigma(r,q)] = 0 \oplus \eta_q$ (Hard)
- Soft Constraints
  - Minimize the size of each term: $\mathcal{S}_{i,r} := (d_{i,r})$; $W(\mathcal{S}_{i,r}) = 1$
  - Minimize mis-classification: $\mathcal{T}_q := (\neg \eta_q)$ $W(\mathcal{T}_q) = 1$

Illustrative Example

- Iris Classification:
- Features: sepal length, sepal width, petal length, and petal width
- MLIC learned $\mathcal{R}=$
  1. (sepal length $\leq$ 6.3 $\wedge$ sepal width $\leq$ 3.0 $\wedge$ petal width $\geq$ 1.5) $\vee$
  2. (sepal width $\geq$ 2.7 $\wedge$ petal length $\leq$ 4.0 $\wedge$ petal width $\leq$ 1.2) $\vee$
  3. (petal length $>$ 5.0)

Accuracy

Dataset	Size	# Features	RIPPER	Log Reg	NN	RF	SVM	MLIC
ionosphere	350	564	0.886 (0.1)	0.909 (0.1)	0.926 (1.2)	0.909 (1.3)	0.886 (0.1)	0.889 (15.04)
parkinsons	190	392	0.868 (0.1)	0.884 (0.1)	0.921 (1.2)	0.895 (1.1)	0.879 (1.6)	0.895 (245)
Trans	740	64	0.78 (0.0)	0.759 (0.0)	0.788 (1.2)	0.788 (1.2)	0.765 (372.3)	0.797 (1177)
WDBC	560	540	0.961 (0.1)	0.936 (0.0)	0.961 (1.3)	0.943 (1.4)	0.955 (3.0)	0.946 (911)

Intepretability

Dataset	Examples	# Features	MLIC
ionosphere	350	564	5.5
parkinsons	190	392	6
Trans	740	64	4
WDBC	560	540	3.5

Scalability

How do we scale to tens of thousands of examples and features?

Primary Bottleneck Size of MaxSAT formula $\mathcal{O}(M \cdot N \cdot K)$ for a formula on M examples, N clauses and K features

Outline

Discretization

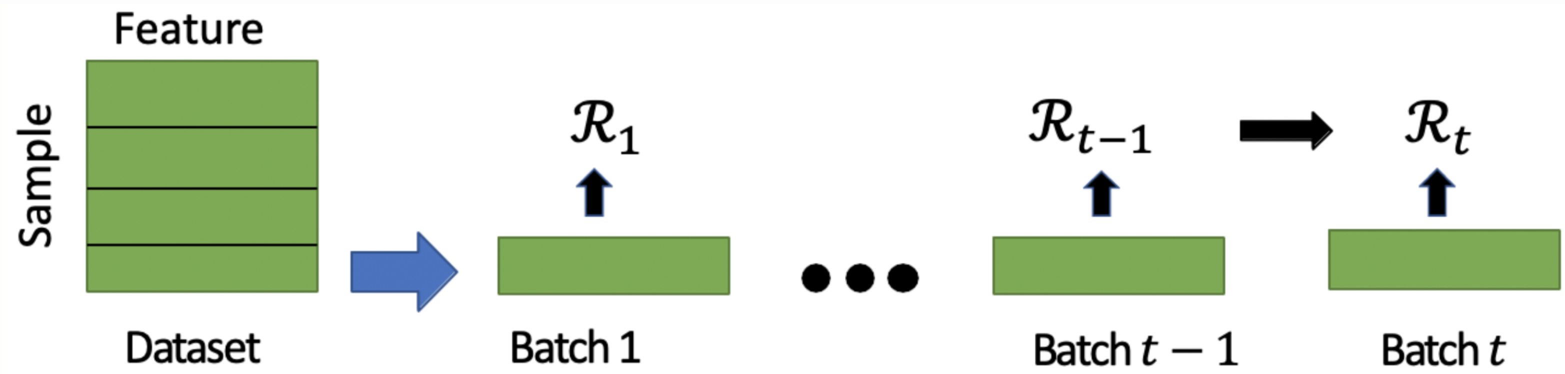
Classification via Decision Sets

Decision Sets via MaxSAT

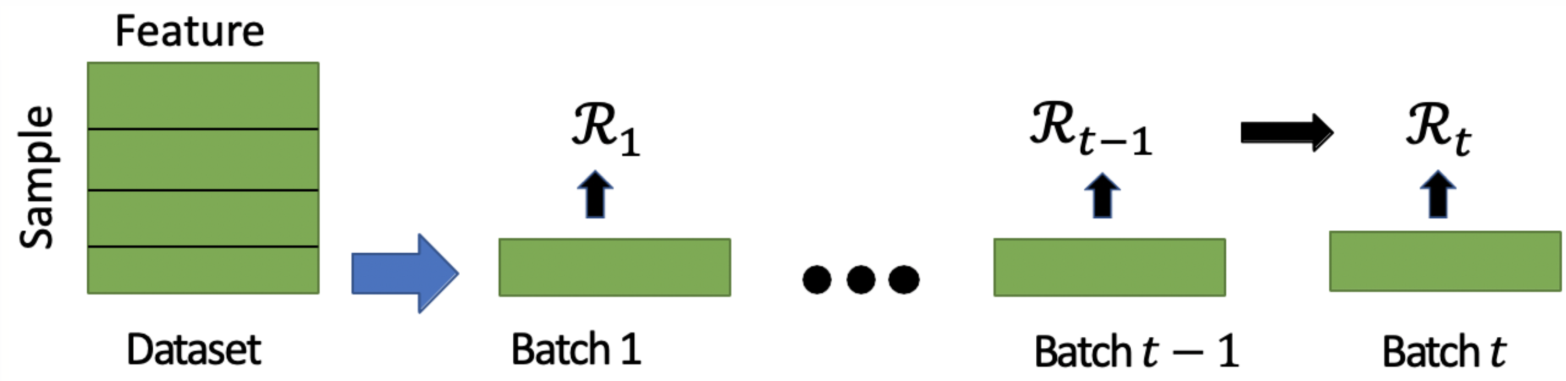
Incremental learning

IMLI: Incremental Rule-learning Approach

- The large formula size of the MaxSAT instance for the poor scalability
- The proposal of mini-batch incremental learning [Ghosh and M., AIES 19]



IMLI: Solution Technique - I



- We propose a mini-batch incremental learning framework with the following objective function on batch t

$$\min \sum_{i,j} (b_{i,j} \cdot I(b_{i,j}) + c_{i,j} \cdot I(c_{i,j}) + d_{i,j} \cdot I(d_{i,j})) + \lambda \sum_{q} \eta_q.$$

where indicator function $I(\cdot)$ is defined as follows.

$$I(b_{i,j}) = \begin{cases} -1 & \text{if } b_{i,j} \in \mathcal{R}_{t-1} \\ 1 & \text{otherwise} \end{cases}$$

Similarly, for $I(c_{i,j})$ and $I(d_{i,j})$

IMLI: Solution Technique - II

$(t-1)$-th batch

we learn assignment

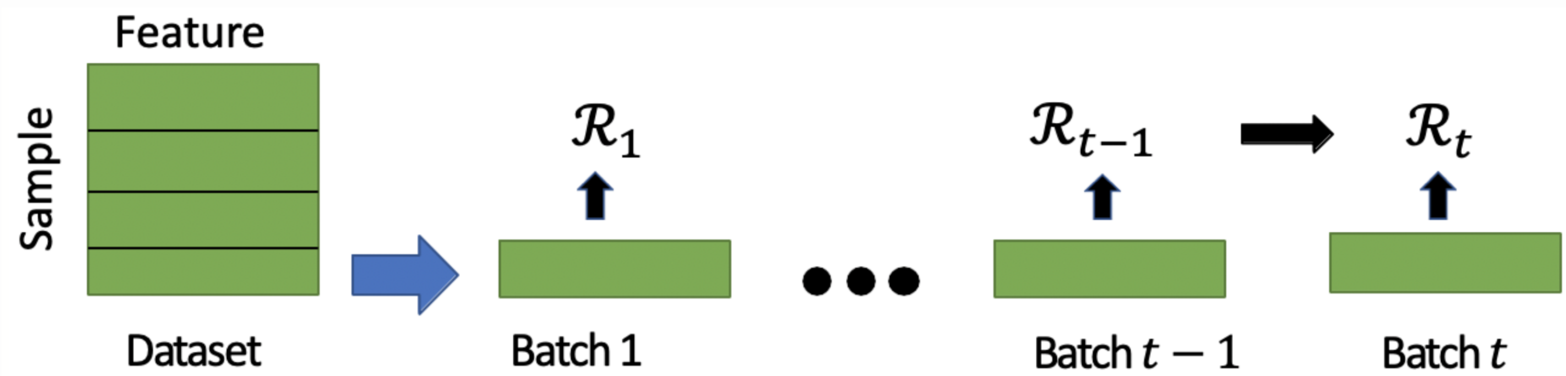
- $b_{1,1} = 0$
- $b_{1,2} = 1$
- $b_{2,1} = 0$
- $b_{2,2} = 1$

t-th batch

we construct soft unit clause

- $\neg b_{1,1}$
- $b_{1,2}$
- $\neg b_{2,1}$
- $b_{2,2}$

IMLI: Solution Technique-III



For M examples, N clauses, and K features,

- The number of clauses for each batch is $\mathcal{O}(\frac{M}{t} \cdot N \cdot K)$
  - Significant reduction from $\mathcal{O}(M \cdot N \cdot K)$

Accuracy and training time of different classifiers

Dataset	Size n	Features m	LR	SVC	RIPPER	IMLI
PIMA	768	134	75.32 (0.3s)	75.32 (0.37s)	75.32 (2.58s)	73.38 (0.74s)
Credit-default	30000	334	80.81 (6.87s)	80.69 (847.93s)	80.97 (20.37s)	79.41 (32.58s)
Twitter	49999	1050	95.67 (3.99s)	Timeout	95.56 (98.21s)	94.69 (59.67s)

Table: Each cell in the last 5 columns refers to test accuracy (%) and training time (s).

MLIC timed out on all the above instances

Size of rules of different rule-based classifiers

Dataset	RIPPER	IMLI
PIMA	8.25	3.5
Twitter	21.6	6
Credit	14.25	3

Table: Average size of the rules of different rule-based models.

IMLI generates shorter rules compared to other rule-based models

Example Rules

Rule for Pima Indians Diabetes Database

Tested positive for diabetes if :=
(Plasma glucose concentration $>$ 125 AND Triceps thickness $\leq$ 35 mm AND Diabetes pedigree function $>$ 0.259 AND Age $>$ 25 years)

Example Rules

Rule for Pima Indians Diabetes Database

Tested positive for diabetes if $:=$
(Plasma glucose concentration $>$ 125 AND Triceps thickness $\leq$ 35 mm AND Diabetes pedigree function $>$ 0.259 AND Age $>$ 25 years)

Rule for Parkinson's Disease Dataset

A person has Parkinson's disease if $:=$
(minimum vocal fundamental frequency $\leq$ 87.57 Hz OR minimum vocal fundamental frequency $>$ 121.38 Hz OR Shimmer:APQ3 $\leq$ 0.01 OR MDVP:APQ $>$ 0.02 OR D2 $\leq$ 1.93 OR NHR $>$ 0.01 OR HNR $>$ 26.5 OR spread2 $>$ 0.3) AND
(Maximum vocal fundamental frequency $\leq$ 200.41 Hz OR HNR $\leq$ 18.8 OR spread2 $>$ 0.18 OR D2 $>$ 2.92)

Recipe so far

- Discretization of the training and test dataset
- Hard Constraints to capture structure of the rules
- Hard Constraints to capture evaluation of rules: A rule must
  - EITHER return True on positive example and False on negative example
  - OR the noise variable is set to True
- Soft Constraints
  - Minimize the size of rules
  - Minimize the number of mis-classifications

From Decisions Sets to Decision Trees

[NIPM, IJCAI-18]

- Hard Constraints to capture structure of the rules
  - A leaf node has no children and is either 0 (False) or 1 (True)
  - A non-leaf node must have a child.
  - If the i-th node is a parent then it must have a child
  - All nodes (except root) must have a parent
  - Left edge corresponding to node with label f_r corresponds to $f_r = 0$
  - Right edge corresponding to node with label f_r corresponds to $f_r = 1$
- Evaluation along a path is just conjunction of edges
- Hard constraints to capture evaluation of rules
  - return True on positive example and False on negative example
- Exploitation of domain specific knowledge to improve encoding
  - Minimize the size of the trees
  - Minimize the number of mis-classifications

Lot of Exciting Research

- Janota, Morgado: SAT-Based Encodings for Optimal Decision Trees with Explicit Paths. SAT 2020: 501-518
- Verhaeghe, Nijssen, Pesant, Quimper, Schaus: Learning optimal decision trees using constraint programming. Constraints An Int. J. 25(3-4): 226-250 (2020)
- Aglin, Nijssen, Schaus: Learning Optimal Decision Trees Using Caching Branch-and-Bound Search. AAAI 2020: 3146-3153
- Aglin, Nijssen, Schaus: PyDL8.5: a Library for Learning Optimal Decision Trees. IJCAI 2020: 5222-5224
- Demirovic, Lukina, Hebrard, Chan, Bailey, Leckie, Ramamohanarao, P Stuckey: MurTree: Optimal Classification Trees via Dynamic Programming and Search. CoRR abs/2007.12652 (2020)
- Hu, Siala, Hebrard, Huguet: Learning Optimal Decision Trees with MaxSAT and its Integration in AdaBoost. IJCAI 2020: 1170-1176
- Avellaneda: Efficient Inference of Optimal Decision Trees. AAAI 2020: 3195-3202

From Decision Sets to Decision Lists

- Rule 1: $((\neg\text{M}, \neg\text{E}), c_1)$
  - Meaning: **if** ¬Meeting and ¬Expo **then** Hike
- Rule 2: $((\text{V}, \neg\text{C}), c_1)$
  - Meaning: **if** Vacation and ¬Concert **then** ¬Hike
- Decision List: Oredered List of Rules
- List A: Rule 1 followed by Rule 2
  - $V = 1, C = 0, M = 0, E = 0$
- List A Evaluation: Hike

From Decision Sets to Decision Lists

- Rule 1: $((\neg \mathrm{M}, \neg \mathrm{E}), c_1)$
  - Meaning: **if** ¬Meeting and ¬Expo **then** Hike
- Rule 2: $((\mathrm{V}, \neg \mathrm{C}), c_1)$
  - Meaning: **if** Vacation and ¬Concert **then** ¬Hike
- Decision List: Oredered List of Rules
- List A: Rule 1 followed by Rule 2
  - $V = 1, C = 0, M = 0, E = 0$
- List A Evaluation: Hike
- List B: Rule 2 followed by Rule 1

From Decision Sets to Decision Lists

- Rule 1: $((\neg\mathrm{M}, \neg\mathrm{E}), c_1)$
  - Meaning: **if** ¬Meeting and ¬Expo **then** Hike
- Rule 2: $((\mathrm{V}, \neg\mathrm{C}), c_1)$
  - Meaning: **if** Vacation and ¬Concert **then** ¬Hike

- Decision List: Oredered List of Rules
- List A: Rule 1 followed by Rule 2
  - $V = 1, C = 0, M = 0, E = 0$
- List A Evaluation: Hike
- List B: Rule 2 followed by Rule 1
- List B Evaluation: ¬Hike

Jinqiang Yu, Alexey Ignatiev, Pierre Le Bodic, Peter J. Stuckey: Optimal Decision Lists using SAT. CoRR abs/2010.09919 (2020)

Exciting Work

-

Conclusions & research directions

- SAT/MaxSAT-based solutions for computing (explainable) decision sets
  - Minimize the number of terms
  - Allows several different objective functions
- Far better than local search based approach

Conclusions & research directions

- SAT/MaxSAT-based solutions for computing (explainable) decision sets
  - Minimize the number of terms
  - Allows several different objective functions
- Far better than local search based approach

- Formalizations beyond Decisions sets and Decision Trees
  - Checklists [GMM, ECAI20]
  - The underlying approach can be applied
  - Exploitation of domain specific knowledge
- Scalability and handling very large data sets.

Tools

- Local search-based:
  git clone `git@github.com:jirifilip/pyIDS.git`
- MaxSAT-based Decision sets
  git clone `https://github.com/alexeyignatiev/minds`
- Noisy and Incremental: pip install rulelearning

Questions?

Part 3. Robustness of ML models

Nina Narodytska

Part 3. Robustness of Deep NNs

Nina Narodytska

Outline

Motivation

Adversarial attacks

Verification methods

SAT-based verification of Binarized NNs

Outline

Motivation

Why robustness?

Robustness of ML models

Interpretability of ML models

Why robustness?

Robustness of ML models

Interpretability of ML models

Why robustness?

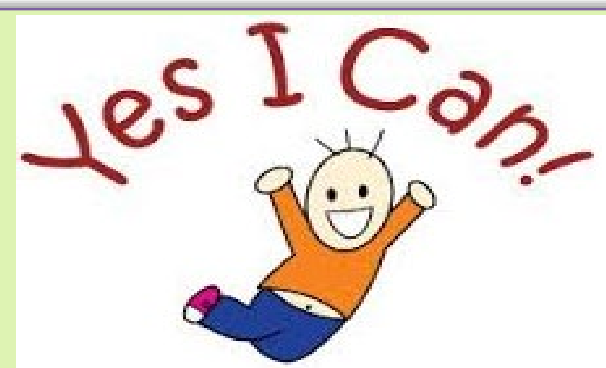
Robustness of ML models

Interpretability of ML models

Why robustness?

Robustness of ML models

Interpretability of ML models



Why robustness?

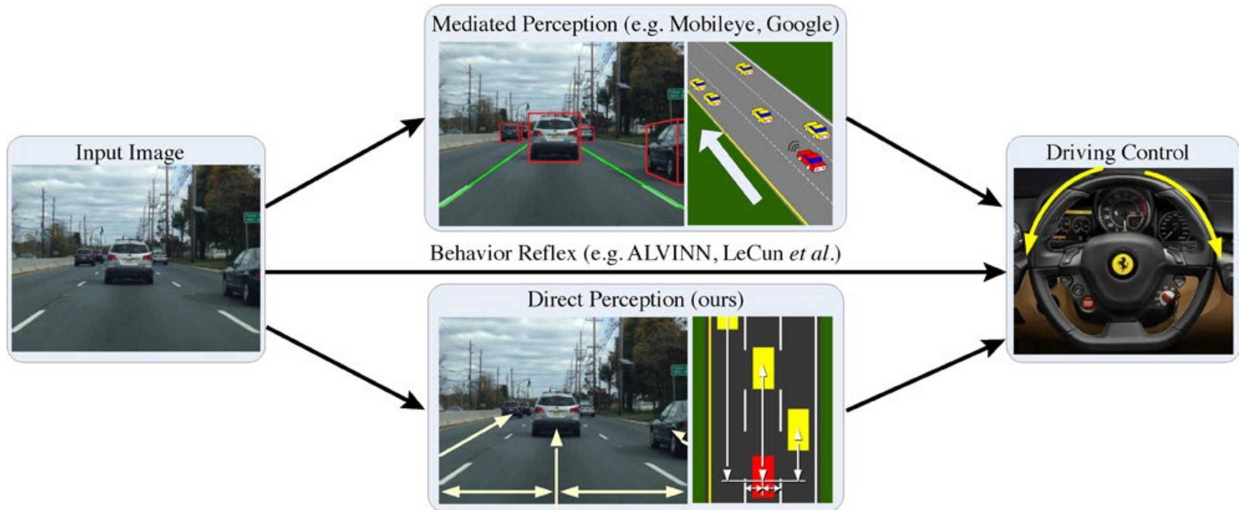
Robustness of ML models

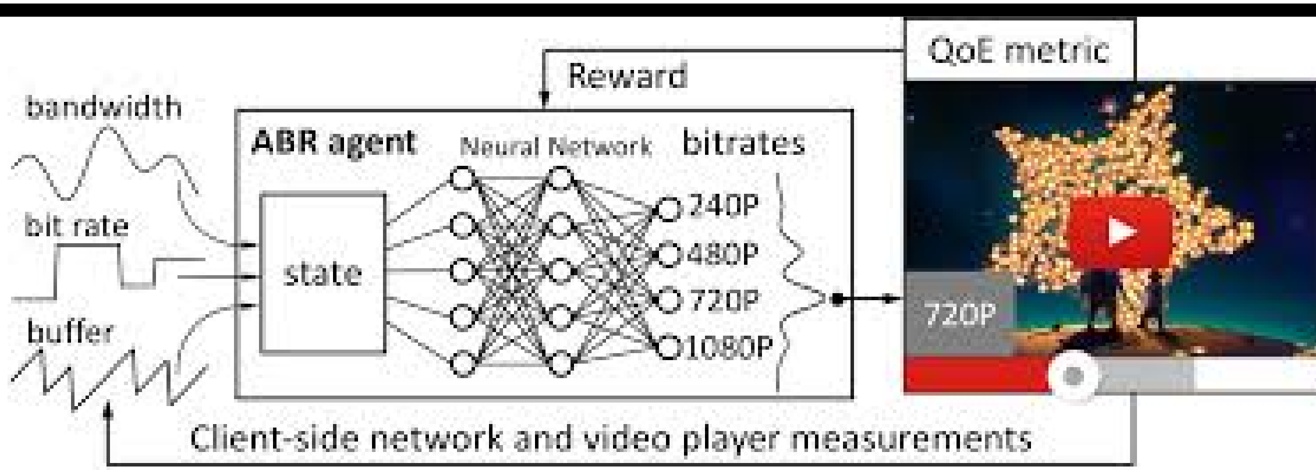
??? Part 5!!!

Interpretability of ML models

Dialogs/chat bots

Control systems





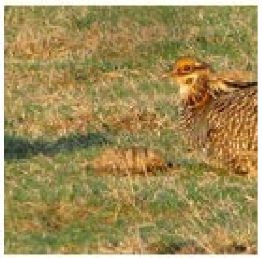
Machine Learning is used on daily basis

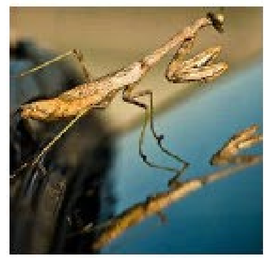
Deep learning-based systems can be fooled

Deep learning-based systems can be fooled

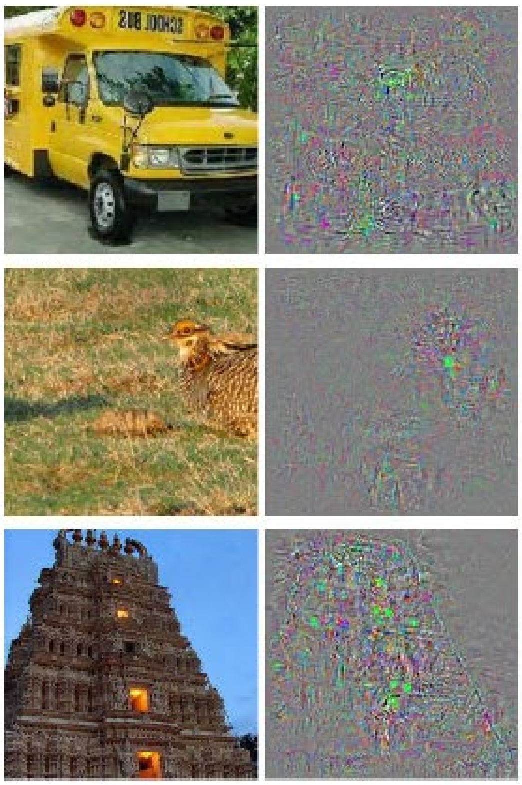
$\boxed{\text{Easily}}$

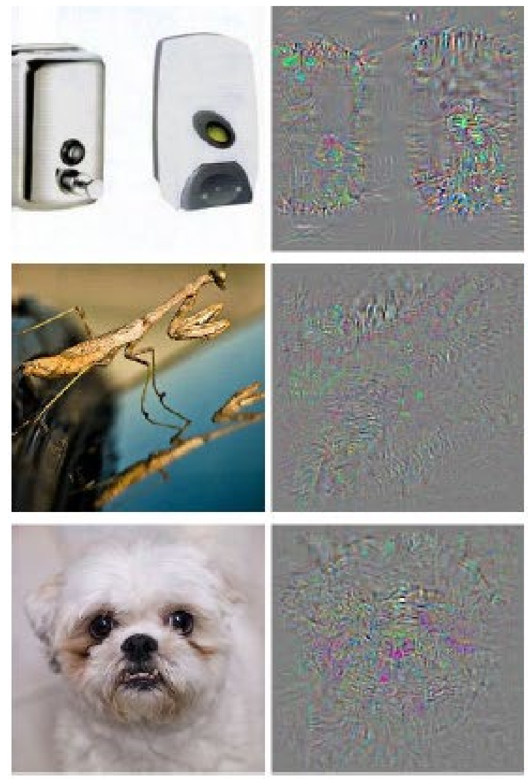
Fooling DL systems



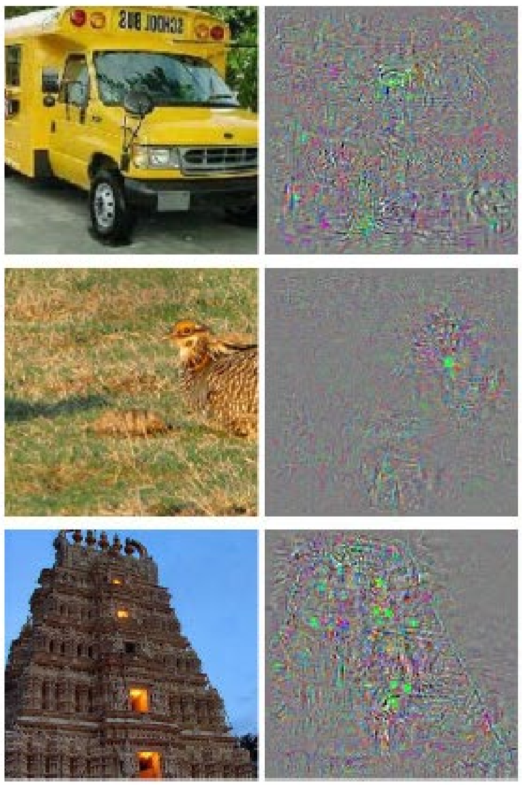


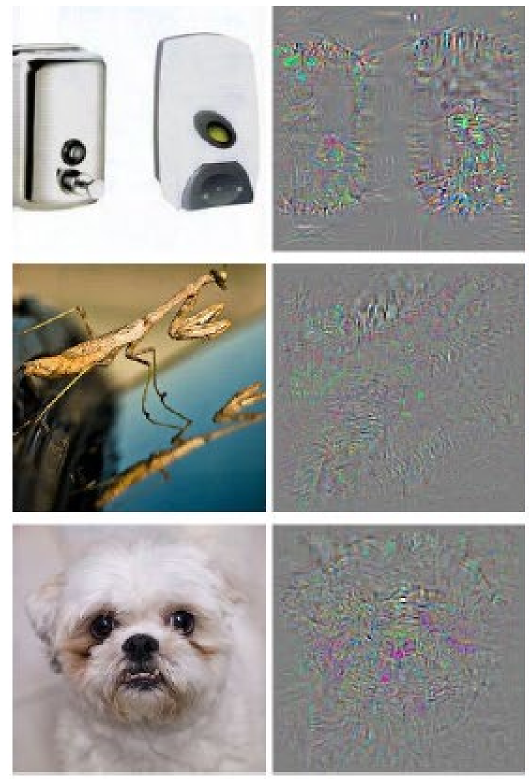
Fooling DL systems



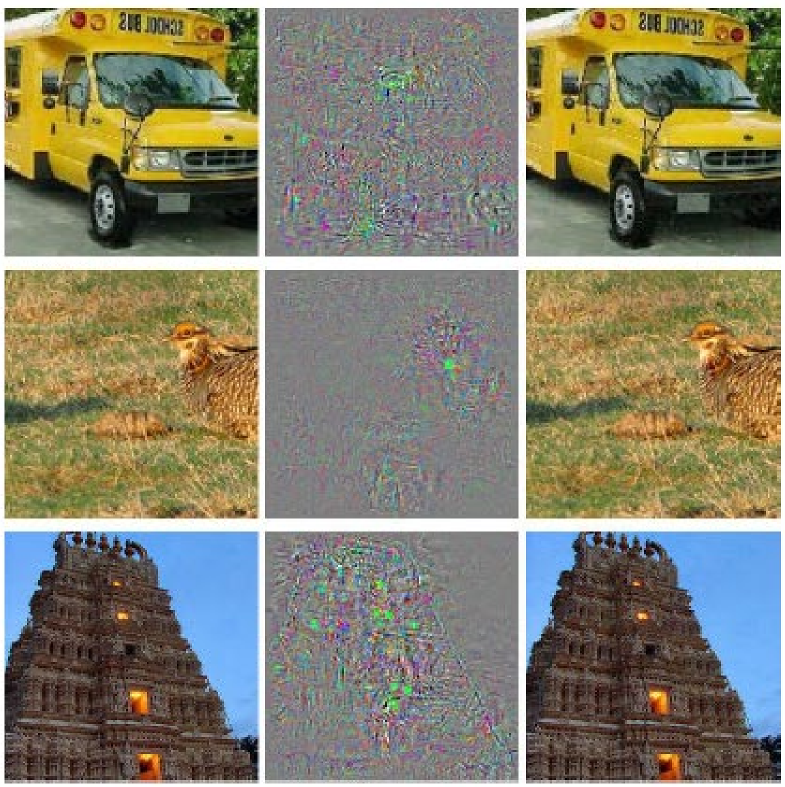


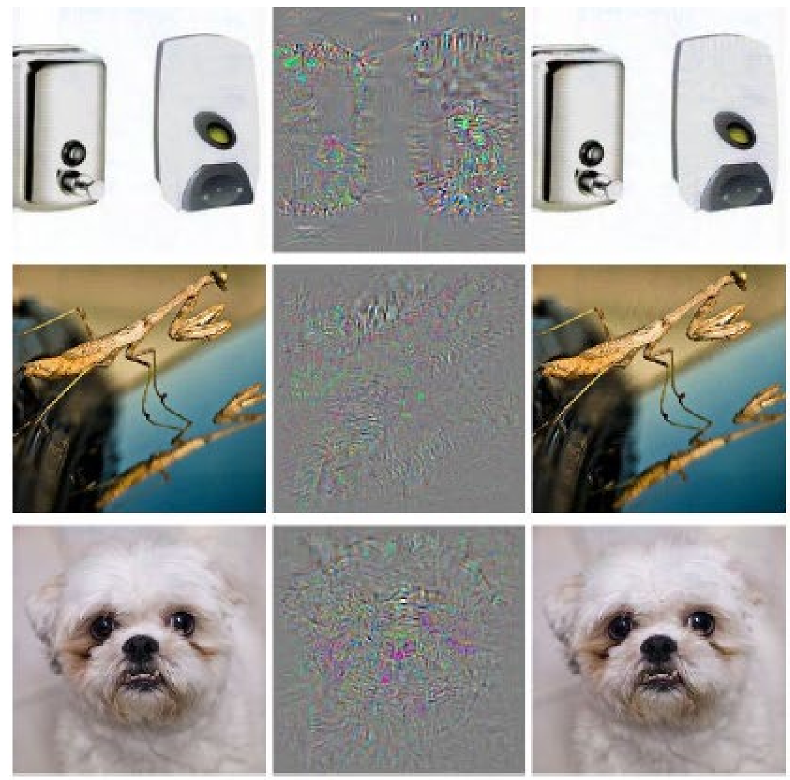
Fooling DL systems



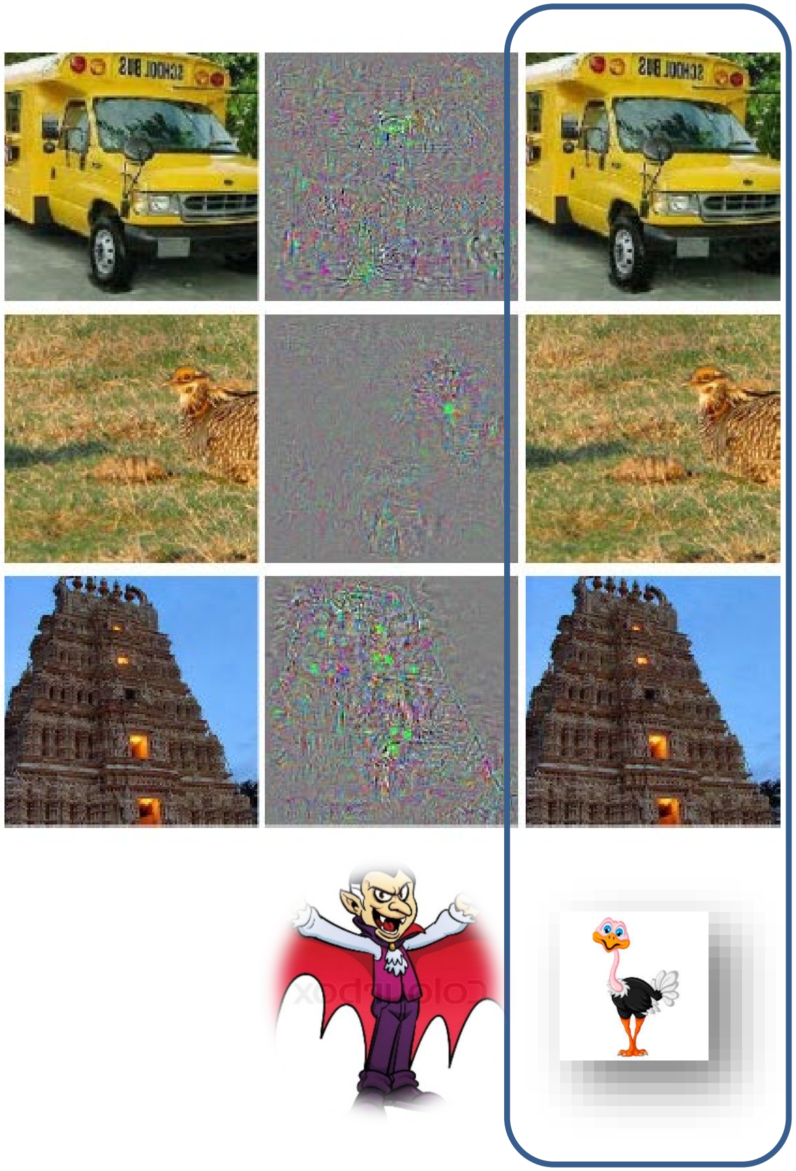


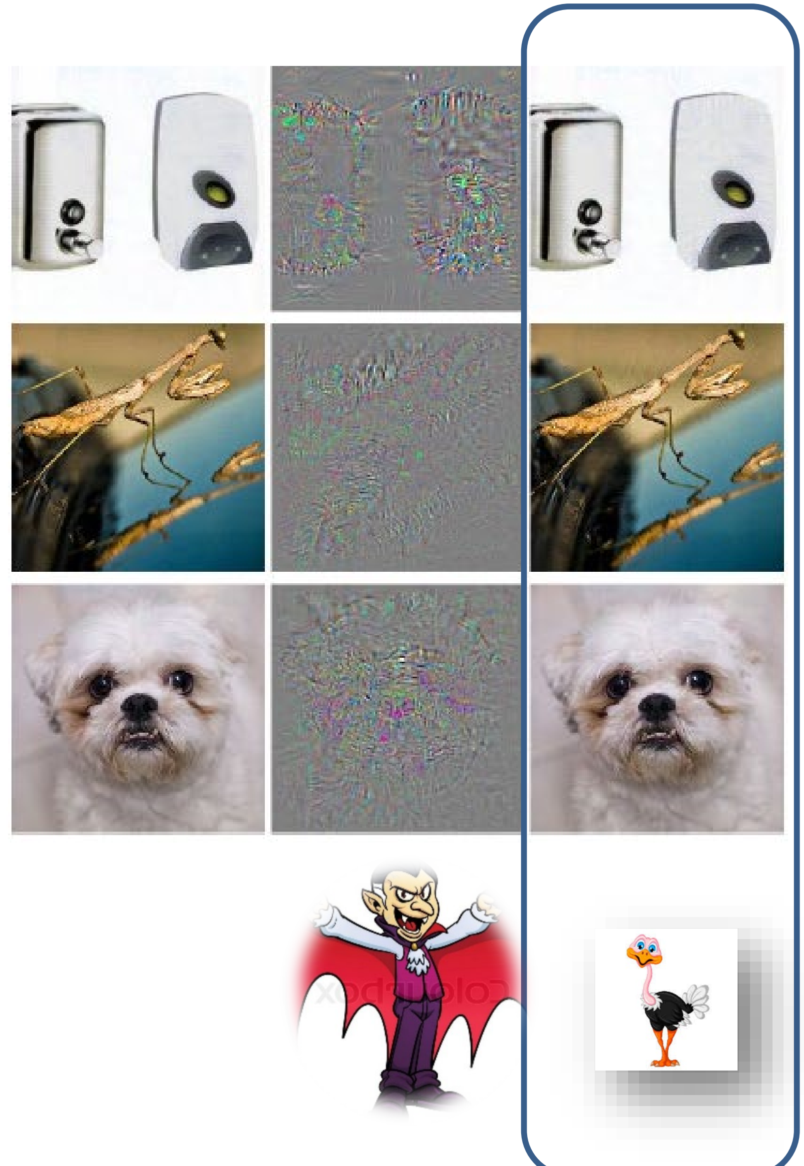
Fooling DL systems





Fooling DL systems





[Szegedy et al.] Intriguing properties of neural networks

Outline

Motivation

Adversarial attacks

Adversarial attacks

Untargeted adversarial examples

Given an input (X, C), an input $X' = X + P$ is an untargeted adversarial example iff NN misclassifies X' and P is small according to some metric.

Untargeted adversarial examples

Given an input $(\mathbf{X},\ \mathbf{C})$, an input $X' = X + P$ is an untargeted adversarial example iff NN misclassifies X' and P is small according to some metric.

Untargeted adversarial examples

Given an input $(\mathbf{X},\ \mathbf{C})$, an input $\mathbf{X'}= \mathbf{X} + \mathbf{P}$ is an untargeted adversarial example iff NN misclassifies X' and P is small according to some metric.

