Logic-Based Explainable Artificial Intelligence

Joao Marques-Silva

ICREA, Univ. Lleida, Catalunya, Spain

ESSAI, Athens, Greece, July 2024

My team's recent & not so recent work...

SAT Solving
(Clause learning, UIPs, ...)

Quantification & CEGAR
(QBF, QMaxSAT, etc.)

Function Synthesis
(Min DNF cover, ...)

Inconsistency
(MUS, MCS, etc.)

Certification of Reasoners

Model Checking, Synthesizing Invariants, ATPG, Reconfiguration

Optimization
(MaxSAT, MinSAT, PBO, WBO, etc.)

Propositional Encodings, Backbones, Autarkies, Minimal models, etc.

Enumeration
(MUSes, MCSes, etc.)

Proof Systems
(DRMaxSAT, etc.)

Primes, Abduction, DLs, etc.

New area of research, since circa 2018...

SAT Solving (Clause learning, UIPs, ...)	Quantification & CEGAR (QBF, QMaxSAT, etc.)	Function Synthesis (Min DNF cover, ...)
Inconsistency (MUS, MCS, etc.)	Certification of Reasoners	Model Checking, Synthesizing Invariants, ATPG, Reconfiguration
Optimization (MaxSAT, MinSAT, PBO, WBO, etc.)	Propositional Encodings, Backbones, Autarkies, Minimal models, etc.	Enumeration (MUSes, MCSes, etc.)
Proof Systems (DRMaxSAT, etc.)	Primes, Abduction, DLs, etc.	Explainability & Interpretability in ML

New area of research, since circa 2018...

- SAT Solving (Clause learning, UIPs, ...)
- Quantification & CEGAR (QBF, QMaxSAT, etc.)
- Function Synthesis (Min DNF cover, ...)
- Inconsistency (MUS, MCS, etc.)
- Certification of Reasoners
- Model Checking, Synthesizing Invariants, ATPG, Reconfiguration
- Optimization (MaxSAT, MinSAT, PBO, WBO, etc.)
- Propositional Encodings, Backbones, Autarkies, Minimal models, etc.
- Enhancing ML by exploiting AR & FM !
- Proof Systems (DRMaxSAT, etc.)
- Primes, Abduction, DLs, etc.
- Explainability & Interpretability in ML

Lecture 01

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

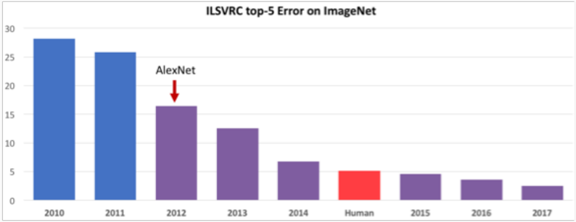
DeepMind

AlphaGo

AlphaGo Zero & Alpha Zero

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

Image & Speech Recognition

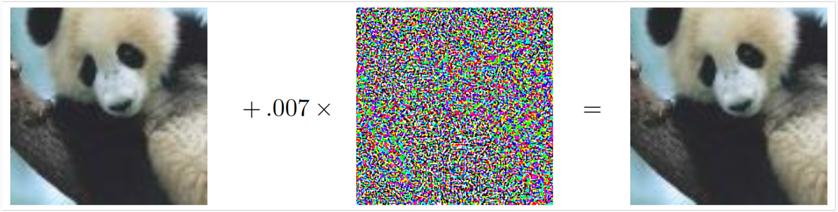


http://gradientscience.org/intro_adversarial/

Can we trust ML models?

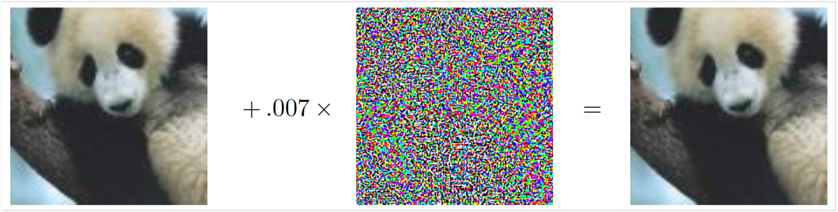
- Accuracy in training/test data
- Complex ML models are brittle
  - Extensive work on finding adversarial examples
  - Extensive work on learning robust ML models
- More recently, complex ML models hallucinate
- One **must** be able to validate operation of ML model, with rigor
  - Explanations; robustness; verification

ML models are brittle — adversarial examples



Panda Gibbon Goodfellow et al., ICLR'15

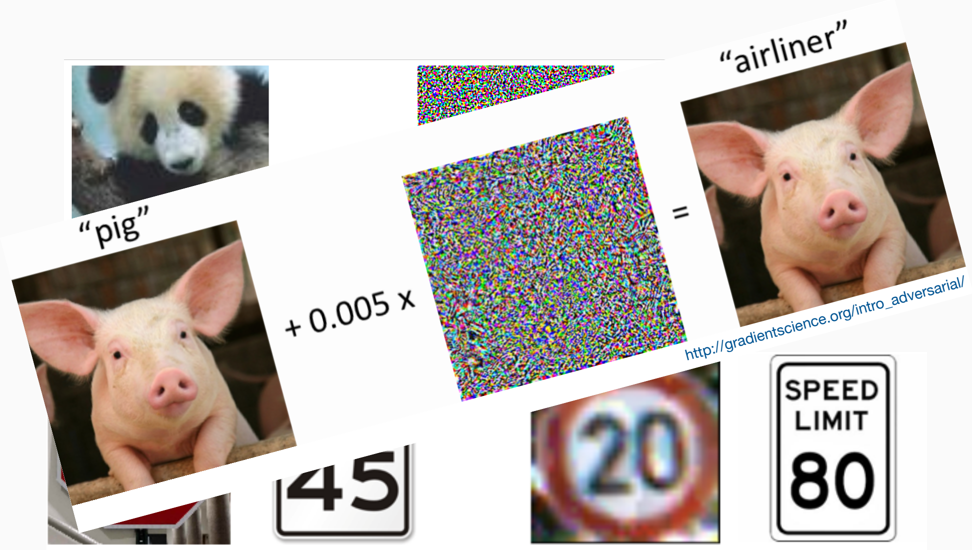
ML models are brittle — adversarial examples



Panda Gibbon Goodfellow et al., ICLR'15

Eykholt et al'18 Aung et al'17

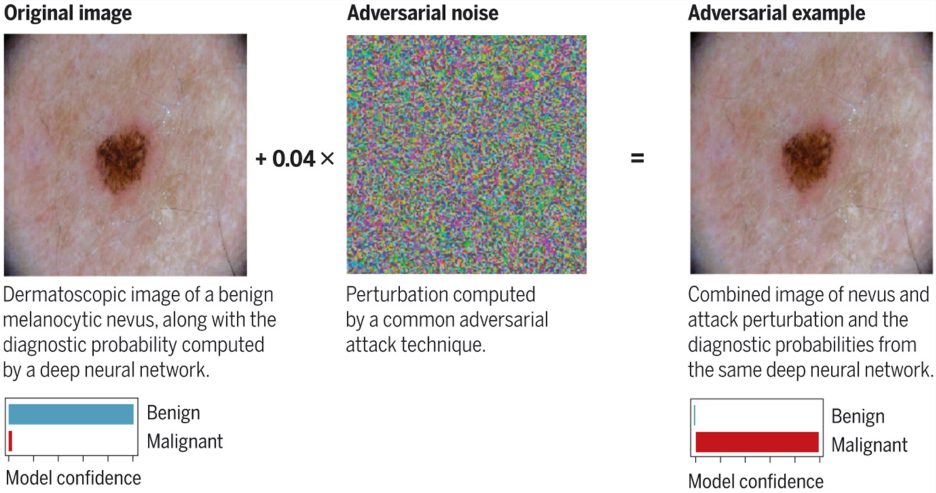
ML models are brittle — adversarial examples



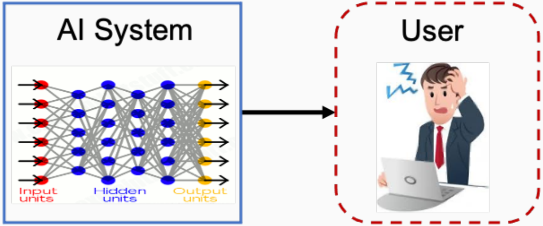
Eykholt et al'18

Aung et al'17

Adversarial examples can be very problematic

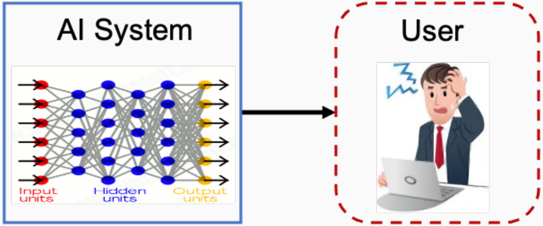


Finlayson et al., Nature 2019



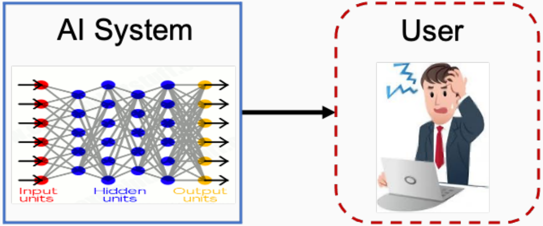
- Complex ML models are **opaque**
- Goal of XAI: **to help humans understand ML models**
- Many questions to address:

eXplainable AI (XAI)



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

eXplainable AI (XAI)



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

Importance of XAI

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

of 27 April 2016

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

(Text with EEA relevance)

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

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

Proposal for a

REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

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

■ *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)

XAI

EXPLAINABLE ARTIFICIAL INTELLIGENCE

FY17 FY18 FY19 FY20 FY21

David Gunning
DARPA/I2O
Program Update November 2017

DARPA

©DARPA

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

Digital Single Market

REPORT / STUDY | 8 April 2019

Ethics guidelines for trustworthy AI

Importance of XAI

REGULATION (EU) 2016/67… THE COUNCIL

on the p… …ata and on the free
movei… …tion Regulation)

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

European Union regulation…
and a "righ…

Bryce Goodma…

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

Proposal for a

…EAN PARLIAMENT AND OF THE COUNCIL

…RULES ON ARTIFICIAL INTELLIGENCE
…ACT) AND AMENDING CERTAIN UNION
…LATIVE ACTS

…e (XAI)

…pdate November 2017

DARPA

©DARPA

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

Digital Single Market

REPORT / STUDY | 8 April 2019

Ethics guidelines for trustworthy AI

XAI & EU guidelines (AI HLEG)

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

XAI & the principle of explicability


European Commission > Strategy > Digital Single Market > Reports and

Digital Single Market

About Artificial intelligence

Blog posts

News

The principle of explicability

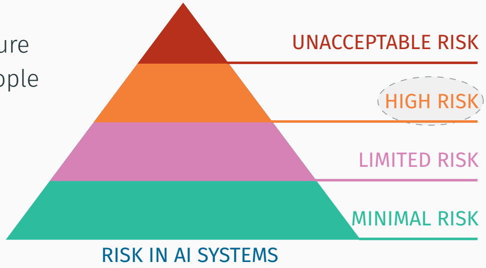
Explicability is crucial for building and maintaining users' trust in AI systems. This means that processes need to be transparent, the capabilities and purpose of AI systems openly communicated, and decisions – to the extent possible – explainable to those directly and indirectly affected. Without such 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]

& thousands of recent papers!

