Logic-Based Explainable Artificial Intelligence

Joao Marques-Silva

ICREA, Univ. Lleida, Catalunya, Spain

ESSAI, Athens, Greece, July 2024

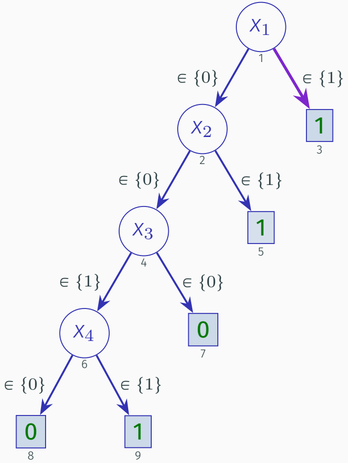
Lecture 04

Recapitulate third lecture

- Logic encoding for explaining DLs
  - And status of (in)tractability in logic-based XAI
- Query: enumeration of explanations
- Query: feature necessity, AXp & CXp
- Query: feature relevancy

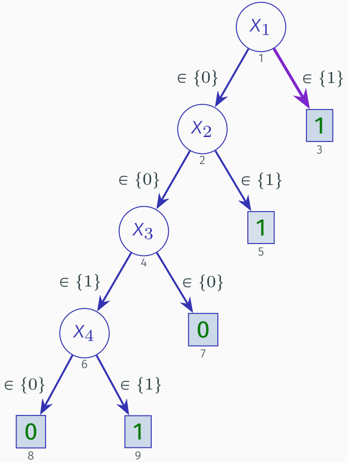
Recap example

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



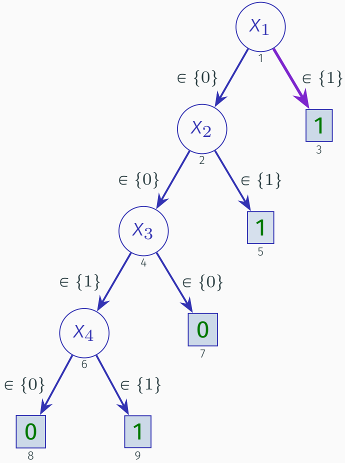
Recap example

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



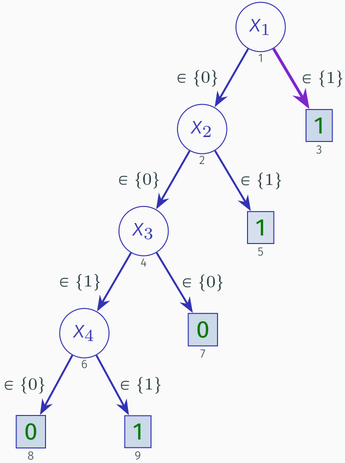
Recap example

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



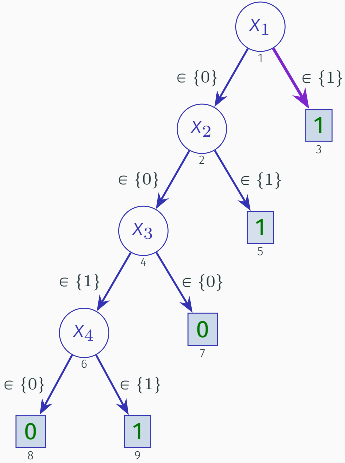
Recap example

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



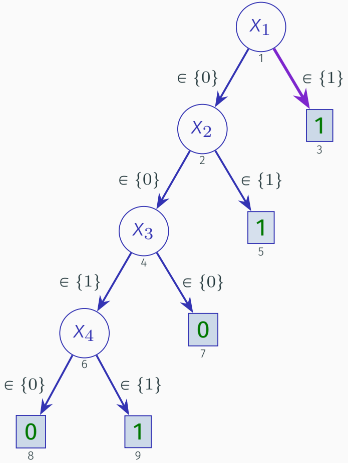
Recap example

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



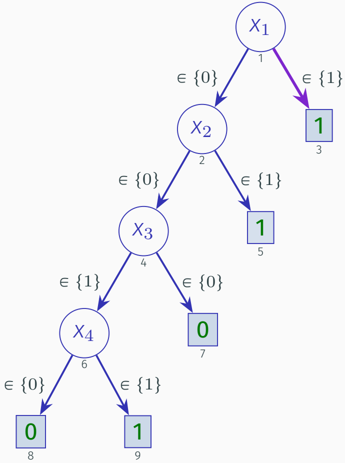
Recap example

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



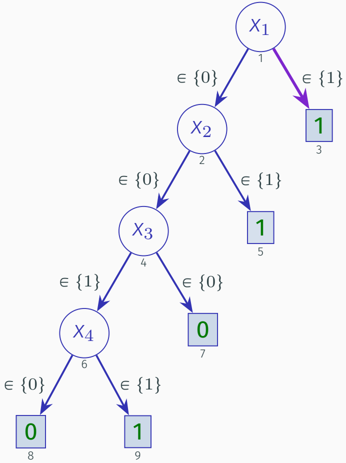
Recap example

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



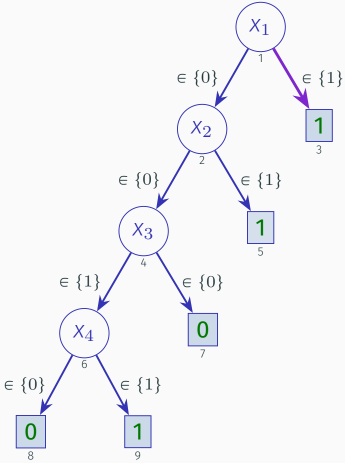
Recap example

- Instance $(\mathbf{v}, c) = ((0,0,0,0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$?
  - Yes! Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$?