Untargeted adversarial examples

Given an input $(\mathbf{X}, \mathbf{C})$, an input $\mathbf{X'} = \mathbf{X} + \mathbf{P}$ is an untargeted adversarial example iff <u>*NN misclassifies*</u> X' and P is small according to some metric.

Untargeted adversarial examples

Original image

88% tabby cat

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

Untargeted adversarial examples

Original image **+** **P**erturbation

88% tabby cat

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

Untargeted adversarial examples

Original image + Perturbation = Perturbed image

88% tabby cat

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

Untargeted adversarial examples

Original image + Perturbation = Perturbed image

88% tabby cat

99% guacamole

[Szegedy *et al.*] *Intriguing properties of neural networks*
[Athalye et al.]Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples

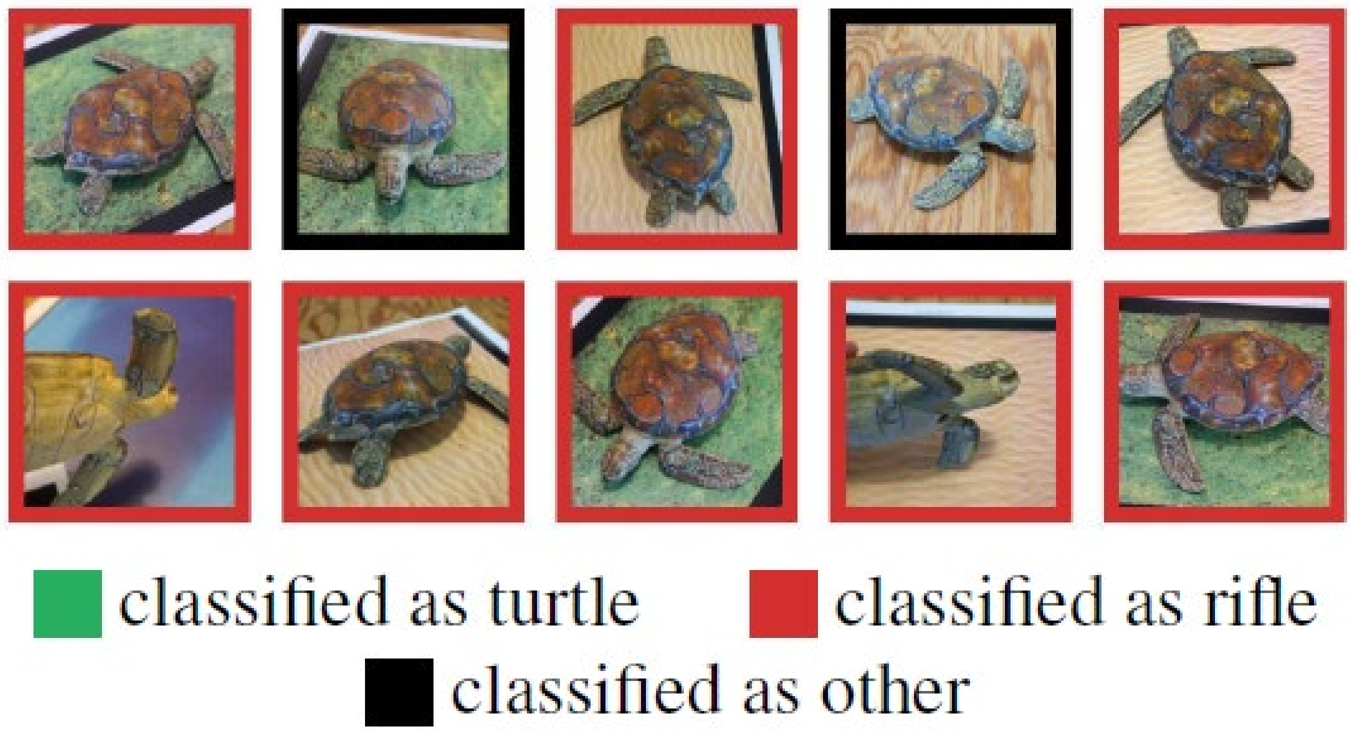
Beyond cats and dogs

[Eykholt at al.] Robust Physical-World Attacks on Deep Learning Visual Classification

Beyond cats and dogs

[Athalye at al.] Synthesizing Robust Adversarial Examples

Beyond cats and dogs

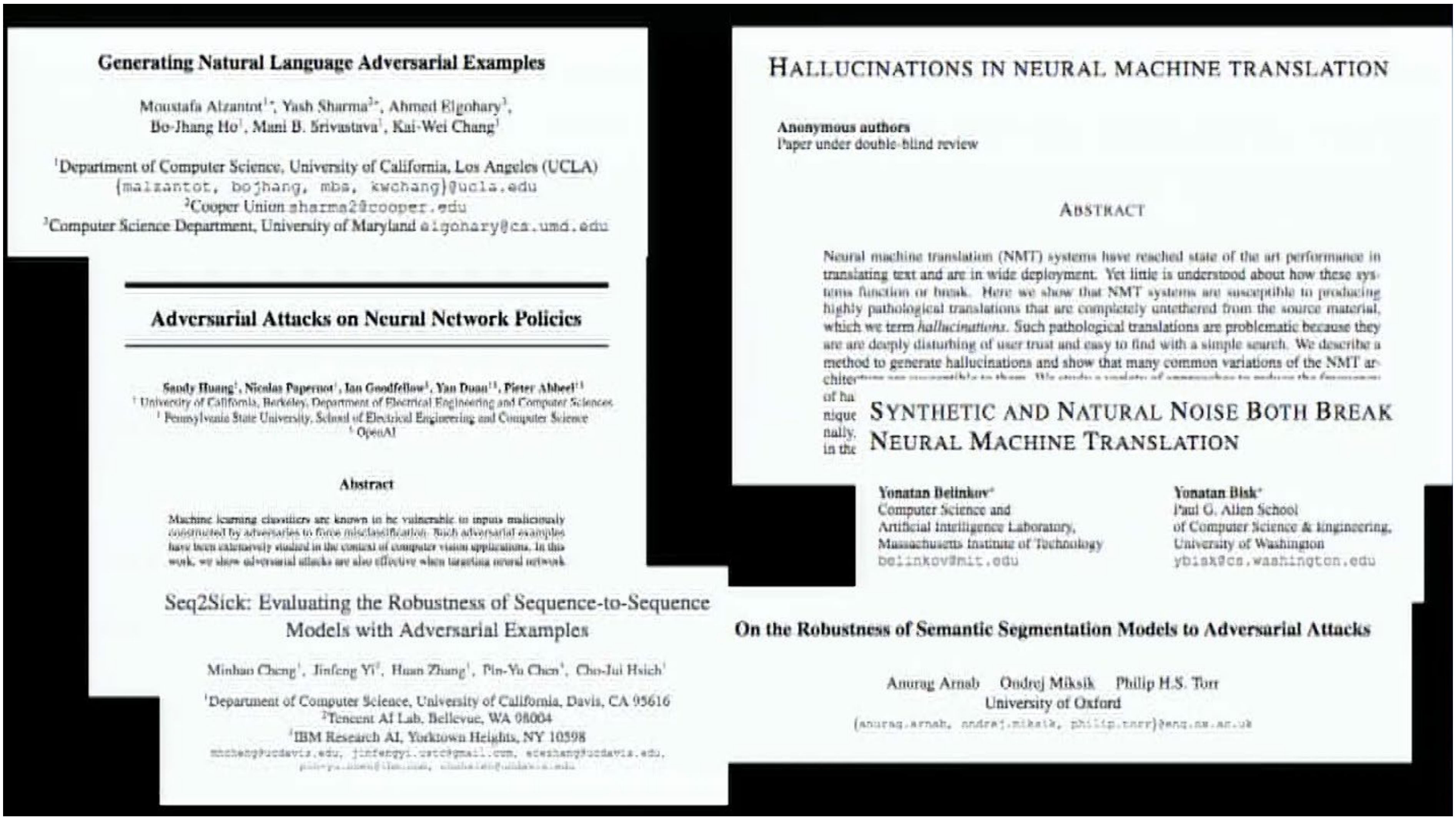


[Athalye at al.] Synthesizing Robust Adversarial Examples

Beyond cats and dogs

[Eykholt at al.] Robust Physical-World Attacks on Deep Learning Visual Classification

Beyond images



Generating Natural Language Adversarial Examples

Moustafa Alzantot[1*], Yash Sharma[2*], Ahmed Elgohary[3], Bo-Jhang Ho[1], Mani B. Srivastava[1], Kai-Wei Chang[1]

[1]Department of Computer Science, University of California, Los Angeles (UCLA) {malzantot, bojhang, mbs, kwchang}@ucla.edu
[2]Cooper Union sharma2@cooper.edu
[3]Computer Science Department, University of Maryland elgohary@cs.umd.edu

Adversarial Attacks on Neural Network Policies

Sandy Huang[1], Nicolas Papernot[2], Ian Goodfellow[3], Yan Duan[1,3], Pieter Abbeel[1,3]
[1] University of California, Berkeley, Department of Electrical Engineering and Computer Sciences
[2] Pennsylvania State University, School of Electrical Engineering and Computer Science
[3] OpenAI

Abstract

Machine learning classifiers are known to be vulnerable to inputs maliciously constructed by adversaries to force misclassification. Such adversarial examples have been extensively studied in the context of computer vision applications. In this work, we show adversarial attacks are also effective when targeting neural network

Seq2Sick: Evaluating the Robustness of Sequence-to-Sequence Models with Adversarial Examples

Minhao Cheng[1], Jinfeng Yi[2], Huan Zhang[1], Pin-Yu Chen[3], Cho-Jui Hsieh[1]

[1]Department of Computer Science, University of California, Davis, CA 95616
[2]Tencent AI Lab, Bellevue, WA 98004
[3]IBM Research AI, Yorktown Heights, NY 10598

HALLUCINATIONS IN NEURAL MACHINE TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Neural machine translation (NMT) systems have reached state of the art performance in translating text and are in wide deployment. Yet little is understood about how these systems function or break. Here we show that NMT systems are susceptible to producing highly pathological translations that are completely untethered from the source material, which we term *hallucinations*. Such pathological translations are problematic because they are are deeply disturbing of user trust and easy to find with a simple search. We describe a method to generate hallucinations and show that many common variations of the NMT architectures

SYNTHETIC AND NATURAL NOISE BOTH BREAK NEURAL MACHINE TRANSLATION

Yonatan Belinkov*
Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology
belinkov@mit.edu

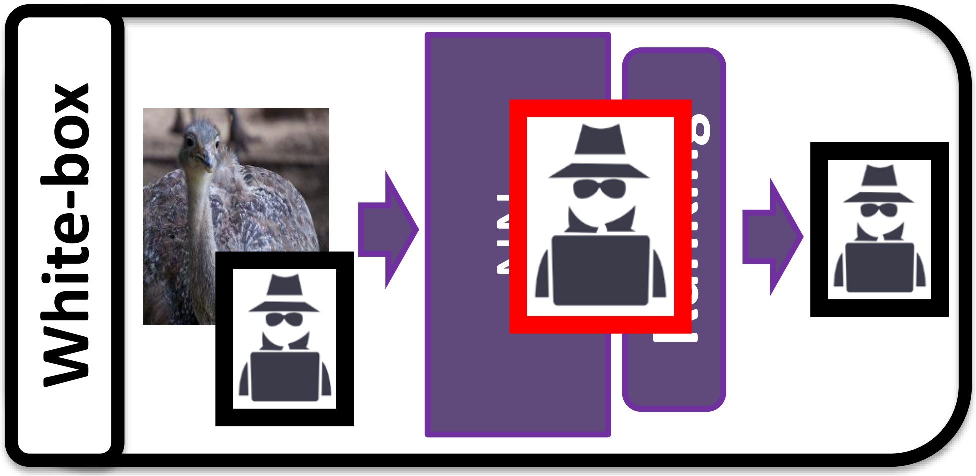
Yonatan Bisk*
Paul G. Allen School of Computer Science & Engineering, University of Washington
ybisk@cs.washington.edu

On the Robustness of Semantic Segmentation Models to Adversarial Attacks

Anurag Arnab, Ondrej Miksik, Philip H.S. Torr
University of Oxford

[Nicholas Carlini] *On (In-)* security of Deep Learning Models

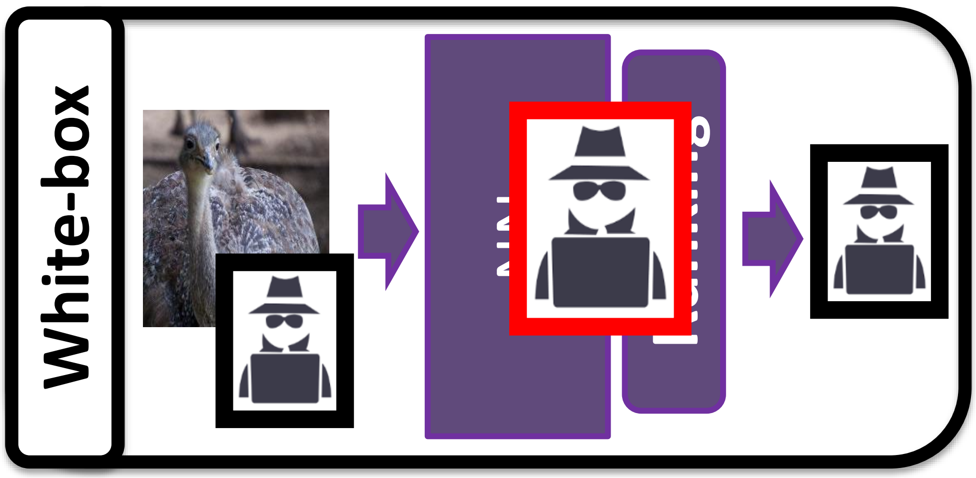
White-box vs Black-box Attacks



[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

White-box vs Black-box Attacks

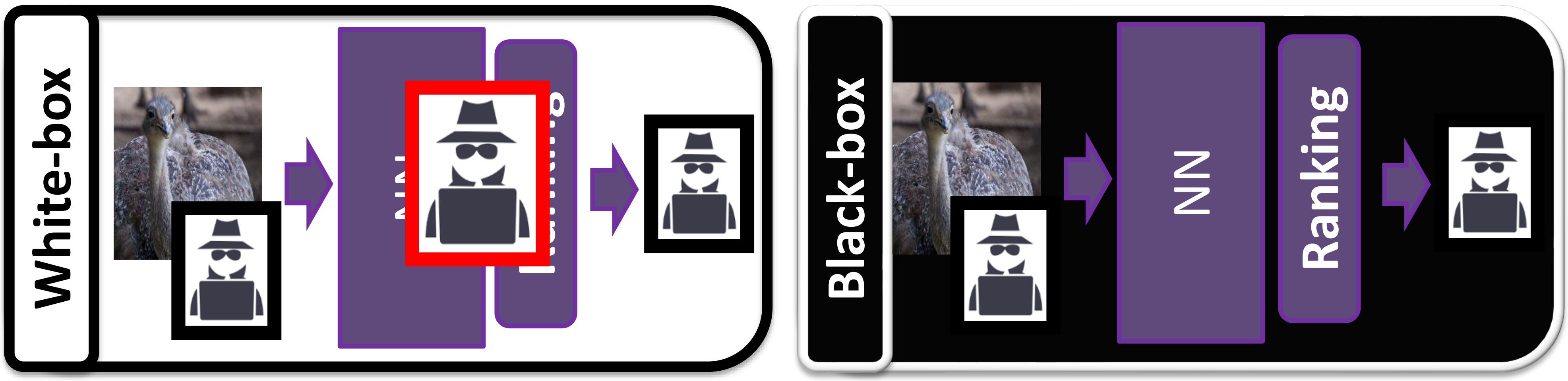


[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

Gradient-based methods that generate adversarial images by perturbing the gradients of the loss function w.r.t. the input image

White-box vs Black-box Attacks

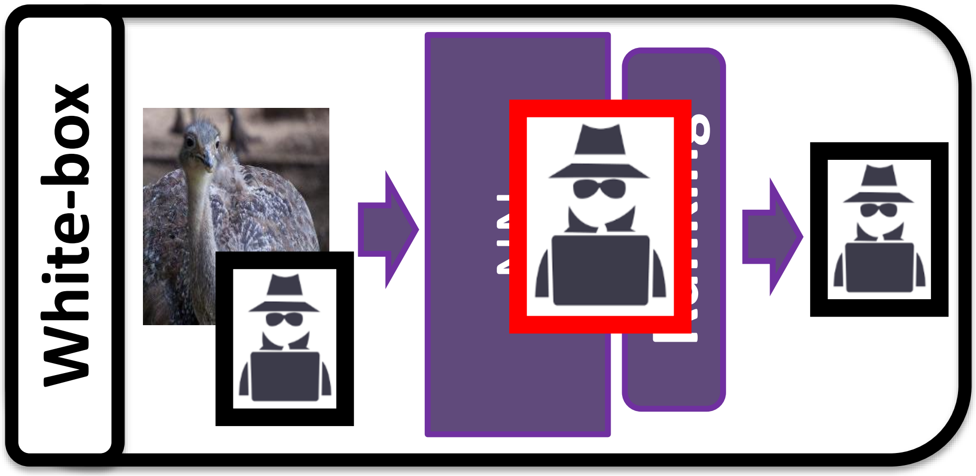


[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

Gradient-based methods that generate adversarial images by perturbing the gradients of the loss function w.r.t. the input image

White-box vs Black-box Attacks



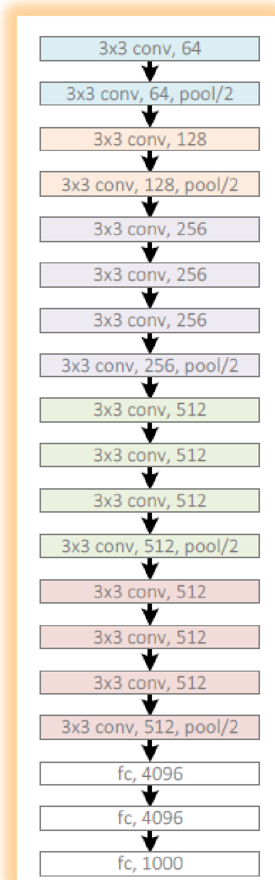
[Goodfellow et al., Szegedy et al.]

[Papernot et al., 2016a, 2016b]

Gradient-based methods that generate adversarial images by perturbing the gradients of the loss function w.r.t. the input image

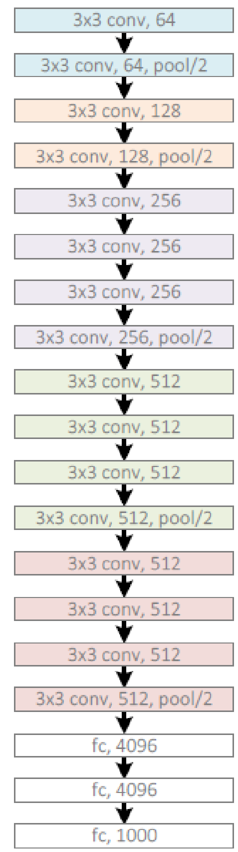
- More realistic and applicable model
- Challenging because of weak adversaries: no knowledge of the network architecture
- Previous attacks require 'transferability' assumption on adversarial examples
- GAN based attacks

New research sub-area



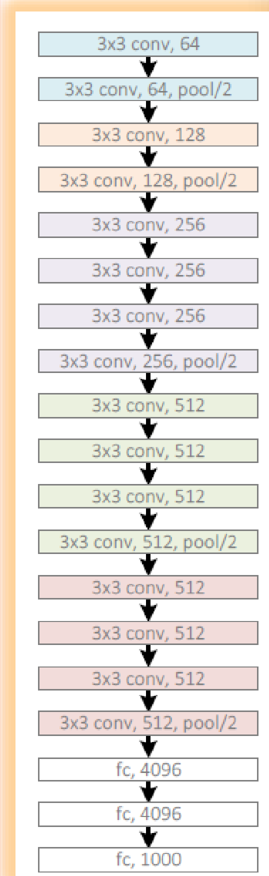
New research sub-area

Attacks



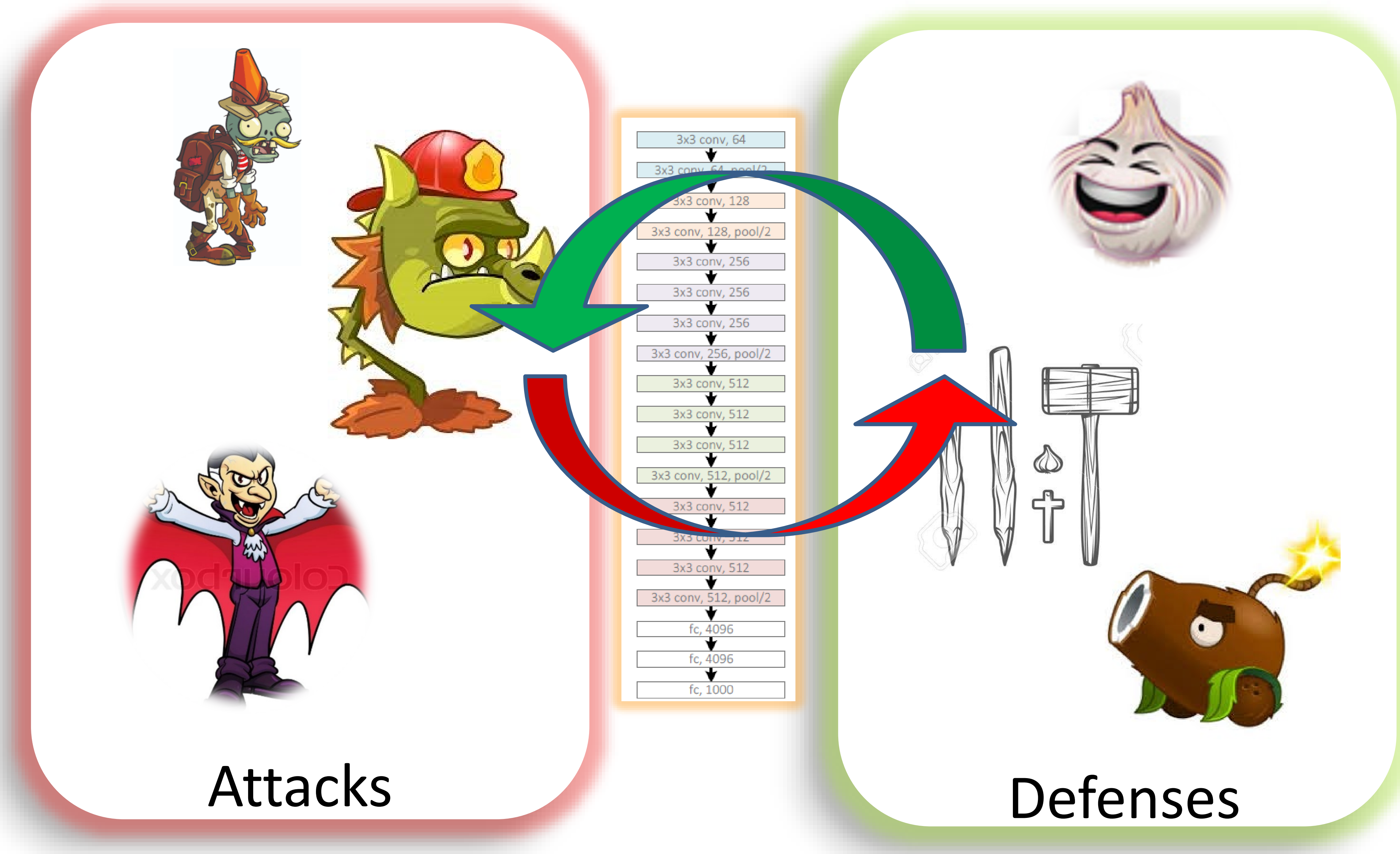
New research sub-area

Attacks

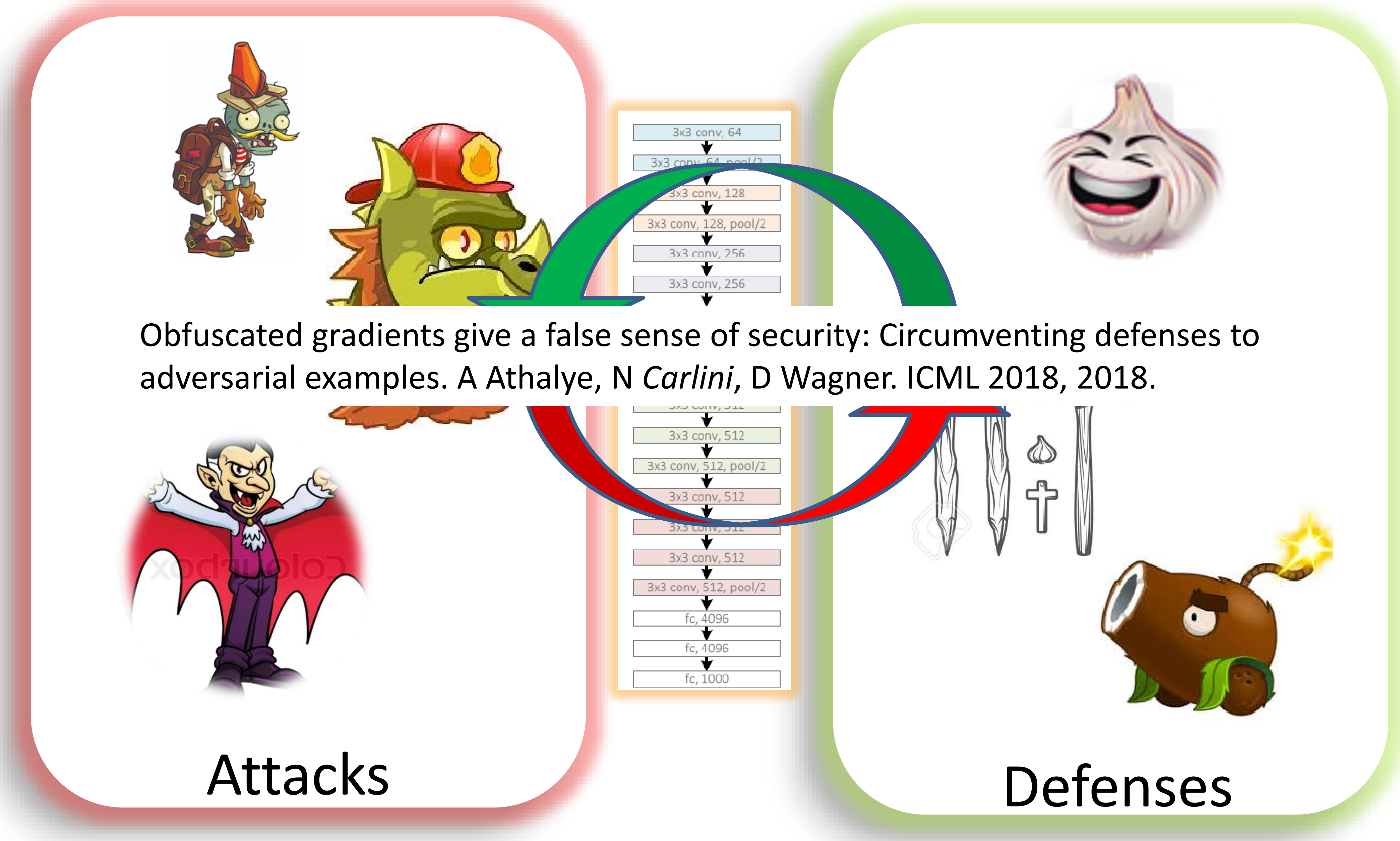


Defenses

New research sub-area

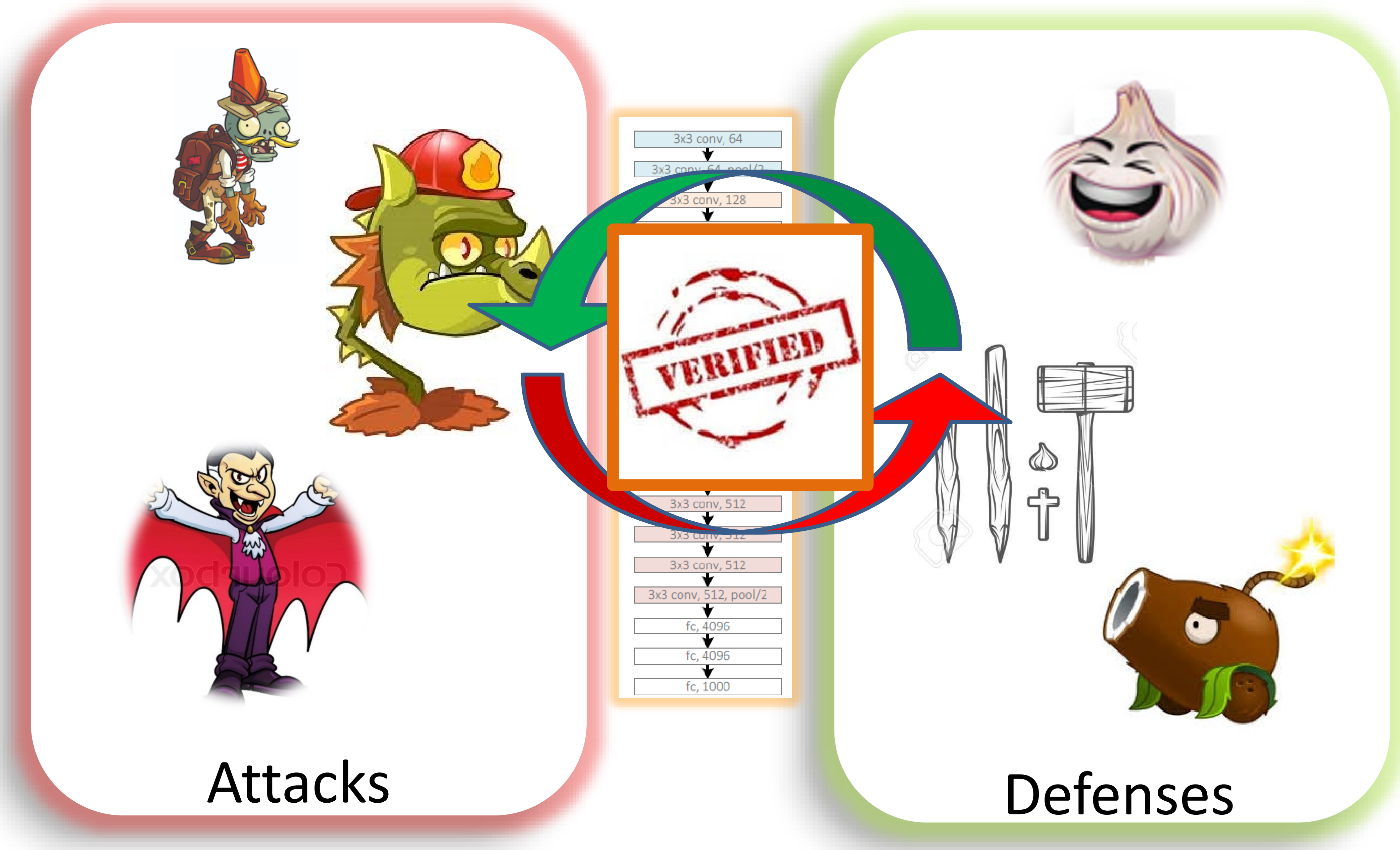


New research sub-area



Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. A Athalye, N *Carlini*, D Wagner. ICML 2018, 2018.

New research sub-area



Outline

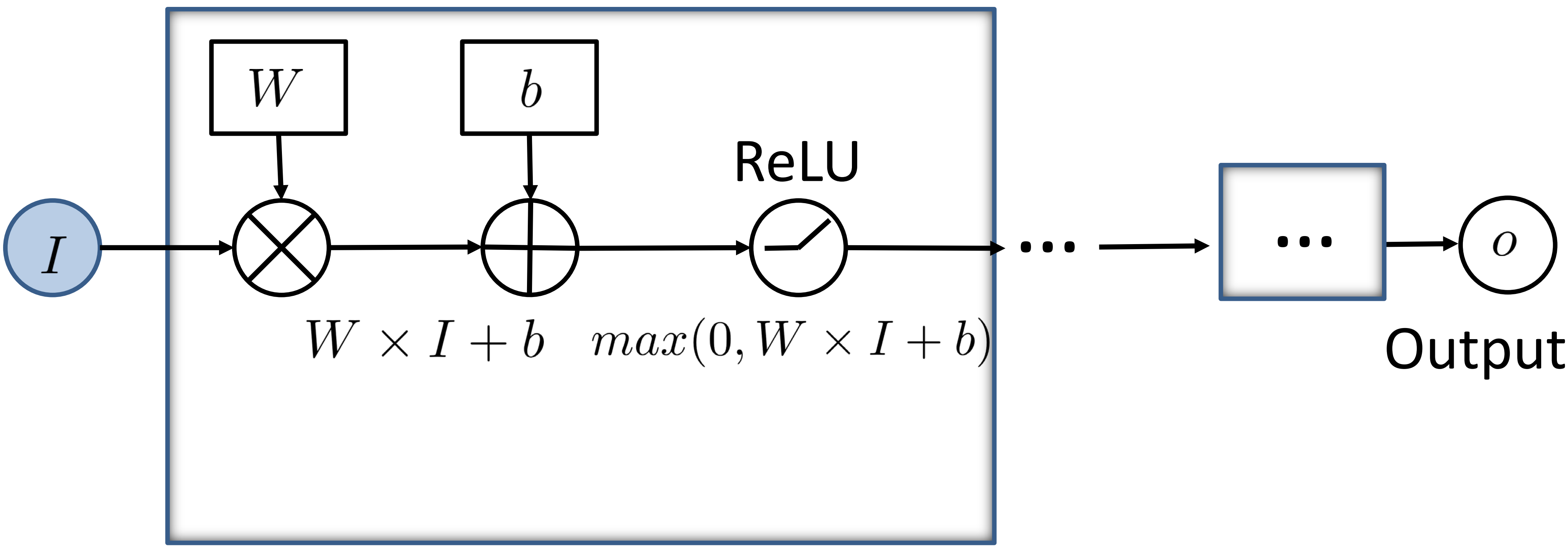
Motivation

Adversarial attacks

Verification methods

Network verification problem

Network verification problem



Input

- features
- images

Network verification problem

NNs is defined as $I^n \rightarrow O^m$

$pre(x)$ and $post(y)$ are logic formulas

pre defines *preconditions* on the inputs

$post$ defines *postconditions* on the output

Network verification problem

Given conditions pre and $post$, a property is:

$$\forall x . \forall y . (pre(x) \wedge y = NN(x)) \implies post(y)$$

Network verification problem

To find a counterexample:

$$pre(x) \land (y = NN(x)) \land \neg post(y)$$

Network verification problem

Let x' is a given  classified as ‘cat’.

$$pre(x) := |x - x'| \leq \epsilon$$

$$post(y) := \text{‘cat’}$$

$$\forall x. \forall y. (pre(x) \wedge y = NN(x)) \implies post(y)$$

Verification methods

Verification

Verification methods

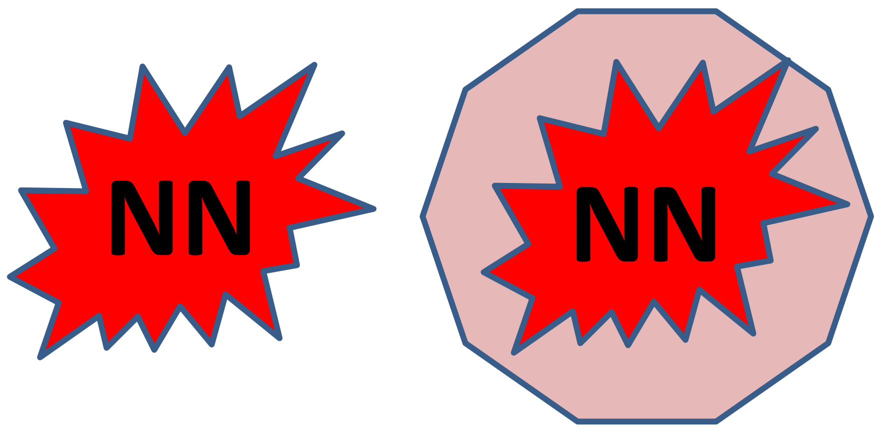
Verification

Verification roadmap

Verification roadmap

Exact Methods

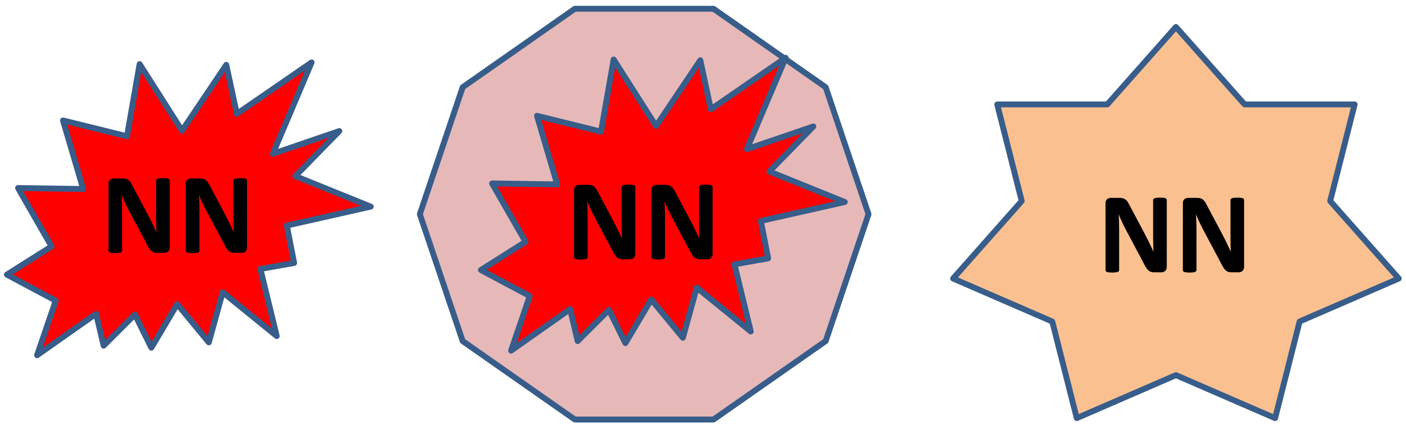
Verification roadmap



Exact Methods

Over-approx methods

Verification roadmap

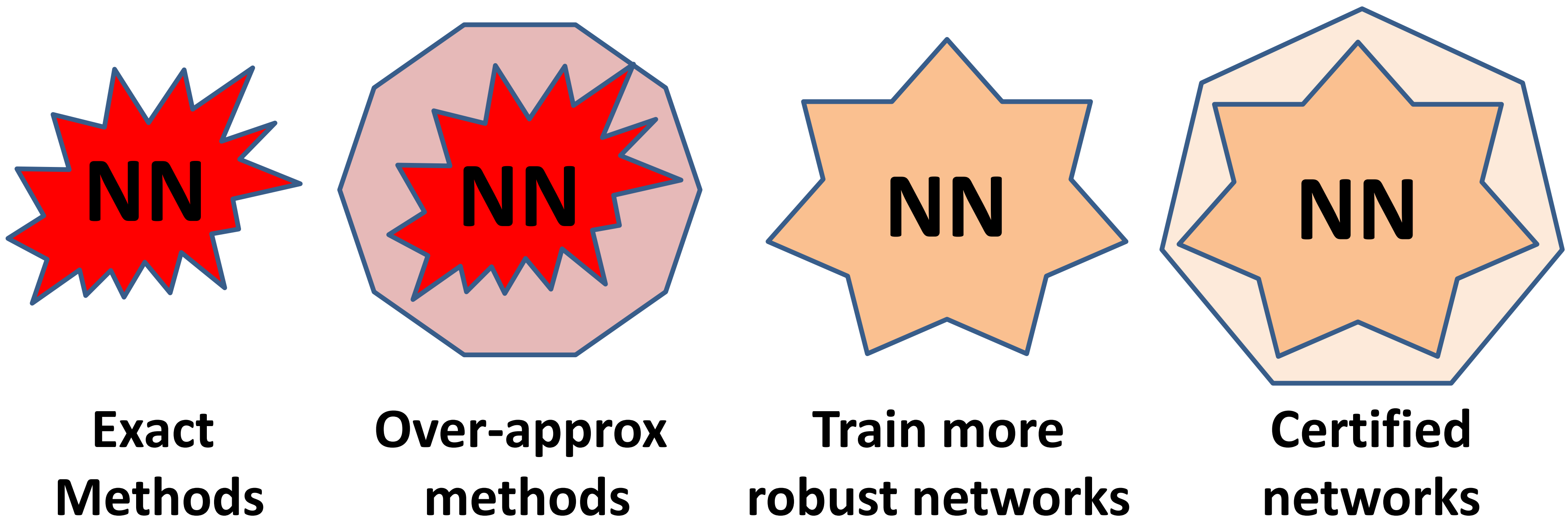


Exact Methods

Over-approx methods

Train more robust networks

Verification roadmap



Do we augment training?