XAI for high-risk & safety-critical applications

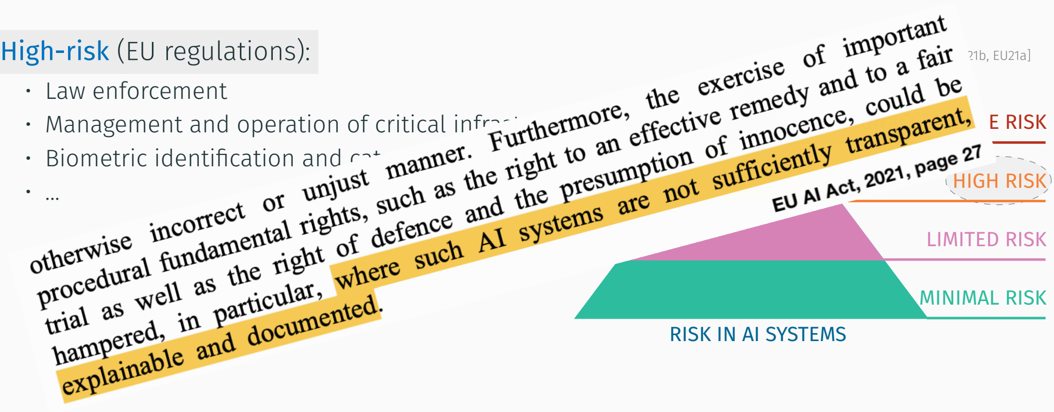
- High-risk (EU regulations):
  - Law enforcement
  - Management and operation of critical infrastructure
  - Biometric identification and categorization of people
  - ...

[EU21b, EU21a]



XAI for high-risk & safety-critical applications

- High-risk (EU regulations): 21b, EU21a]
  - Law enforcement
  - Management and operation of critical infra
  - Biometric identification and cat
  - ...

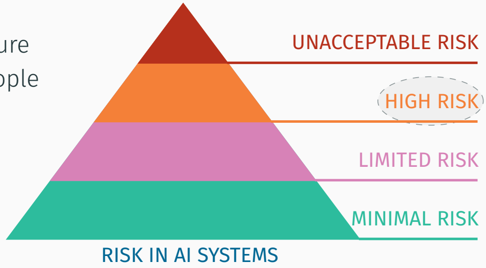

otherwise incorrect or unjust manner. Furthermore, the exercise of important procedural fundamental rights, such as the right to an effective remedy and to a fair trial as well as the right of defence and the presumption of innocence, could be hampered, in particular, where such AI systems are not sufficiently transparent, explainable and documented.

EU AI Act, 2021, page 27

XAI for high-risk & safety-critical applications

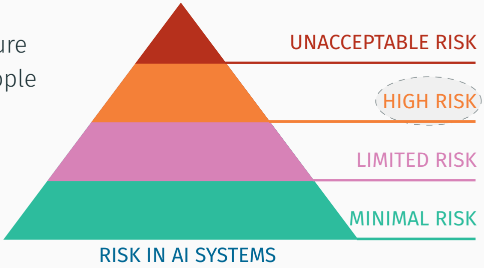
- High-risk (EU regulations):
  - Law enforcement
  - Management and operation of critical infrastructure
  - Biometric identification and categorization of people
  - ...
- And safety-critical:
  - Self-driving cars
  - Autonomous vehicles
  - Autonomous aereal devices
  - ...

[EU21b, EU21a]



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

[EU21b, EU21a]




PERSPECTIVE
https://doi.org/10.1038/s42256-019-0048-x
nature machine intelligence

Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead

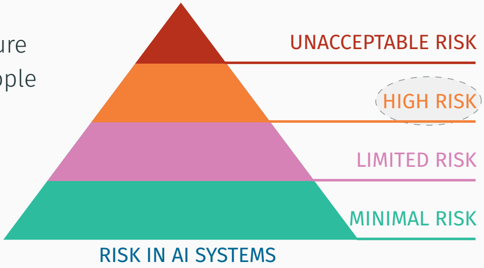
Cynthia Rudin

May 2019

XAI for high-risk & safety-critical applications

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

[EU21b, EU21a]




PERSPECTIVE
https://doi.org/10.1038/s42256-019-0048-x
nature machine intelligence

Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead

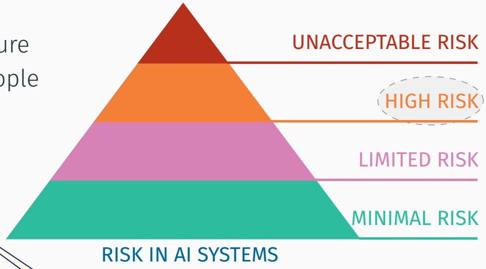
Cynthia Rudin

May 2019

XAI for high-risk & safety-critical applications

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

[EU21b, EU21a]



Main motivation for our work! (since 2019)

Can we trust (non-symbolic) XAI? – some questions

- Many proposed **solutions** for XAI
  - Most, and the better-known, are heuristic
  - I.e. no guarantees of rigor
- Many proposed **uses** of XAI
- Regular complaints about issues with existing (heuristic) methods of XAI

Can we trust (non-symbolic) XAI? – some questions

- Many proposed **solutions** for XAI
  - Most, and the better-known, are heuristic
  - I.e. no guarantees of rigor
- Many proposed **uses** of XAI
- Regular complaints about issues with existing (heuristic) methods of XAI

- **Q:** Can heuristic XAI be trusted in high-risk and/or safety-critical domains?
- **Q:** Can we validate results of heuristic XAI?

What have we been up to? 1. Created the field of symbolic (formal) XAI – I

[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa

What have we been up to? 1. Created the field of symbolic (formal) XAI – I

[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms

What have we been up to? 1. Created the field of symbolic (formal) XAI – I

[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms
- Intractability results
  - Devised efficient methods
  - Links with automated reasoners

What have we been up to? 1. Created the field of symbolic (formal) XAI – I

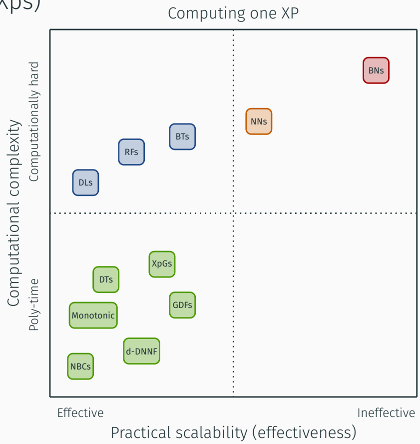
[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms
- Intractability results
  - Devised efficient methods
  - Links with automated reasoners
- Wealth of computational problems related with AXps/CXps

What have we been up to? 1. Created the field of symbolic (formal) XAI – I

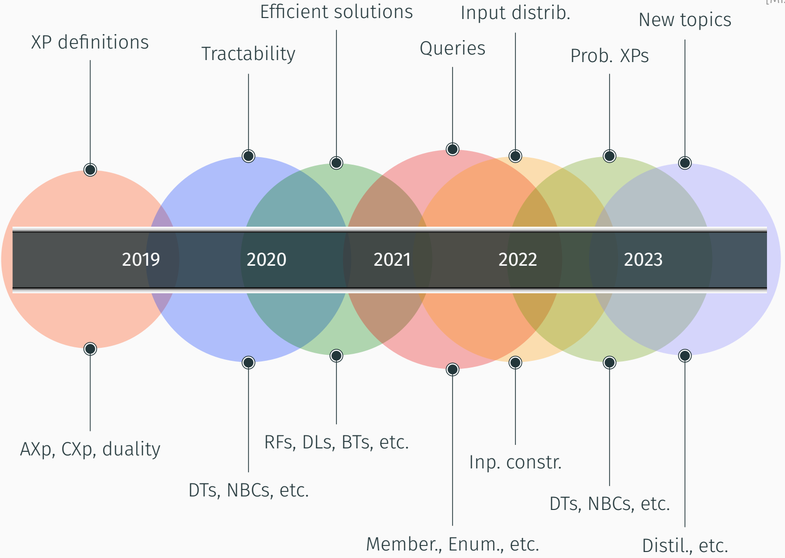
[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms
- Intractability results
  - Devised efficient methods
  - Links with automated reasoners
- Wealth of computational problems related with AXps/CXps



What have we been up to? 1. Created the field of symbolic (formal) XAI – II

[MI22, Mar22, MS23, Mar24]



[RSG16, LL17, RSG18, Rud19]

LIME

"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

X

SHAP

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg
Paul G. Allen School of Computer Science
University of Washington
Seattle, WA 98105
slund1@cs.washington.edu

Su-In Lee
Paul G. Allen School of Computer Science
Department of Genome Sciences
University of Washington
Seattle, WA 98105
suinlee@cs.washington.edu

X

PERSPECTIVE

https://doi.org/10.1038/s42256-019-0048-x

nature machine intelligence

Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead

Cynthia Rudin

X

Intrinsic Interpretability

Anchors: High-Precision Model-Agnostic Explanations

X

Anchor

Marco Tulio Ribeiro
University of Washington
marcotcr@cs.washington.edu

Sameer Singh
University of California, Irvine
sameer@uci.edu

Carlos Guestrin
University of Washington
guestrin@cs.washington.edu

What have we been up to? 2. Uncovered key myths of non-symbolic XAI – II

[MSH24, HMS24, HM23]

research and advances

Check for updates

DOI:10.1145/3635301

When the decisions of ML models impact people, one should expect explanations to offer the strongest guarantees of rigor. However, the most popular XAI approaches offer none.