  - No! Thus, feature 3 is not AXp-necessary
- Are there CXp-necessary features?
  - No! There are no singleton AXps



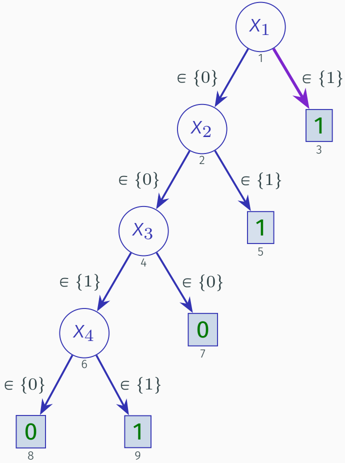
Recap example

- Instance $(\mathbf{v}, c) = ((0,0,0,0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$?
  - **Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$?
  - **No!** Thus, feature 3 is not AXp-necessary
- Are there CXp-necessary features?
  - **No!** There are no singleton AXps
- Confirmation:



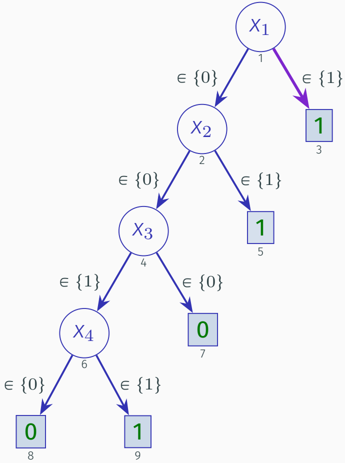
Recap example

- Instance $(\mathbf{v}, c) = ((0,0,0,0),0)$
- Is feature 1 AXp-necessary?
  - Does there exist u_1, such that $\kappa(u_1,0,0,0) \neq \kappa(0,0,0,0)$?
  - Yes! Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist u_3, such that $\kappa(0,0,u_3,0) \neq \kappa(0,0,0,0)$?
  - No! Thus, feature 3 is not AXp-necessary
- Are there CXp-necessary features?
  - No! There are no singleton AXps
- Confirmation:
  - CXps: $\{\{1\},\{2\},\{3,4\}\}$ (2 is also AXp-necessary)
  - AXps: $\{\{1,2,3\},\{1,2,4\}\}$



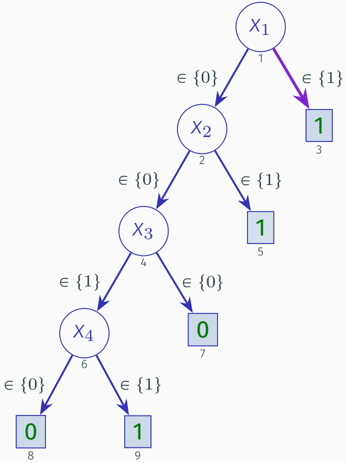
Recap example – a different instance

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



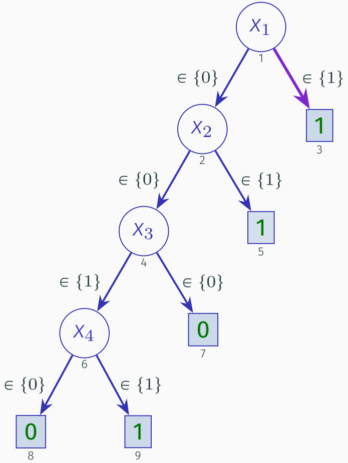
Recap example – a different instance

- Instance $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Are there CXp-necessary features?