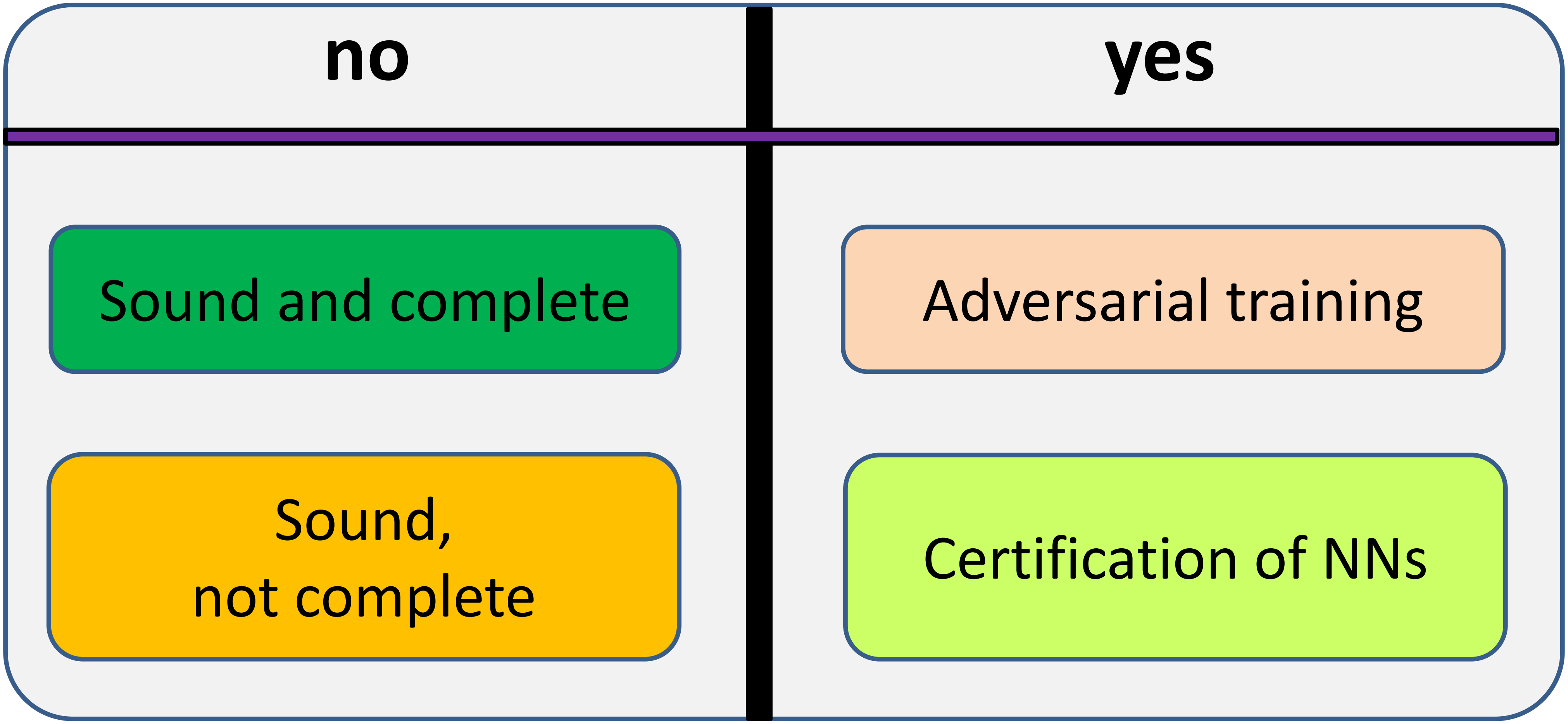
Do we augment training?

no	yes

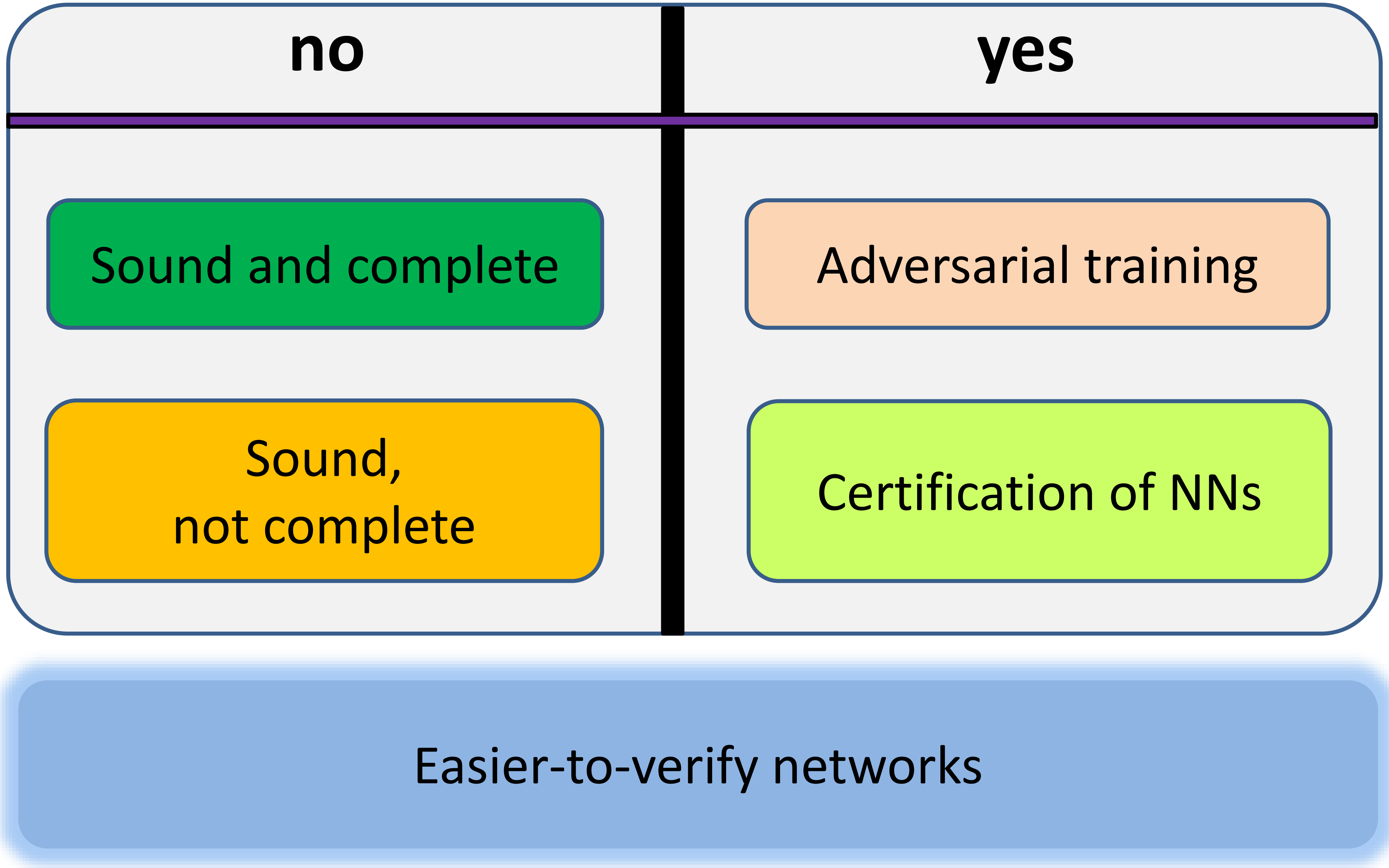
Do we augment training?

no	yes
Sound and complete	
Sound, not complete	

Do we augment training?



Do we augment training?



Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

Sound and complete methods

Strength: Prove whether a property holds

- R. Ehlers. Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks,2017
- R. Bunel, I. Turksaslan, P. Torr, P. Kohli, and P. Kumar. Piecewise Linear Neural Network Verification: A Comparative Study, 2017.
- G. Katz, C. Barrett, D. Dill, K. Julian, and M. Kochenderfer. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks.2017
- A. Lomuscio and L. Maganti. An Approach to Reachability Analysis for Feed-Forward ReLU Neural Networks, 2017.

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

$$\Downarrow$$

$$\mathrm{SMT}(pre(x)) \wedge \mathrm{SMT}(y = NN(x)) \wedge \mathrm{SMT}(\neg post(y))$$

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

$$\mathrm{SMT}(pre(x)) \wedge \mathrm{SMT}(y = NN(x)) \wedge \mathrm{SMT}(\neg post(y))$$

SMT solver

Sound and complete methods

$$pre(x) \wedge (y = NN(x)) \wedge \neg post(y)$$

(will discuss for BNNs+SAT)

$$\mathrm{SMT}(pre(x)) \wedge \mathrm{SMT}(y = NN(x)) \wedge \mathrm{SMT}(\neg post(y))$$

SMT solver

Sound and complete methods

$$pre(x) \land (y = NN(x)) \land \neg post(y)$$

$$\Downarrow$$

$$\mathrm{SMT}(pre(x)) \land \mathrm{SMT}(y = NN(x)) \land \mathrm{SMT}(\neg post(y))$$

$$\Downarrow$$

SMT solver (or Marabou, Planet, etc)

Sound and complete methods

Limitation: scalability (up to 2000 neurons)

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

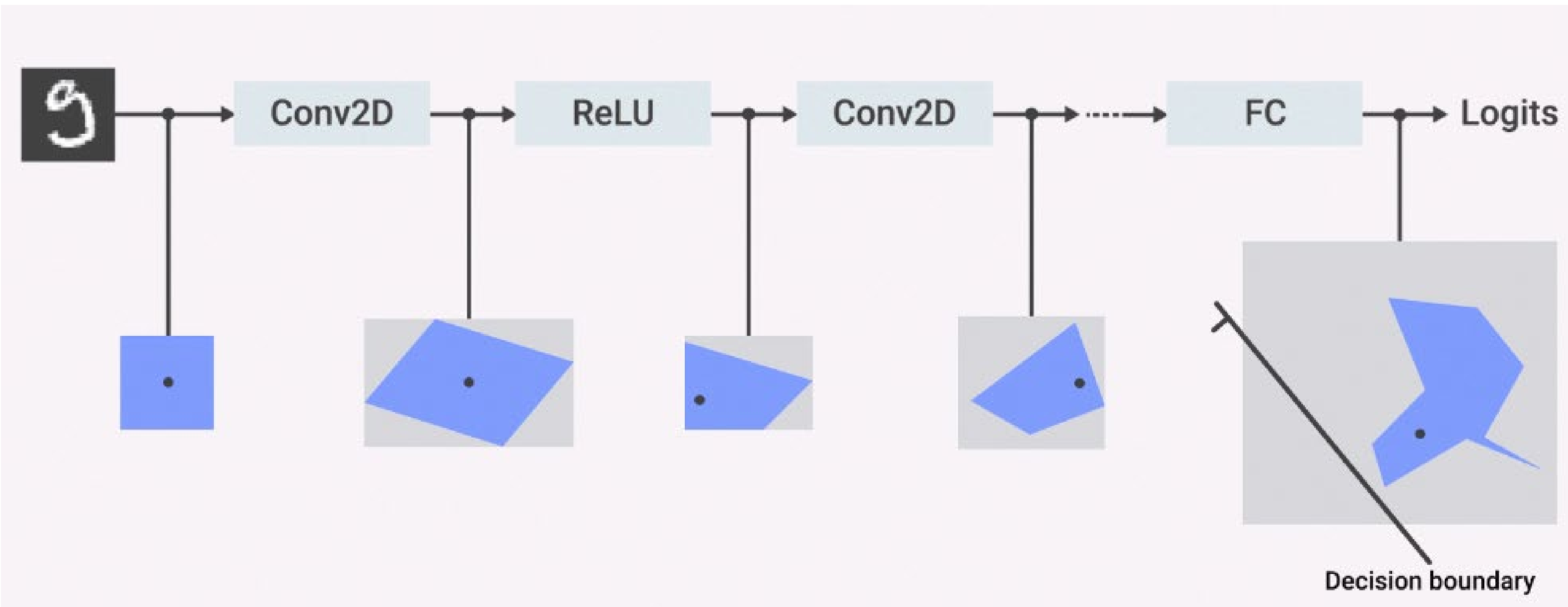
Sound and incomplete methods

Strength: Prove that a property holds
(can return `*do not know*')

- Singh, G., Gehr, T., Mirman, M., Puschel, M., and Vechev, M. T. Fast and effective robustness certification.
- Zhang, H., Weng, T., Chen, P., Hsieh, C., and Daniel, L. Efficient neural network robustness certification with general activation functions.
- Weng, T., Zhang, H., Chen, H., Song, Z., Hsieh, C., Daniel, L., Boning, D. S., and Dhillon, I. S. Towards fast computation of certified robustness for relu networks
- T. Gehr, M. Mirman, D. Drachsler-Cohen, E. Tsankov, S. Chaudhuri, and M. Vechev. AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation.

Sound and incomplete methods

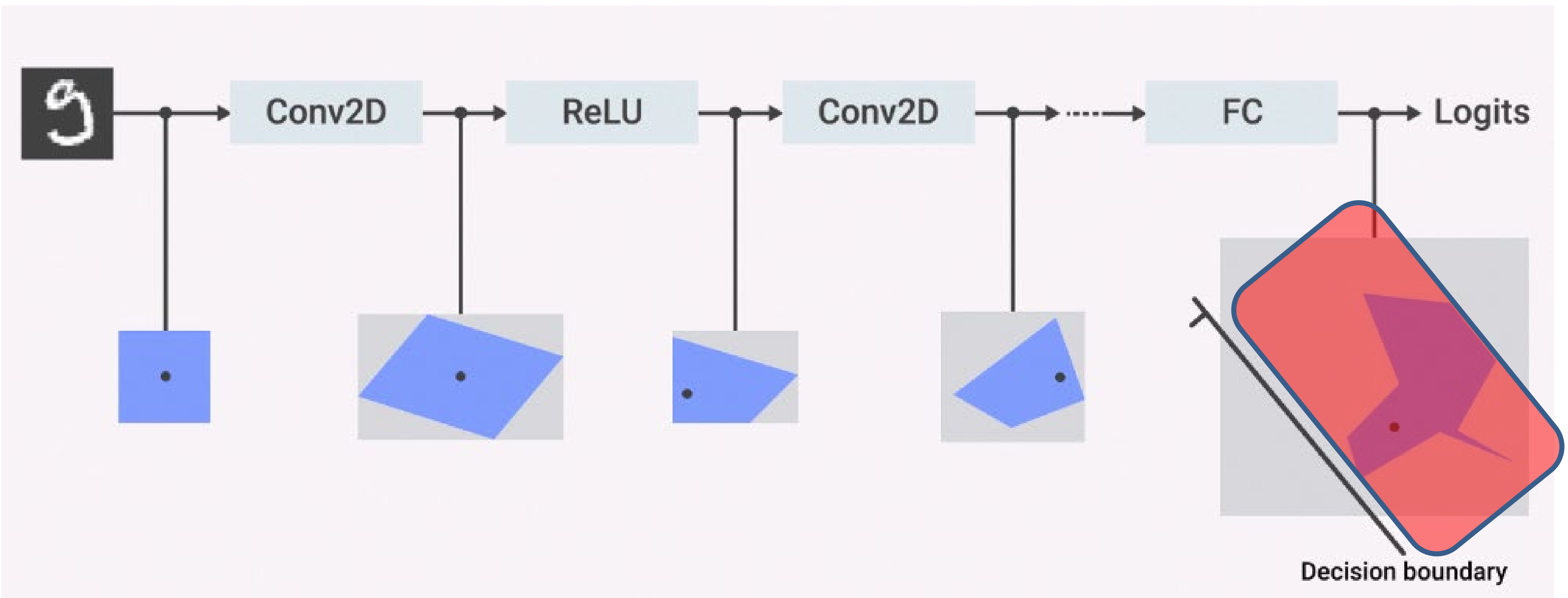
Based on over-approximation of the output space



https://medium.com/@deepmindsafetyresearch/towards-robust-and-verified-ai-specification-testing-robust-training-and-formal-verification-69bd1bc48bda

Sound and incomplete methods

Based on over-approximation of the output space

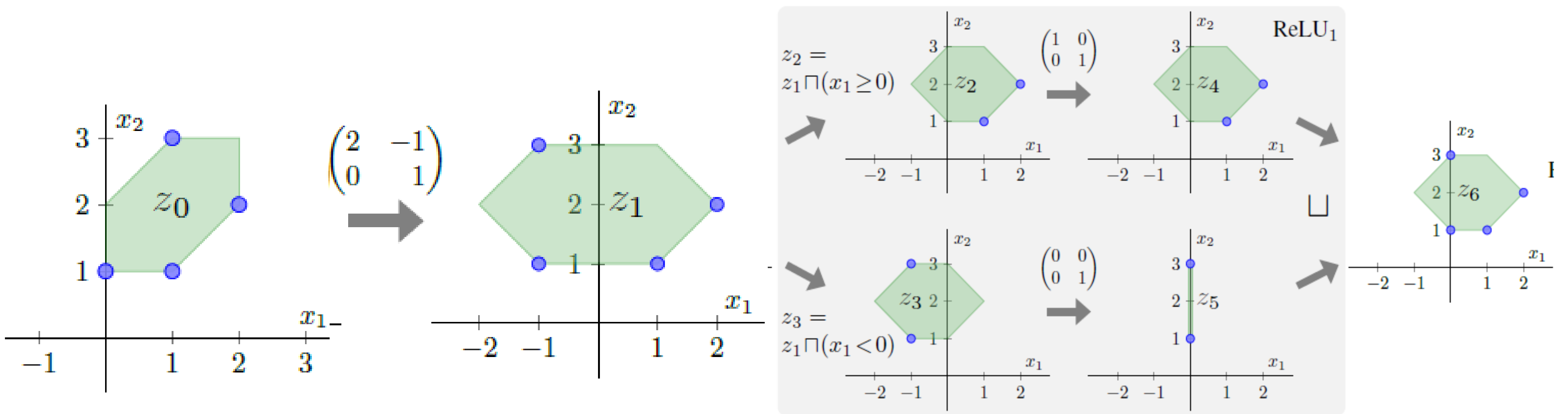


https://medium.com/@deepmindsafetyresearch/towards-robust-and-verified-ai-specification-testing-robust-training-and-formal-verification-69bd1bc48bda

Sound and incomplete methods

Based on over-approximation of the output space

*Input *Linear Transformer *ReLU

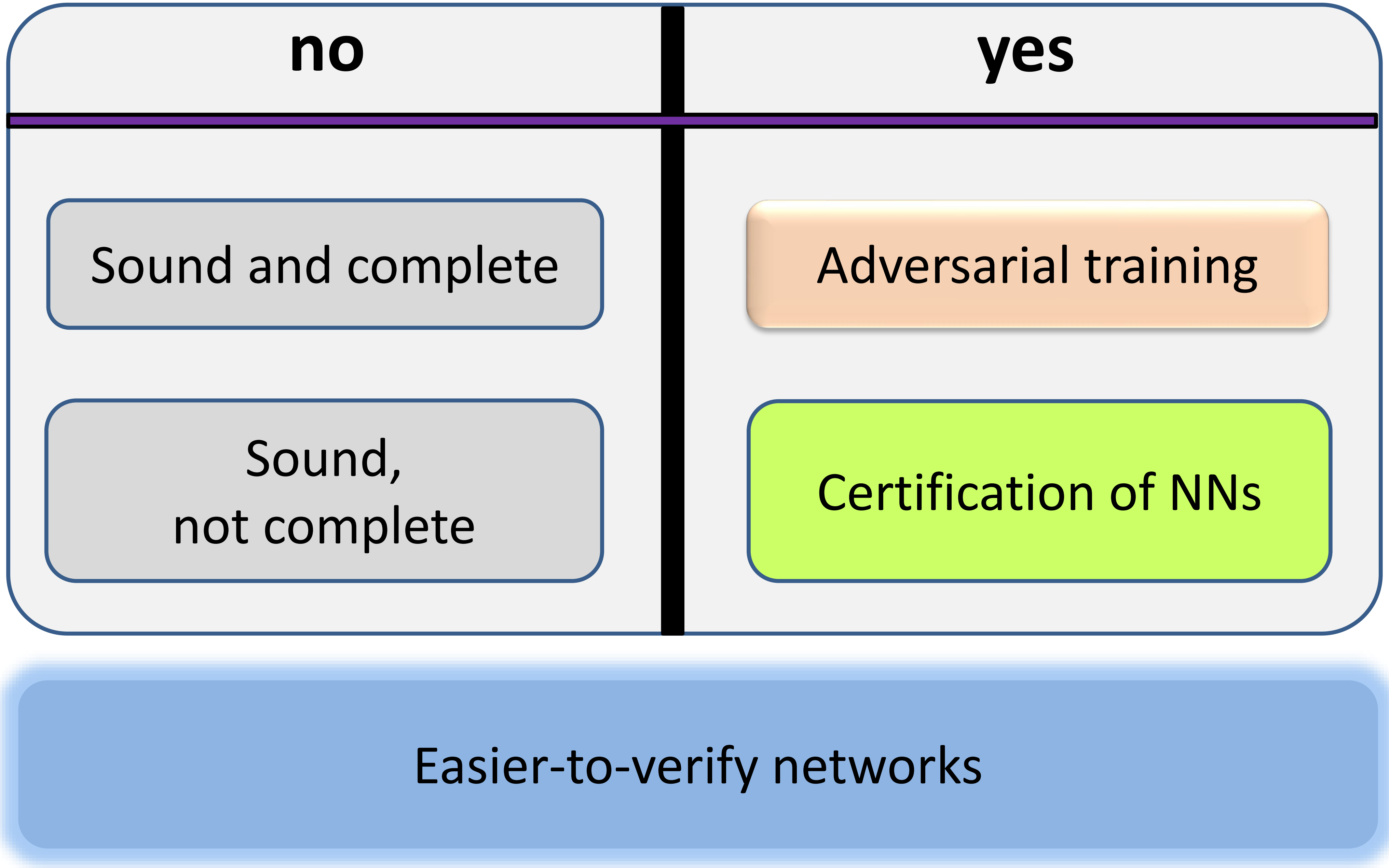


[Gehr et al.] AI2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation

Sound and incomplete methods

Limitation: scalability (up to 10000 neurons)

Do we augment training?



Adversarial training methods

Strength: (empirically) improve robustness of NNs

- Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale, 2017.
- Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples.2017
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu.Towards deep learning models resistant to adversarial attacks, 2018.

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \max_{\delta \in \Delta} Loss(I + \delta, L, W)$$

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \max_{\delta \in \Delta} Loss(I + \delta, L, W)$$

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

- Use gradient-based search, e.g. PGD, to solve inner optimization

Adversarial training methods

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I+\delta, L, W)}$$

1. Select minibatch B
2. For each (I,L) ∈ B compute an adversarial example δ*
3. Update parameters at I+ δ*

Adversarial training methods

Limitation: no guarantees on robustness

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

Certified training methods

Strength: prove that a property holds
(but can produce false negatives)

- Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope, 2018
- Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. 2018
- Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. 2018

Certification of NNs

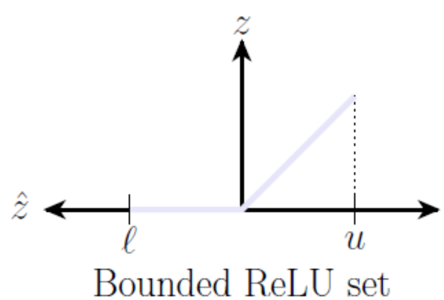
$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

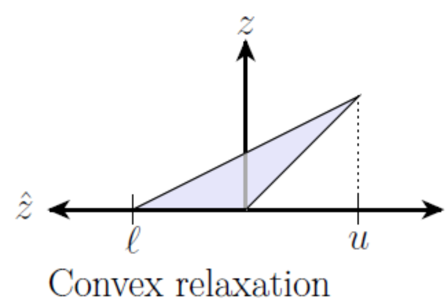
Certification of NNs

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$

Certification of NNs

$$\min_{W} \sum_{I,L \in \mathcal{D}} \boxed{\max_{\delta \in \Delta} Loss(I + \delta, L, W)}$$





- Use a convex relaxation inner optimization
- Use gradients of this relaxation in the training procedure

Certification of NNs

Limitation:

- work with relaxation, an upper bound on the can be quite loose
- the loss is much more complex than in a non-adv training

(accuracy drops, scalability issues)

Do we augment training?

no	yes
Sound and complete	Adversarial training
Sound, not complete	Certification of NNs

Easier-to-verify networks

Easier-to-verify networks

Strength: train a network that is easier to verify for existing decision procedures

- Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
  Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry, ICLR'19
- In Search for a SAT-friendly Binarized Neural Network Architecture
  Nina Narodytska, Hongce Zhang, Aarti Gupta, Toby Walsh, ICLR20

Easier-to-verify networks

Limitation: no guarantees on robustness

Do we augment training?

no	yes
Sound and complete	

Easier-to-verify networks

Do we augment training?

no	yes
Sound and complete	

Binarized Neural Networks

Easier-to-verify networks

Why BNNs?

Binarized neural networks: Training deep **neural networks** with weights and activations constrained to+ 1 or-1

M Courbariaux, I Hubara, D Soudry, R El-Yaniv… - arXiv preprint arXiv …, 2016 - arxiv.org

We introduce a method to train Binarized Neural Networks (BNNs)-neural networks with binary weights and activations at run-time. At training-time the binary weights and activations are used for computing the parameters gradients. During the forward pass, BNNs drastically …

Cited by 925 Related articles All 9 versions

Binarized neural networks

I Hubara, M Courbariaux, D Soudry… - Advances in **neural** …, 2016 - papers.nips.cc

We introduce a method to train Binarized Neural Networks (BNNs)-neural networks with binary weights and activations at run-time. At train-time the binary weights and activations are used for computing the parameter gradients. During the forward pass, BNNs drastically …

Cited by 470 Related articles All 5 versions

Xnor-net: Imagenet classification using binary convolutional **neural networks**

M Rastegari, V Ordonez, J Redmon… - European Conference on …, 2016 - Springer

… Because, at inference we only perform forward propagation with the **binarized** weights … Similar to **binarization** in the forward pass, we can **binarize** \(g^{in}\) in the backward pass … Our **binarization** technique is general, we can use any CNN architecture …

Cited by 1373 Related articles All 8 versions

Compactness

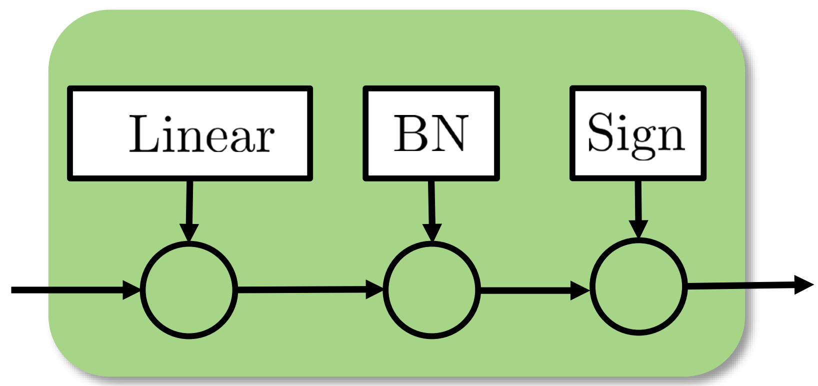
- Only 1 bit per weight, {-1,1}
- Can be deployed on embedded devices

Inference efficiency

- fast binary matrix multiplication
  (7X speed up on GPU)
- “Accelerating Binarized Neural Networks: Comparison of FPGA, CPU, GPU, and ASIC” IEEE’2016

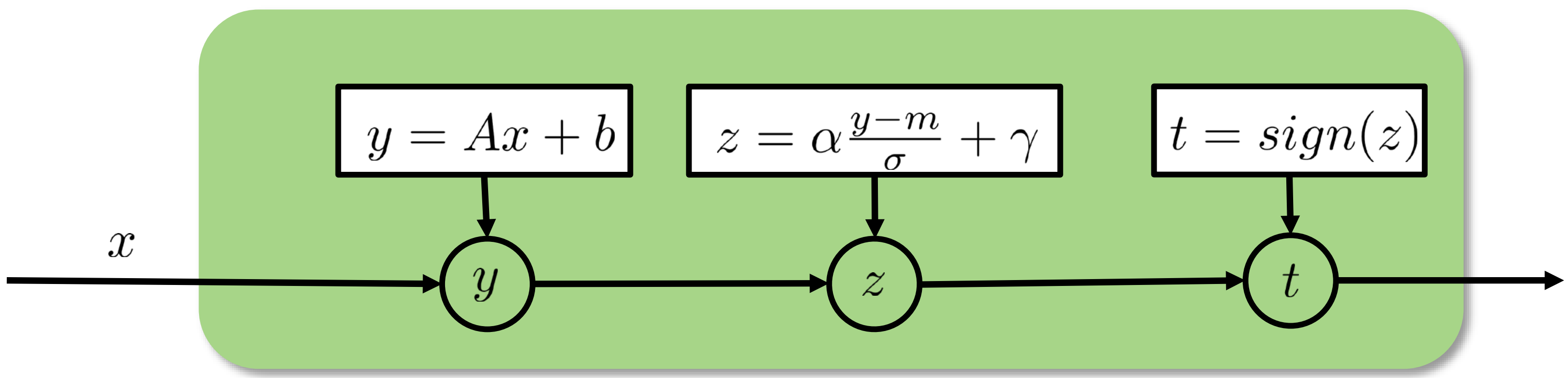
Structure of BNNs

Binarized Neural Networks

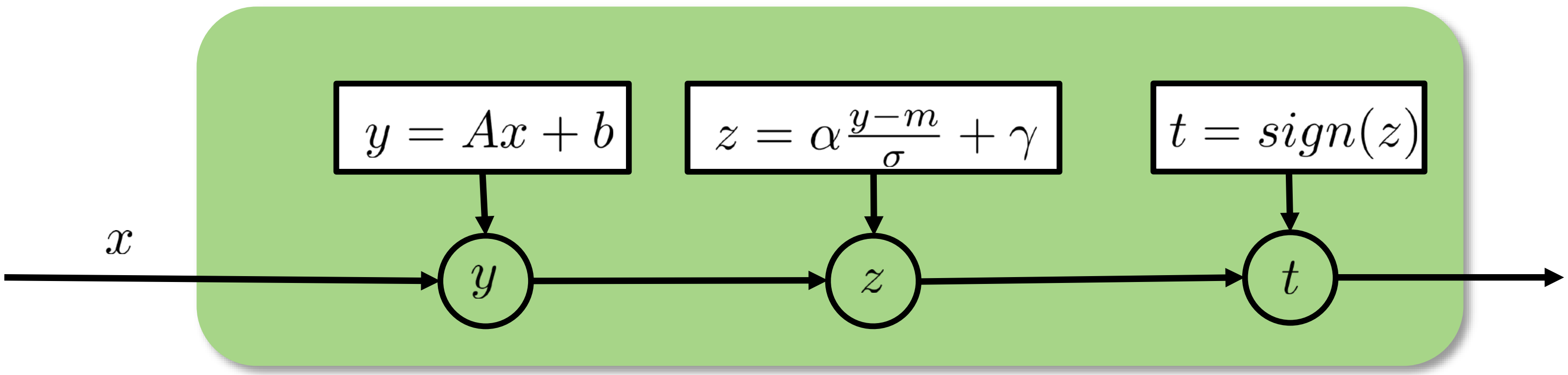


Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1
Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, Yoshua Bengio

Binarized Neural Networks

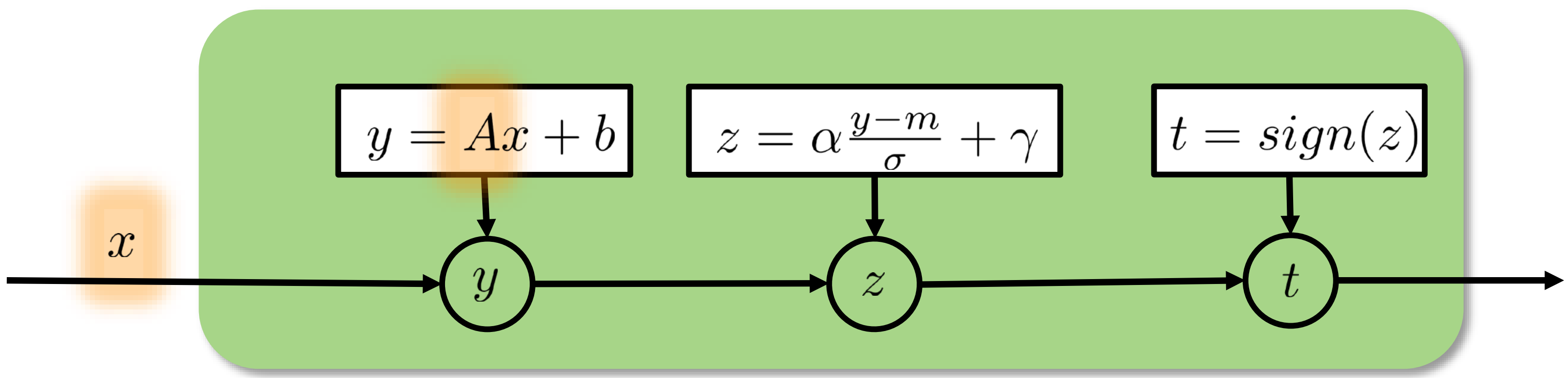


Binarized Neural Networks



$x, a_{i,j} \in \{-1, 1\}$ $\qquad$ $b, \alpha, m, \sigma, \gamma \in \mathbf{R}$

Binarized Neural Networks



$x, a_{i,j} \in \{-1, 1\}$ $\quad$ $b, \alpha, m, \sigma, \gamma \in \mathbf{R}$

BNNs and logic-based reasoning

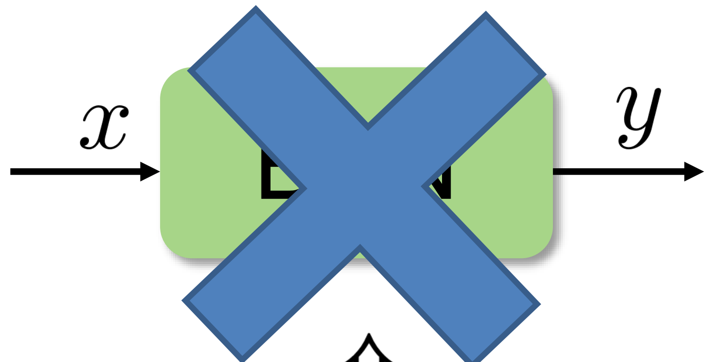
BNNs and Logic

BNNs and Logic

$$\Updownarrow$$

$$SAT(y = BNN(x))$$

BNNs and Logic



$$\Updownarrow$$

$$SAT(y = BNN(x))$$

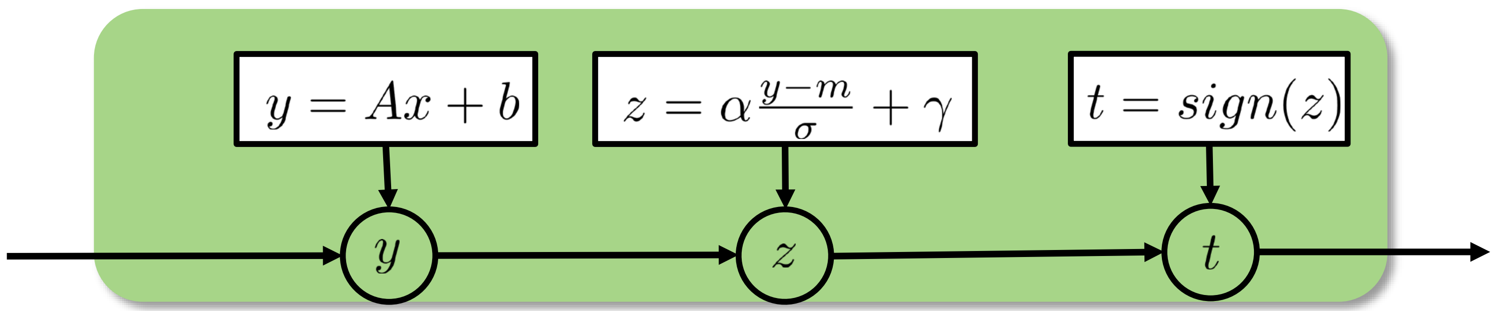
BNNs and Logic

$$SAT(y = BNN(x))$$

BNNs and Logic

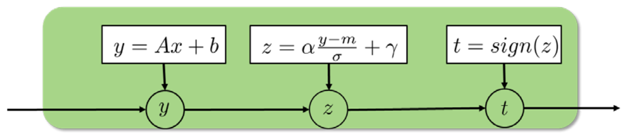
$$BinBNN(x, y) :=$$
$$SAT(y = BNN(x))$$

Translation: BNN to SAT



$$(l_1 + \ldots + l_n \geq k) \Leftrightarrow t_i = 1$$

Translation: BNN to SAT

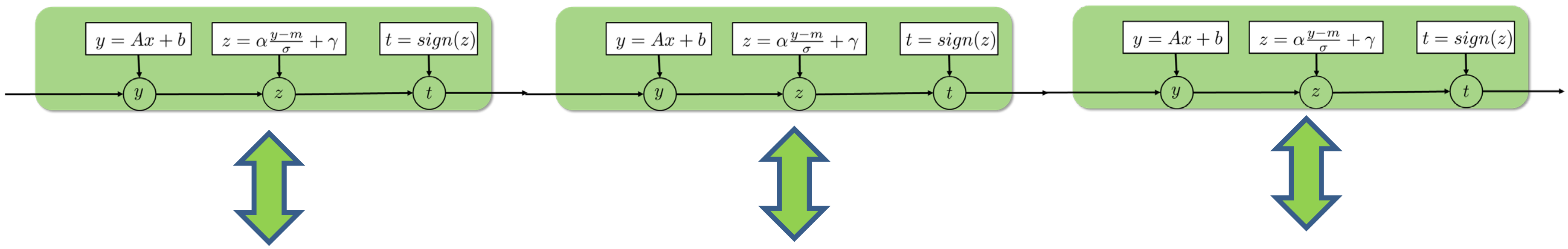


$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

$$\vdots$$

$$(l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m = 1$$

Translation: BNN to SAT



$$\begin{pmatrix} (l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^1 = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^1 = 1 \end{pmatrix} \wedge \begin{pmatrix} (l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1 \\ \vdots \quad . \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m = 1 \end{pmatrix} \wedge \begin{pmatrix} (l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^p = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^p = 1 \end{pmatrix}$$

BinBNN

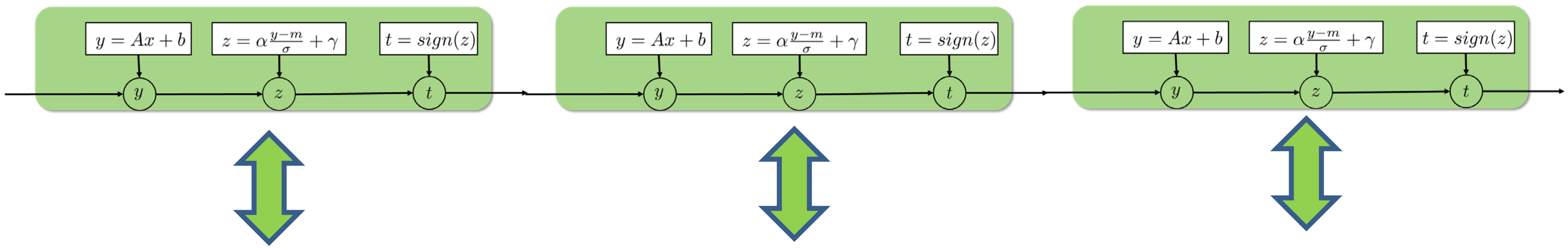
Work with small networks

In Search for a SAT-friendly Binarized Neural Network Architecture

ICLR’20

N Narodytska, H Zhang, A Gupta, T Walsh

Translation: BNN to SAT



$$\begin{array}{c} (l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^1 = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^1 = 1 \end{array} \wedge \begin{array}{c} (l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m = 1 \end{array} \wedge \begin{array}{c} (l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1^p = 1 \\ \vdots \\ (l_{m,1} + \ldots + l_{m,n} \geq k_m) \Leftrightarrow t_m^p = 1 \end{array}$$

BinBNN

“Neuron” constraint

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

$\ldots$

“Neuron” constraint

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

...