BY JOAO MARQUES-SILVA AND XUANXIANG HUANG

Explainability Is *Not* a Game

» key insights

- Shapley values find extensive uses in explaining machine learning models and serve to assign importance to the features of the model.
- Shapley values for explainability also find ever-increasing uses in high-risk and safety-critical domains, for example, medical diagnosis.
- This article proves that the existing definition of Shapley values for explainability can produce misleading information regarding feature importance, and so can induce human decision makers in error.

66 COMMUNICATIONS OF THE ACM | JULY 2024 | VOL. 67 | NO. 7

Plan for this course

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

Unit #01

Foundations

Classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, m\}$, each feature i taking values from domain D_i
  - Features can be categorical, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{m} D_i$
- Set of classes $\mathcal{K} = \{c_1, \ldots, c_K\}$

Classification problems

- Set of features $\mathcal{F} = \{1, 2, \ldots, m\}$, each feature i taking values from domain D_i
  - Features can be categorical, discrete or real-valued
  - Feature space: $\mathbb{F} = \Pi_{i=1}^{m} D_i$
- Set of classes $\mathcal{K} = \{c_1, \ldots, c_K\}$
- ML model $\mathcal{M}_C$ computes a (non-constant) classification function $\kappa : \mathbb{F} \to \mathcal{K}$
  - $\mathcal{M}_C$ is a tuple $(\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$

Classification problems

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

Regression problems

- For regression problems:
    - Codomain: $\mathbb{V}$
    - Regression function: $\rho : \mathbb{F} \to \mathbb{V}$ (non-constant)
    - ML model: $\mathcal{M}_R$ is a tuple $(\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$

Regression problems

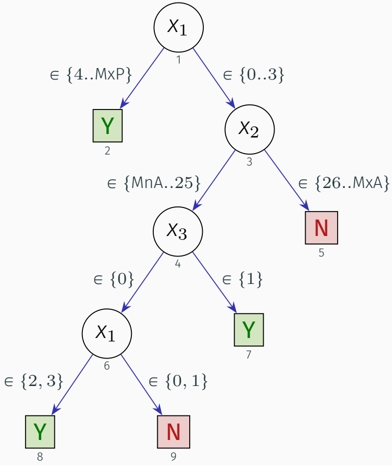
- For regression problems:
  - Codomain: $\mathbb{V}$
  - Regression function: $\rho : \mathbb{F} \to \mathbb{V}$ (non-constant)
  - ML model: $\mathcal{M}_R$ is a tuple $(\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$

- General ML model:
  - $\mathbb{T}$: range of possible predictions
  - Non-constant function $\tau : \mathbb{F} \to \mathbb{T}$
  - ML model: $\mathcal{M}$ is a tuple $(\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$

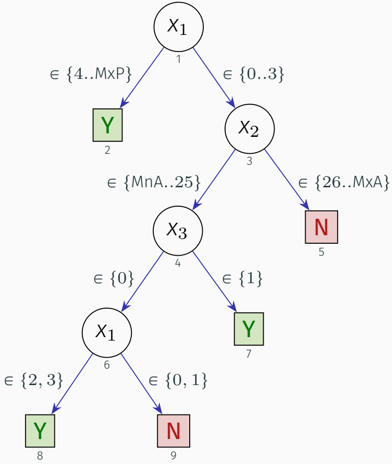
Regression problems

- For regression problems:
  - Codomain: $\mathbb{V}$
  - Regression function: $\rho : \mathbb{F} \to \mathbb{V}$ (non-constant)
  - ML model: $\mathcal{M}_R$ is a tuple $(\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
- General ML model:
  - $\mathbb{T}$: range of possible predictions
  - Non-constant function $\tau : \mathbb{F} \to \mathbb{T}$
  - ML model: $\mathcal{M}$ is a tuple $(\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$
- Instance: $(\mathbf{v}, q)$, $q \in \mathbb{T}$

Example ML models – classification – decision trees (DTs)

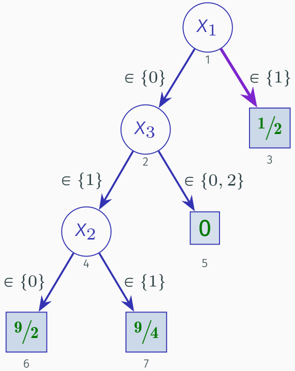


Example ML models – classification – decision trees (DTs)



- Literals in DTs can use $=$ or $\in$

Example ML models – regression – regression trees (RTs)



- Literals in RTs can use $=$ or $\in$

Example ML models – classification – rules

- Ordered rules – decision lists (DLs):

IF	$x_1 \land x_2$	THEN	predict Y
ELSE IF	$\neg x_2 \lor x_3$	THEN	predict N
ELSE		THEN	predict Y

$\mathcal{F} = \{1, 2, 3\}; \mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \{0, 1\}; \mathcal{K} = \{\mathsf{Y}, \mathsf{N}\}$

Example ML models – classification – rules

- Ordered rules – decision lists (DLs):

$$
\begin{array}{llll}
\text{IF} & x_1 \land x_2 & \text{THEN} & \text{predict } \mathsf{Y} \\
\text{ELSE IF} & \neg x_2 \lor x_3 & \text{THEN} & \text{predict } \mathsf{N} \\
\text{ELSE} & & \text{THEN} & \text{predict } \mathsf{Y}
\end{array}
$$

$$\mathcal{F} = \{1,2,3\}; \mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \{0,1\}; \mathcal{K} = \{\mathsf{Y}, \mathsf{N}\}$$

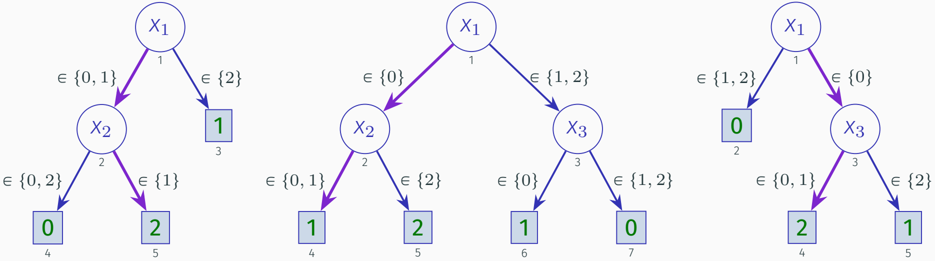
- Unordered rules – decision sets (DSs):

$$
\begin{array}{llll}
\text{IF} & x_1 + x_2 \geqslant 0 & \text{THEN} & \text{predict } \boxplus \\
\text{IF} & x_1 + x_2 < 0 & \text{THEN} & \text{predict } \boxminus
\end{array}
$$

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

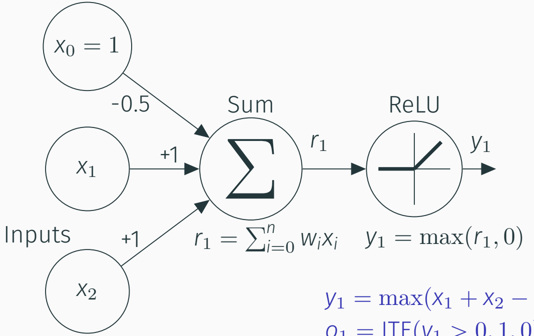
  - Issues of DSs: overlap; incomplete coverage

Example ML models – classification – random forests (RFs)



- For each input, each DT picks a class
- Result uses majority or weighted voting of the DTs

Example ML models – classification – neural networks (NNs)



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

Outline – Unit #01

Basics of (non-symbolic) XAI – more detail later

- Feature attribution:
  - LIME [RSG16]
  - SHAP [LL17]
  - ...

Basics of (non-symbolic) XAI – more detail later

- Feature attribution: assign relative importance to features
  - LIME [RSG16]
  - SHAP [LL17]
  - ...

Basics of (non-symbolic) XAI – more detail later

- Feature attribution: assign relative importance to features
  - LIME [RSG16]
  - SHAP [LL17]
  - ...
- Feature selection:
  - Anchors [RSG18]
  - ...

Basics of (non-symbolic) XAI – more detail later

- Feature attribution: assign relative importance to features
  - LIME [RSG16]
  - SHAP [LL17]
  - ...
- Feature selection: select set of features
  - Anchors [RSG18]
  - ...

Basics of (non-symbolic) XAI – more detail later

- Feature attribution: assign relative importance to features
  - LIME [RSG16]
  - SHAP [LL17]
  - ...
- Feature selection: select set of features
  - Anchors [RSG18]
  - ...
- Hybrid approaches:
  - Saliency maps [BBM+15]
  - ...

Basics of (non-symbolic) XAI – more detail later

- Feature attribution: assign relative importance to features
  - LIME [RSG16]
  - SHAP [LL17]
  - ...
- Feature selection: select set of features
  - Anchors [RSG18]
  - ...
- Hybrid approaches:
  - Saliency maps [BBM+15]
  - ...
- Intrinsic interpretability: [Mol20, Rud19]
  - DTs, DLs, ...

Basics of (non-symbolic) XAI – more detail later

- Feature attribution: assign relative importance to features
  - LIME [RSG16]
  - SHAP [LL17]
  - ...
- Feature selection: select set of features
  - Anchors [RSG18]
  - ...
- Hybrid approaches:
  - Saliency maps [BBM+15]
  - ...
- Intrinsic interpretability: the (interpretable) model is the explanation [Mol20, Rud19]
  - DTs, DLs, ...

Some examples

- Anchors: [RSG18]

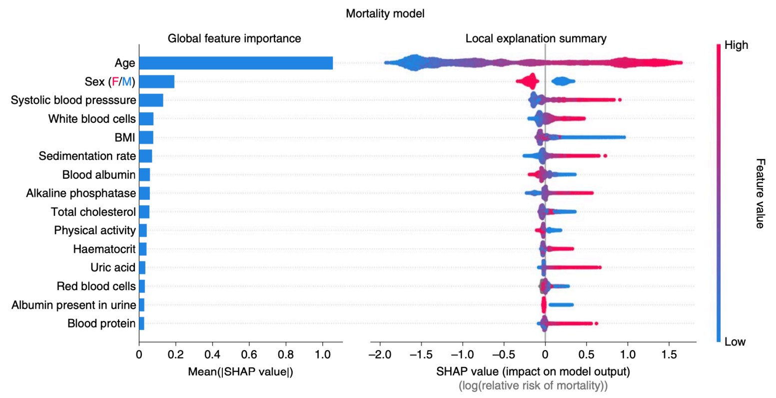
IF Country = United-States **AND** Capital Loss = Low
AND Race = White **AND** Relationship = Husband
AND Married **AND** $28 < \text{Age} \leq 37$
AND Sex = Male **AND** High School grad
AND Occupation = Blue-Collar
THEN PREDICT Salary $> \$50K$

Some examples

- Anchors: [RSG18]

IF Country = United-States **AND** Capital Loss = Low
AND Race = White **AND** Relationship = Husband
AND Married **AND** $28 < \text{Age} \leq 37$
AND Sex = Male **AND** High School grad
AND Occupation = Blue-Collar
THEN PREDICT Salary > \$50K

- SHAP: [LL17, LEC+20]



Outline – Unit #01

What explanations do we seek? I.e. how to answer Why? questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”

What explanations do we seek? I.e. how to answer Why? questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN $\kappa(\mathbf{x}) = \mathbf{c}$

What explanations do we seek? I.e. how to answer Why? questions?

- How to answer a **Why?** question? I.e. " Why (the prediction)? "
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN $\kappa(\mathbf{x}) = \mathbf{c}$

  - Explanation: set of literals (or just features) in <COND>; irreducibility matters!
    - <COND> is sufficient for the prediction

What explanations do we seek? I.e. how to answer Why? questions?

- How to answer a **Why?** question? I.e. " Why (the prediction)? "
  - Our answer to a **Why?** question is a **rule**:

    IF <COND> THEN $\kappa(\mathbf{x}) = \mathbf{c}$

  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**
    - <COND> is **sufficient** for the prediction
  - **Obs:** rules are used in tools like Anchors [RSG16]
    - An anchor is a "high-precision rule" [RSG16]

What explanations do we seek? I.e. how to answer Why? questions?

- How to answer a **Why?** question? I.e. " Why (the prediction)? "
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN $\kappa(\mathbf{x}) = \mathbf{c}$

  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**
    - <COND> is **sufficient** for the prediction
  - **Obs:** rules are used in tools like Anchors [RSG16]
    - An anchor is a "high-precision rule" [RSG16]
  - We seek a **rigorous** definition of rules for answering **Why?** questions such that,

What explanations do we seek? I.e. how to answer Why? questions?

- How to answer a **Why?** question? I.e. " Why (the prediction)? "
  - Our answer to a **Why?** question is a **rule**:

$$\text{IF} \quad \text{<COND>} \quad \text{THEN} \quad \kappa(\mathbf{x}) = \mathbf{c}$$

  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**
    - <COND> is sufficient for the prediction
  - **Obs:** rules are used in tools like Anchors [RSG16]
    - An anchor is a "high-precision rule" [RSG16]
  - We seek a **rigorous** definition of rules for answering **Why?** questions such that,
    - <COND> is sufficient for the prediction
    - <COND> is irreducible

What explanations do we seek? I.e. how to answer Why? questions?

- How to answer a **Why?** question? I.e. " Why (the prediction)? "
  - Our answer to a **Why?** question is a **rule**:

$$\text{IF} \quad \text{<COND>} \quad \text{THEN} \quad \kappa(\mathbf{x}) = \mathbf{c}$$

  - **Explanation**: set of literals (or just features) in <COND>; **irreducibility matters!**
    - <COND> is **sufficient** for the prediction
  - **Obs:** rules are used in tools like Anchors [RSG16]
    - An anchor is a "high-precision rule" [RSG16]
  - We seek a **rigorous** definition of rules for answering **Why?** questions such that,
    - <COND> is **sufficient** for the prediction
    - <COND> is **irreducible**
  - We also seek the algorithms for the rigorous computation of such rules

A decision list example

IF	$\neg x_1 \land x_2$	THEN	predict Y
ELSE IF	$\neg x_1 \land x_3$	THEN	predict Y
ELSE IF	$x_4 \land x_5$	THEN	predict N
ELSE		THEN	predict Y

A decision list example

$$
\begin{array}{llll}
\text{IF} & \neg x_1 \land x_2 & \text{THEN} & \text{predict } \mathsf{Y} \\
\text{ELSE IF} & \neg x_1 \land x_3 & \text{THEN} & \text{predict } \mathsf{Y} \\
\text{ELSE IF} & x_4 \land x_5 & \text{THEN} & \text{predict } \mathsf{N} \\
\text{ELSE} & & \text{THEN} & \text{predict } \mathsf{Y}
\end{array}
$$

- Explanation for why $\kappa(1, 1, 1, 1, 1) = \mathsf{N}$?

A decision list example

IF	$\neg x_1 \land x_2$	THEN	predict Y
ELSE IF	$\neg x_1 \land x_3$	THEN	predict Y
ELSE IF	$x_4 \land x_5$	THEN	predict **N**
ELSE		THEN	predict Y

- Explanation for why $\kappa(1,1,1,1,1) = \mathbf{N}$?
    - Given $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$,
      IF $(x_1 = 1) \land (x_4 = 1) \land (x_5 = 1)$ **THEN** $\kappa(\mathbf{x}) = \mathbf{N}$
    - I.e. $\{x_1 = 1, x_4 = 1, x_5 = 1\}$ suffice for DL to predict **N**

A decision list example

IF	$\neg x_1 \land x_2$	THEN	predict Y
ELSE IF	$\neg x_1 \land x_3$	THEN	predict Y
ELSE IF	$x_4 \land x_5$	THEN	predict N
ELSE		THEN	predict Y

- Explanation for why $\kappa(1,1,1,1,1) = \text{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$,
    IF $(x_1 = 1) \land (x_4 = 1) \land (x_5 = 1)$ **THEN** $\kappa(\mathbf{x}) = \text{N}$
  - I.e. $\{x_1 = 1, x_4 = 1, x_5 = 1\}$ suffice for DL to predict **N**
- Explanation for why $\kappa(1,0,0,0,0) = \text{Y}$?

A decision list example

IF	$\neg x_1 \land x_2$	THEN	predict Y
ELSE IF	$\neg x_1 \land x_3$	THEN	predict Y
ELSE IF	$x_4 \land x_5$	THEN	predict N
ELSE		THEN	predict Y

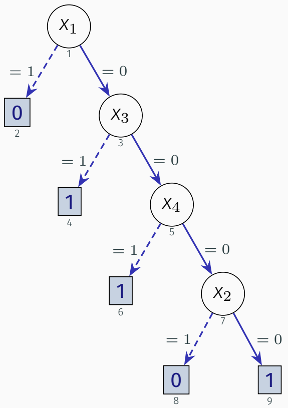
- Explanation for why $\kappa(1,1,1,1,1) = \text{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$,
    IF $(x_1 = 1) \land (x_4 = 1) \land (x_5 = 1)$ **THEN** $\kappa(\mathbf{x}) = \text{N}$
  - I.e. $\{x_1 = 1, x_4 = 1, x_5 = 1\}$ suffice for DL to predict **N**
- Explanation for why $\kappa(1,0,0,0,0) = \text{Y}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$,
    IF $(x_4 = 0)$ **THEN** $\kappa(\mathbf{x}) = \text{Y}$
  - I.e. $\{x_4 = 0\}$ suffices for DL to predict Y

A decision list example

IF	$\neg x_1 \land x_2$	THEN	predict Y
ELSE IF	$\neg x_1 \land x_3$	THEN	predict Y
ELSE IF	$x_4 \land x_5$	THEN	predict N
ELSE		THEN	predict Y

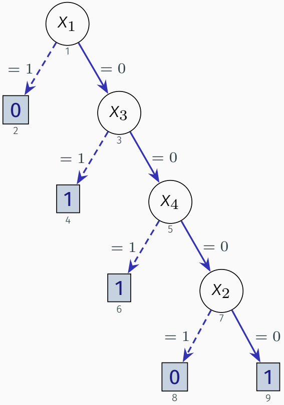
- Explanation for why $\kappa(1,1,1,1,1) = \text{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$,
    IF $(x_1 = 1) \land (x_4 = 1) \land (x_5 = 1)$ THEN $\kappa(\mathbf{x}) = \text{N}$
  - I.e. $\{x_1 = 1, x_4 = 1, x_5 = 1\}$ suffice for DL to predict N
- Explanation for why $\kappa(1,0,0,0,0) = \text{Y}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$,
    IF $(x_4 = 0)$ THEN $\kappa(\mathbf{x}) = \text{Y}$
  - I.e. $\{x_4 = 0\}$ suffices for DL to predict Y
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$,
    IF $(x_5 = 0)$ THEN $\kappa(\mathbf{x}) = \text{Y}$
  - I.e. $\{x_5 = 0\}$ also suffices for DL to predict Y

A decision tree example



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

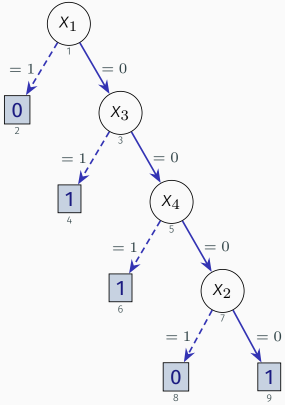
A decision tree example



- Explanation for why $\kappa(0, 0, 0, 0) = 1$?