Number of variables

Reification means no propagation!

"Neuron" constraint

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

...

Number of variables

Reification means no propagation!

+ reduce #vars

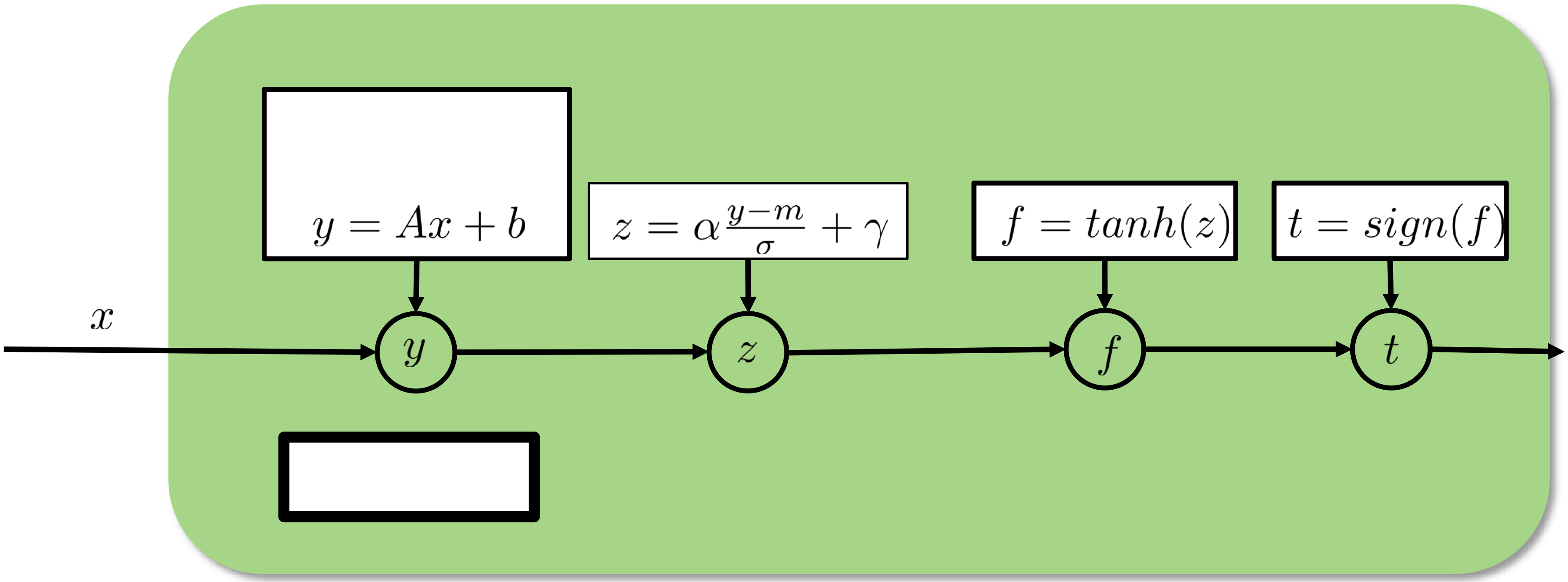
+ eliminate reifications

We can train a BNN so that

+ reduce #vars

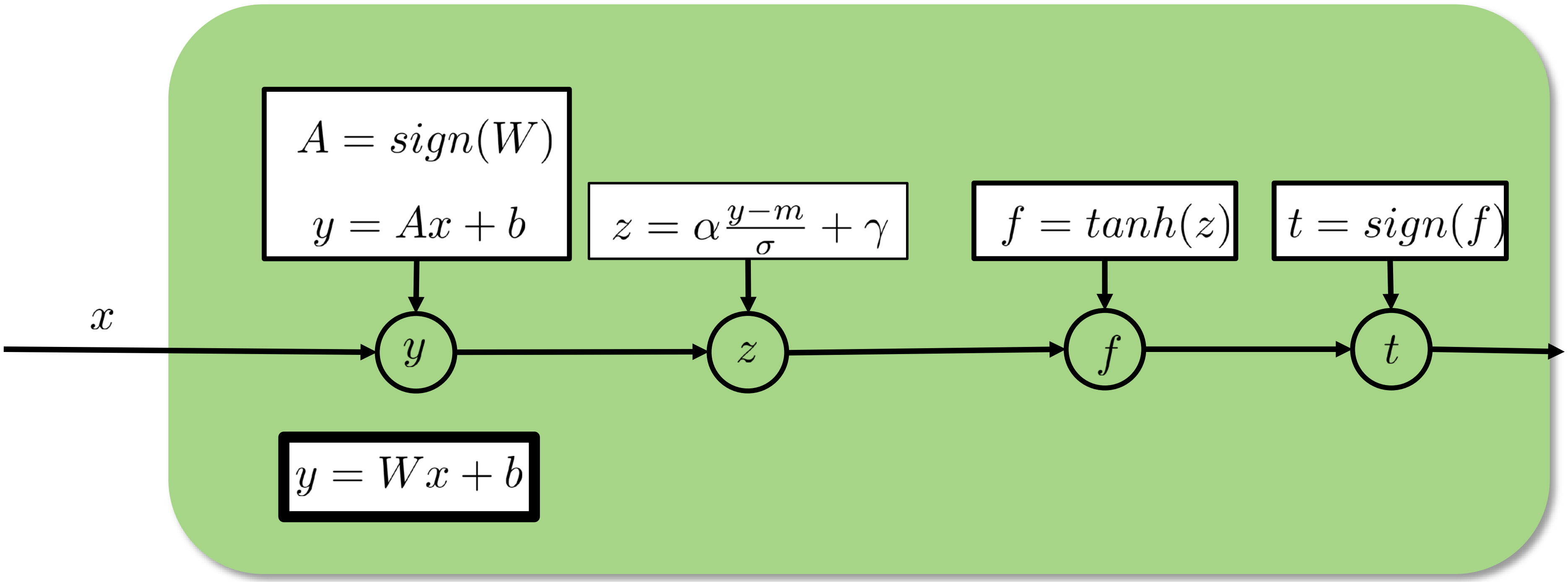
+ eliminate reifications

Binarized Neural Network



$x, a_{i,j} \in \{-1, 1\}$ $\qquad$ $b, \alpha, m, \sigma, \gamma, W \in \mathbf{R}$

Binarized Neural Network



$x, a_{i,j} \in \{-1, 1\}$ $\qquad b, \alpha, m, \sigma, \gamma, W \in \mathbf{R}$

Ternary quantization

BNN+: Improved Binary Network Training
Sajad Darabi, Mouloud Belbahri, Matthieu Courbariaux, Vahid Partovi Nia

Ternary quantization

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

where
$a_{i,j} = 1 \Rightarrow l_j = x_j,$
$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

Ternary quantization

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

where

$a_{i,j} = 1 \Rightarrow l_j = x_j,$

$a_{i,j} = 0 \Rightarrow l_j = 0,$

$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

L1+Ternary quantization

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

where

$a_{i,j} = 1 \Rightarrow l_j = x_j,$

$a_{i,j} = 0 \Rightarrow l_j = 0,$

$a_{i,j} = -1 \Rightarrow l_j = \bar{x}_j$

Add L1 regularization

L1+Ternary quantization

1. Train a BNN
2. Build a distribution of absolute values of weights
3. Select a percentile (40%, 60%), t= 0.03
4. Train a ternary BNN with the two-sided threshold t

$$a_{i,j} = \begin{cases} 0 & \text{if } |w_{i,j}| \leq t \\ sign(w_{i,j}) & \text{otherwise} \end{cases}$$

Stabilization of SIGN

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$LB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} \geq 0$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$LB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} \geq 0 \qquad t_1 = 1$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$UB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} < 0$$

Stabilization of SIGN

$$(l_{1,1} + \ldots + l_{1,n} - k_1 \geq 0) \Leftrightarrow t_1 = 1$$

$$UB_{(l_{1,1}+\ldots+l_{1,n}-k_1)} < 0 \qquad\qquad t_1 = 0$$

Stabilization of SIGN

Encourage LB and UB of a neurons to take the same sign:

$$sign(UB_{i,j}) = sign(LB_{i,j})$$

Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry

Stabilization of SIGN

Encourage LB and UB of a neurons to take the same sign:

$$-sign(UB_{i,j}) * sign(LB_{i,j})$$

Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry

Stabilization of SIGN

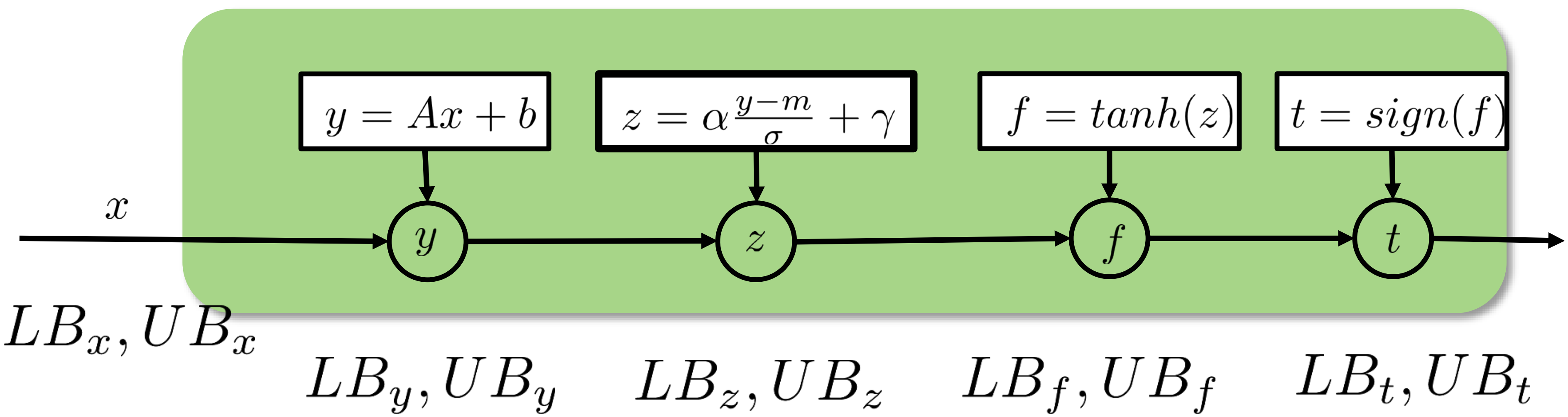
Encourage LB and UB of a neurons to take the same sign:

$$\cancel{-sign(UB_{i,j}) * sign(LB_{i,j})}$$

$$-tanh(1 + UB_{ij}LB_{ij})$$

Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability
Kai Y. Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, Aleksander Madry

Stabilization of SIGN



Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	$623K$	82.1	$623K$	74.3	$623K$
Sparse	96.4	$32K$	84.1	$37K$	78.2	$41K$
Sparse+Stable	95.9	$32K$	83.2	$37K$	78.3	$38K$
Sparse+L1	96.0	$20K$	83.7	$35K$	78.4	$36K$
Sparse+L1+Stable	95.2	$20K$	82.9	$37K$	80.0	$34K$

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	623K	82.1	623K	74.3	623K
Sparse	96.4	32K	84.1	37K	78.2	41K
Sparse+Stable	95.9	32K	83.2	37K	78.3	38K
Sparse+L1	96.0	20K	83.7	35K	78.4	36K
Sparse+L1+Stable	95.2	20K	82.9	37K	80.0	34K

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	$623K$	82.1	$623K$	74.3	$623K$
Sparse	96.4	$32K$	84.1	$37K$	78.2	$41K$
Sparse+Stable	95.9	$32K$	83.2	$37K$	78.3	$38K$
Sparse+L1	96.0	$20K$	83.7	$35K$	78.4	$36K$
Sparse+L1+Stable	95.2	$20K$	82.9	$37K$	80.0	$34K$

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	$623K$	82.1	$623K$	74.3	$623K$
Sparse	96.4	$32K$	84.1	$37K$	78.2	$41K$
Sparse+Stable	95.9	$32K$	83.2	$37K$	78.3	$38K$
Sparse+L1	96.0	$20K$	83.7	$35K$	78.4	$36K$
Sparse+L1+Stable	95.2	$20K$	82.9	$37K$	80.0	$34K$

Sparse+L1+StableSign

BNNs	MNIST		FASHION		MNISTBG	
	%	#prms	%	#prms	%	#prms
Vanilla	96.5	623K	82.1	623K	74.3	623K
Sparse	96.4	32K	84.1	37K	78.2	41K
Sparse+Stable	95.9	32K	83.2	37K	78.3	38K
Sparse+L1	96.0	20K	83.7	35K	78.4	36K
Sparse+L1+Stable	95.2	20K	82.9	37K	80.0	34K

Sparse+L1+StableSign

BNNs	MNIST #vars/#cls	FASHION #vars/#cls	MNISTBG #vars/#cls
Sparse	63K/224K	34K/116K	24K/80K
Sparse+Stable	42K/146K	19K/58K	12K/36K
Sparse+L1	8K/20K	34K/115K	17K/53K
Sparse+Stable+L1	11K/33K	12K/33K	10K/28K

Efficient Exact Verification of Binarized Neural Networks

Kai Jia, Martin Rinard

Neurips'20

1. Improved sparsity

Ternary quantization

Balanced ternary quantization

2. Friendly reified cardinality

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

...

Number of variables

Reification means no propagation!

2. Friendly reified cardinality

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

...

Number of variable

Reification means no propagation!

Force k to me small!

2. Friendly reified cardinality

$$(l_{1,1} + \ldots + l_{1,n} \geq k_1) \Leftrightarrow t_1 = 1$$

...

Number of variable

Reification means no propagation!

Force k to me small!

3. Improved adversarial training

Improved the backpropagation procedure to make PGD attacks more effective

4. Improved the SAT solver

Keep cardinality constraints natively

Impressive preformence

		Mean Time (s)			Accuracy			Timeout
		Build	Solve	Total	Verifiable	Natural	PGD	
MNIST $\epsilon = 0.1$	EEV S	0.0158	0.0004	0.0162	89.29%	97.44%	93.47%	0
	EEV L	0.1090	0.0025	0.1115	91.68%	97.46%	95.47%	0
	Xiao et al. [63] S	4.98	0.49	5.47	94.33%	98.68%	95.13%	0.05%
	Xiao et al. [63] L*	156.74	0.27	157.01	95.6%	98.95%	96.58%	0
MNIST $\epsilon = 0.3$	EEV S	0.0140	0.0006	0.0146	66.42%	94.31%	80.70%	0
	EEV L	0.1140	0.0039	0.1179	77.59%	96.36%	87.90%	0
	Xiao et al. [63] S	4.34	2.78	7.12	80.68%	97.33%	92.05%	1.02%
	Xiao et al. [63] L*	166.39	37.45	203.84	59.6%	97.54%	93.25%	24.1%
CIFAR10 $\epsilon = \frac{2}{255}$	EEV S	0.0258	0.0013	0.0271	26.13%	46.58%	33.70%	0
	EEV L	0.1653	0.0097	0.1750	30.49%	47.35%	38.22%	0
	Xiao et al. [63] S	52.58	13.50	66.08	45.93%	61.12%	49.92%	1.86%
	Xiao et al. [63] L*	335.97	29.88	365.85	41.4%	61.41%	50.61%	9.6%
CIFAR10 $\epsilon = \frac{8}{255}$	EEV S	0.0313	0.0014	0.0327	18.93%	37.75%	24.60%	0
	EEV L	0.1691	0.0090	0.1781	22.55%	35.00%	26.41%	0
	Xiao et al. [63] S	38.34	22.33	60.67	20.27%	40.45%	26.78%	2.47%
	Xiao et al. [63] L*	401.72	20.14	421.86	19.8%	42.81%	28.69%	5.4%

Impressive preformence

		Mean Time (s)			Accuracy			Timeout
		Build	Solve	Total	Verifiable	Natural	PGD	
MNIST $\epsilon = 0.1$	EEV S	0.0158	0.0004	0.0162	89.29%	97.44%	93.47%	0
	EEV L	0.1090	0.0025	0.1115	91.68%	97.46%	95.47%	0
	Xiao et al. [63] S	4.98	0.49	5.47	94.33%	98.68%	95.13%	0.05%
	Xiao et al. [63] L*	156.74	0.27	157.01	95.6%	98.95%	96.58%	0
MNIST $\epsilon = 0.3$	EEV S	0.0140	0.0006	0.0146	66.42%	94.31%	80.70%	0
	EEV L	0.1140	0.0039	0.1179	77.59%	96.36%	87.90%	0
	Xiao et al. [63] S	4.34	2.78	7.12	80.68%	97.33%	92.05%	1.02%
	Xiao et al. [63] L*	166.39	37.45	203.84	59.6%	97.54%	93.25%	24.1%
CIFAR10 $\epsilon = \frac{2}{255}$	EEV S	0.0258	0.0013	0.0271	26.13%	46.58%	33.70%	0
	EEV L	0.1653	0.0097	0.1750	30.49%	47.35%	38.22%	0
	Xiao et al. [63] S	52.58	13.50	66.08	45.93%	61.12%	49.92%	1.86%
	Xiao et al. [63] L*	335.97	29.88	365.85	41.4%	61.41%	50.61%	9.6%
CIFAR10 $\epsilon = \frac{8}{255}$	EEV S	0.0313	0.0014	0.0327	18.93%	37.75%	24.60%	0
	EEV L	0.1691	0.0090	0.1781	22.55%	35.00%	26.41%	0
	Xiao et al. [63] S	38.34	22.33	60.67	20.27%	40.45%	26.78%	2.47%
	Xiao et al. [63] L*	401.72	20.14	421.86	19.8%	42.81%	28.69%	5.4%

Outline

Motivation

Adversarial attacks

Verification methods

SAT-based verification of Binarized NNs

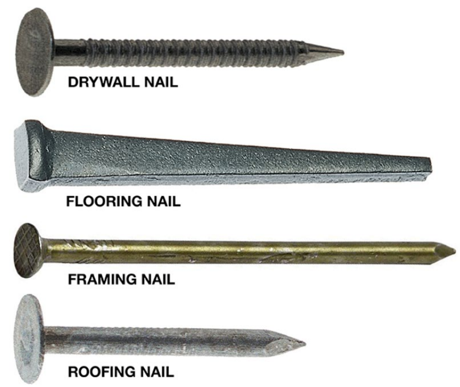
Where we are

Verification methods

Nails

Where we are

Verification methods



Nails

Where we are

VNN-LIB
Verification of Neural Networks

HOME ABOUT NEWS STANDARD BENCHMARKS SOFTWARE CREDITS

Home

VNN-LIB is an international initiative whose aim is to encourage collaboration and facilitate research and development in Verification of Neural Networks (VNN).

The goals of VNN-LIB are:

- Develop a cohesive community around VNN by connecting developers and researchers working in this domain.
- Establish a common format for the exchange of Neural Networks and their properties.
- Provide the community with a library of established common benchmarks for VNN tools.
- Provide and maintain a common repository for tools and resources useful to the VNN community.

The initiative and this site are still in their embryonal stages: your collaboration is essential to grow and improve VNN-LIB, so do not hesitate to send us feedback, comments and suggestions.

Where we are

VNN 2020

Home · Program · Call for Papers and Benchmarks · **VNN-COMP**

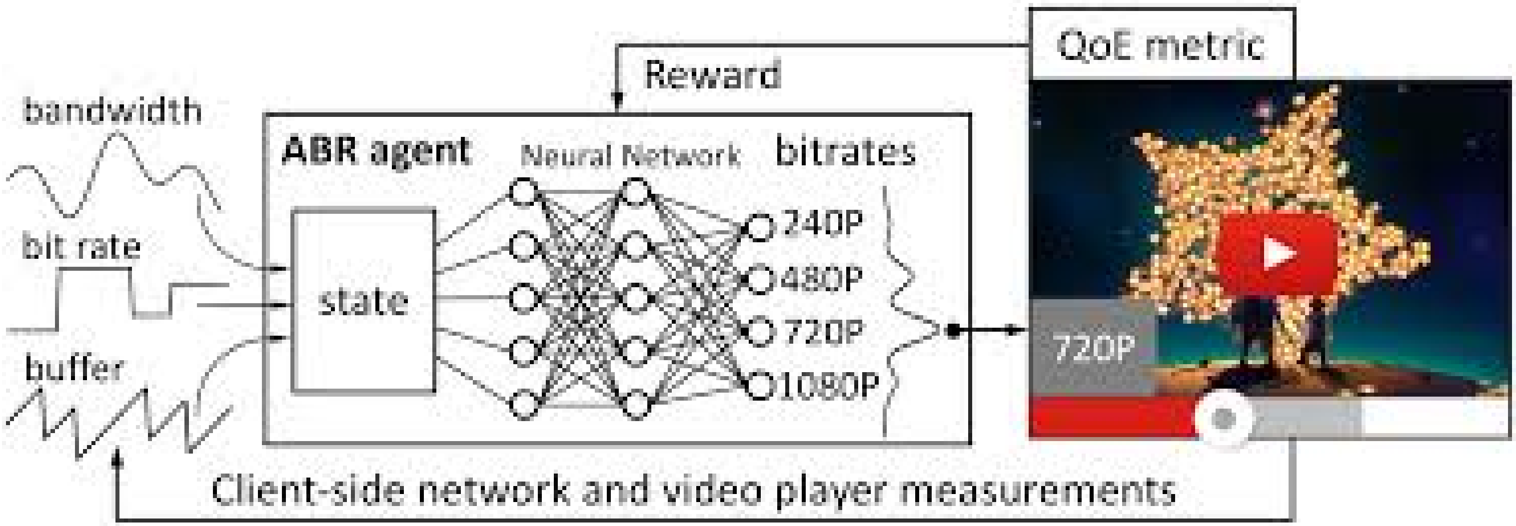
VNN-COMP

VNN-COMP 2020 Report

A draft (read only) version of the report is available on Overleaf here: https://www.overleaf.com/read/rbcfnbyhymmy

VNN-COMP 2020 Call for Participation

What is next?


QoE metric
Reward
bandwidth
ABR agent
Neural Network
bitrates
bit rate
state
240P
480P
720P
1080P
buffer
720P
Client-side network and video player measurements

What is next?

1. Verification is a very important tool to analyze NNs

2. Smaller networks are useful in many practical applications

Thanks!

Logic-Enabled Verification and Explanation of ML Models

Part 4

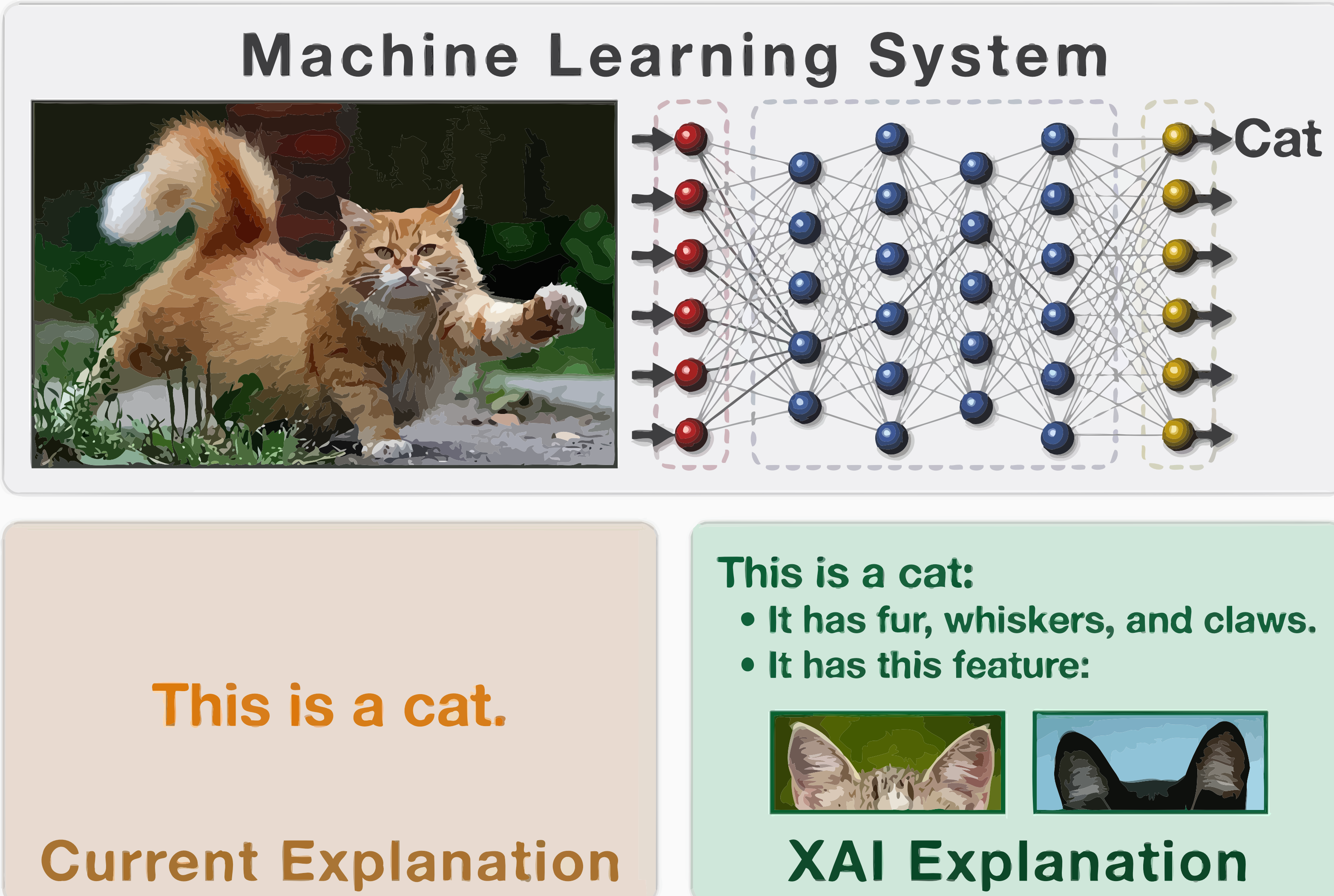
A. Ignatiev, J. Marques-Silva, K. Meel & N. Narodytska

Monash Univ, ANITI@Univ. Toulouse, NU Singapore & VMWare Research

January 08, 2021 | IJCAI Tutorial T22

Computing Explanations

What do we want to achieve?



©DARPA

interpretable ML models

(decision trees, lists, sets)

A recap: approaches to XAI

interpretable ML models

(decision trees, lists, sets)

explanation of ML models “on the fly”

(post-hoc explanation)

Why? or Why not? explanations

why?   why not?

*(**why** did (**not**) I get a loan?)*

Why? or Why not? explanations

why? **why not?**

*(**why** did (**not**) I get a loan?)*

abductive ***contrastive***

Heuristic approaches exist

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

- **local** explanations

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

- **local** explanations
- **no** guarantees

State of the art (heuristics)

heuristic approaches exist

(e.g. **LIME**, **Anchor**, or **SHAP**) [RSG16, RSG18, LL17]

- **local** explanations
- **no** guarantees

(un-)**reliable**?

Rigorous approaches

State of the art (rigorous approaches)

alternative is to use logic

alternative is to use logic

(reasoning over formal models)

State of the art (rigorous approaches)

alternative is to use logic

(reasoning over formal models)

⇩

- **search**

State of the art (rigorous approaches)

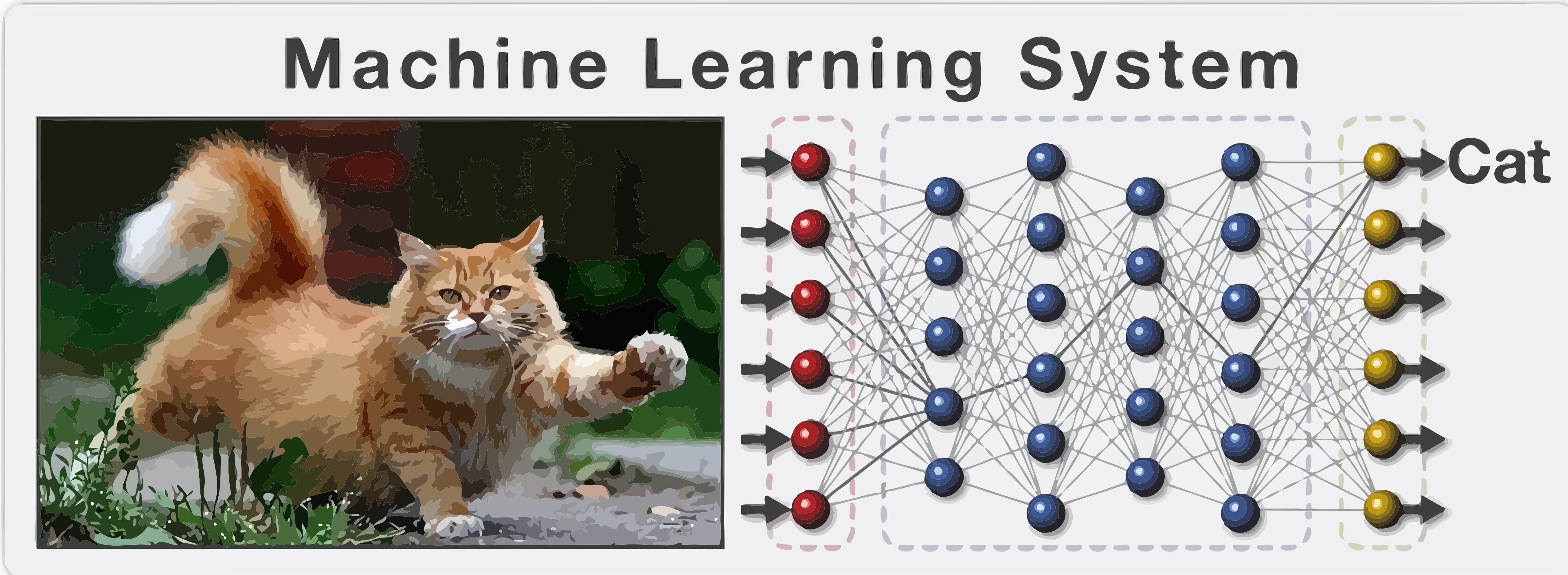
alternative is to use logic

(reasoning over formal models)

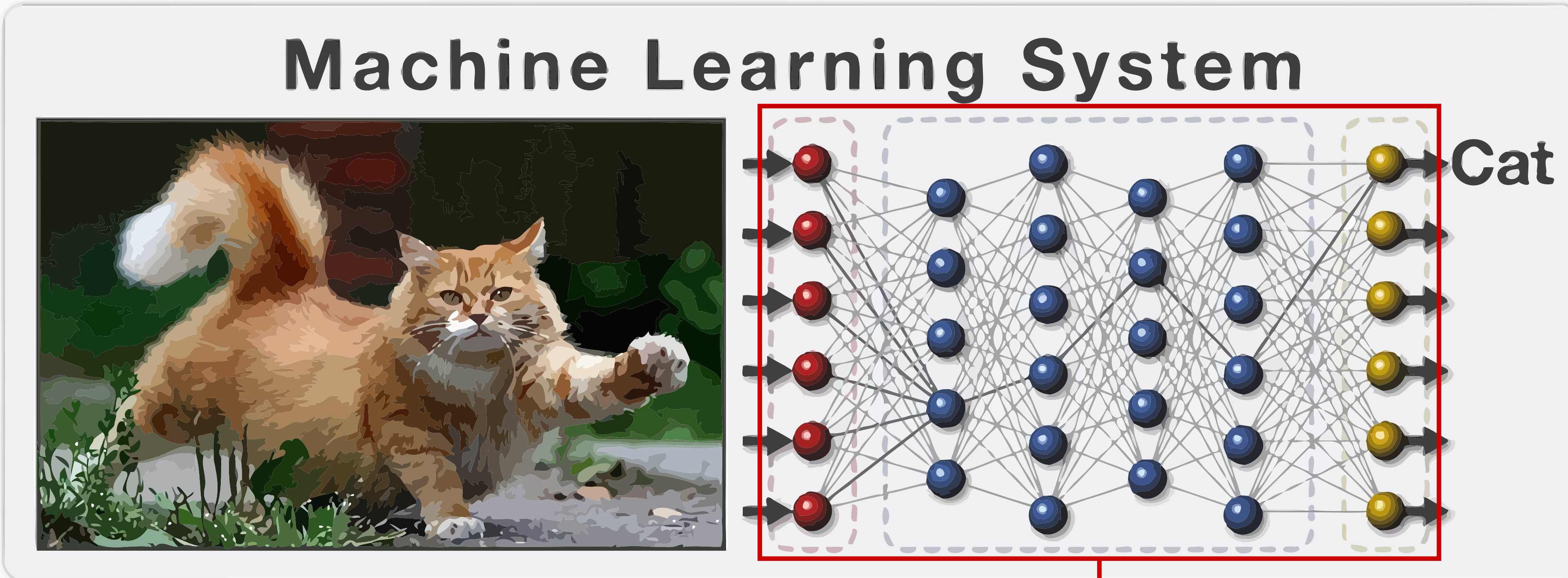
- **search**
- **compilation**

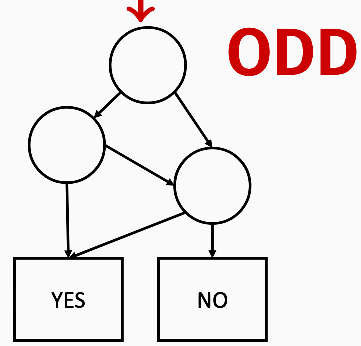
Compilation-based approach

Compiling a classifier

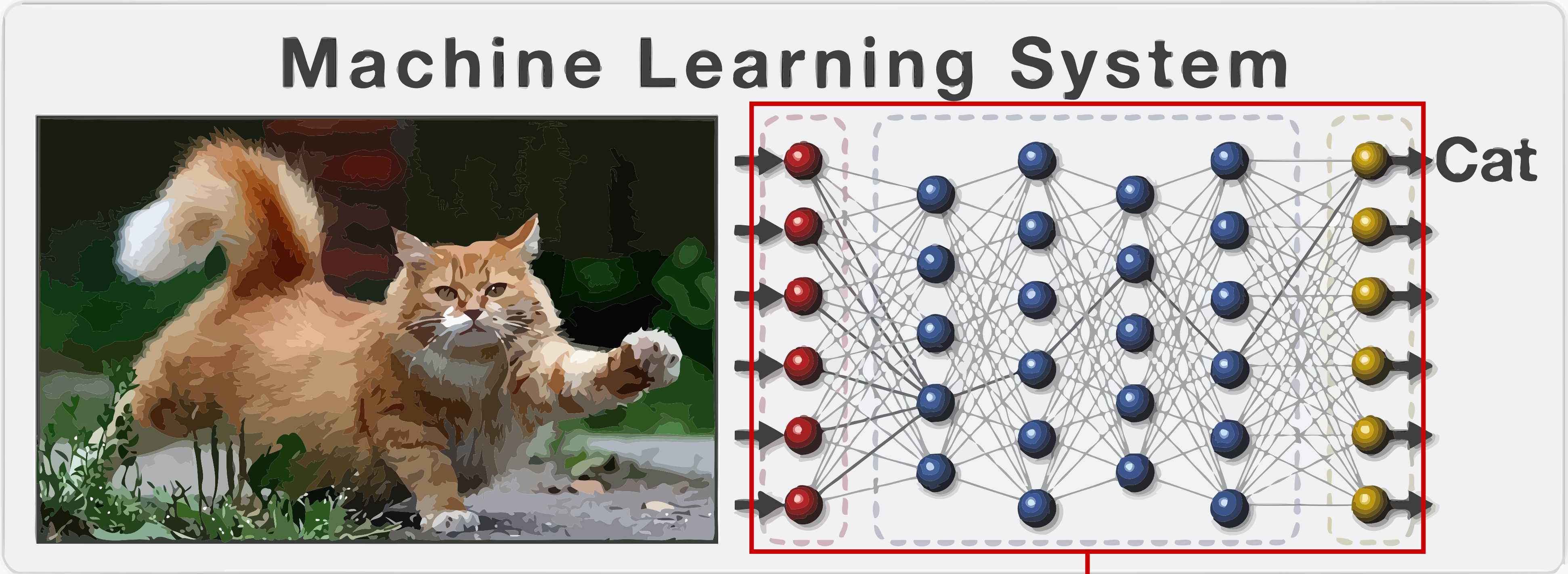


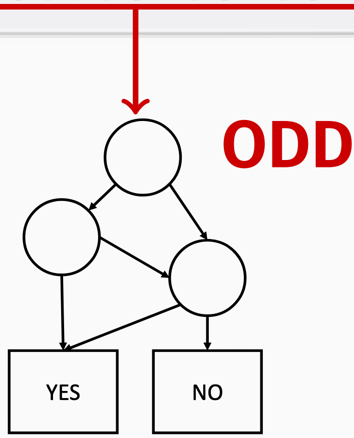
Compiling a classifier





Compiling a classifier





perform operations on
tractable representation

The idea is that

once you have an ODD:

once you have an ODD:

- compute **MC-explanations** [SCD18]

  "Which positive features are responsible for a yes decision?"