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

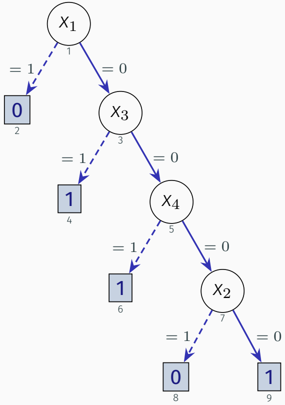
A decision tree example



- Explanation for why $\kappa(0,0,0,0) = 1$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$,
    IF $(x_1 = 0) \wedge (x_2 = 0)$ THEN $\kappa(\mathbf{x}) = 1$
  - I.e. $\{x_1 = 0, x_2 = 0\}$ suffice for DT to predict 1

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

A decision tree example

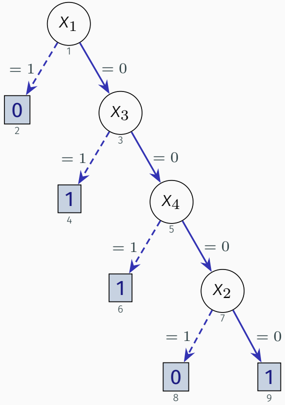


- Explanation for why $\kappa(0,0,0,0) = 1$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$,
    IF $(x_1 = 0) \wedge (x_2 = 0)$ THEN $\kappa(\mathbf{x}) = 1$
  - I.e. $\{x_1 = 0, x_2 = 0\}$ suffice for DT to predict 1

- Explanation for why $\kappa(1,1,1,1) = 0$?

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

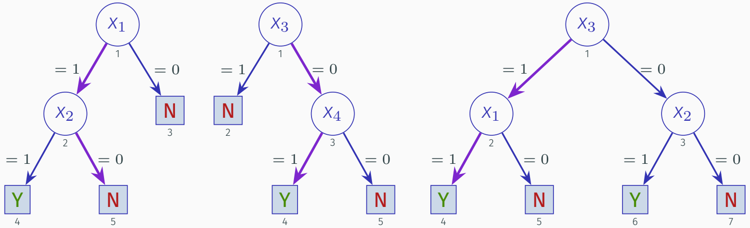
A decision tree example



- Explanation for why $\kappa(0,0,0,0) = 1$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$,
    IF $(x_1 = 0) \land (x_2 = 0)$ THEN $\kappa(\mathbf{x}) = 1$
  - I.e. $\{x_1 = 0, x_2 = 0\}$ suffice for DT to predict 1
- Explanation for why $\kappa(1,1,1,1) = 0$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$,
    IF $(x_1 = 1)$ THEN $\kappa(\mathbf{x}) = 0$
  - I.e. $\{x_1 = 1\}$ suffices for DT to predict 0

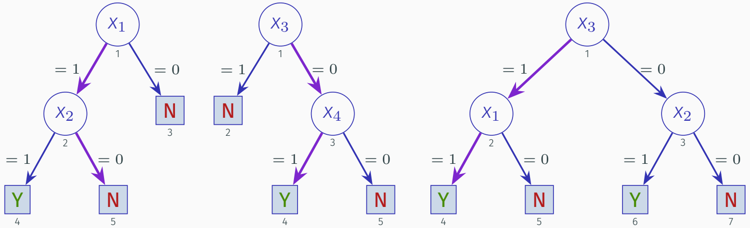
x_1	x_2	x_3	x_4	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

A random forest example



x_1	x_2	x_3	x_4	T_1	T_2	T_3	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

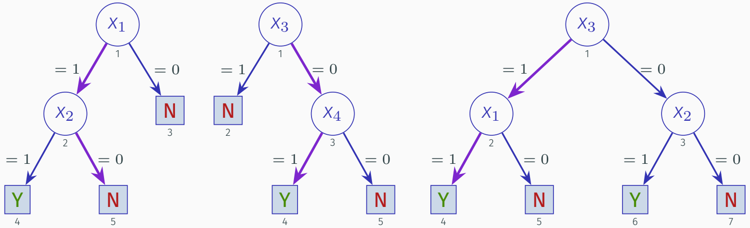
A random forest example



- Explanation for why $\kappa(1,0,0,1) = \text{N}$?

x_1	x_2	x_3	x_4	T_1	T_2	T_3	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

A random forest example

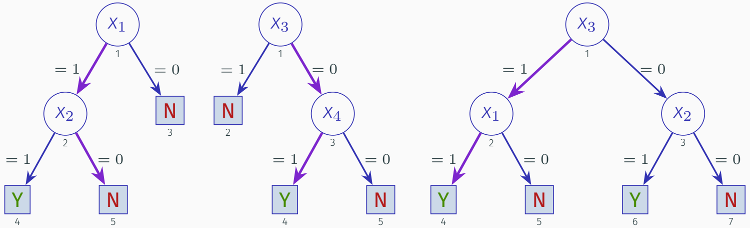


x_1	x_2	x_3	x_4	T_1	T_2	T_3	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

- Explanation for why $\kappa(1,0,0,1) = \text{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_2 = 0)$ THEN $\kappa(\mathbf{x}) = \text{N}$
  - I.e. $\{x_2 = 0\}$ suffices for DT to predict N

A random forest example

[IMS21]

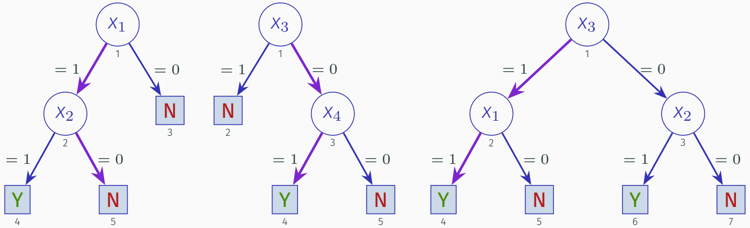


- Explanation for why $\kappa(1,0,0,1) = \text{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_2 = 0)$ THEN $\kappa(\mathbf{x}) = \text{N}$
  - I.e. $\{x_2 = 0\}$ suffices for DT to predict N
- Explanation for why $\kappa(1,1,1,1) = \text{Y}$?

x_1	x_2	x_3	x_4	T_1	T_2	T_3	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

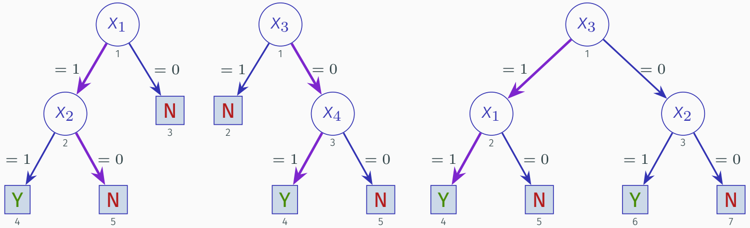
A random forest example

[IMS21]



- Explanation for why $\kappa(1,0,0,1) = \text{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_2 = 0)$ THEN $\kappa(\mathbf{x}) = \text{N}$
  - I.e. $\{x_2 = 0\}$ suffices for DT to predict N
- Explanation for why $\kappa(1,1,1,1) = \text{Y}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_1 = 1) \wedge (x_2 = 1)$ THEN $\kappa(\mathbf{x}) = \text{Y}$
  - I.e. $\{x_1 = 1, x_2 = 1\}$ suffice for DT to predict Y

x_1	x_2	x_3	x_4	T_1	T_2	T_3	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

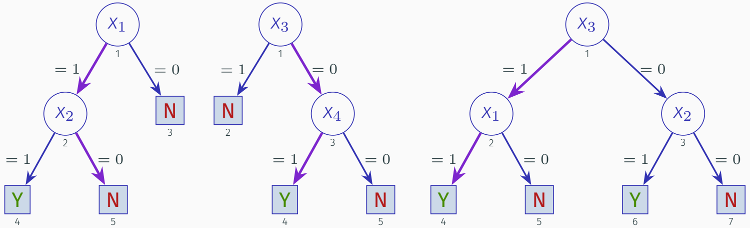


x_1	x_2	x_3	x_4	T_1	T_2	T_3	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

- Explanation for why $\kappa(1,0,0,1) = \mathsf{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_2 = 0)$ THEN $\kappa(\mathbf{x}) = \mathsf{N}$
  - I.e. $\{x_2 = 0\}$ suffices for DT to predict N
- Explanation for why $\kappa(1,1,1,1) = \mathsf{Y}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_1 = 1) \land (x_2 = 1)$ THEN $\kappa(\mathbf{x}) = \mathsf{Y}$
  - I.e. $\{x_1 = 1, x_2 = 1\}$ suffice for DT to predict Y
- Explanation for why $\kappa(0,1,1,1) = \mathsf{N}$?

A random forest example

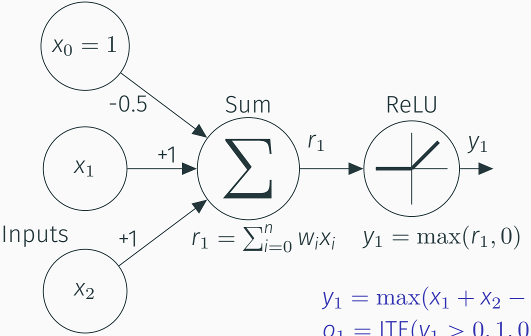
[IMS21]



x_1	x_2	x_3	x_4	T_1	T_2	T_3	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

- Explanation for why $\kappa(1,0,0,1) = \mathsf{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_2 = 0)$ THEN $\kappa(\mathbf{x}) = \mathsf{N}$
  - I.e. $\{x_2 = 0\}$ suffices for DT to predict N
- Explanation for why $\kappa(1,1,1,1) = \mathsf{Y}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_1 = 1) \land (x_2 = 1)$ THEN $\kappa(\mathbf{x}) = \mathsf{Y}$
  - I.e. $\{x_1 = 1, x_2 = 1\}$ suffice for DT to predict Y
- Explanation for why $\kappa(0,1,1,1) = \mathsf{N}$?
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$, IF $(x_1 = 0) \land (x_2 = 1) \land (x_3 = 1)$ THEN $\kappa(\mathbf{x}) = \mathsf{N}$
  - I.e. $\{x_1 = 0, x_2 = 1, x_3 = 1\}$ suffices for DT to predict N