  "Which negative features are responsible for a no decision?"

once you have an ODD:

- compute **MC-explanations** [SCD18]

  "Which positive features are responsible for a yes decision?"

  "Which negative features are responsible for a no decision?"

- compute **PI-explanations** [SCD18, DH20]

  "Which features (+ or -) make the other features irrelevant?"

once you have an ODD:

- compute **MC-explanations** [SCD18]
  "Which positive features are responsible for a yes decision?"
  "Which negative features are responsible for a no decision?"

- compute **PI-explanations** [SCD18, DH20]
  "Which features (+ or -) make the other features irrelevant?"

- perform **verification queries** [SDC19]
  counting of counterexamples, computing their probabilites and common characteristics

What ML models can we compile?

- **Naïve Bayes** [CD03]

What ML models can we compile?

- **Naïve Bayes** [CD03]
- **Latent Tree** [SCD18]

What ML models can we compile?

- **Naïve Bayes** [CD03]
- **Latent Tree** [SCD18]
- **General BN** [SCD19]

What ML models can we compile?

- **Naïve Bayes** [CD03]
- **Latent Tree** [SCD18]
- **General BN** [SCD19]
- **BNN and CNN** [SDC19]

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

but

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

but

difficult to compute an ODD

Pros and cons of ML model compilation

reasoning about explanations in polynomial time

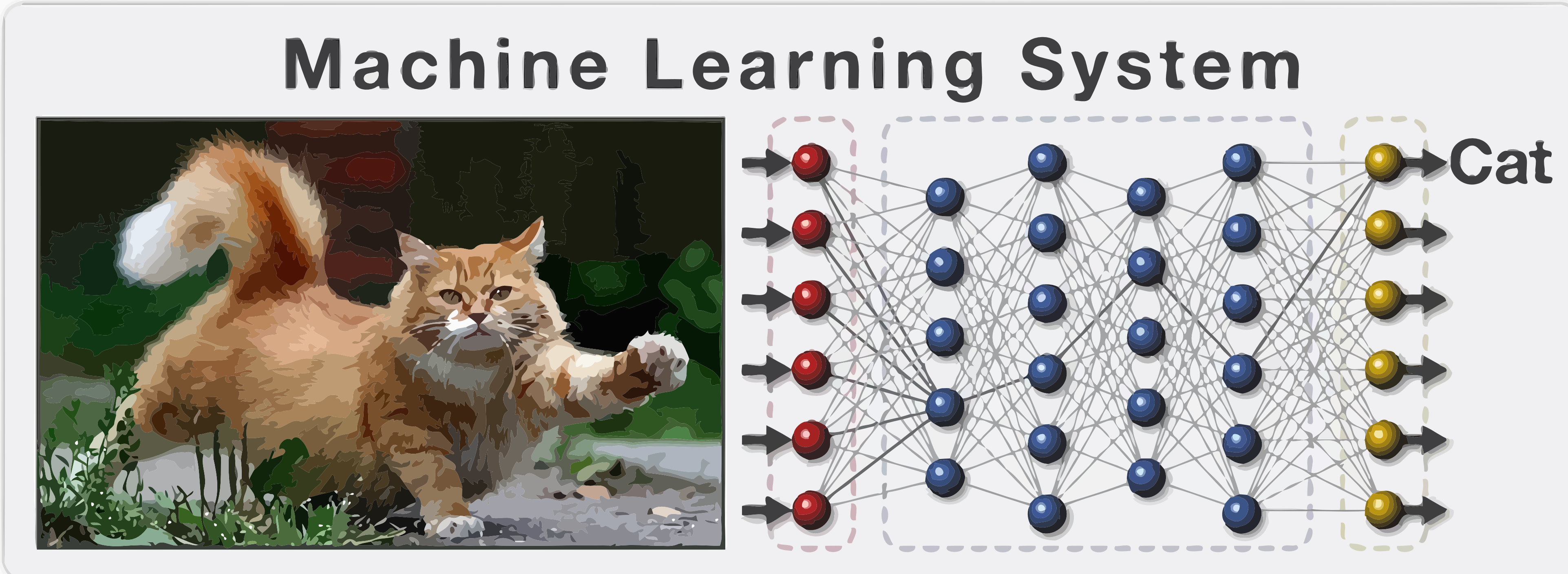
but

difficult to compute an ODD

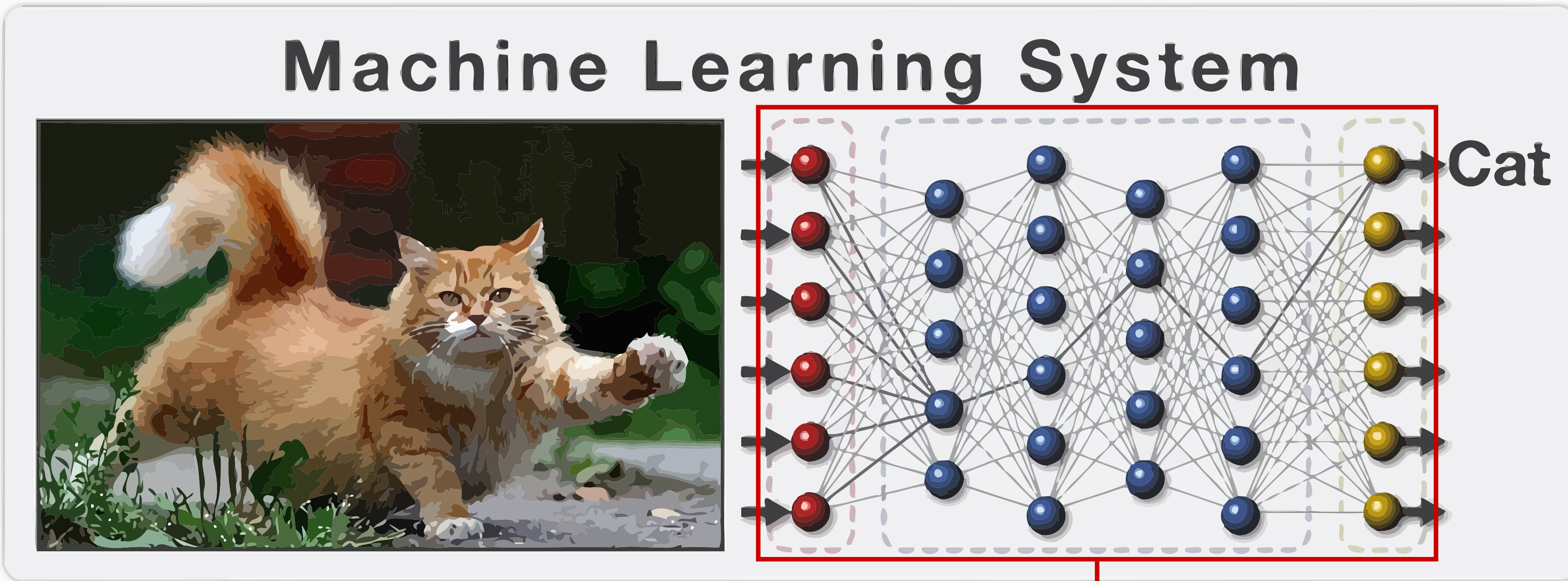
ODD can be *large*

Search-based explanations

From ML model to logic

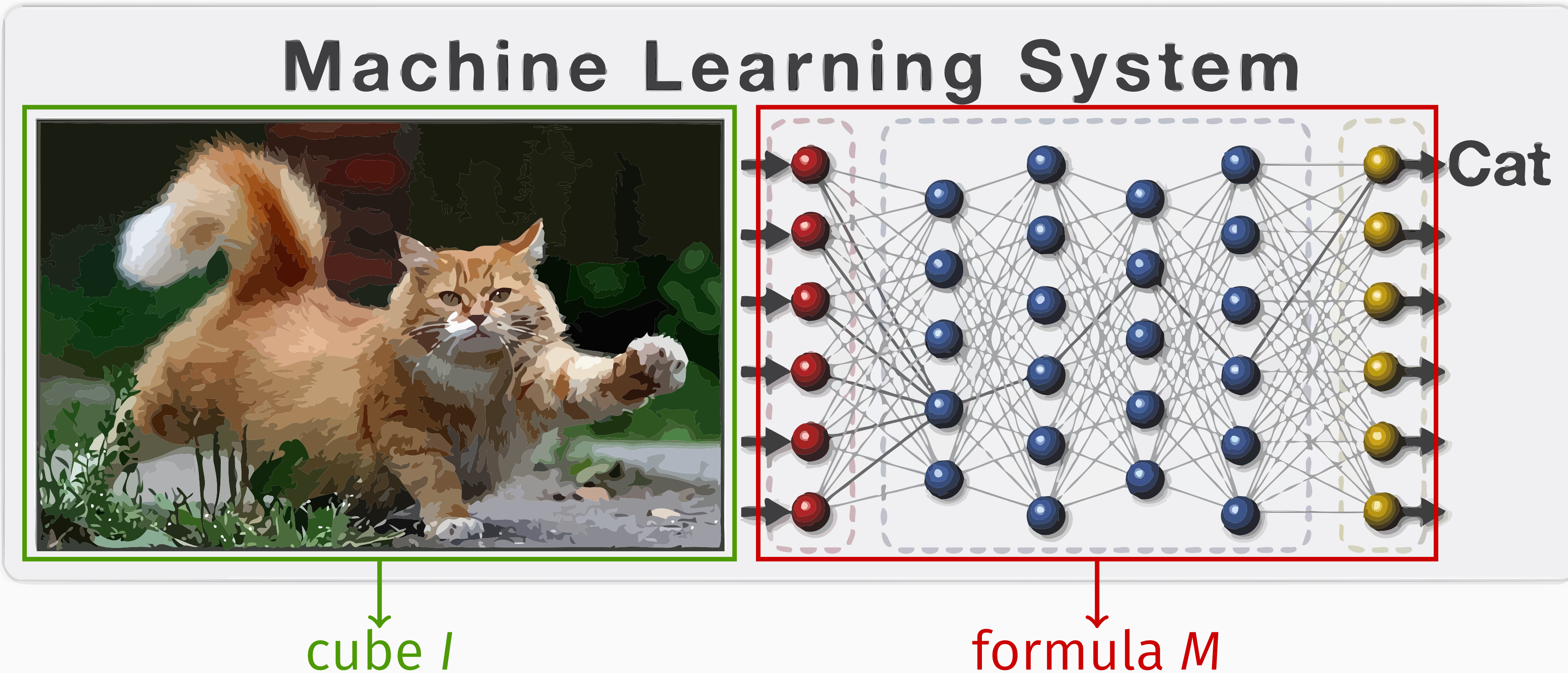


From ML model to logic

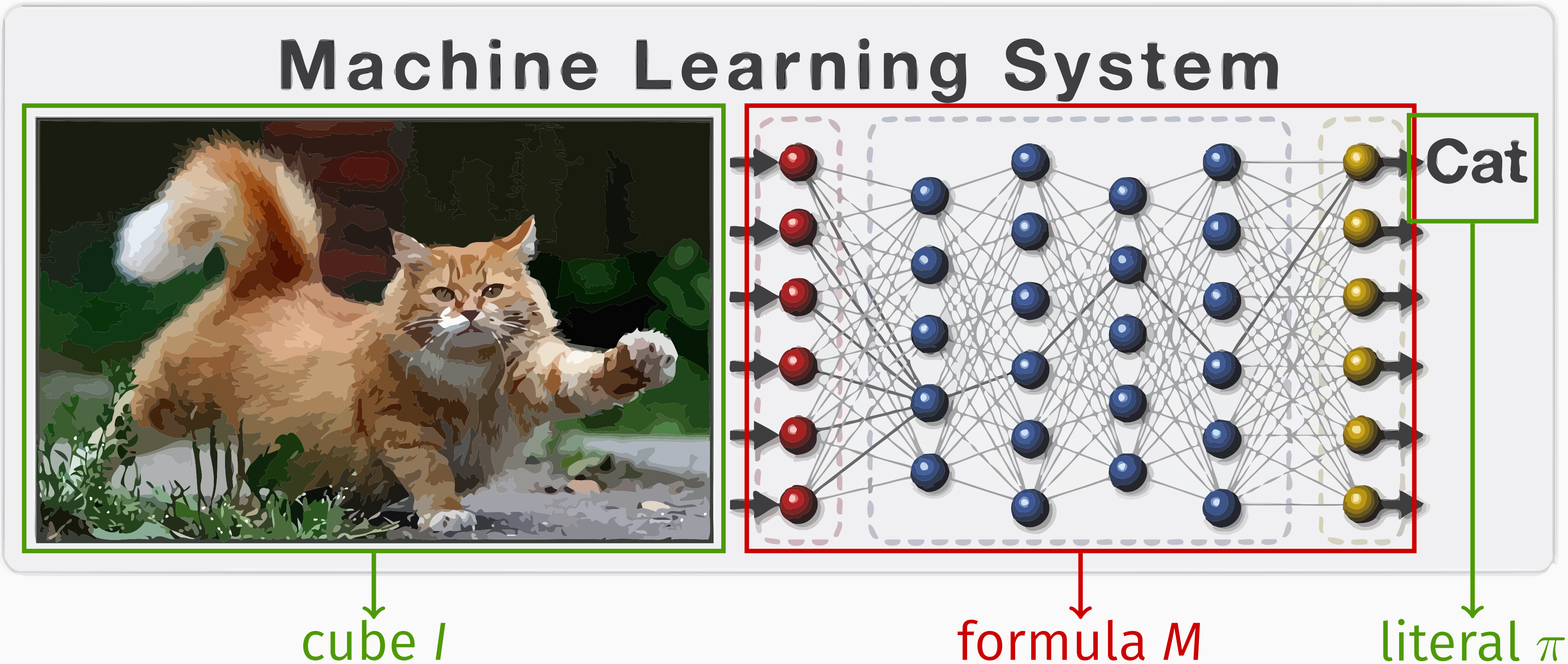


formula M

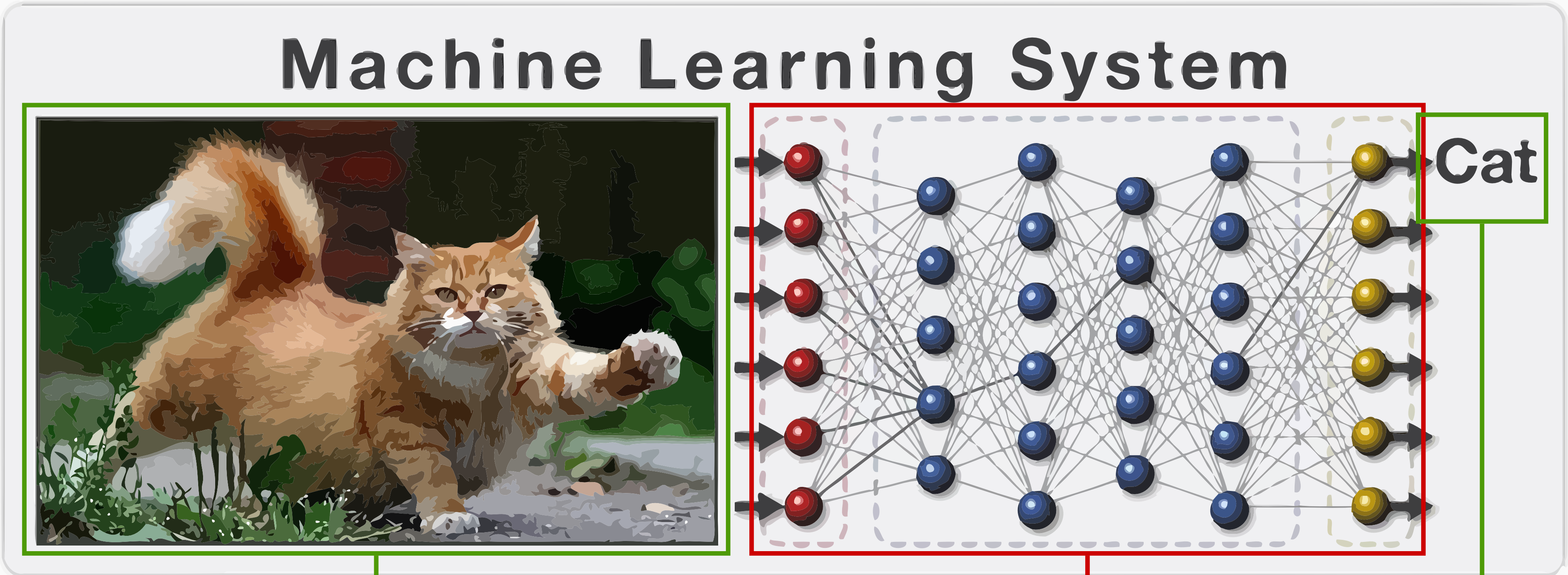
From ML model to logic



From ML model to logic



From ML model to logic



cube I formula M literal π

$$I \land M \vDash \pi$$

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

compute a (**cardinality- or subset-**) minimal $E_m \subseteq I$ s.t.

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

compute a (**cardinality- or subset-**) minimal $E_m \subseteq I$ s.t.

$$E_m \wedge M \not\models \perp$$

and

$$E_m \wedge M \models \pi$$

Abductive explanations of ML models

[INMS19]

given a ***classifier*** M, a ***cube*** I and a ***prediction*** π,

compute a (**cardinality- or subset-**) minimal $E_m \subseteq I$ s.t.

$$E_m \wedge M \not\models \perp$$

and

$$E_m \wedge M \models \pi$$

⇩

iterative **explanation procedure**

Computing primes

1. $E_m \wedge M \not\models \bot$

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***
2. $E_m \wedge M \models \pi$

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***
2. $E_m \wedge M \models \pi \iff E_m \models (M \rightarrow \pi)$

Computing primes

1. $E_m \wedge M \not\models \bot$ — ***tautology***
2. $E_m \wedge M \models \pi \iff E_m \models (M \rightarrow \pi)$

$\Downarrow$

E_m is a ***prime implicant*** of $M \rightarrow \pi$

Computing one subset-minimal explanation

Input: model M, initial cube I, prediction π
Output: ***Subset-minimal*** explanation E_m

```
begin
   for l ∈ I :
      if Entails(I \ {l}, M → π) :          # make an (entailment) oracle call
         I ← I \ {l}
   return I
end
```

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

(following **implicit-hitting set** based approach) [IMM16]

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

(following **implicit-hitting set** based approach) [IMM16]

⇩

but it is **hard for Σ_2^P** [INMS19]

Computing one cardinality-minimal explanation

cardinality-minimal explanations can be computed

(following **implicit-hitting set** based approach) [IMM16]

but it is **hard for Σ_2^P** [INMS19]

(**worst-case exponential** number of oracle queries)

Experimental setup

- implementation in Python
  - supports **SMT** solvers through PySMT
    - **Yices2** used
  - supports **CPLEX 12.8.0**

Experimental setup

- implementation in Python
  - supports **SMT** solvers through PySMT
    - **Yices2** used
  - supports **CPLEX 12.8.0**
- **ReLU-based** neural networks [FJ18]
  - one *hidden* layer with $i \in \{10, 15, 20\}$ neurons

Experimental setup

- implementation in Python
  - supports **SMT** solvers through PySMT
    - **Yices2** used
  - supports **CPLEX 12.8.0**
- **ReLU-based** neural networks [FJ18]
  - one *hidden* layer with $i \in \{10, 15, 20\}$ neurons
- benchmarks selected from:
  - **UCI** Machine Learning Repository
  - **Penn** Machine Learning Benchmarks
  - **MNIST** Digits Database

Experimental setup

- implementation in Python
  - supports **SMT** solvers through PySMT
    - **Yices2** used
  - supports **CPLEX 12.8.0**
- **ReLU-based** neural networks [FJ18]
  - one *hidden* layer with $i \in \{10, 15, 20\}$ neurons
- benchmarks selected from:
  - **UCI** Machine Learning Repository
  - **Penn** Machine Learning Benchmarks
  - **MNIST** Digits Database
- Machine configuration:
  - Intel Core i7 2.8GHz, 8GByte
  - time limit — 1800s
  - memory limit — 4GByte

Some of the experimental results

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

Some of the experimental results

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

Some of the experimental results

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

Some of the experimental results

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

Some of the experimental results

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

Comparing quality to compilation-based approach

- **“Congressional Voting Records”** dataset

Comparing quality to compilation-based approach

- **"Congressional Voting Records"** dataset
- `(0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 1)` — data sample **(16 features)**

Comparing quality to compilation-based approach

- **“Congressional Voting Records”** dataset
- `(0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 1)` — data sample **(16 features)**

smallest size explanations computed by **compilation for BN**: [SCD18]

- `(    0 1 1   0 0 0       1 1 0  )` — **9 literals**
- `(    0 1 1 1   0 0       1 1 0  )` — **9 literals**

Comparing quality to compilation-based approach

- **"Congressional Voting Records"** dataset
- `(0 1 0 1 1 1 0 0 0 0 0 0 1 1 0 1)` — data sample **(16 features)**

smallest size explanations computed by **compilation for BN**: [SCD18]

- `(    0 1 1   0 0 0       1 1 0  )` — **9 literals**
- `(    0 1 1 1   0 0       1 1 0  )` — **9 literals**

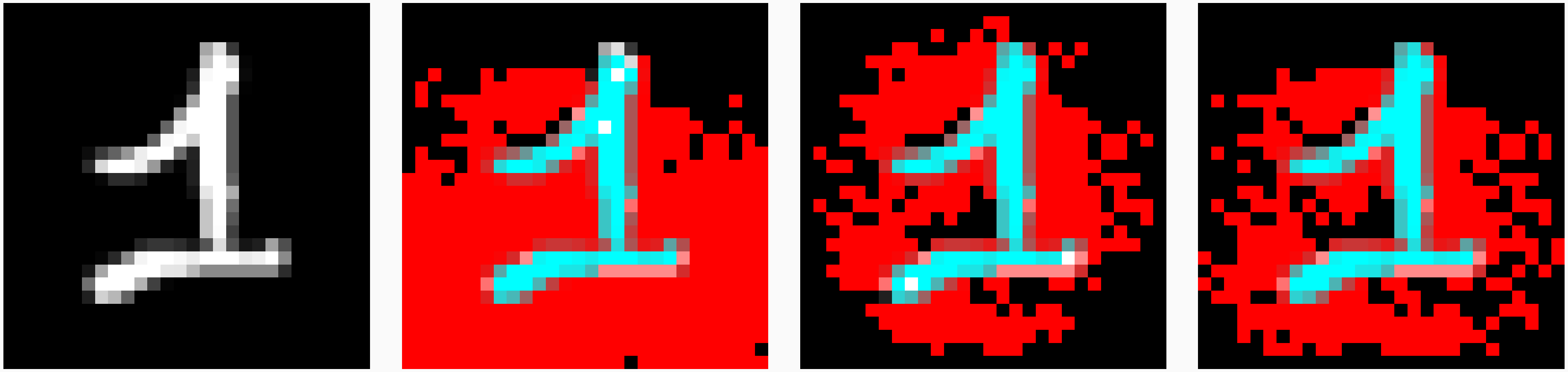
subset-minimal explanations computed by **search for ReLU-NNs**: [INMS19]

- `(      1         0   0       0  )` — **4 literals**
- `(      1         0   0          )` — **3 literals**
- `(    0 1         0   0       0  )` — **5 literals**
- `(    0 1         0   0         1)` — **5 literals**

What does it mean?

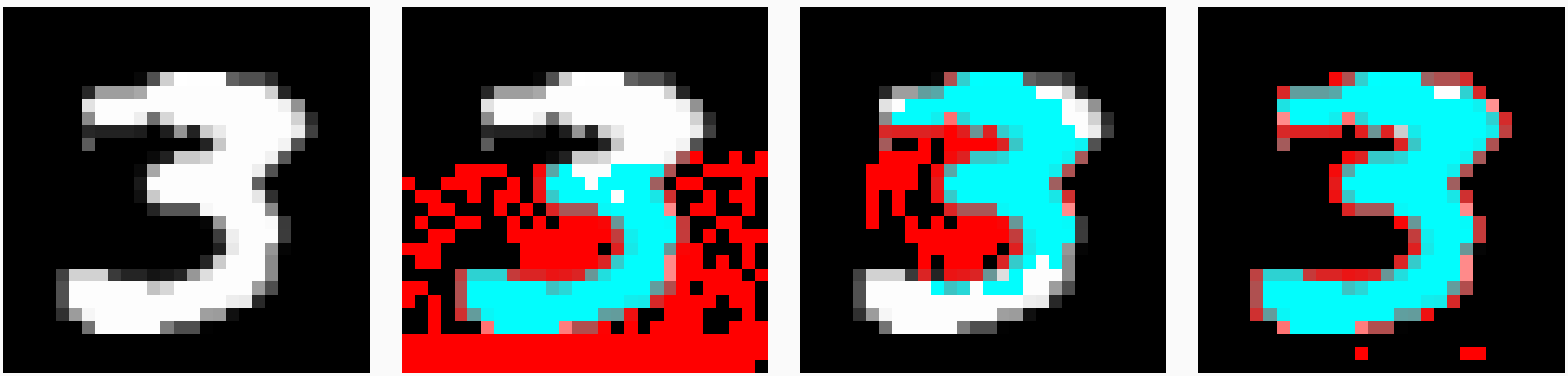
explanations can *hint* on the classifier quality!

MNIST examples



(a) (b) (c) (d)

Figure 1: Possible minimal explanations for digit one.



(a) (b) (c) (d)

And so what?

explanations are not equally good!

Summary on search-based explanations

principled approach to XAI

Summary on search-based explanations

principled approach to XAI

based on **abductive reasoning**

Summary on search-based explanations

principled approach to XAI

based on **abductive reasoning**

applies a **reasoning oracle**, e.g. SMT or MILP

Summary on search-based explanations

principled approach to XAI

based on **abductive reasoning**

applies a **reasoning oracle**, e.g. SMT or MILP

provides **minimality guarantees**

principled approach to XAI

based on **abductive reasoning**

applies a **reasoning oracle**, e.g. SMT or MILP

provides **minimality guarantees**

global explanations!

What next?

What next?

enumeration of **explanations**?

What next?

enumeration of **explanations**?

preferences over explanations?

What next?

enumeration of **explanations**?

preferences over explanations?

reasoning about explanations!

(assessment of heuristic approaches)

Assessing heuristic approaches

Heuristic approaches – a recap

heuristic approaches
(e.g. **LIME**, **Anchor**, **SHAP**) [RSG16, RSG18, LL17]

Heuristic approaches – a recap

heuristic approaches
(e.g. **LIME**, **Anchor**, **SHAP**) [RSG16, RSG18, LL17]

local explanations

heuristic approaches
(e.g. **LIME**, **Anchor**, **SHAP**) [RSG16, RSG18, LL17]

local explanations
no minimality **guarantees**

Assessment setup

[INM19]

how good are heuristic explanations?

Assessment setup

[INM19]

how good are heuristic explanations?

let's check for boosted trees [CG16]

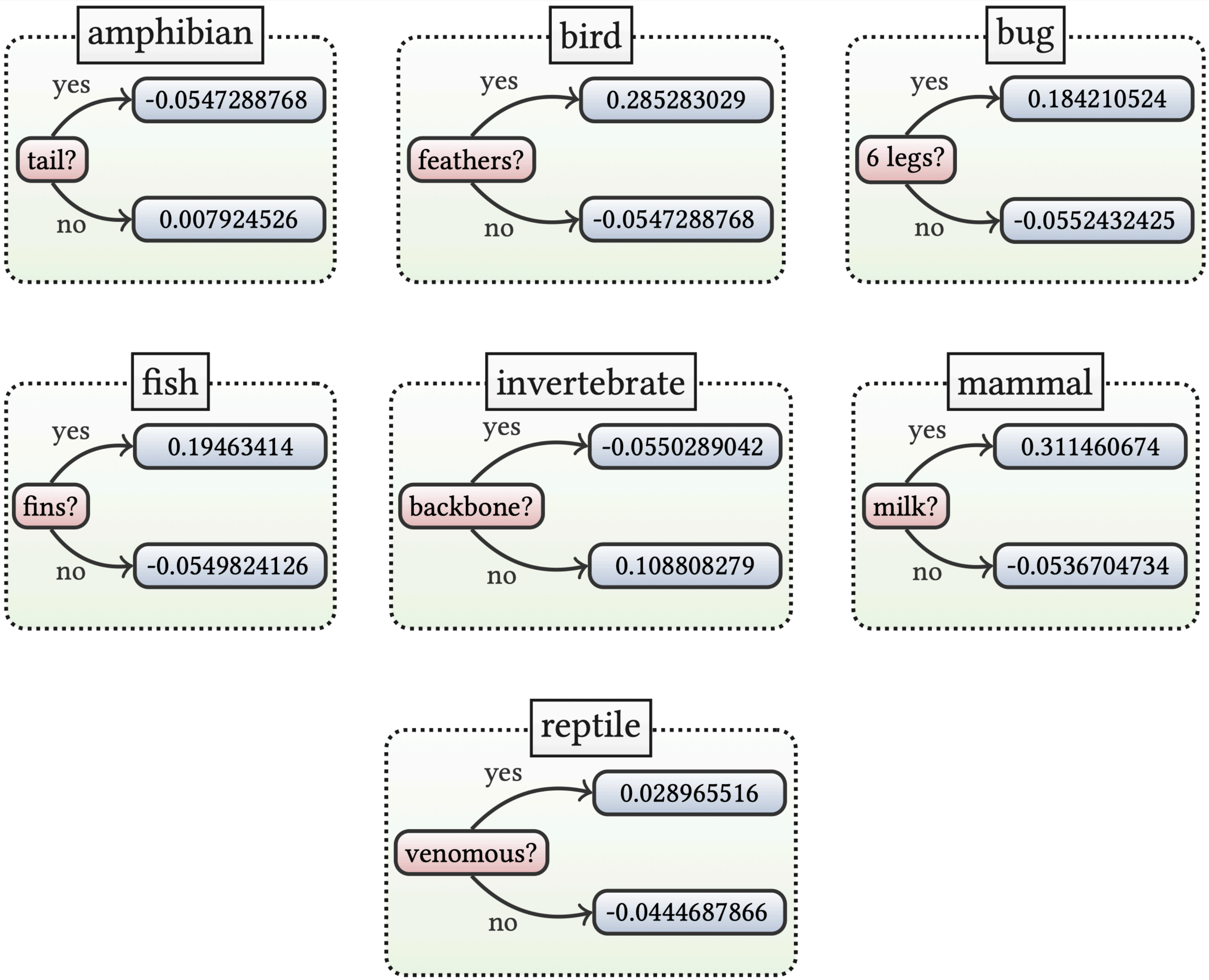
Assessment setup

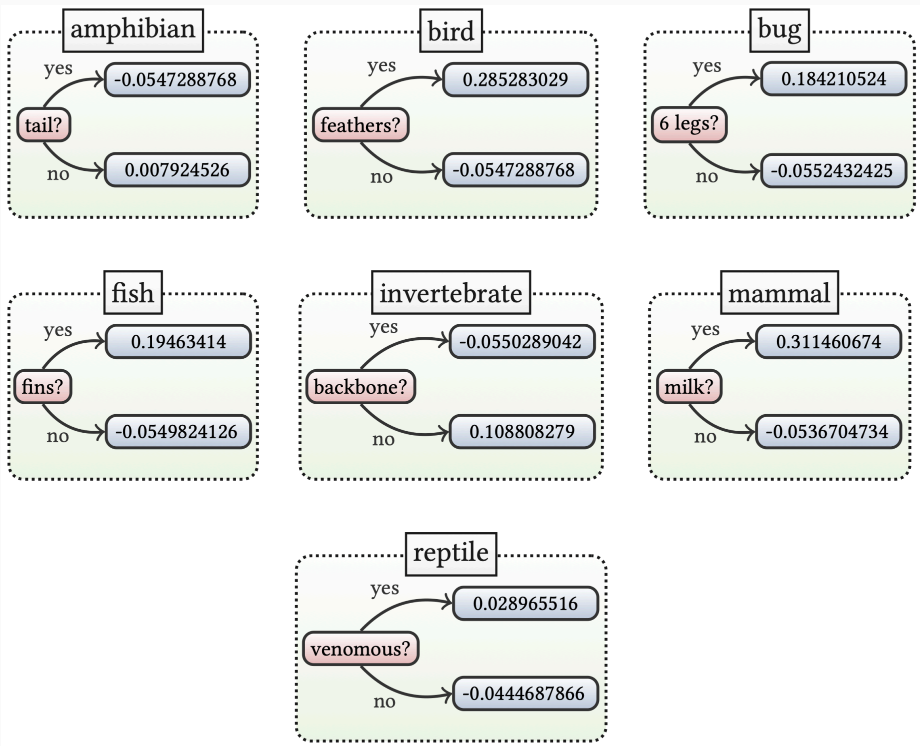
[INM19]

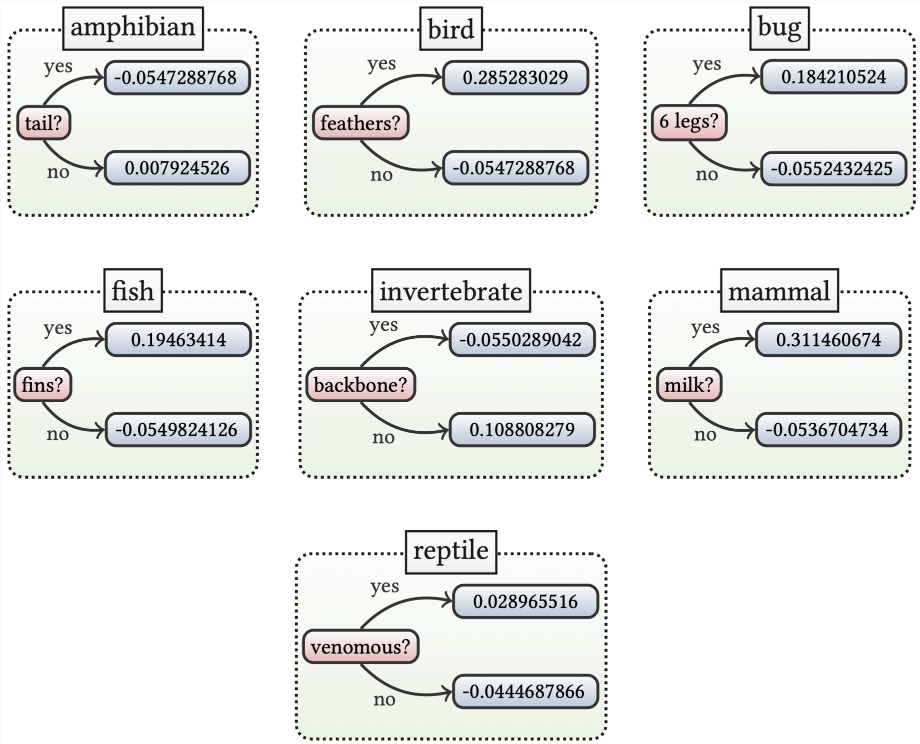
how good are heuristic explanations?