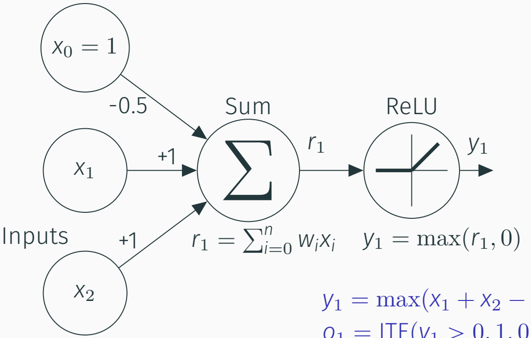
A neural network example



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

x_1	x_2	r_1	y_1	$\kappa(\mathbf{x})$
0	0	-0.5	0	0
0	1	0.5	0.5	1
1	0	0.5	0.5	1
1	1	1.5	1.5	1

A neural network example

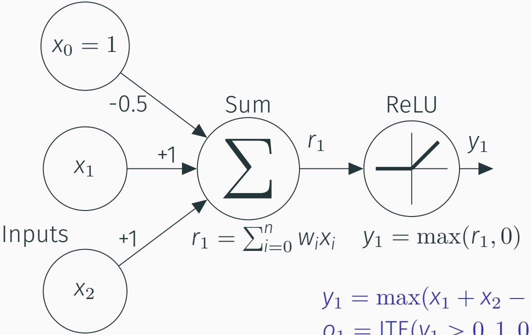


$$y_1 = \max(x_1 + x_2 - 0.5, 0)$$
$$o_1 = \mathsf{ITE}(y_1 > 0, 1, 0)$$

x_1	x_2	r_1	y_1	$\kappa(\mathbf{x})$
0	0	-0.5	0	0
0	1	0.5	0.5	1
1	0	0.5	0.5	1
1	1	1.5	1.5	1

- Explanation for why $\kappa(1,1) = 1$?

A neural network example

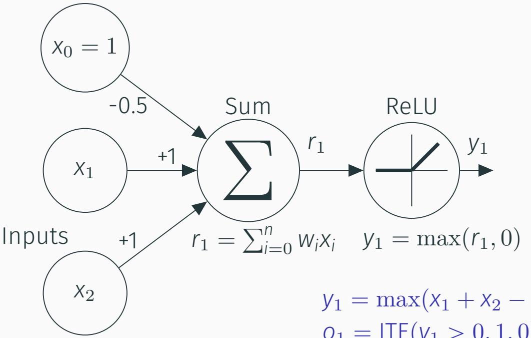


x_1	x_2	r_1	y_1	$\kappa(\mathbf{x})$
0	0	-0.5	0	**0**
0	1	0.5	0.5	**1**
1	0	0.5	0.5	**1**
1	1	1.5	1.5	**1**

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

- Explanation for why $\kappa(1, 1) = 1$?
  - Given $\mathbf{x} = (x_1, x_2)$, **IF** $(x_1 = 1)$ **THEN** $\kappa(\mathbf{x}) = 1$
  - I.e. $\{x_1 = 1\}$ suffices for NN to predict **1**

A neural network example



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

x_1	x_2	r_1	y_1	$\kappa(\mathbf{x})$
0	0	-0.5	0	**0**
0	1	0.5	0.5	**1**
1	0	0.5	0.5	**1**
1	1	1.5	1.5	**1**

- Explanation for why $\kappa(1,1) = 1$?
  - Given $\mathbf{x} = (x_1, x_2)$, **IF** $(x_1 = 1)$ **THEN** $\kappa(\mathbf{x}) = 1$
  - I.e. $\{x_1 = 1\}$ suffices for NN to predict **1**
  - Given $\mathbf{x} = (x_1, x_2)$, **IF** $(x_2 = 1)$ **THEN** $\kappa(\mathbf{x}) = 1$
  - I.e. $\{x_2 = 1\}$ suffices for NN to predict Y

An arbitrary classifier

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

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

An arbitrary classifier

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

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

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

An arbitrary classifier

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

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

- Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$,
  IF $(x_1 = 0) \wedge (x_3 = 0)$ THEN $\kappa(\mathbf{x}) = 1$
- I.e. $\{x_1 = 0, x_3 = 0\}$ suffices for DT to predict 1

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

Outline – Unit #01

Standard tools of the trade

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

Standard tools of the trade

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

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

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

SAT/SMT/MILP/CP solvers used as oracles – more detail later

- Deciding satisfiability, entailment
- Computing prime implicants/implicates
- Computing MUSes, MCSes
  - Algorithms: Deletion, QuickXplain, Progression, Dichotomic, etc. [MM20]
- Enumeration of MUSes, MCSes
  - Algorithms: Marco, Camus, etc. [LS08, LPMM16]
- Solving MaxSAT, MaxSMT
  - Algorithms: Core-guided, Minimum hitting sets, branch&bound, etc. [MHL+13]
- Solving quantification problems, e.g. QBF
  - Algorithms: Abstraction refinement [JKMC16]

Basic definitions in propositional logic

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

Basic definitions in propositional logic

- Atoms ($\{x, x_1, \ldots\}$) & literals (x_1, $\neg x_1$)
- Well-formed formulas using $\neg$, $\wedge$,$\vee$, $\ldots$
- Clause: disjunction of literals
- Term: conjunction of literals
- Conjunctive normal form (CNF): conjunction of clauses
- Disjunctive normal form (DNF): disjunction of terms
- Simple to generalize to more expressive domains
- CO$(\psi(\mathbf{x}))$ decides whether $\psi(\mathbf{x})$ is satisfiable (i.e. whether it is consistent), using an oracle for SAT/SMT/MILP/CP/etc.

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:

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

  - We say that $\tau(\mathbf{x})$ is **sufficient** for $\varphi(\mathbf{x})$

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:

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

  - We say that $\tau(\mathbf{x})$ is **sufficient** for $\varphi(\mathbf{x})$
- To decide entailment:
  - $\tau \vDash \varphi$ if $\tau(\mathbf{x}) \land \neg\varphi(\mathbf{x})$ is **not** consistent, i.e. $\mathsf{CO}(\tau(\mathbf{x}) \land \neg\varphi(\mathbf{x}))$ does not hold

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:

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

  - We say that $\tau(\mathbf{x})$ is **sufficient** for $\varphi(\mathbf{x})$
- To decide entailment:
  - $\tau \vDash \varphi$ if $\tau(\mathbf{x}) \land \neg\varphi(\mathbf{x})$ is **not** consistent, i.e. **CO**$(\tau(\mathbf{x}) \land \neg\varphi(\mathbf{x}))$ does not hold

- An example:
  - $\mathbb{F} = \{0, 1\}^2$
  - $\varphi(x_1, x_2) = x_1 \lor \neg x_2$
  - Clearly, $x_1 \vDash \varphi$ and $\neg x_2 \vDash \varphi$
  - Also, **CO**$(x_1 \land (\neg x_1 \land x_2))$ does not hold

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:

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

  - We say that $\tau(\mathbf{x})$ is **sufficient** for $\varphi(\mathbf{x})$
- To decide entailment:
  - $\tau \vDash \varphi$ if $\tau(\mathbf{x}) \land \neg\varphi(\mathbf{x})$ is **not** consistent, i.e. $\mathsf{CO}(\tau(\mathbf{x}) \land \neg\varphi(\mathbf{x}))$ does not hold

- An example:
  - $\mathbb{F} = \{0,1\}^2$
  - $\varphi(x_1, x_2) = x_1 \lor \neg x_2$
  - Clearly, $x_1 \vDash \varphi$ and $\neg x_2 \vDash \varphi$
  - Also, $\mathsf{CO}(x_1 \land (\neg x_1 \land x_2))$ does not hold
- Another example:
  - $\mathbb{F} = \{0,1\}^3$
  - $\varphi(x_1, x_2, x_3) = x_1 \land x_2 \lor x_1 \land x_3$
  - Clearly, $x_1 \land x_2 \vDash \varphi$ and $x_1 \land x_3 \vDash \varphi$
  - Also, $\mathsf{CO}(x_1 \land x_2 \land ((\neg x_1 \lor \neg x_2) \land (\neg x_1 \lor \neg x_3)))$ does not hold

Entailment & explanations – how do we construct explanations?

- Classification function:

  $$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

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

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

Entailment & explanations – how do we construct explanations?

- Classification function:

  $$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

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

- Localized explanation: any irreducible conjunction of literals, consistent with $\mathbf{v}$, and that entails the prediction

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

- Classification function:

  $$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

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

- Localized explanation: any irreducible conjunction of literals, consistent with $\mathbf{v}$, and that entails the prediction
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$,
    IF $(x_1 = 0) \wedge (x_3 = 0)$ THEN $\kappa(\mathbf{x}) = 1$

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

- Classification function:

  $$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

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

- Localized explanation: any irreducible conjunction of literals, consistent with $\mathbf{v}$, and that entails the prediction
  - Given $\mathbf{x} = (x_1, x_2, x_3, x_4)$,
    IF $(x_1 = 0) \wedge (x_3 = 0)$ THEN $\kappa(\mathbf{x}) = 1$

- Global explanation: any irreducible conjunction of literals, that is consistent, and that entails the prediction

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

Entailment & explanations – how do we construct explanations?

- Classification function:

  $$\kappa(x_1,x_2,x_3,x_4) = \neg x_1 \land \neg x_2 \lor x_1 \land x_2 \land x_4 \lor \neg x_1 \land x_2 \land \neg x_3 \lor \neg x_2 \land x_3 \land x_4$$

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

- Localized explanation: any irreducible conjunction of literals, consistent with $\mathbf{v}$, and that entails the prediction
  - Given $\mathbf{x} = (x_1,x_2,x_3,x_4)$,
    IF $(x_1 = 0) \land (x_3 = 0)$ THEN $\kappa(\mathbf{x}) = 1$

- Global explanation: any irreducible conjunction of literals, that is consistent, and that entails the prediction
  - Given $\mathbf{x} = (x_1,x_2,x_3,x_4)$,
    IF $(x_1 = 0) \land (x_2 = 0)$ THEN $\kappa(\mathbf{x}) = 1$

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

Outline – Unit #01

Decision sets with boolean features

- Example ML model:

  Features: $x_1, x_2, x_3, x_4 \in \{0, 1\}$ (boolean)

  Rules:

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

Decision sets with boolean features

- Example ML model:

  Features: $x_1, x_2, x_3, x_4 \in \{0, 1\}$ (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, i.e. is there overlap?