let's check for boosted trees [CG16]

(easy to encode) [BLM15, LMB17, VZY17, INM19]


amphibian
tail?
yes
-0.0547288768
no
0.007924526
bird
feathers?
yes
0.285283029
no
-0.0547288768
bug
6 legs?
yes
0.184210524
no
-0.0552432425
fish
fins?
yes
0.19463414
no
-0.0549824126
invertebrate
backbone?
yes
-0.0550289042
no
0.108808279
mammal
milk?
yes
0.311460674
no
-0.0536704734
reptile
venomous?
yes
0.028965516
no
-0.0444687866

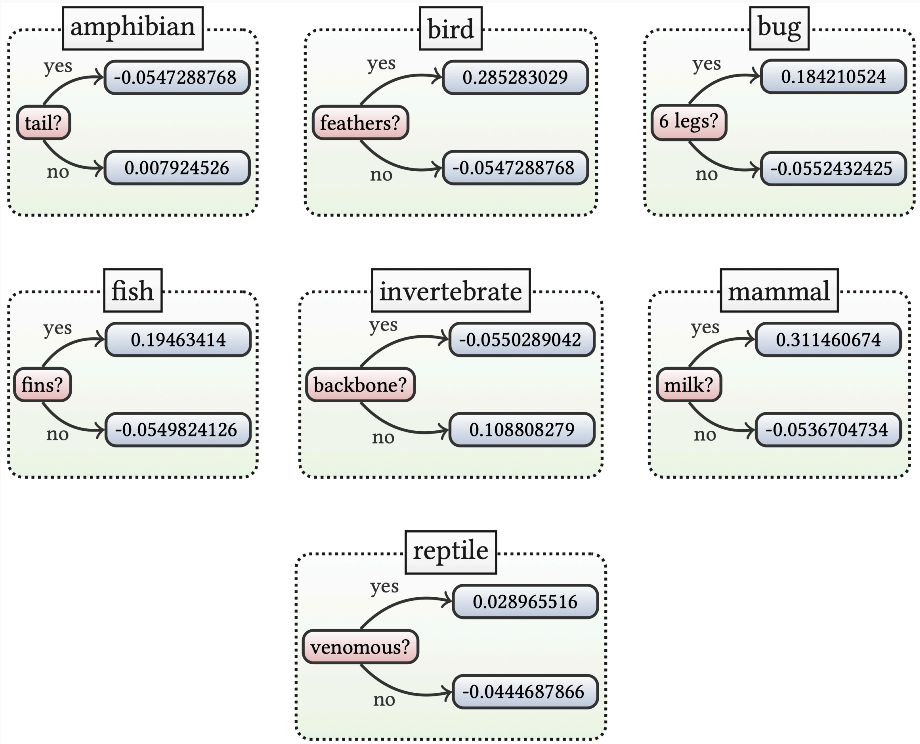

amphibian
yes
-0.0547288768
tail?
no
0.007924526
bird
yes
0.285283029
feathers?
no
-0.0547288768
bug
yes
0.184210524
6 legs?
no
-0.0552432425
fish
yes
0.19463414
fins?
no
-0.0549824126
invertebrate
yes
-0.0550289042
backbone?
no
0.108808279
mammal
yes
0.311460674
milk?
no
-0.0536704734
reptile
yes
0.028965516
venomous?
no
-0.0444687866



input instance:

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

THEN (class = reptile)



input instance:

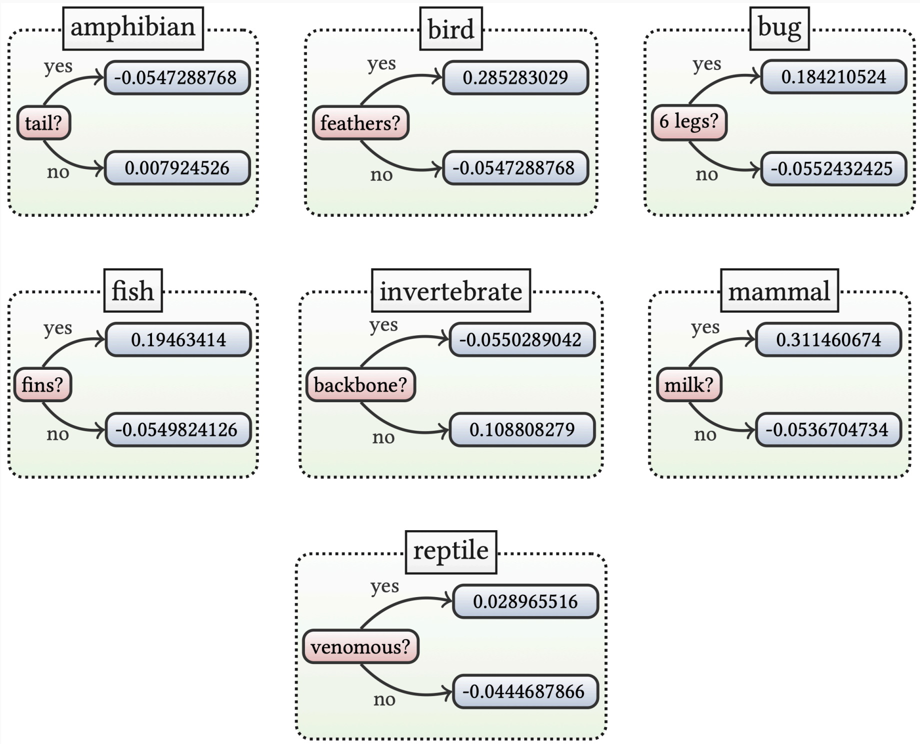
IF (animal_name = pitviper) $\wedge$ ¬hair $\wedge$ ¬feathers $\wedge$ eggs $\wedge$ ¬milk $\wedge$ ¬airborne $\wedge$ ¬aquatic $\wedge$ predator $\wedge$ ¬toothed $\wedge$ ¬fins $\wedge$ (legs = 0) $\wedge$ tail $\wedge$ ¬domestic $\wedge$ ¬catsize

THEN (class = reptile)

Anchor's explanation:

IF ¬hair $\wedge$ ¬milk $\wedge$ ¬toothed $\wedge$ ¬fins

THEN (class = reptile)



input instance:

IF (animal_name = pitviper) $\wedge$ ¬hair ¬feathers $\wedge$ eggs $\wedge$ ¬milk $\wedge$ ¬airborne$\wedge$ ¬aquatic $\wedge$ predator $\wedge$ ¬toothed $\wedge$ ¬fins$\wedge$ (legs $= 0$) $\wedge$ tail $\wedge$ ¬domestic $\wedge$ ¬catsize
THEN (class = reptile)

Anchor's explanation:

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

counterexample!

IF (animal_name = toad) $\wedge$ ¬hair ¬feathers $\wedge$ eggs $\wedge$ ¬milk $\wedge$ ¬airborne$\wedge$ ¬aquatic $\wedge$ ¬predator $\wedge$ ¬toothed $\wedge$ ¬fins$\wedge$ (legs $= 4$) $\wedge$ ¬tail $\wedge$ ¬domestic $\wedge$ ¬catsize
THEN (class = amphibian)

how?

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \models (\mathcal{M} \to \pi)$

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \vDash (\mathcal{M} \rightarrow \pi)$

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \vDash (\mathcal{M} \rightarrow \pi)$

$\mathcal{E}_h \wedge \mathcal{M} \wedge \neg\pi$ — **satisfiable**

how?

given $\mathcal{E}_h$, $\mathcal{E}_h \models (\mathcal{M} \to \pi)$

$$\mathcal{E}_h \wedge \mathcal{M} \wedge \neg\pi$$ — **satisfiable**

(in fact, this formula can have **many models**)

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (I_1, I_2) ← (I \ E_h, E_h)
    for l ∈ I_1 :
        if Entails(I_1 ∪ I_2 \ {l}, M → π) :
            I_1 ← I_1 \ {l}
    for l ∈ I_2 :
        if Entails(I_1 ∪ I_2 \ {l}, M → π) :
            I_2 ← I_2 \ {l}
    return I_1 ∪ I_2
end
```

begin

$(\mathcal{I}_1, \mathcal{I}_2) \leftarrow (\mathcal{I} \setminus \mathcal{E}_h, \mathcal{E}_h)$

for $l \in \mathcal{I}_1$ **:**

if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \to \pi)$ **:**

$\mathcal{I}_1 \leftarrow \mathcal{I}_1 \setminus \{l\}$

for $l \in \mathcal{I}_2$ **:**

if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \to \pi)$ **:**

$\mathcal{I}_2 \leftarrow \mathcal{I}_2 \setminus \{l\}$

return $\mathcal{I}_1 \cup \mathcal{I}_2$

end

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (I1, I2) ← (I \ Eh, Eh)
    for l ∈ I1 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I1 ← I1 \ {l}
    for l ∈ I2 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I2 ← I2 \ {l}
    return I1 ∪ I2
end
```

$$(\mathcal{I}_1, \mathcal{I}_2) \leftarrow (\mathcal{I} \setminus \mathcal{E}_h, \mathcal{E}_h)$$

for $l \in \mathcal{I}_1$ **:**
  if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \rightarrow \pi)$ **:**
    $\mathcal{I}_1 \leftarrow \mathcal{I}_1 \setminus \{l\}$

for $l \in \mathcal{I}_2$ **:**
  if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \rightarrow \pi)$ **:**
    $\mathcal{I}_2 \leftarrow \mathcal{I}_2 \setminus \{l\}$

return $\mathcal{I}_1 \cup \mathcal{I}_2$

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (I1, I2) ← (I \ Eh, Eh)
    for l ∈ I1 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I1 ← I1 \ {l}
    for l ∈ I2 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I2 ← I2 \ {l}
    return I1 ∪ I2
end
```

$(\mathcal{I}_1, \mathcal{I}_2) \leftarrow (\mathcal{I} \setminus \mathcal{E}_h, \mathcal{E}_h)$

for $l \in \mathcal{I}_1$: if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \rightarrow \pi)$: $\mathcal{I}_1 \leftarrow \mathcal{I}_1 \setminus \{l\}$

for $l \in \mathcal{I}_2$: if Entails$(\mathcal{I}_1 \cup \mathcal{I}_2 \setminus \{l\}, \mathcal{M} \rightarrow \pi)$: $\mathcal{I}_2 \leftarrow \mathcal{I}_2 \setminus \{l\}$

return $\mathcal{I}_1 \cup \mathcal{I}_2$

Repairing heuristic explanations

Input: model $\mathcal{M}$, initial cube $\mathcal{I}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
    (I1, I2) ← (I \ Eh, Eh)
    for l ∈ I1 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I1 ← I1 \ {l}
    for l ∈ I2 :
        if Entails(I1 ∪ I2 \ {l}, M → π) :
            I2 ← I2 \ {l}
    return I1 ∪ I2
end
```

$$(\mathcal{I}_1, \mathcal{I}_2) \leftarrow (\mathcal{I} \setminus \mathcal{E}_h, \mathcal{E}_h)$$

incorrect explanation

IF $\neg\text{hair} \land \neg\text{milk} \land \neg\text{toothed} \land \neg\text{fins}$

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

incorrect explanation

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

repaired explanation

IF $\neg$feathers $\wedge$ $\neg$milk $\wedge$ backbone$\wedge$
$\neg$fins $\wedge$ (legs $= 0$) $\wedge$ tail
THEN (class $=$ reptile)

Refining heuristic explanations

Input: model $\mathcal{M}$, heuristic explanation $\mathcal{E}_h$, prediction π
Output: ***Subset-minimal*** explanation $\mathcal{E}_m$

```
begin
  for l ∈ E_h :
    if Entails(E_h \ {l}, M → π) :
      E_h ← E_h \ {l}
  return E_h
end
```

begin

  for $l \in \mathcal{E}_h$ **:**

    if $\texttt{Entails}(\mathcal{E}_h \setminus \{l\}, \mathcal{M} \to \pi)$ **:**

      $\mathcal{E}_h \leftarrow \mathcal{E}_h \setminus \{l\}$

  return $\mathcal{E}_h$

end

Assessment experiment

3 datasets from Anchor [RSG18]

Assessment experiment

3 datasets from Anchor [RSG18]

2 additional datasets from **FairML** and **ProPublica** [Fai16, ALMK16] [FSV15, FFM+15, FSV+19]

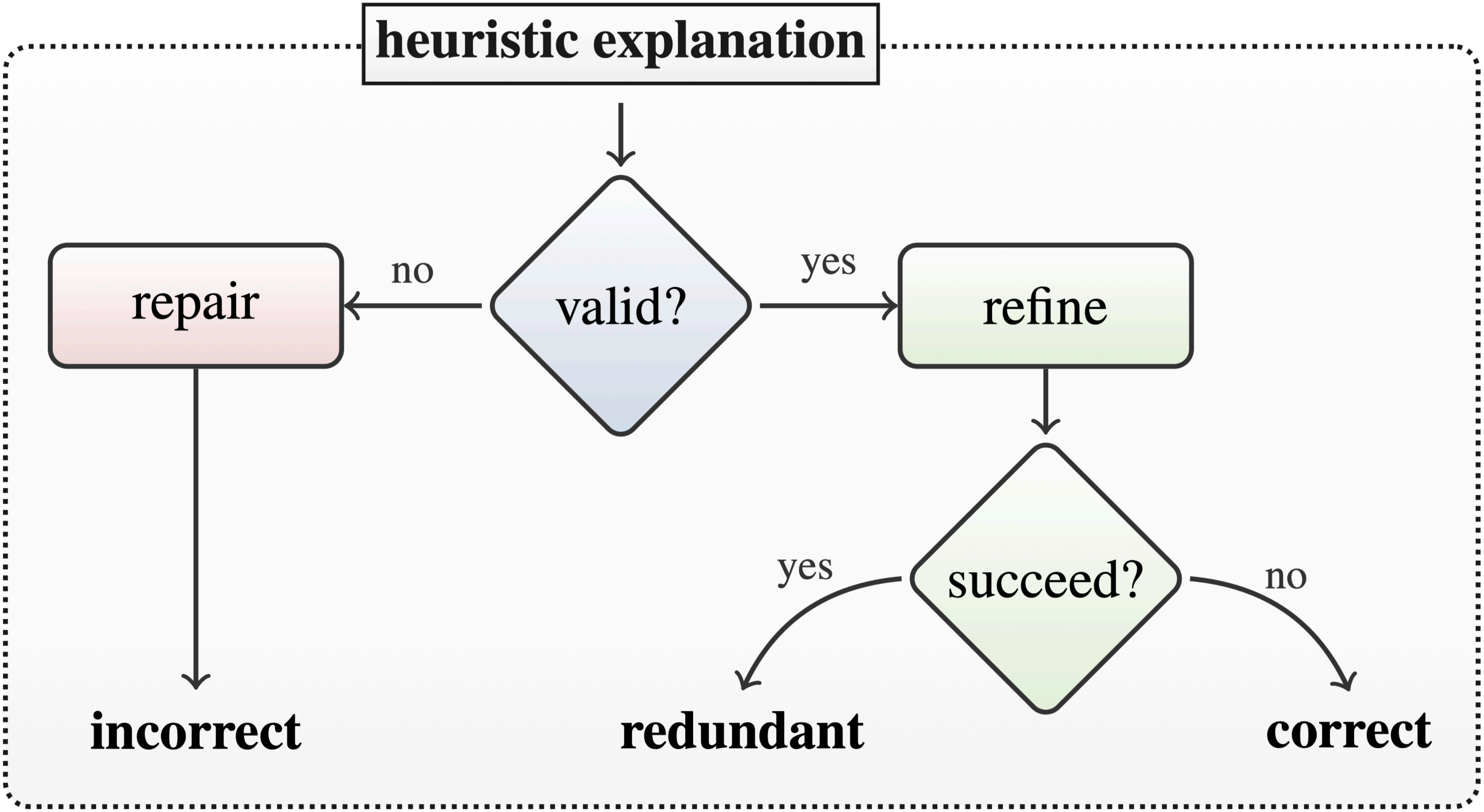
Assessment experiment

3 datasets from Anchor [RSG18]

2 additional datasets from **FairML** and **ProPublica** [Fai16, ALMK16]
[FSV15, FFM+15, FSV+19]

target **all data samples**

Assessment experiment



Assessment experiment

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

Assessment experiment

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

so should we trust heuristic approaches?

Assessment experiment

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

so should we trust heuristic approaches?

or better not?

let’s go further!

let's go further!

what about measuring precision **of Anchor's explanations?** [NSM+19]

What about measuring precision of Anchor's explanations? [NSM+19]

given model $\mathcal{M}$, input $\mathcal{I}$, prediction π, and explanation $\mathcal{E}$:

$$\mathit{prec}(\mathcal{E}) = \mathbb{E}_{\mathcal{D}(\mathcal{I}' \supset \mathcal{E})}[\mathcal{M}(\mathcal{I}') = \pi]$$

What about measuring precision of Anchor's explanations? [NSM+19]

given model $\mathcal{M}$, input $\mathcal{I}$, prediction π, and explanation $\mathcal{E}$:

$$\mathit{prec}(\mathcal{E}) = \mathbb{E}_{\mathcal{D}(\mathcal{I}' \supset \mathcal{E})}[\mathcal{M}(\mathcal{I}') = \pi]$$

alternatively, do approximate model counting for:

$$\mathcal{E} \wedge \mathcal{M} \wedge \neg \pi$$

What about measuring precision of Anchor's explanations? [NSM+19]

given model $\mathcal{M}$, input $\mathcal{I}$, prediction π, and explanation $\mathcal{E}$:

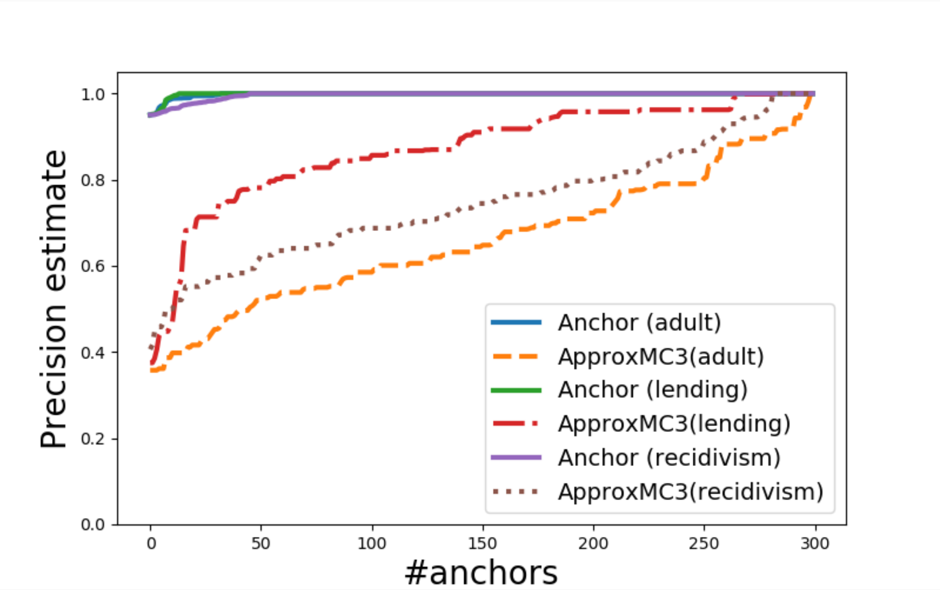
$$\mathit{prec}(\mathcal{E}) = \mathbb{E}_{\mathcal{D}(\mathcal{I}' \supset \mathcal{E})}[\mathcal{M}(\mathcal{I}') = \pi]$$

alternatively, do approximate model counting for:

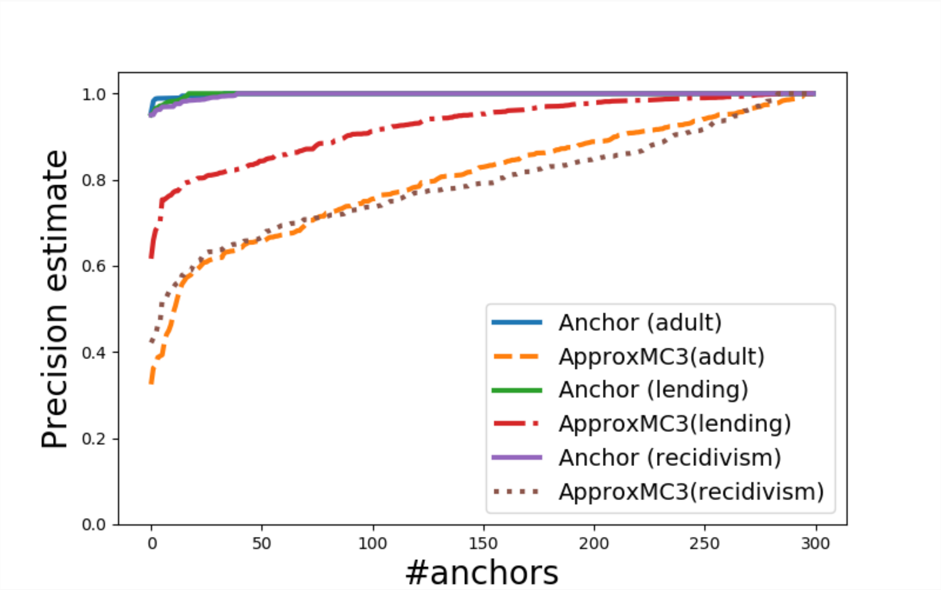
$$\mathcal{E} \wedge \mathcal{M} \wedge \neg\pi$$

(in fact, a bit more complicated but the idea is here)

Assessing heuristic explanations[1]



unconstrained feature space



samples with $\leq$ 50% difference

Dataset	Unconstrained inputs		Constrainted inputs	
	Anchor	**ApproxMC3**	**Anchor**	**ApproxMC3**
adult	0.99	0.67	0.99	0.81
lending	0.99	0.87	0.99	0.92
recidivism	0.99	0.75	0.99	0.80

Summary

Summary

logic is helpful in XAI!

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

trustable explanations

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

trustable explanations

minimality guarantees

Summary

logic is helpful in XAI!

(for computing explanations but also assessing heuristic appoaches)

rigorous approach

trustable explanations

minimality guarantees

(if one can encode and check entailment!)

challenges

Future

challenges

scalability

(**search** or **compilation**?)

Future

challenges

scalability

(**search** or **compilation**?)

other ML **models**, **reasoners**, **methods**?

challenges

scalability

(**search** or **compilation**?)

other ML **models**, **reasoners**, **methods**?

other types of explanations?

challenges

scalability

(**search** or **compilation**?)

other ML **models**, **reasoners**, **methods**?

other types of explanations?

what about **other heuristic approaches?**

Future

challenges

scalability

(**search** or **compilation**?)

other ML **models**, **reasoners**, **methods**?

other types of explanations?

what about **other heuristic approaches?**

hybrid approaches?

Further insights (see next)

generic oracle-based approach but...

Further insights (see next)

generic oracle-based approach but...

poly time algorithms for some ML models!

Further insights (see next)

generic oracle-based approach but...

poly time algorithms for some ML models!

$+$

'why?' **vs** ***'why not?'***

XAI vs verification

References i

[ALMK16] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner.
Machine bias.
`http://tiny.cc/dd7mjz`, 2016.

[BLM15] Alessio Bonfietti, Michele Lombardi, and Michela Milano.
Embedding decision trees and random forests in constraint programming.
In *CPAIOR*, pages 74–90, 2015.

[CD03] Hei Chan and Adnan Darwiche.
Reasoning about Bayesian network classifiers.
In *UAI*, pages 107–115, 2003.

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

[DH20] Adnan Darwiche and Auguste Hirth.
On the reasons behind decisions.
In *ECAI*, pages 712–720, 2020.

[Fai16] Auditing black-box predictive models.
`http://tiny.cc/6e7mjz`, 2016.

References ii

[FFM+15] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian.
Certifying and removing disparate impact.
In *KDD*, pages 259–268, 2015.

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

[FSV15] Sorelle Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian.
On algorithmic fairness, discrimination and disparate impact.
2015.

[FSV+19] Sorelle A. Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P. Hamilton, and Derek Roth.
A comparative study of fairness-enhancing interventions in machine learning.
In *FAT*, pages 329–338, 2019.

[IMM16] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva.
Propositional abduction with implicit hitting sets.
In *ECAI*, pages 1327–1335, 2016.

[INM19] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
On validating, repairing and refining heuristic ML explanations.
CoRR, abs/1907.02509, 2019.

References iii

[INMS19] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.
Abduction-based explanations for machine learning models.
In *AAAI*, pages 1511–1519, 2019.

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

[LMB17] Michele Lombardi, Michela Milano, and Andrea Bartolini.
Empirical decision model learning.
Artif. Intell., 244:343–367, 2017.

[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, 2018.

References iv

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

[SCD19] Andy Shih, Arthur Choi, and Adnan Darwiche.
Compiling Bayesian network classifiers into decision graphs.
In *AAAI*, pages 7966–7974, 2019.

[SDC19] Andy Shih, Adnan Darwiche, and Arthur Choi.
Verifying binarized neural networks by Angluin-style learning.
In *SAT*, pages 354–370, 2019.

[VZY17] Sicco Verwer, Yingqian Zhang, and Qing Chuan Ye.
Auction optimization using regression trees and linear models as integer programs.
Artif. Intell., 244:368–395, 2017.

Logic-Enabled Learning, Verification & Explanation of ML Models

A. Ignatiev, J. Marques-Silva, K. Meel & N. Narodytska

Monash Univ, ANITI/IRIT/CNRS, NU Singapore & VMWare Research

January 08, 2021 | IJCAI Tutorial T22

Part 5

Tractability, Duality, Fairness & Wrap-up

Outline

Tractability

Duality

Links with Fairness

Research Directions

Outline

Tractability

- Explaining Decision Trees
- Explaining NBCs & LCs

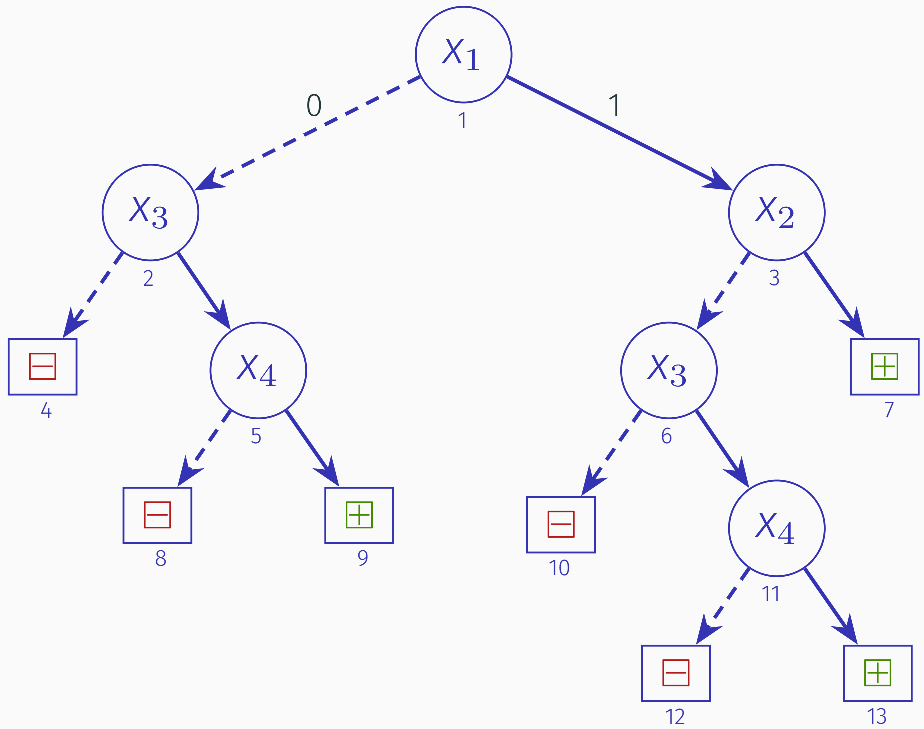
Duality

Links with Fairness

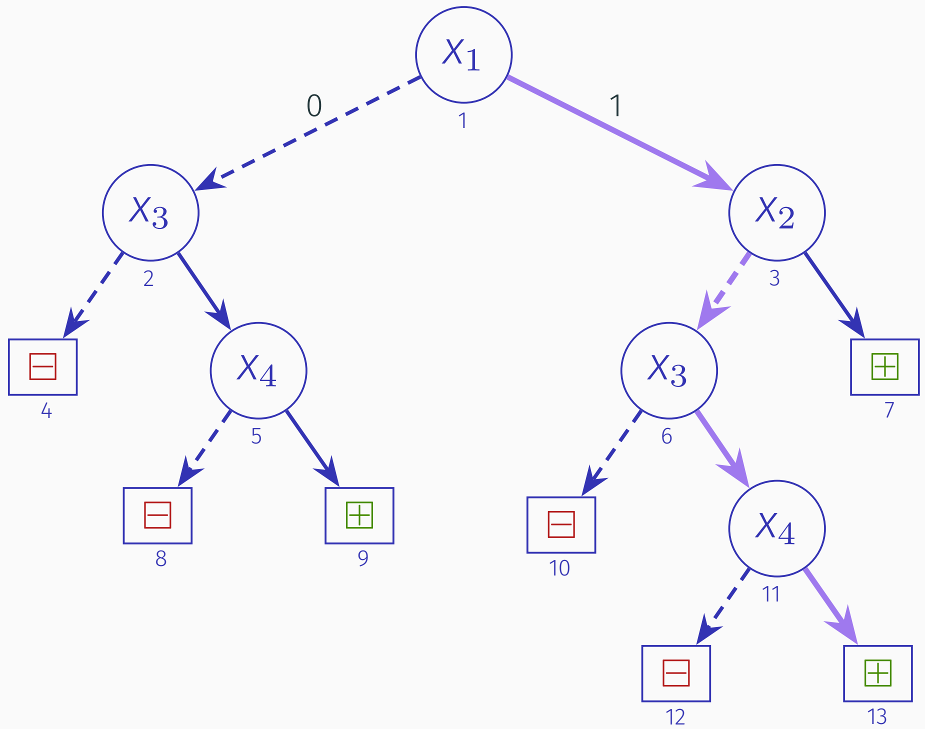
Research Directions

Why PI-explanations for DTs?

[IIM20]



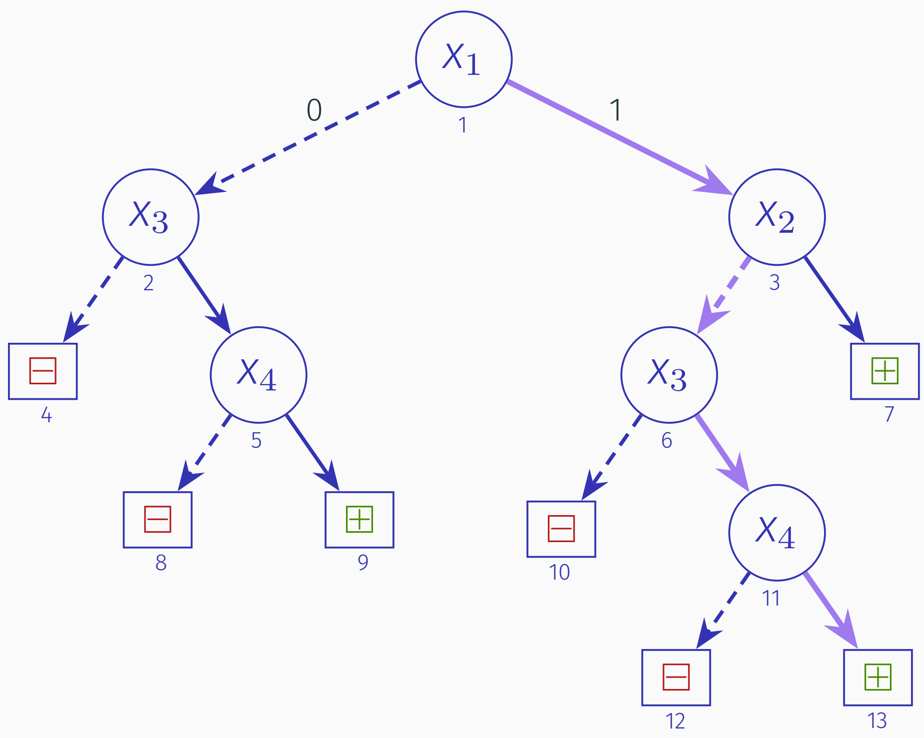
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$

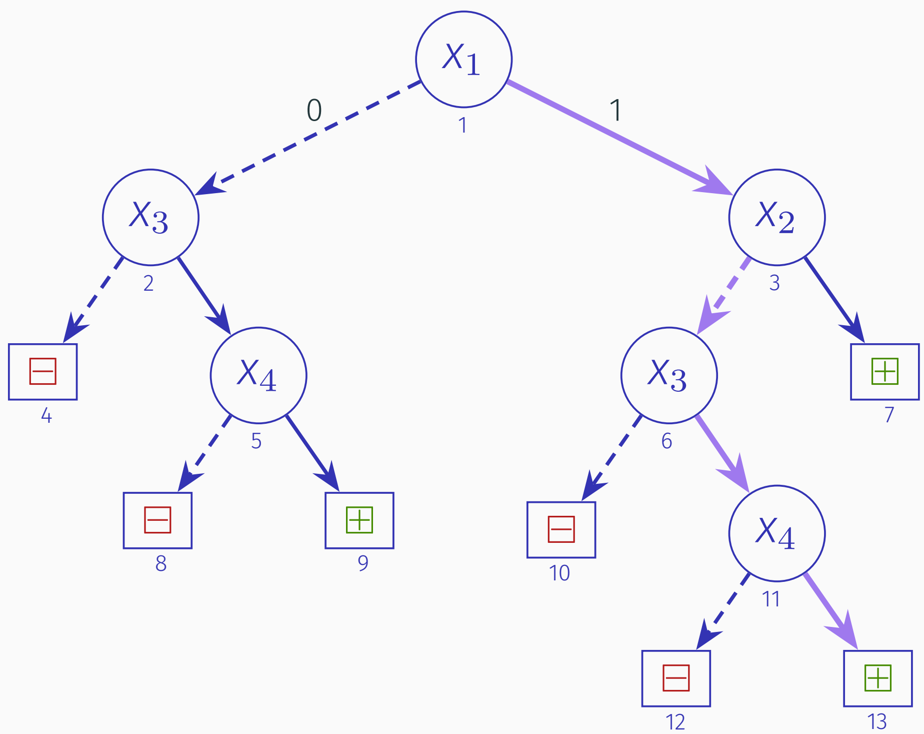
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction ⊞?
    - PI-explanation for prediction ⊞ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?

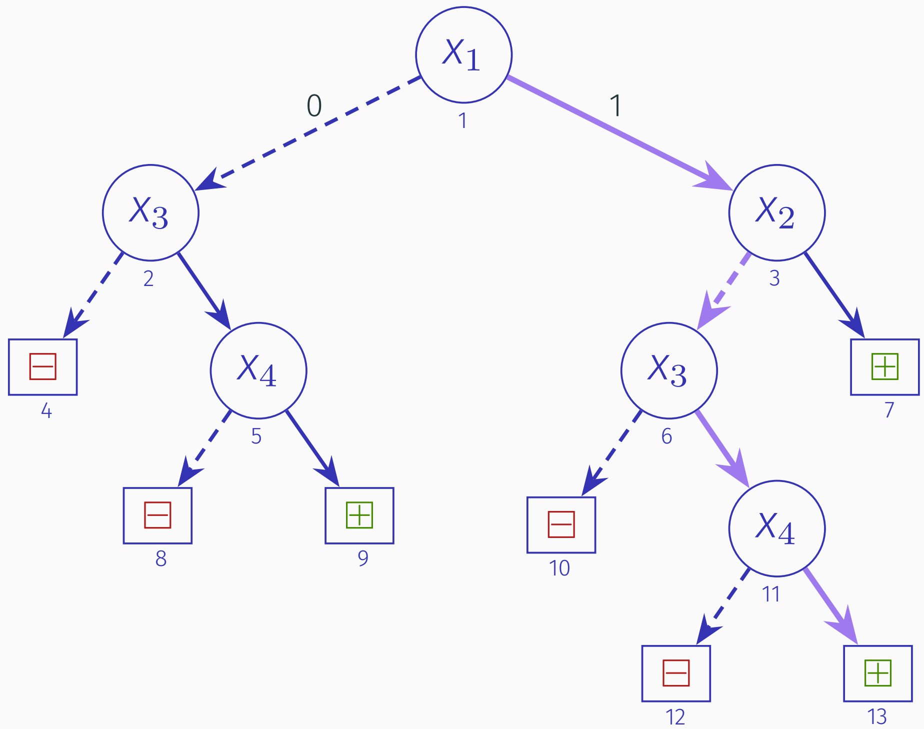
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction ⊞?
    - PI-explanation for prediction ⊞ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?
- Analysis:

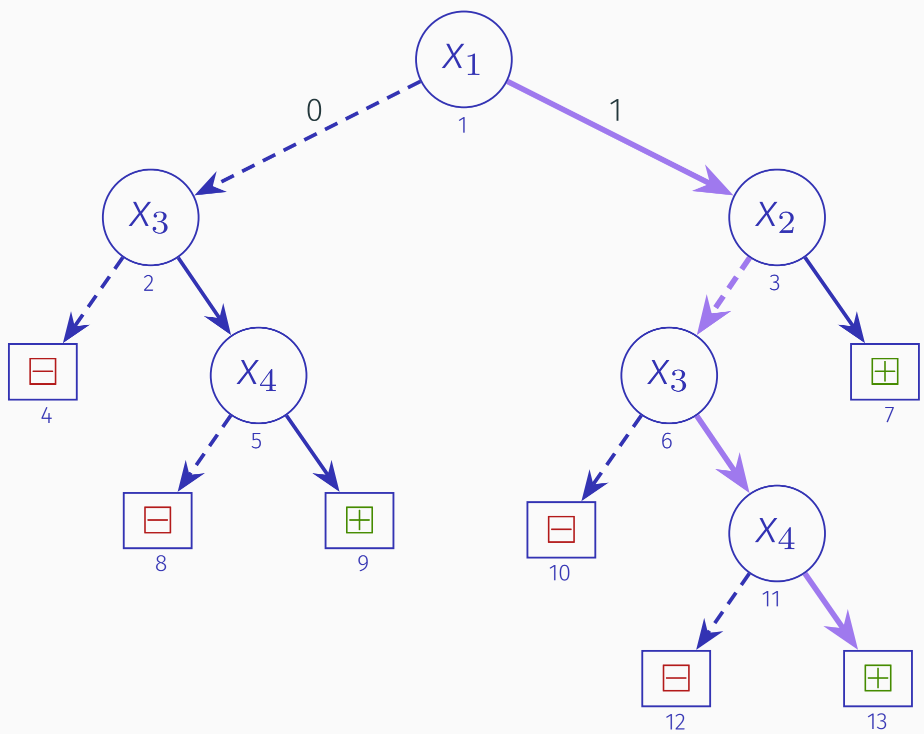
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction $\boxplus$?
  - PI-explanation for prediction $\boxplus$ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?
- Analysis:
  - Prediction changes if x_1 can take any value in $\{0, 1\}$?