Decision sets with boolean features

- Example ML model:

  Features: $x_1, x_2, x_3, x_4 \in \{0, 1\}$ (boolean)

  Rules:

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

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

Decision sets with boolean features

- Example ML model:

  Features: $x_1, x_2, x_3, x_4 \in \{0, 1\}$ (boolean)

  Rules:

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

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

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

  ... and solve with SAT solver (after clausification) [Tse68, PG86]

  Or use PySAT [IMM18]

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

Decision sets with ordinal features

- Example ML model:

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

  Rules:

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

Decision sets with ordinal features

- Example ML model:

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

  Rules:

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

- **Q:** Can the model predict both $\boxplus$ and $\boxminus$ for some instance, i.e. is there overlap?

Decision sets 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 $2x_1 - x_2 \leqslant 0$ THEN predict $\boxminus$

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

Decision sets 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 $2x_1 - x_2 \leqslant 0$ THEN predict $\boxminus$

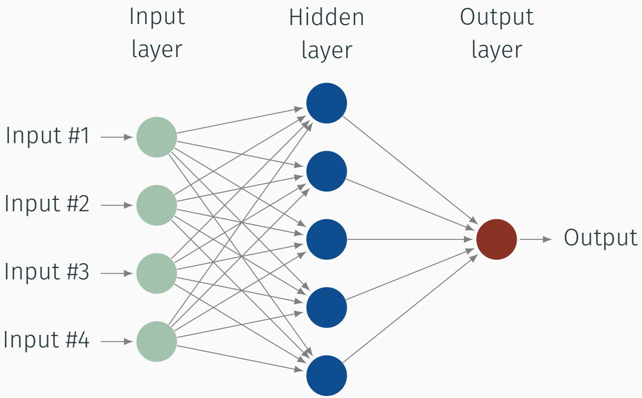
- **Q:** Can the model predict both $\boxplus$ and $\boxminus$ for some instance, i.e. is there overlap?
  - Yes, of course: pick $x_1 = 0$ and $x_2 = 1$
  - A formalization:

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

… and solve with SMT solver (many alternatives)

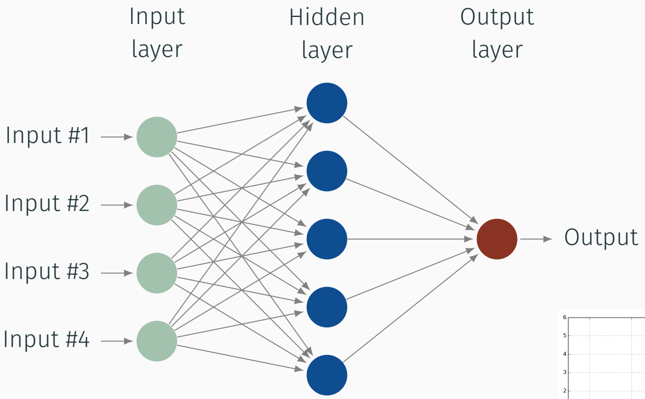
$\therefore$ There exists a model iff there exists a point in feature space yielding both predictions

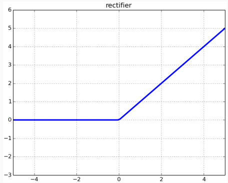
Neural networks



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

Neural networks





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

[NH10]

Encoding NNs using MILP

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

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

Encoding NNs using MILP

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

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

Encoding each **block**: [FJ18]

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

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

Encoding NNs using MILP

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

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

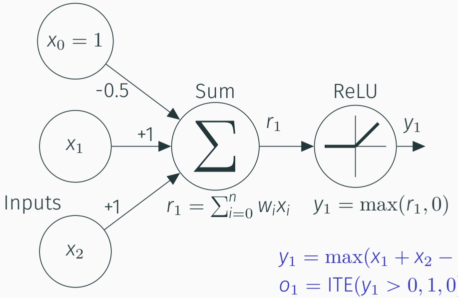
Modeling ML models with logic is not only possible but also simple !

Encoding each **block**: [FJ18]

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

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

Example – encoding a simple NN in MILP



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

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

MILP encoding:

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

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

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

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

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

Outline – Unit #01

What is intrinsic interpretability?

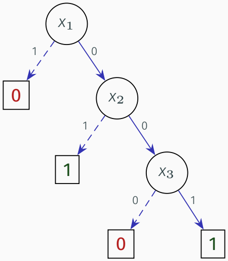
- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC^{+}22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*

What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]

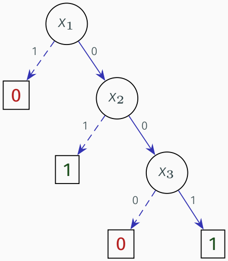
What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]



What is intrinsic interpretability?

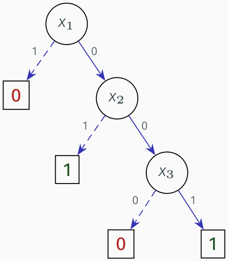
- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]



- What is an explanation for $((0, 0, 1), 1)$?

What is intrinsic interpretability?

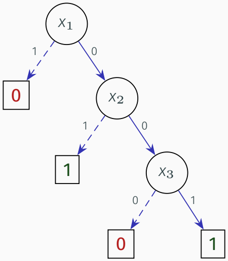
- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]



- What is an explanation for $((0, 0, 1), 1)$?
- Clearly, IF $\neg x_1 \wedge \neg x_2 \wedge x_3$ THEN $\kappa(\mathbf{x}) = 1$

What is intrinsic interpretability?

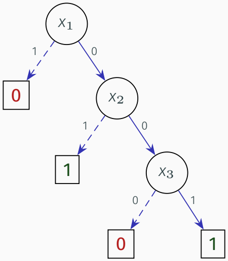
- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]



- What is an explanation for $((0,0,1),1)$?
- Clearly, IF $\neg x_1 \land \neg x_2 \land x_3$ THEN $\kappa(\mathbf{x}) = 1$
  - $\{\neg x_1, \neg x_2, x_3\}$ or $\{1,2,3\}$ is an explanation

What is intrinsic interpretability?

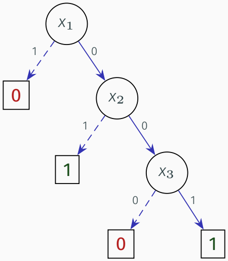
- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]



- What is an explanation for $((0,0,1),1)$?
- Clearly, IF $\neg x_1 \land \neg x_2 \land x_3$ THEN $\kappa(\mathbf{x}) = 1$
  - $\{\neg x_1, \neg x_2, x_3\}$ or $\{1,2,3\}$ is an explanation **Really?**

What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]

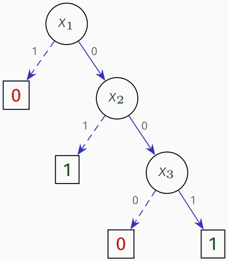


- What is an explanation for $((0,0,1),1)$?
- Clearly, IF $\neg x_1 \wedge \neg x_2 \wedge x_3$ THEN $\kappa(\mathbf{x}) = 1$
  - $\{\neg x_1, \neg x_2, x_3\}$ or $\{1,2,3\}$ is a weak explanation!
- It is the case that: IF $\neg x_1 \wedge x_3$ THEN $\kappa(\mathbf{x}) = 1$

  $\therefore$ $\{1,3\}$ is also sufficient for the prediction!

What is intrinsic interpretability?

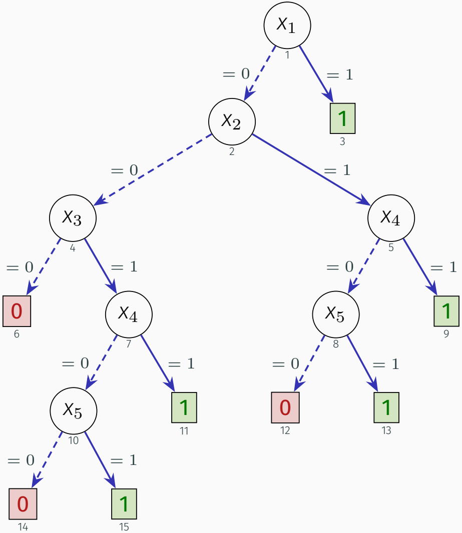
- Goal is to deploy *interpretable* ML models [Rud19, Mol20, RCC+22, Rud22]
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But**: definition of interpretability is rather subjective... [Lip18]



- What is an explanation for $((0, 0, 1), 1)$?
- Clearly, IF $\neg x_1 \wedge \neg x_2 \wedge x_3$ THEN $\kappa(\mathbf{x}) = 1$
  - $\{\neg x_1, \neg x_2, x_3\}$ or $\{1, 2, 3\}$ is a weak explanation!
- It is the case that: IF $\neg x_1 \wedge x_3$ THEN $\kappa(\mathbf{x}) = 1$

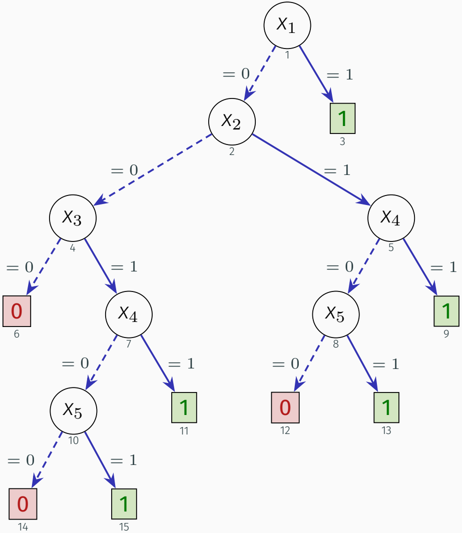
  $\therefore$ $\{1, 3\}$ is also sufficient for the prediction!
  - $\{1, 3\}$ is easier to grasp; also, it is irreducible

Are *interpretable* models really interpretable? – DTs

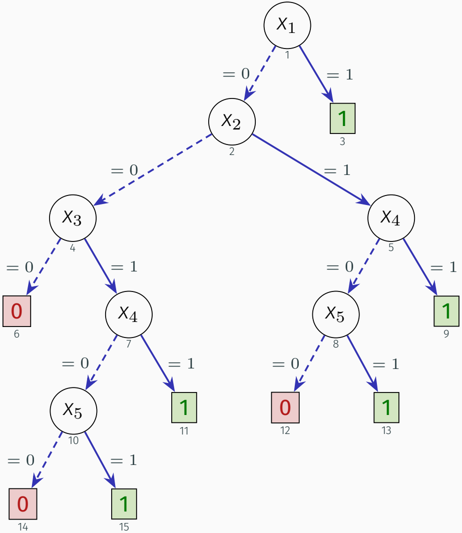


- Case of optimal decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?