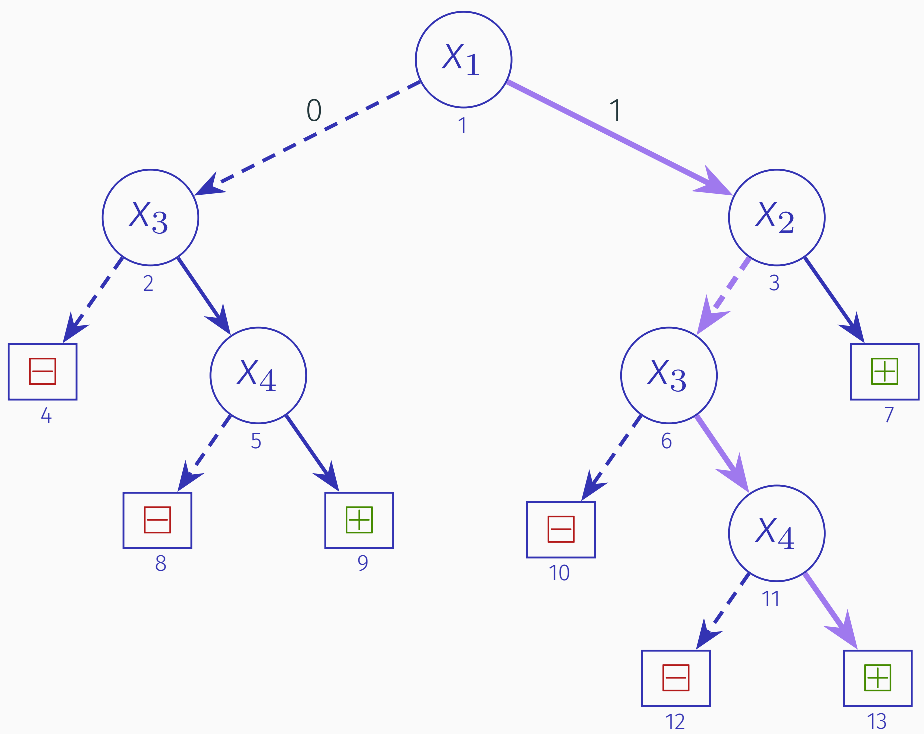
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction ⊞?
  - PI-explanation for prediction ⊞ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?
- Analysis:
  - Prediction changes if x_1 can take any value in $\{0, 1\}$? No

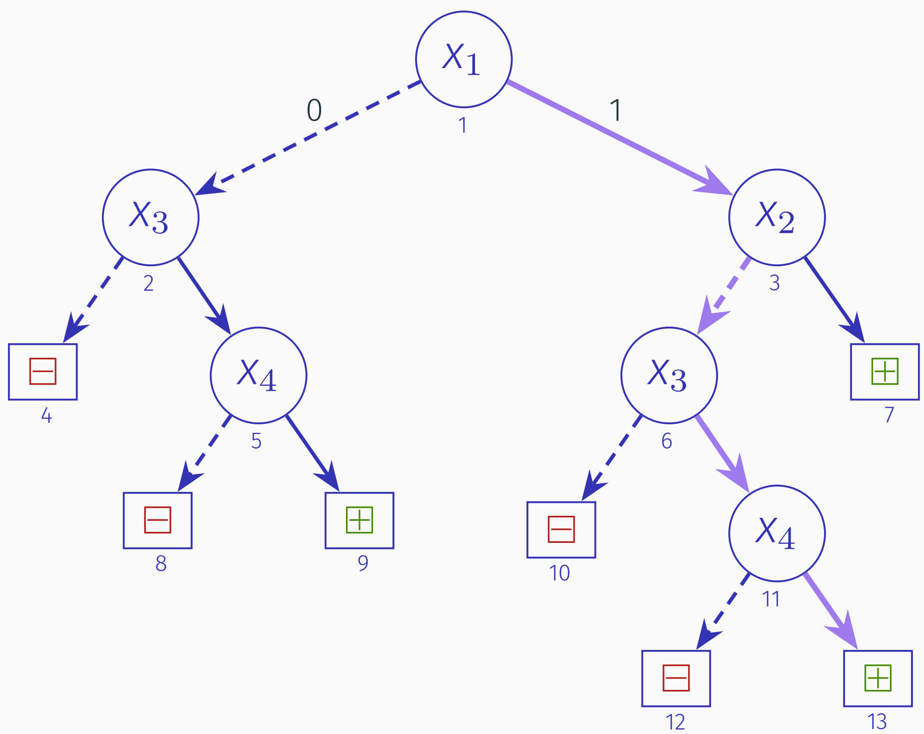
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction ⊞?
  - PI-explanation for prediction ⊞ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?
- Analysis:
  - Prediction changes if x_1 can take any value in $\{0, 1\}$? No
  - Prediction changes if x_2 and x_1 can take any value in $\{0, 1\}$?

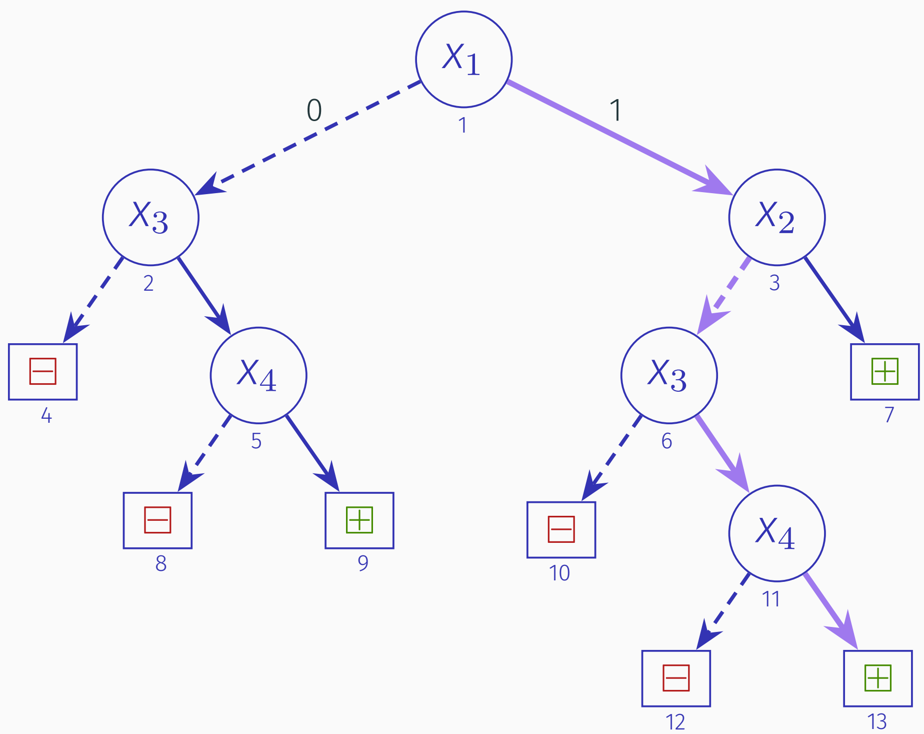
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction $\boxplus$?
  - PI-explanation for prediction $\boxplus$ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?
- Analysis:
  - Prediction changes if x_1 can take any value in $\{0, 1\}$? No
  - Prediction changes if x_2 and x_1 can take any value in $\{0, 1\}$? No

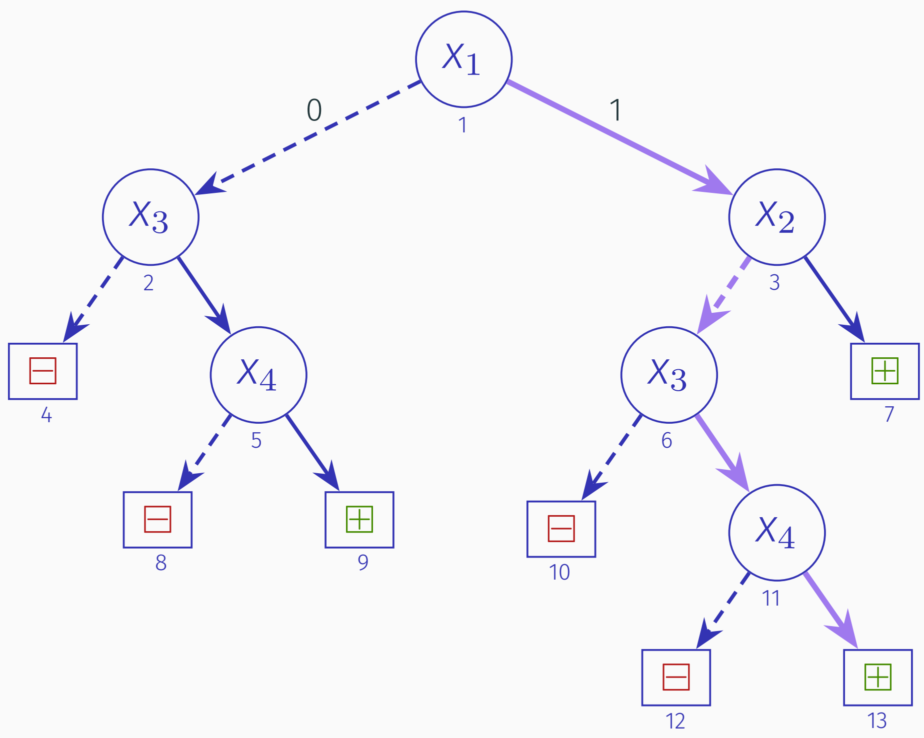
Why PI-explanations for DTs?



[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction ⊞?
  - PI-explanation for prediction ⊞ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?
- Analysis:
  - Prediction changes if x_1 can take any value in $\{0, 1\}$? No
  - Prediction changes if x_2 and x_1 can take any value in $\{0, 1\}$? No
  - PI-explanation: $(x_3 = 1) \wedge (x_4 = 1)$

Why PI-explanations for DTs?

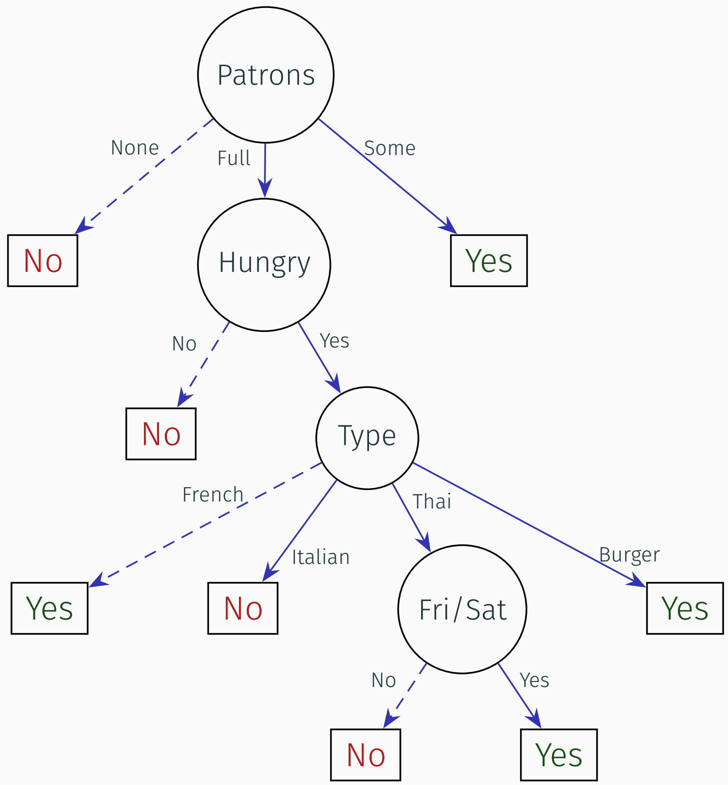


[IIM20]

- Instance: $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- Why is prediction $\boxplus$?
  - PI-explanation for prediction $\boxplus$ given instance $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$?
- Analysis:
  - Prediction changes if x_1 can take any value in $\{0, 1\}$? No
  - Prediction changes if x_2 and x_1 can take any value in $\{0, 1\}$? No
  - PI-explanation: $(x_3 = 1) \wedge (x_4 = 1)$
  - **Obs:** There are functions for which some paths grows with number of features, and PI-explanation is of constant-size

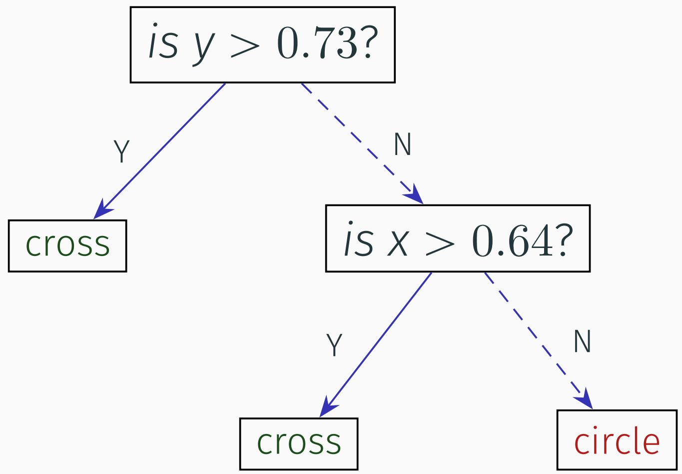
Need for PI-explanations in DTs is ubiquitous– Russell&Norving's book

[RN10]



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

Need for PI-explanations in DTs is ubiquitous– Zhou's book



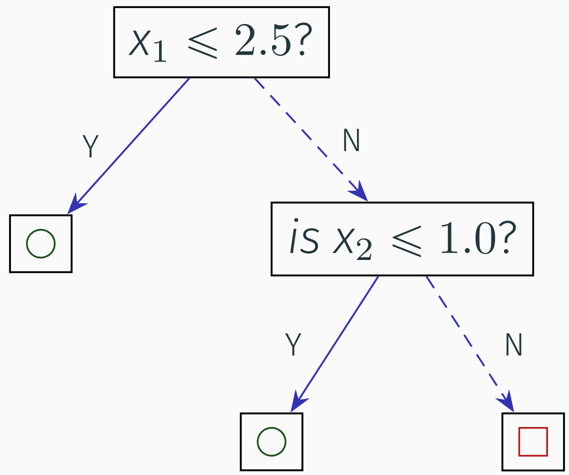
[Zho12]

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

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

Need for PI-explanations in DTs is ubiquitous– Alpaydin's book

[Alp14]

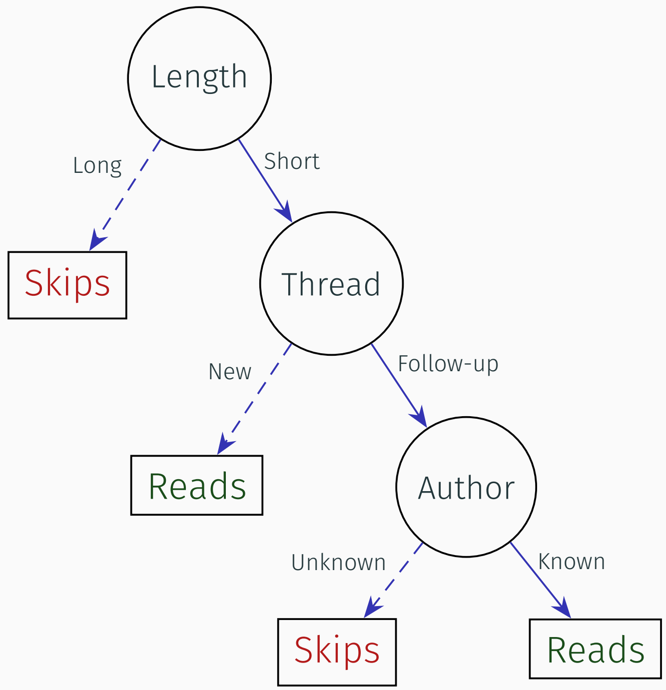


- PI-explanation for $(x_1, x_2) = (3.14, 0.87)$?

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

Need for PI-explanations in DTs is ubiquitous– Poole&Mackworth's book

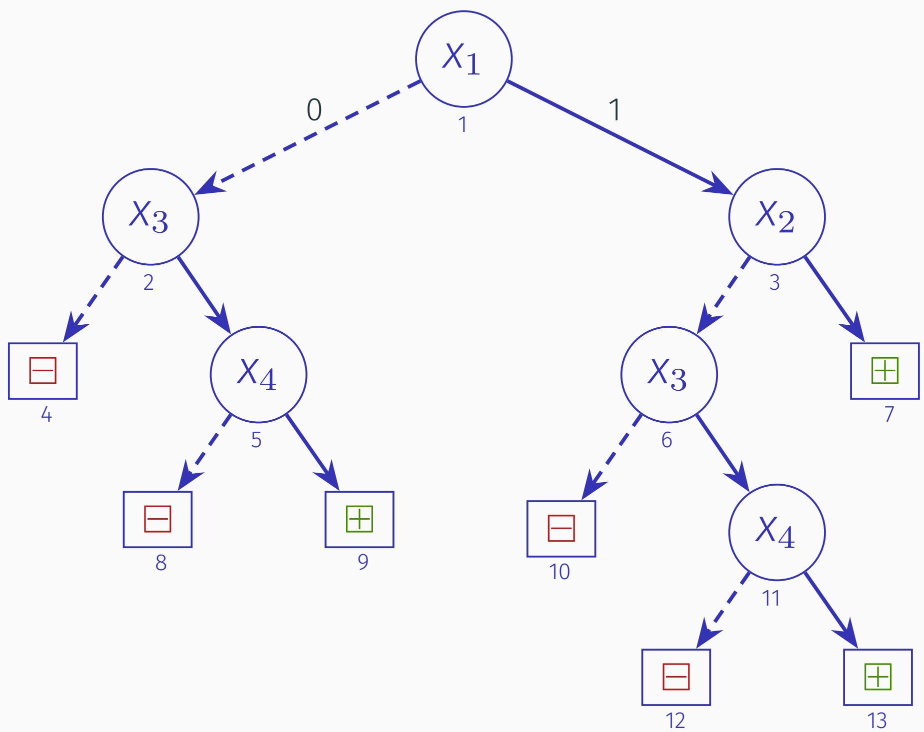
[PM17]



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

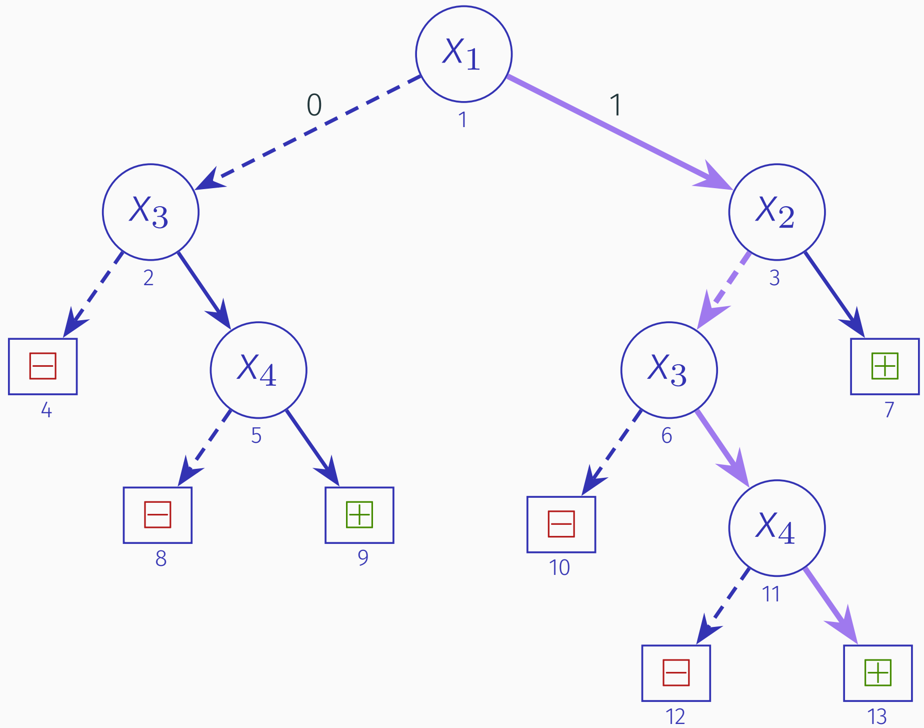
DT explanations

[IIM20]



DT explanations

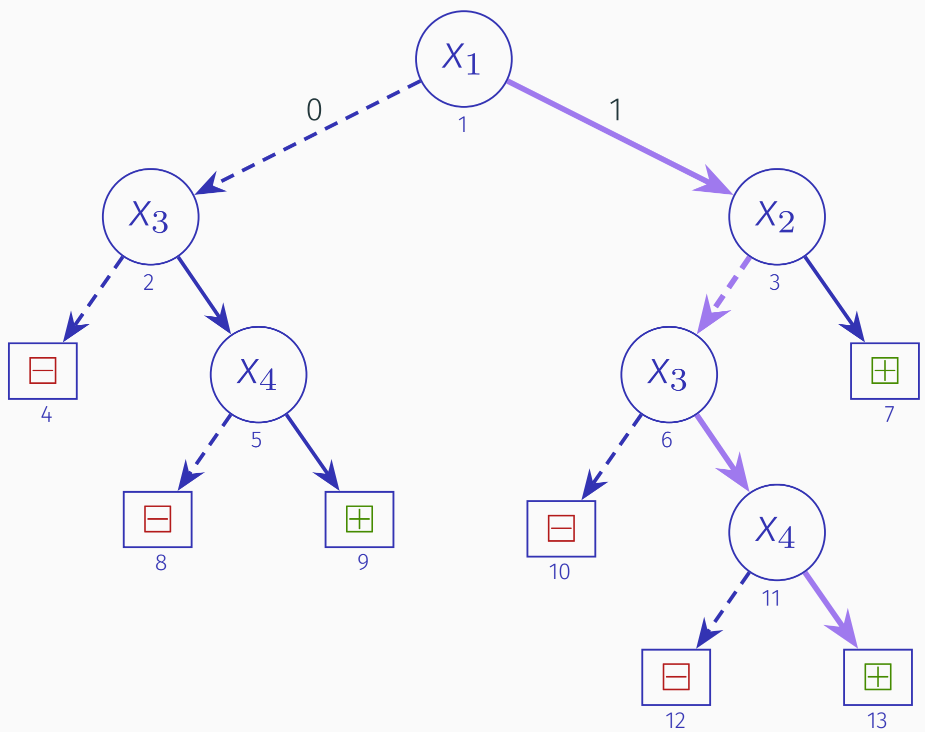
[IIM20]



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

DT explanations

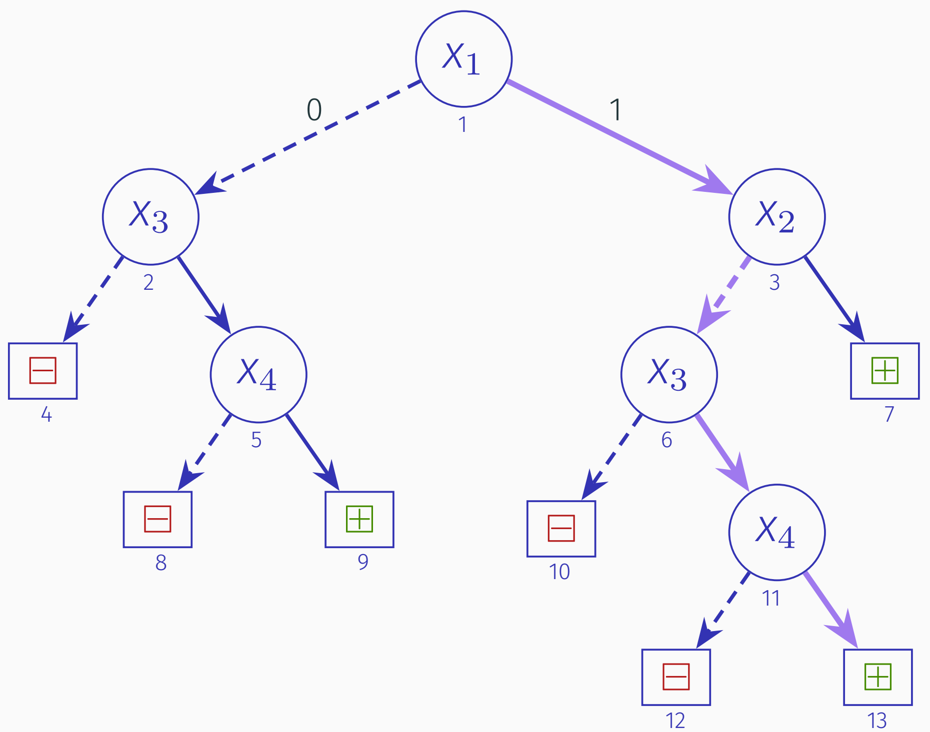
[IIM20]



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

DT explanations in polynomial time

[IIM20]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- **For prediction ⊞, it suffices to ensure all ⊟ paths remain inconsistent**
  - I.e. find a subset-minimal hitting set of all ⊟ paths; these are the features to keep
  - Well-known to be solvable in polynomial time [EG95]

Experimental evidence

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

Outline

Tractability

Explaining Decision Trees

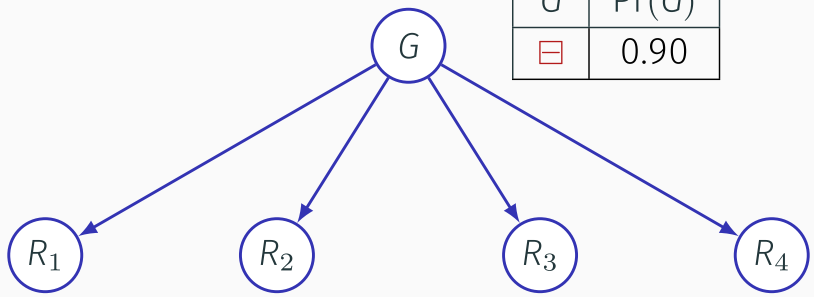
Explaining NBCs & LCs

Duality

Links with Fairness

Research Directions

Key concepts & outcomes – NBCs & lPr



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1 \vert G)$
$\boxplus$	0.95
$\boxminus$	0.03

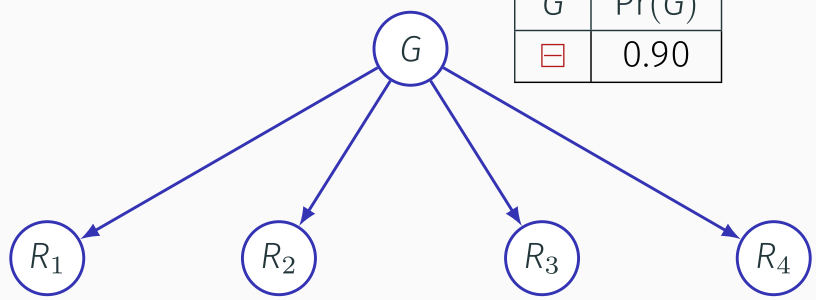
G	$\Pr(R_2 \vert G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \vert G)$
$\boxplus$	0.02
$\boxminus$	0.34

G	$\Pr(R_4 \vert G)$
$\boxplus$	0.20
$\boxminus$	0.75

NBC classifier (def): $\tau(\mathbf{e}) = \text{argmax}_{c \in \mathcal{K}}(\Pr(c|\mathbf{e}))$

Key concepts & outcomes – NBCs & lPr



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1 \mid G)$
$\boxplus$	0.95
$\boxminus$	0.03

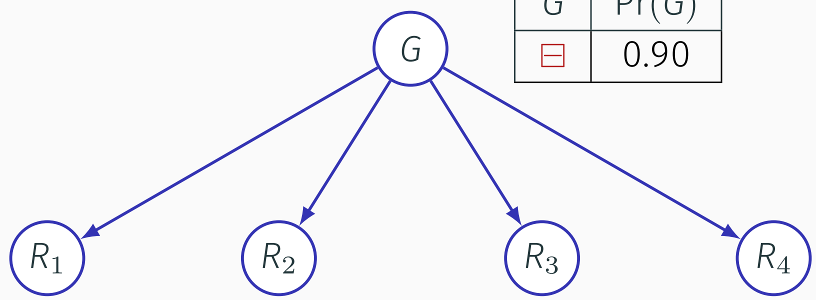
G	$\Pr(R_2 \mid G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \mid G)$
$\boxplus$	0.02
$\boxminus$	0.34

G	$\Pr(R_4 \mid G)$
$\boxplus$	0.20
$\boxminus$	0.75

NBC classifier (def): $\tau(\mathbf{e}) = \text{argmax}_{c \in \mathcal{K}}(\Pr(c \mid \mathbf{e})) = \text{argmax}_{c \in \mathcal{K}}\left(\Pr(c) \times \prod_i \Pr(e_i \mid c)\right)$

Key concepts & outcomes – NBCs & lPr



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1 \mid G)$
$\boxplus$	0.95
$\boxminus$	0.03

G	$\Pr(R_2 \mid G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \mid G)$
$\boxplus$	0.02
$\boxminus$	0.34

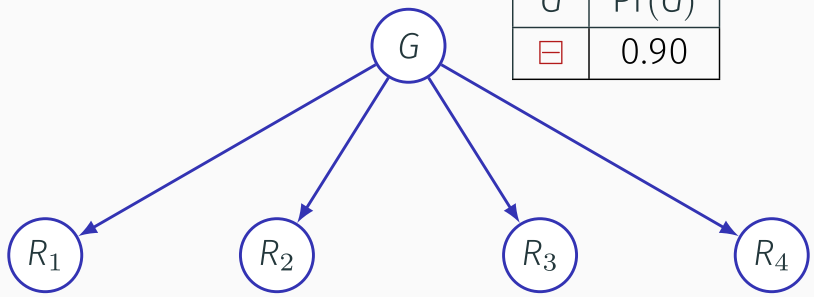
G	$\Pr(R_4 \mid G)$
$\boxplus$	0.20
$\boxminus$	0.75

NBC classifier (def): $\tau(\mathbf{e}) = \mathrm{argmax}_{c \in \mathcal{K}}(\Pr(c \mid \mathbf{e})) = \mathrm{argmax}_{c \in \mathcal{K}}\left(\Pr(c) \times \prod_i \Pr(e_i \mid c)\right)$

NBC classifier (alt): $\tau(\mathbf{e}) = \mathrm{argmax}_{c \in \mathcal{K}}\left((\mathbb{T} + \log \Pr(c)) + \sum_i (\mathbb{T} + \log \Pr(e_i \mid c))\right)$

Key concepts & outcomes – NBCs & lPr

G	$\Pr(G)$
$\boxminus$	0.90



G	$\Pr(R_1 \mid G)$
$\boxplus$	0.95
$\boxminus$	0.03

G	$\Pr(R_2 \mid G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \mid G)$
$\boxplus$	0.02
$\boxminus$	0.34

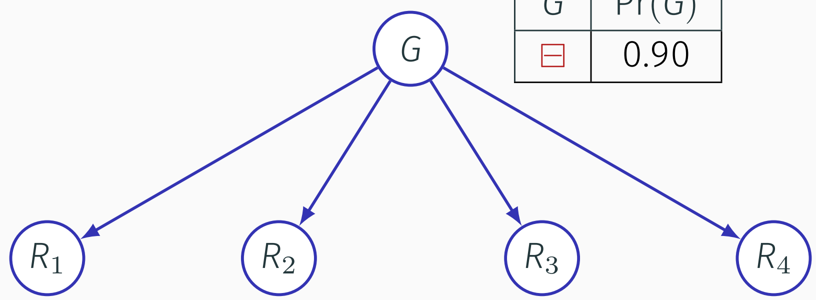
G	$\Pr(R_4 \mid G)$
$\boxplus$	0.20
$\boxminus$	0.75

NBC classifier (def): $\tau(\mathbf{e}) = \operatorname{argmax}_{c \in \mathcal{K}}(\Pr(c \mid \mathbf{e})) = \operatorname{argmax}_{c \in \mathcal{K}} \left(\Pr(c) \times \prod_i \Pr(e_i \mid c)\right)$

NBC classifier (alt): $\tau(\mathbf{e}) = \operatorname{argmax}_{c \in \mathcal{K}} \left((\mathbb{T} + \log \Pr(c)) + \sum_i (\mathbb{T} + \log \Pr(e_i \mid c))\right)$

Using oper. $\mathrm{lPr}(\cdot)$: $\tau(\mathbf{e}) = \operatorname{argmax}_{c \in \mathcal{K}}(\mathrm{lPr}(c \mid \mathbf{e})) = \operatorname{argmax}_{c \in \mathcal{K}} \left((\mathrm{lPr}(c)) + \sum_i (\mathrm{lPr}(e_i \mid c))\right)$

Key concepts & outcomes – working with lPr



G	Pr(G)
$\boxminus$	0.90

G	Pr($R_1\|G$)
$\boxplus$	0.95
$\boxminus$	0.03

G	Pr($R_2\|G$)
$\boxplus$	0.05
$\boxminus$	0.95

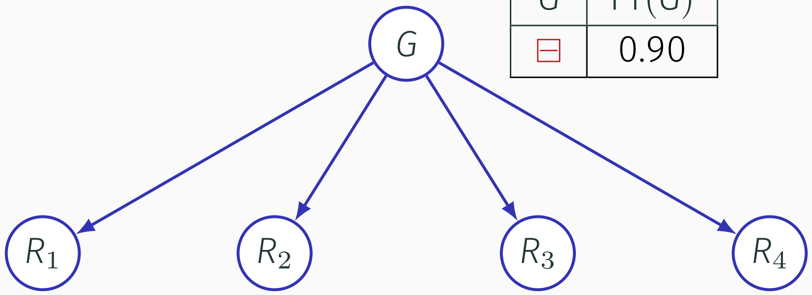
G	Pr($R_3\|G$)
$\boxplus$	0.02
$\boxminus$	0.34

G	Pr($R_4\|G$)
$\boxplus$	0.20
$\boxminus$	0.75

$\mathbf{a} = (1, 0, 1, 0)$	Pr($\boxplus$)	Pr($r_1\| \boxplus$)	Pr($\neg r_2\| \boxplus$)	Pr($r_3\| \boxplus$)	Pr($\neg r_4\| \boxplus$)	lPr($\boxplus$ $\|\mathbf{a}$)
Pr($\cdot$)	0.10	0.95	0.95	0.02	0.80	
lPr($\cdot$)	1.70	3.95	3.95	0.09	3.78	13.47

$\mathbf{a} = (1, 0, 1, 0)$	Pr($\boxminus$)	Pr($r_1\| \boxminus$)	Pr($\neg r_2\| \boxminus$)	Pr($r_3\| \boxminus$)	Pr($\neg r_4\| \boxminus$)	lPr($\boxminus$ $\|\mathbf{a}$)
Pr($\cdot$)	0.90	0.03	0.05	0.34	0.25	
lPr($\cdot$)	3.89	0.49	1.00	2.92	2.61	10.91

Key concepts & outcomes – working with lPr



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1 \mid G)$
$\boxplus$	0.95
$\boxminus$	0.03

G	$\Pr(R_2 \mid G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \mid G)$
$\boxplus$	0.02
$\boxminus$	0.34

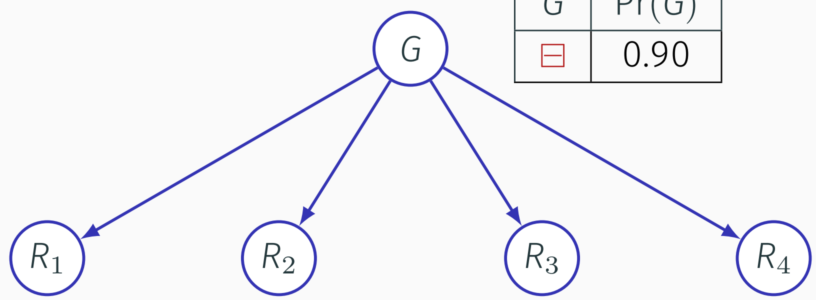
G	$\Pr(R_4 \mid G)$
$\boxplus$	0.20
$\boxminus$	0.75

Pick class $\boxplus$!