Are *interpretable* models really interpretable? – DTs



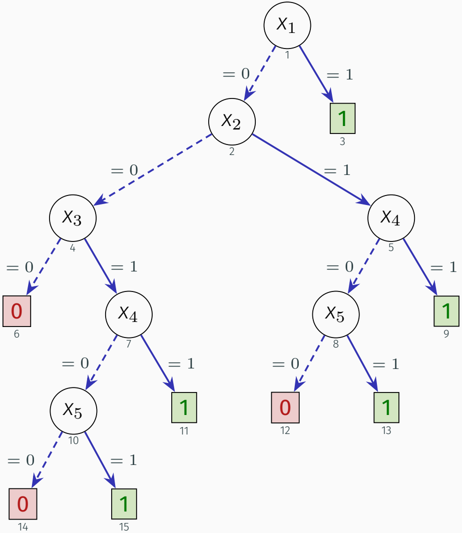
- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?
  - Clearly, IF $\neg x_1 \land \neg x_2 \land x_3 \land \neg x_4 \land x_5$ THEN $\kappa(\mathbf{x}) = 1$



- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?
  - Clearly, IF $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5$ THEN $\kappa(\mathbf{x}) = 1$
  - But, x_1, x_2, x_4 are irrelevant for the prediction:

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

Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for $(0, 0, 1, 0, 1)$, with prediction 1?
  - Clearly, IF $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5$ THEN $\kappa(\mathbf{x}) = 1$
  - But, x_1, x_2, x_4 are irrelevant for the prediction:

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

$\therefore$ fixing $\{3, 5\}$ suffices for the prediction
Compare with $\{1, 2, 3, 4, 5\}$...

Are *interpretable* models really interpretable? – DLs

[MSI23]

R_1 :	IF	$(x_1 \land x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_2 :	ELSE IF	$(x_2 \land x_4 \land x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_3 :	ELSE IF	$(\neg x_1 \land x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_4 :	ELSE IF	$(x_4 \land x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_5 :	ELSE IF	$(\neg x_1 \land \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_6 :	ELSE IF	(x_6)	THEN	$\kappa(\mathbf{x}) = 0$
$\mathsf{R}_{\mathsf{DEF}}$:	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance: $((0, 1, 0, 1, 0, 1), 0)$, i.e. rule R_2 fires

Are *interpretable* models really interpretable? – DLs

[MSI23]

R_1 :	IF	$(x_1 \land x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_2 :	ELSE IF	$(x_2 \land x_4 \land x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_3 :	ELSE IF	$(\neg x_1 \land x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_4 :	ELSE IF	$(x_4 \land x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_5 :	ELSE IF	$(\neg x_1 \land \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_6 :	ELSE IF	(x_6)	THEN	$\kappa(\mathbf{x}) = 0$
$\mathsf{R}_{\mathsf{DEF}}$:	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance: $((0,1,0,1,0,1),0)$, i.e. rule R_2 fires
- What is an explanation for the prediction?

Are *interpretable* models really interpretable? – DLs

[MSI23]

R_1 :	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_2 :	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_3 :	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_4 :	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_5 :	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_6 :	ELSE IF	(x_6)	THEN	$\kappa(\mathbf{x}) = 0$
$\mathrm{R}_{\mathrm{DEF}}$:	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance: $((0, 1, 0, 1, 0, 1), 0)$, i.e. rule R_2 fires
- What is an explanation for the prediction?
- Fixing $\{3, 4, 6\}$ suffices for the prediction

Are *interpretable* models really interpretable? – DLs

[MSI23]

R_1 :	IF	$(x_1 \land x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_2 :	ELSE IF	$(x_2 \land x_4 \land x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_3 :	ELSE IF	$(\neg x_1 \land x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_4 :	ELSE IF	$(x_4 \land x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_5 :	ELSE IF	$(\neg x_1 \land \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_6 :	ELSE IF	(x_6)	THEN	$\kappa(\mathbf{x}) = 0$
$\mathrm{R}_{\mathrm{DEF}}$:	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance: $((0,1,0,1,0,1),0)$, i.e. rule R_2 fires
- What is an explanation for the prediction?
- Fixing $\{3,4,6\}$ suffices for the prediction
  - Why?
    - We need 3 (or 1) so that R_1 cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then R_4 is guaranteed to fire

Are *interpretable* models really interpretable? – DLs

[MSI23]

R_1 :	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_2 :	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_3 :	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_4 :	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_5 :	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_6 :	ELSE IF	(x_6)	THEN	$\kappa(\mathbf{x}) = 0$
R_{DEF} :	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance: $((0,1,0,1,0,1),0)$, i.e. rule R_2 fires
- What is an explanation for the prediction?
- Fixing $\{3,4,6\}$ suffices for the prediction
  - Why?
    - We need 3 (or 1) so that R_1 cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then R_4 is guaranteed to fire
  - Some questions:
    - Would average human decision maker be able to understand the irreducible set $\{3,4,6\}$?

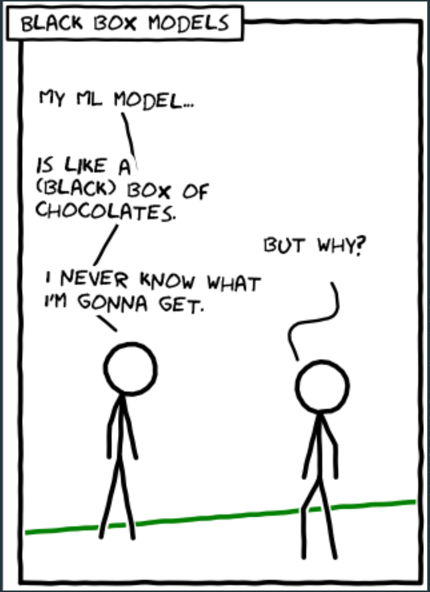
Are *interpretable* models really interpretable? – DLs

[MSI23]

R_1 :	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_2 :	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_3 :	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_4 :	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
R_5 :	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
R_6 :	ELSE IF	(x_6)	THEN	$\kappa(\mathbf{x}) = 0$
$\mathsf{R}_{\mathsf{DEF}}$:	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance: $((0,1,0,1,0,1),0)$, i.e. rule R_2 fires
- What is an explanation for the prediction?
- Fixing $\{3,4,6\}$ suffices for the prediction
  - **Why?**
    - We need 3 (or 1) so that R_1 cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then R_4 is guaranteed to fire
  - **Some questions**:
    - Would average human decision maker be able to understand the irreducible set $\{3,4,6\}$?
    - Would he/she be able to compute the set $\{3,4,6\}$, by manual inspection?

Questions?



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

References i

[BBM+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek.
On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation.
PloS one, 10(7):e0130140, 2015.

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

[EU21a] EU.
European Artificial Intelligence Act.
`https://eur-lex.europa.eu/eli/reg/2024/1689/oj`, 2021.

[EU21b] EU.
European Artificial Intelligence Act – Proposal.
`https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1623335154975&uri=CELEX%3A52021PC0206`, 2021.

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

References ii

[HM23] Xuanxiang Huang and João Marques-Silva.
The inadequacy of Shapley values for explainability.
CoRR, abs/2302.08160, 2023.

[HMS24] Xuanxiang Huang and Joao Marques-Silva.
On the failings of Shapley values for explainability.
International Journal of Approximate Reasoning, page 109112, 2024.

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

[IMM18] Alexey Ignatiev, António Morgado, and João Marques-Silva.
PySAT: A python toolkit for prototyping with SAT oracles.
In *SAT*, pages 428–437, 2018.

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

[JKMC16] Mikolás Janota, William Klieber, Joao Marques-Silva, and Edmund M. Clarke.
Solving QBF with counterexample guided refinement.
Artif. Intell., 234:1–25, 2016.

[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.

[LEC+20] Scott M. Lundberg, Gabriel G. Erion, Hugh Chen, Alex J. DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee.
From local explanations to global understanding with explainable AI for trees.
Nat. Mach. Intell., 2(1):56–67, 2020.

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

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

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

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

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

[Mar24] Joao Marques-Silva.
Logic-based explainability: Past, present & future.
CoRR, abs/2406.11873, 2024.

[MHL+13] António Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and Joao Marques-Silva.
Iterative and core-guided MaxSA solving: A survey and assessment.
Constraints, 18(4):478–534, 2013.

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

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

References v

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

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

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

[MSH24] Joao Marques-Silva and Xuanxiang Huang.
Explainability is *Not* a game.
Commun. ACM, 67(7):66–75, jul 2024.

[MSI23] Joao Marques-Silva and Alexey Ignatiev.
No silver bullet: interpretable ml models must be explained.
Frontiers in Artificial Intelligence, 6, 2023.

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

[PG86] David A. Plaisted and Steven Greenbaum.
A structure-preserving clause form translation.
J. Symb. Comput., 2(3):293–304, 1986.

[RCC+22] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong.
Interpretable machine learning: Fundamental principles and 10 grand challenges.
Statistics Surveys, 16:1–85, 2022.

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

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

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

[Rud22] Cynthia Rudin.
Why black box machine learning should be avoided for high-stakes decisions, in brief.
Nature Reviews Methods Primers, 2(1):1–2, 2022.

[Tse68] G.S. Tseitin.
On the complexity of derivations in the propositional calculus.
In H.A.O. Slesenko, editor, *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pages 115–125, 1968.