$\mathbf{a} = (1, 0, 1, 0)$	$\Pr(\boxplus)$	$\Pr(r_1 \mid \boxplus)$	$\Pr(\neg r_2 \mid \boxplus)$	$\Pr(r_3 \mid \boxplus)$	$\Pr(\neg r_4 \mid \boxplus)$	$\mathrm{lPr}(\boxplus \mid \mathbf{a})$
$\Pr(\cdot)$	0.10	0.95	0.95	0.02	0.80	
$\mathrm{lPr}(\cdot)$	1.70	3.95	3.95	0.09	3.78	13.47

$\mathbf{a} = (1, 0, 1, 0)$	$\Pr(\boxminus)$	$\Pr(r_1 \mid \boxminus)$	$\Pr(\neg r_2 \mid \boxminus)$	$\Pr(r_3 \mid \boxminus)$	$\Pr(\neg r_4 \mid \boxminus)$	$\mathrm{lPr}(\boxminus \mid \mathbf{a})$
$\Pr(\cdot)$	0.90	0.03	0.05	0.34	0.25	
$\mathrm{lPr}(\cdot)$	3.89	0.49	1.00	2.92	2.61	10.91

Key concepts & outcomes – XLCs



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1 \mid G)$
$\boxplus$	0.95
$\boxminus$	0.03

G	$\Pr(R_2 \mid G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \mid G)$
$\boxplus$	0.02
$\boxminus$	0.34

G	$\Pr(R_4 \mid G)$
$\boxplus$	0.20
$\boxminus$	0.75

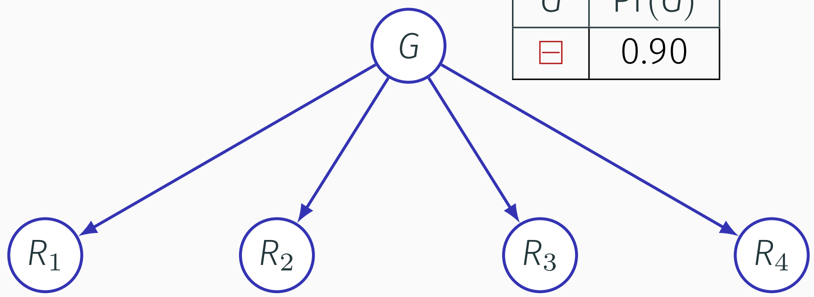
NBC classifier (def): $\tau(\mathbf{e}) = \mathrm{argmax}_{c \in \mathcal{K}} \left(\Pr(c) \times \prod_i \Pr(e_i|c)\right)$

NBC classifier (alt): $\tau(\mathbf{e}) = \mathrm{argmax}_{c \in \mathcal{K}} \left((\mathbb{T} + \log \Pr(c)) + \sum_i (\mathbb{T} + \log \Pr(e_i|c))\right)$

Using oper. $\mathrm{l}\Pr(\cdot)$: $\tau(\mathbf{e}) = \mathrm{argmax}_{c \in \mathcal{K}} \left((\mathrm{l}\Pr(c)) + \sum_i (\mathrm{l}\Pr(e_i|c))\right)$

XLC classifier: $\nu(\mathbf{e}) \triangleq w_0 + \sum_{i \in \mathcal{R}} w_i e_i + \sum_{j \in \mathcal{C}} \sigma(e_j, v_j^1, v_j^2, \ldots, v_j^{d_j})$

Key concepts & outcomes – XLCs



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1\|G)$
$\boxplus$	0.95
$\boxminus$	0.03

G	$\Pr(R_2\|G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3\|G)$
$\boxplus$	0.02
$\boxminus$	0.34

G	$\Pr(R_4\|G)$
$\boxplus$	0.20
$\boxminus$	0.75

Can reduce NBC to XLC

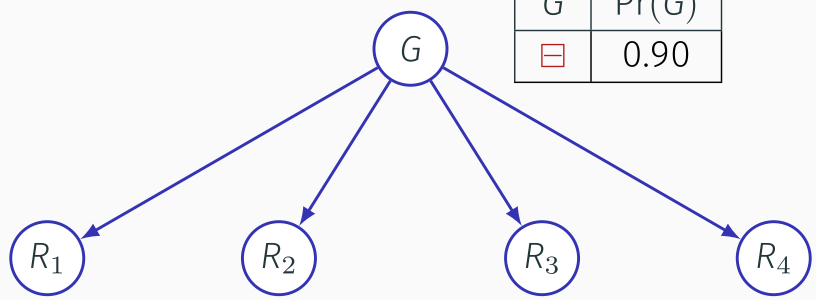
NBC classifier (def): $\tau(\mathbf{e}) = \mathrm{argmax}_{c\in\mathcal{K}}\left(\Pr(c) \times \prod_i \Pr(e_i|c)\right)$

NBC classifier (alt): $\tau(\mathbf{e}) = \mathrm{argmax}_{c\in\mathcal{K}}\left((\mathbb{T} + \log\Pr(c)) + \sum_i(\mathbb{T} + \log\Pr(e_i|c))\right)$

Using oper. $\mathrm{lPr}(\cdot)$: $\tau(\mathbf{e}) = \mathrm{argmax}_{c\in\mathcal{K}}\left((\mathrm{lPr}(c)) + \sum_i(\mathrm{lPr}(e_i|c))\right)$

XLC classifier: $\nu(\mathbf{e}) \triangleq w_0 + \sum_{i\in\mathcal{R}} w_i e_i + \sum_{j\in\mathcal{C}} \sigma(e_j, v_j^1, v_j^2, \ldots, v_j^{d_j})$

Key concepts & outcomes – NBC to XLC



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1 \mid G)$
$\boxplus$	0.95
$\boxminus$	0.03

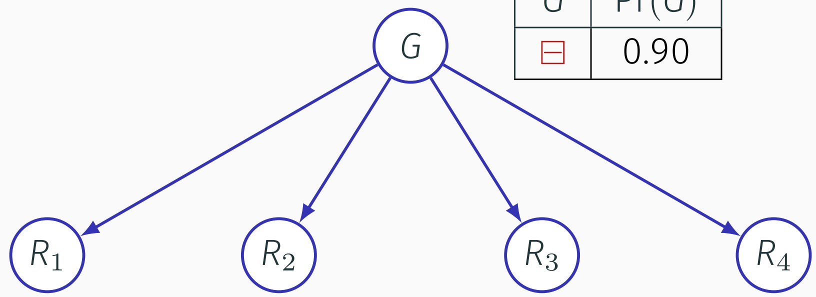
G	$\Pr(R_2 \mid G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \mid G)$
$\boxplus$	0.02
$\boxminus$	0.34

G	$\Pr(R_4 \mid G)$
$\boxplus$	0.20
$\boxminus$	0.75

Eliminate argmax:
$$\mathrm{lPr}(\boxplus) - \mathrm{lPr}(\boxminus) + \sum_{i=1}^{n} (\mathrm{lPr}(\neg e_i \mid \boxplus) - \mathrm{lPr}(\neg e_i \mid \boxminus)) \neg e_i + \sum_{i=1}^{n} (\mathrm{lPr}(e_i \mid \boxplus) - \mathrm{lPr}(e_i \mid \boxminus)) e_i > \mathbf{0}$$

Key concepts & outcomes – NBC to XLC



G	$\Pr(G)$
$\boxminus$	0.90

G	$\Pr(R_1 \mid G)$
$\boxplus$	0.95
$\boxminus$	0.03

G	$\Pr(R_2 \mid G)$
$\boxplus$	0.05
$\boxminus$	0.95

G	$\Pr(R_3 \mid G)$
$\boxplus$	0.02
$\boxminus$	0.34

G	$\Pr(R_4 \mid G)$
$\boxplus$	0.20
$\boxminus$	0.75

Eliminate argmax:

$$\mathrm{l}\Pr(\boxplus) - \mathrm{l}\Pr(\boxminus) + \sum_{i=1}^{n}(\mathrm{l}\Pr(\neg e_i \mid \boxplus) - \mathrm{l}\Pr(\neg e_i \mid \boxminus))\neg e_i + \sum_{i=1}^{n}(\mathrm{l}\Pr(e_i \mid \boxplus) - \mathrm{l}\Pr(e_i \mid \boxminus))e_i > \mathbf{0}$$

Mapping to XLC:

$$w_0 \triangleq \mathrm{l}\Pr(\boxplus) - \mathrm{l}\Pr(\boxminus)$$

$$v_j^1 \triangleq \mathrm{l}\Pr(\neg e_j \mid \boxplus) - \mathrm{l}\Pr(\neg e_j \mid \boxminus)$$

$$v_j^2 \triangleq \mathrm{l}\Pr(e_j \mid \boxplus) - \mathrm{l}\Pr(e_j \mid \boxminus)$$

Key concepts & outcomes – minding the gap

	$\Pr(\boxplus)$	$\Pr(\neg r_1 \mid \boxplus)$	$\Pr(r_1 \mid \boxplus)$	$\Pr(\neg r_2 \mid \boxplus)$	$\Pr(r_2 \mid \boxplus)$	$\Pr(\neg r_3 \mid \boxplus)$	$\Pr(r_3 \mid \boxplus)$	$\Pr(\neg r_4 \mid \boxplus)$	$\Pr(r_4 \mid \boxplus)$
$\Pr(\cdot)$	0.10	0.05	0.95	0.95	0.05	0.98	0.02	0.80	0.20
$\mathrm{lPr}(\cdot)$	1.70	1.00	3.95	3.95	1.00	3.98	0.09	3.78	2.39

	$\Pr(\boxminus)$	$\Pr(\neg r_1 \mid \boxminus)$	$\Pr(r_1 \mid \boxminus)$	$\Pr(\neg r_2 \mid \boxminus)$	$\Pr(r_2 \mid \boxminus)$	$\Pr(\neg r_3 \mid \boxminus)$	$\Pr(r_3 \mid \boxminus)$	$\Pr(\neg r_4 \mid \boxminus)$	$\Pr(r_4 \mid \boxminus)$
$\Pr(\cdot)$	0.90	0.97	0.03	0.05	0.95	0.66	0.34	0.25	0.75
$\mathrm{lPr}(\cdot)$	3.89	3.97	0.49	1.00	3.95	3.58	2.92	2.61	3.71

Gap value: $\Gamma^a \triangleq \nu(\mathbf{a}) = w_0 + \sum_{j\in\mathcal{C}} \sigma(a_j, v_j^1, v_j^2, \ldots, v_i^{d_j}) \; > \mathbf{0}$

Worst-case gap: $\Gamma^\omega \triangleq w_0 + \sum_{j\in\mathcal{C}} v_j^\omega \; < \mathbf{0}$

Relate Γ^a and Γ^ω: $\Gamma^\omega = w_0 + \sum_{j\in\mathcal{C}} v_j^{a_j} - \sum_{j\in\mathcal{C}} (v_j^{a_j} - v_j^\omega) = \Gamma^a - \sum_{j\in\mathcal{C}} \delta_j = -\Phi$

where, $\delta_j \triangleq v_j^{a_j} - v_j^\omega = v_j^{a_j} - \min\{v_j^1, v_j^2, \ldots\}$

Worst-case, given some min. $\mathcal{P}$: $w_0 + \sum_{j\in\mathcal{P}} v_j^{a_j} + \sum_{j\notin\mathcal{P}} v_j^\omega = -\Phi + \sum_{j\in\mathcal{P}} \delta_j > 0$

Key concepts & outcomes – 0-1 ILP

	$\Pr(\boxplus)$	$\Pr(\neg r_1 \mid \boxplus)$	$\Pr(r_1 \mid \boxplus)$	$\Pr(\neg r_2 \mid \boxplus)$	$\Pr(r_2 \mid \boxplus)$	$\Pr(\neg r_3 \mid \boxplus)$	$\Pr(r_3 \mid \boxplus)$	$\Pr(\neg r_4 \mid \boxplus)$	$\Pr(r_4 \mid \boxplus)$
$\Pr(\cdot)$	0.10	0.05	0.95	0.95	0.05	0.98	0.02	0.80	0.20
$\mathrm{lPr}(\cdot)$	1.70	1.00	3.95	3.95	1.00	3.98	0.09	3.78	2.39

	$\Pr(\boxminus)$	$\Pr(\neg r_1 \mid \boxminus)$	$\Pr(r_1 \mid \boxminus)$	$\Pr(\neg r_2 \mid \boxminus)$	$\Pr(r_2 \mid \boxminus)$	$\Pr(\neg r_3 \mid \boxminus)$	$\Pr(r_3 \mid \boxminus)$	$\Pr(\neg r_4 \mid \boxminus)$	$\Pr(r_4 \mid \boxminus)$
$\Pr(\cdot)$	0.90	0.97	0.03	0.05	0.95	0.66	0.34	0.25	0.75
$\mathrm{lPr}(\cdot)$	3.89	3.97	0.49	1.00	3.95	3.58	2.92	2.61	3.71

Optimization problem:

$$\begin{array}{ll} \min & \sum_{i=1}^{n} p_i \\ \text{s.t.} & \sum_{i=1}^{n} \delta_i p_i > \Phi \\ & p_i \in \{0,1\} \end{array}$$

Key concepts & outcomes – 0-1 ILP

	$\Pr(\boxplus)$	$\Pr(\neg r_1 \mid \boxplus)$	$\Pr(r_1 \mid \boxplus)$	$\Pr(\neg r_2 \mid \boxplus)$	$\Pr(r_2 \mid \boxplus)$	$\Pr(\neg r_3 \mid \boxplus)$	$\Pr(r_3 \mid \boxplus)$	$\Pr(\neg r_4 \mid \boxplus)$	$\Pr(r_4 \mid \boxplus)$
$\Pr(\cdot)$	0.10	0.05	0.95	0.95	0.05	0.98	0.02	0.80	0.20
$\mathrm{lPr}(\cdot)$	1.70	1.00	3.95	3.95	1.00	3.98	0.09	3.78	2.39

	$\Pr(\boxminus)$	$\Pr(\neg r_1 \mid \boxminus)$	$\Pr(r_1 \mid \boxminus)$	$\Pr(\neg r_2 \mid \boxminus)$	$\Pr(r_2 \mid \boxminus)$	$\Pr(\neg r_3 \mid \boxminus)$	$\Pr(r_3 \mid \boxminus)$	$\Pr(\neg r_4 \mid \boxminus)$	$\Pr(r_4 \mid \boxminus)$
$\Pr(\cdot)$	0.90	0.97	0.03	0.05	0.95	0.66	0.34	0.25	0.75
$\mathrm{lPr}(\cdot)$	3.89	3.97	0.49	1.00	3.95	3.58	2.92	2.61	3.71

Optimization problem:

$$\begin{array}{ll} \min & \sum_{i=1}^{n} p_i \\ \text{s.t.} & \sum_{i=1}^{n} \delta_i p_i > \Phi \\ & p_i \in \{0,1\} \end{array}$$

Special case of knapsack;
can solve in log-linear time

Key concepts & outcomes – 0-1 ILP

	$\Pr(\boxplus)$	$\Pr(\neg r_1 \mid \boxplus)$	$\Pr(r_1 \mid \boxplus)$	$\Pr(\neg r_2 \mid \boxplus)$	$\Pr(r_2 \mid \boxplus)$	$\Pr(\neg r_3 \mid \boxplus)$	$\Pr(r_3 \mid \boxplus)$	$\Pr(\neg r_4 \mid \boxplus)$	$\Pr(r_4 \mid \boxplus)$
$\Pr(\cdot)$	0.10	0.05	0.95	0.95	0.05	0.98	0.02	0.80	0.20
$\mathrm{lPr}(\cdot)$	1.70	1.00	3.95	3.95	1.00	3.98	0.09	3.78	2.39

	$\Pr(\boxminus)$	$\Pr(\neg r_1 \mid \boxminus)$	$\Pr(r_1 \mid \boxminus)$	$\Pr(\neg r_2 \mid \boxminus)$	$\Pr(r_2 \mid \boxminus)$	$\Pr(\neg r_3 \mid \boxminus)$	$\Pr(r_3 \mid \boxminus)$	$\Pr(\neg r_4 \mid \boxminus)$	$\Pr(r_4 \mid \boxminus)$
$\Pr(\cdot)$	0.90	0.97	0.03	0.05	0.95	0.66	0.34	0.25	0.75
$\mathrm{lPr}(\cdot)$	3.89	3.97	0.49	1.00	3.95	3.58	2.92	2.61	3.71

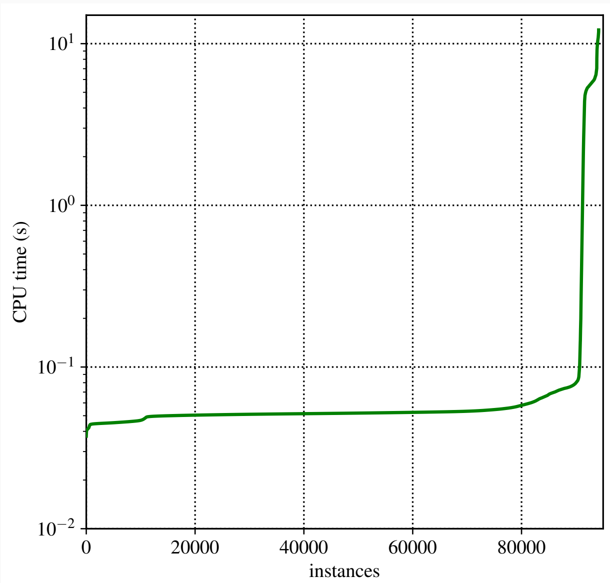
Can enumerate min. sols w/ log-linear delay

Optimization problem:

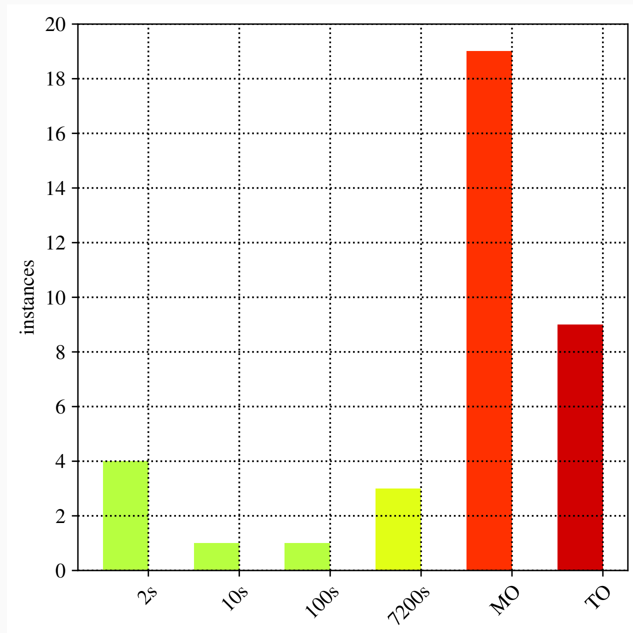
$$\begin{array}{ll} \min & \sum_{i=1}^{n} p_i \\ \text{s.t.} & \sum_{i=1}^{n} \delta_i p_i > \Phi \\ & p_i \in \{0,1\} \end{array}$$

Special case of knapsack; can solve in log-linear time

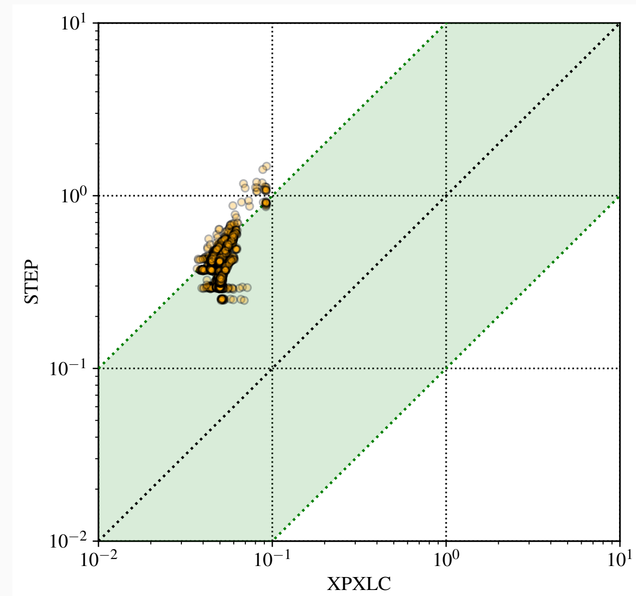
Overview of experimental results



(a) Raw performance of XPXLC



(b) Performance of STEP (with MOs & TOs)



(c) XPXLC vs STEP (no comp. time)

Our work (XPXLC) vs. STEP [SCD18, DH20]

Outline

Tractability

Duality

Links with Fairness

Research Directions

Outline

Tractability

Duality

- Abductive vs. Contrastive Explanations
- Global Explanations vs. Adversarial Examples

Links with Fairness

Research Directions

Main result

- Definitions:
  - Abductive explanation $\mathcal{X}$ (AXp, PI-explanation): [SCD18, INM19a]
    - Minimal set of literals sufficient for prediction

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

  - Contrastive explanation $\mathcal{Y}$ (CXp): [Mil19, INAM20]
    - Minimal set of literals sufficient for changing prediction

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

Main result

- Definitions:
    - Abductive explanation $\mathcal{X}$ (AXp, PI-explanation): [SCD18, INM19a]
        - Minimal set of literals sufficient for prediction

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

    - Contrastive explanation $\mathcal{Y}$ (CXp): [Mil19, INAM20]
        - Minimal set of literals sufficient for changing prediction

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

- Relating AXp's with CXp's: [INAM20]

AXp's are MHSes of CXp's and vice-versa

Main result

- Definitions:
  - Abductive explanation $\mathcal{X}$ (AXp, PI-explanation): [SCD18, INM19a]
    - Minimal set of literals sufficient for prediction

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

  - Contrastive explanation $\mathcal{Y}$ (CXp): [Mil19, INAM20]
    - Minimal set of literals sufficient for changing prediction

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

- Relating AXp's with CXp's: [INAM20]

AXp's are MHSes of CXp's and vice-versa

- Why bother?

Main result

- Definitions:
  - Abductive explanation $\mathcal{X}$ (AXp, PI-explanation): [SCD18, INM19a]
    - Minimal set of literals sufficient for prediction

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

  - Contrastive explanation $\mathcal{Y}$ (CXp): [Mil19, INAM20]
    - Minimal set of literals sufficient for changing prediction

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

- Relating AXp's with CXp's: [INAM20]

AXp's are MHSes of CXp's and vice-versa

- Why bother? **Can compute AXp's from CXp's, and vice-versa !**
  - E.g. one can **enumerate** AXp's+CXp's concurrently

Main result

- Definitions:
  - Abductive explanation $\mathcal{X}$ (AXp, PI-explanation): [SCD18, INM19a]
    - Minimal set of literals sufficient for prediction

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

  - Contrastive explanation $\mathcal{Y}$ (CXp): [Mil19, INAM20]
    - Minimal set of literals sufficient for changing prediction

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

- Relating AXp's with CXp's: [INAM20]

AXp's are MHSes of CXp's and vice-versa

- Why bother? **Can compute AXp's from CXp's, and vice-versa !**
  - E.g. one can **enumerate** AXp's+CXp's concurrently
- Work exploits hitting set duality, first studied in model-based diagnosis [Rei87]

Outline

Tractability

Duality

Abductive vs. Contrastive Explanations

Global Explanations vs. Adversarial Examples

Links with Fairness

Research Directions

Overview

- Vast body of work on computing explanations (XPs)
  - Mostly heuristic approaches, with recent rigorous solutions
- Vast body of work on coping with adversarial examples (AEs)
  - Both heuristic and rigorous approaches

Overview

- Vast body of work on computing explanations (XPs)
  - Mostly heuristic approaches, with recent rigorous solutions
- Vast body of work on coping with adversarial examples (AEs)
  - Both heuristic and rigorous approaches
- Can XPs and AEs be somehow related?
  - Recent work observed that some connection existed, but formal connection has been elusive

Overview

- Vast body of work on computing explanations (XPs)
  - Mostly heuristic approaches, with recent rigorous solutions
- Vast body of work on coping with adversarial examples (AEs)
  - Both heuristic and rigorous approaches
- Can XPs and AEs be somehow related?
  - Recent work observed that some connection existed, but formal connection has been elusive
- Recent proposal of a (first) link between XPs and AEs [INM19b]
  - Work exploits hitting set duality, first studied in model-based diagnosis [Rei87]

A well-known example

[RN10]

Example	Input Attributes										Goal
	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
x_1	Yes	No	No	Yes	Some	$$$	No	Yes	French	0–10	$y_1 =$ Yes
x_2	Yes	No	No	Yes	Full	$	No	No	Thai	30–60	$y_2 =$ No
x_3	No	Yes	No	No	Some	$	No	No	Burger	0–10	$y_3 =$ Yes
x_4	Yes	No	Yes	Yes	Full	$	Yes	No	Thai	10–30	$y_4 =$ Yes
x_5	Yes	No	Yes	No	Full	$$$	No	Yes	French	>60	$y_5 =$ No
x_6	No	Yes	No	Yes	Some	$$	Yes	Yes	Italian	0–10	$y_6 =$ Yes
x_7	No	Yes	No	No	None	$	Yes	No	Burger	0–10	$y_7 =$ No
x_8	No	No	No	Yes	Some	$$	Yes	Yes	Thai	0–10	$y_8 =$ Yes
x_9	No	Yes	Yes	No	Full	$	Yes	No	Burger	>60	$y_9 =$ No
x_{10}	Yes	Yes	Yes	Yes	Full	$$$	No	Yes	Italian	10–30	$y_{10} =$ No
x_{11}	No	No	No	No	None	$	No	No	Thai	0–10	$y_{11} =$ No
x_{12}	Yes	Yes	Yes	Yes	Full	$	No	No	Burger	30–60	$y_{12} =$ Yes

A well-known example (Cont.)

- 10 features:

$$\{\text{A(lternate)}, \text{B(ar)}, \text{W(eekend)}, \text{H(ungry)}, \text{Pa(trons)}, \text{Pr(ice)}, \text{Ra(in)}, \text{Re(serv.)}, \text{T(ype)}, \text{E(stim.)}\}$$

- Example instance (x_1, with outcome $y_1 = \text{Yes}$):

$$\{\text{A}, \neg\text{B}, \neg\text{W}, \text{H}, (\text{Pa} = \text{Some}), (\text{Pr} = \$\$\$), \neg\text{Ra}, \text{Re}, (\text{T} = \text{French}), (\text{E} = 0\text{–}10)\}$$

- A possible decision set (obtained with some off-the-shelf tool, & function*):

$$\begin{array}{llllr}
\textbf{IF} & (\text{Pa} = \text{Some}) \wedge \neg(\text{E} = {>}60) & \textbf{THEN} & (\text{Wait} = \text{Yes}) & (\text{R1}) \\
\textbf{IF} & \text{W} \wedge \neg(\text{Pr} = \$\$\$) \wedge \neg(\text{E} = {>}60) & \textbf{THEN} & (\text{Wait} = \text{Yes}) & (\text{R2}) \\
\textbf{IF} & \neg\text{W} \wedge \neg(\text{Pa} = \text{Some}) & \textbf{THEN} & (\text{Wait} = \text{No}) & (\text{R3}) \\
\textbf{IF} & (\text{E} = {>}60) & \textbf{THEN} & (\text{Wait} = \text{No}) & (\text{R4}) \\
\textbf{IF} & \neg(\text{Pa} = \text{Some}) \wedge (\text{Pr} = \$\$\$) & \textbf{THEN} & (\text{Wait} = \text{No}) & (\text{R5})
\end{array}$$

Counterexamples & breaks

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \models (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \nparallel \pi$

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \models (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \nparallel \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\text{Wait} = \text{Yes})$:

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\text{Wait} = \text{Yes})$:
  - Using (R1) (and assuming a consistent instance), an explanation is:

$$(\text{Pa} = \text{Some}) \wedge \neg(\text{E} = {>}60)$$

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \vDash (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\text{Wait} = \text{Yes})$:
  - Using (R1) (and assuming a consistent instance), an explanation is:

$$(\text{Pa} = \text{Some}) \wedge \neg(\text{E} = {>}60)$$

  - Due to (R5), a counterexample is:

$$\neg(\text{Pa} = \text{Some}) \wedge (\text{Pr} = \$\$\$)$$

Counterexamples & breaks

- Counterexamples:

  A subset-minimal set $\mathcal{C}$ of literals is a counterexample (CEx) to a prediction π, if $\mathcal{C} \models (\mathcal{M} \rightarrow \rho)$, with $\rho \in \mathbb{K} \wedge \rho \neq \pi$

- Breaks:

  A literal τ_i breaks a set of literals $\mathcal{S}$ (each denoting a different feature) if $\mathcal{S}$ contains a literal inconsistent with τ_i

- Back to the example, consider prediction $(\text{Wait} = \text{Yes})$:
  - Using (R1) (and assuming a consistent instance), an explanation is:

    $$(\text{Pa} = \text{Some}) \wedge \neg(\text{E} = {>}60)$$

  - Due to (R5), a counterexample is:

    $$\neg(\text{Pa} = \text{Some}) \wedge (\text{Pr} = \$\$\$)$$

  - XP $\mathcal{S}_1 = \{(\text{Pa} = \text{Some}), \neg(\text{E} = {>}60)\}$ breaks CEx $\mathcal{S}_2 = \{\neg(\text{Pa} = \text{Some}), (\text{Pr} = \$\$\$)\}$ and vice-versa

Some preliminary results

1. Relationship between XPs with CEx's:

Some preliminary results

1. Relationship between XPs with CEx's:
    - Each XP breaks every CEx

Some preliminary results

1. Relationship between XPs with CEx’s:
    - Each XP breaks every CEx
    - Each CEx breaks every XP

Some preliminary results

1. Relationship between XPs with CEx's:
   - Each XP breaks every CEx
   - Each CEx breaks every XP

   $\therefore$ XPs can be computed from all CEx's (by HSD) and vice-versa

Some preliminary results

1. Relationship between XPs with CEx's:
   - Each XP breaks every CEx
   - Each CEx breaks every XP

   $\therefore$ XPs can be computed from all CEx's (by HSD) and vice-versa

2. Given instance $\mathcal{I}$, an AE can be computed from closest CEx

Revisiting the example

- Restaurant dataset
- ML model is decision set (shown earlier)
- Prediction is $(\mathrm{Wait} = \mathrm{Yes})$

- Global explanations:
  1. $(\mathrm{Pa} = \mathrm{Some}) \land \neg(\mathrm{E} = {>}60)$
  2. $\mathrm{W} \land \neg(\mathrm{Pr} = \$\$\$) \land \neg(\mathrm{E} = {>}60)$

- Counterexamples:
  1. $\neg\mathrm{W} \land \neg(\mathrm{Pa} = \mathrm{Some})$
  2. $(\mathrm{E} = {>}60)$
  3. $\neg(\mathrm{Pa} = \mathrm{Some}) \land (\mathrm{Pr} = \$\$\$)$

- The XP's break the CEx's and vice-versa

Outline

Tractability

Duality

Links with Fairness

Research Directions

Some questions regarding fairness

[ICS+20]

- What should be the criterion for fairness of a model (a classifier)?
- What should be the criterion for dataset bias?
- What should be the criterion for fairness of a particular decision?
- How to learn a fair model from biased data?

Basic definitions

- Classifier: boolean function $\varphi(\mathbf{x}, \mathbf{y}) \in \{0, 1\}$, where
  - $\mathbf{x}$: values of **non-protected** features (salary, age, ...), and
  - $\mathbf{y}$: values of **protected** features (gender, race, ...).
- Dataset: set of tuples $\langle \mathbf{x}, \mathbf{y}, c \rangle$ with $c \in \{0, 1\}$
- Examples:
  1. Should a bank approve a loan to a customer?
  2. Should a judge release a prisoner on probation?

Outline

Tractability

Duality

Links with Fairness

- Fairness Through Unawareness
- Relating Fairness with Explanations

Research Directions

Criterion: fairness through unawareness (FTU)

- FTU: φ is a function only of the non-protected features $\mathbf{x}$
- FTU criterion for testing unfairness of model:

$$\exists \mathbf{x}\, \exists (\mathbf{y}_1, \mathbf{y}_2).\, [\mathbf{y}_1 \neq \mathbf{y}_2 \;\wedge\; \varphi(\mathbf{x}, \mathbf{y}_1) \neq \varphi(\mathbf{x}, \mathbf{y}_2)]$$

E.g. Alice and Bob are identical (same salary, age, ...), Alice is refused a loan but Bob isn't

- Optimisation: only need to test criterion for $\mathbf{y}_1, \mathbf{y}_2$ which differ on a single feature

Criterion: fairness through unawareness (FTU)

- **FTU**: φ is a function only of the non-protected features $\mathbf{x}$
- FTU criterion for testing unfairness of model:

$$\exists \mathbf{x}\, \exists (\mathbf{y}_1, \mathbf{y}_2).\, [\mathbf{y}_1 \neq \mathbf{y}_2 \ \wedge \ \varphi(\mathbf{x}, \mathbf{y}_1) \neq \varphi(\mathbf{x}, \mathbf{y}_2)]$$

E.g. Alice and Bob are identical (same salary, age, ...), Alice is refused a loan but Bob isn't

- Optimisation: only need to test criterion for $\mathbf{y}_1, \mathbf{y}_2$ which differ on a single feature

Possible drawbacks of **FTU:**

- There may be correlations between protected and non-protected features

  E.g.: the bank isn't unfair to women, they just don't give loans to people who are pregnant!
- Positive discrimination may be a good thing

  E.g.: height restrictions for army recruits are less strict for women

FTU as a criterion for dataset bias

- FTU criterion for testing bias of a dataset $\mathcal{D}$:

$$\exists \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2.[\mathbf{y}_1 \neq \mathbf{y}_2 \ \wedge \ \langle \mathbf{x}, \mathbf{y}_1, 0 \rangle, \langle \mathbf{x}, \mathbf{y}_2, 1 \rangle \in \mathcal{D}]$$

  - Criterion can be applied even if $\mathcal{D}$ is inconsistent (i.e. $\exists \mathbf{x}, \mathbf{y}[\langle \mathbf{x}, \mathbf{y}, 0 \rangle, \langle \mathbf{x}, \mathbf{y}, 1 \rangle \in \mathcal{D}]$)
  - Criterion can be tested in linear time (using hash tables) since it is equivalent to: $\exists \mathbf{x}$ such that

$$\begin{aligned} |\{c : \exists \mathbf{y}, \langle \mathbf{x}, \mathbf{y}, c \rangle \in \mathcal{D}\}| &> 1 \\ |\{\mathbf{y} : \exists c, \langle \mathbf{x}, \mathbf{y}, c| \rangle \in \mathcal{D}\}| &> 1 \end{aligned}$$

- Recent work showed that FTU is unique is respecting a number of desirable fairness properties [ICS+20]

Outline

Tractability

Duality

Links with Fairness

- Fairness Through Unawareness
- Relating Fairness with Explanations

Research Directions

Local fairness: fairness of a particular decision

- An example:
  - Emma wants to know if she was refused a loan because she is a woman
  - The bank uses a simple model: refuse a loan if the client is unemployed or if they are a woman
  - This model is clearly unfair with respect to gender, but
    - The bank claims that the *decision* is fair since they refused the loan because Emma is unemployed
    - Emma points out there are two possible explanations for the refusal:

      (1) she is unemployed, or that
      (2) she is a woman,

      and hence the decision should be considered unfair

Local fairness: fairness of a particular decision

- An example:
  - Emma wants to know if she was refused a loan because she is a woman
  - The bank uses a simple model: refuse a loan if the client is unemployed or if they are a woman
  - This model is clearly unfair with respect to gender, but
    - The bank claims that the *decision* is fair since they refused the loan because Emma is unemployed
    - Emma points out there are two possible explanations for the refusal:

      (1) she is unemployed, or that
      (2) she is a woman,

      and hence the decision should be considered unfair
  - Who is right?

Fairness of a particular decision from explanations

- **Recap:** a PI-explanation $\mathcal{E}$ of a prediction $\varphi(\mathbf{z}) = c$ is a subset-minimal set of literals from the literals $\mathcal{Z}$ of $\mathbf{z} \in \mathbb{F}$, which entails the prediction c:

$$\forall(\mathbf{x} \in \mathbb{F}).\,[\mathcal{E}(\mathbf{x}) \rightarrow (\varphi(\mathbf{x}) = c)]$$

- An explanation is fair if it includes no protected features
- A prediction $\varphi(\mathbf{z}) = c$ is:
    - Universally fair: if all of its explanations are fair
    - Existentially fair: if at least one of its explanations is fair

Fairness of a particular decision from explanations

- **Recap:** a PI-explanation $\mathcal{E}$ of a prediction $\varphi(\mathbf{z}) = c$ is a subset-minimal set of literals from the literals $\mathcal{Z}$ of $\mathbf{z} \in \mathbb{F}$, which entails the prediction c:

$$\forall(\mathbf{x} \in \mathbb{F}).\, [\mathcal{E}(\mathbf{x}) \rightarrow (\varphi(\mathbf{x}) = c)]$$

- An explanation is fair if it includes no protected features
- A prediction $\varphi(\mathbf{z}) = c$ is:
  - Universally fair: if all of its explanations are fair
  - Existentially fair: if at least one of its explanations is fair
- Back to the example:
  Emma's loan refusal decision is existentially fair but not universally fair

Complexity of checking fairness

- A model φ is fair iff all its decisions are universally fair
    - Checking fairness of a model is in co-NP
- Checking existential fairness of a decision $\varphi(\mathbf{z}) = c$ is in co-NP
    - It can be solved by exhaustive search over only the protected features
- Checking universal fairness of a decision $\varphi(\mathbf{z}) = c$ is in Π_2^p

Outline

Tractability

Duality

Links with Fairness

Research Directions

Many challenges

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?

- More efficient (and still rigorous) alternatives to prime-based explanations?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?
- Scaling the learning of interpretable models?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?
- Scaling the learning of interpretable models?
  - **Q:** How to target large datasets?
  - **Q:** Mechanisms for avoiding overfitting?

Many challenges

- Scalability, scalability, scalability...
  - Rigorous methods still lacking in reasoning about NNs
  - **Q:** How to improve performance of sound & complete methods for assessing robustness?
  - **Q:** Alternatives to NNs in some settings?
- More efficient (and still rigorous) alternatives to prime-based explanations?
  - **Q:** Basis for developing safe heuristics?
- Scaling the learning of interpretable models?
  - **Q:** How to target large datasets?
  - **Q:** Mechanisms for avoiding overfitting?
- Exploiting logic in learning black-box models [FBD+19]

Questions?

References i

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

[DH20] Adnan Darwiche and Auguste Hirth.
On the reasons behind decisions.
In *ECAI*, pages 712–720, 2020.

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

[FBD+19] Marc Fischer, Mislav Balunovic, Dana Drachsler-Cohen, Timon Gehr, Ce Zhang, and Martin T. Vechev.
DL2: training and querying neural networks with logic.
In *ICML*, pages 1931–1941, 2019.

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

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

References ii

[INAM20] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva.
On relating 'why?' and 'why not?' explanations.
CoRR, abs/2012.11067, 2020.

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

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

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

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

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

References iii

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

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

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