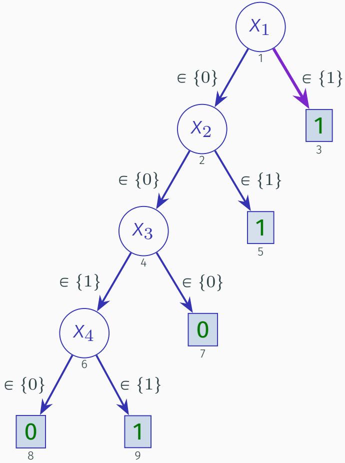
Recap example – a different instance

- Instance $(\mathbf{v}, c) = ((1,1,1,1),1)$
- Are there CXp-necessary features?
  - Yes! Features 1 and 2 (i.e. singleton AXps)



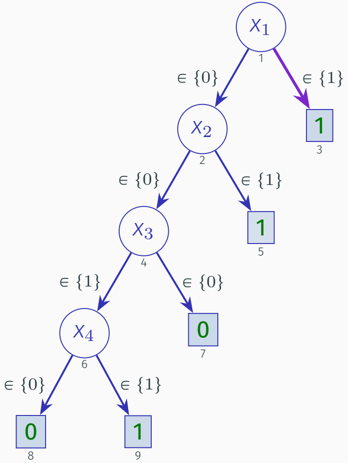
Recap example – a different instance

- Instance $(\mathbf{v}, c) = ((1,1,1,1),1)$
- Are there CXp-necessary features?
    - **Yes!** Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?



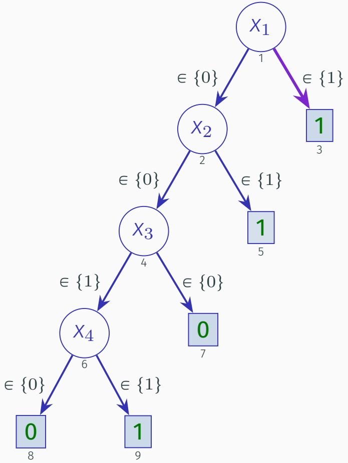
Recap example – a different instance

- Instance $(\mathbf{v}, c) = ((1,1,1,1),1)$
- Are there CXp-necessary features?
  - Yes! Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?
  - No! There are no singleton CXps



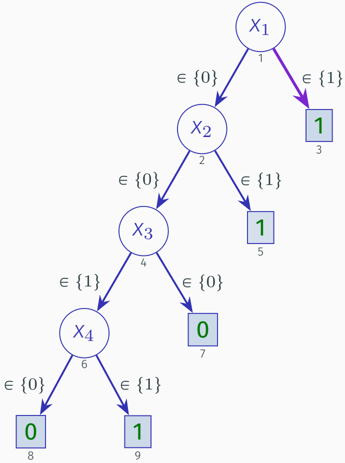
Recap example – a different instance

- Instance $(\mathbf{v}, c) = ((1,1,1,1),1)$
- Are there CXp-necessary features?
  - Yes! Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?
  - No! There are no singleton CXps
- Confirmation:



Recap example – a different instance

- Instance $(\mathbf{v}, c) = ((1,1,1,1),1)$
- Are there CXp-necessary features?
  - **Yes!** Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?
  - **No!** There are no singleton CXps
- Confirmation:
  - AXps: $\{\{1\},\{2\},\{3,4\}\}$
  - CXps: $\{\{1,2,3\},\{1,2,4\}\}$



Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

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

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1, 2, 3, 4, 5\}$; $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 5$; $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1, 1, 1, 1, 1), 1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1, 2, 3, 4, 5\}$; $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 5$; $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1, 1, 1, 1, 1), 1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table
- All AXps:

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table
- All AXps: $\{\{1,2\},\{1,3\}\}$
- All CXps:

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table
- All AXps: $\{\{1,2\},\{1,3\}\}$
- All CXps: $\{\{1\},\{2,3\}\}$
- AXp-necessary:

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1, 2, 3, 4, 5\}$; $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 5$; $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1, 1, 1, 1, 1), 1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table
- All AXps: $\{\{1, 2\}, \{1, 3\}\}$
- All CXps: $\{\{1\}, \{2, 3\}\}$
- AXp-necessary: $\{1\}$ (singleton CXp)
- CXp-necessary:

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table
- All AXps: $\{\{1,2\},\{1,3\}\}$
- All CXps: $\{\{1\},\{2,3\}\}$
- AXp-necessary: $\{1\}$ (singleton CXp)
- CXp-necessary: $\emptyset$
- Relevant:

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table
- All AXps: $\{\{1,2\},\{1,3\}\}$
- All CXps: $\{\{1\},\{2,3\}\}$
- AXp-necessary: $\{1\}$ (singleton CXp)
- CXp-necessary: $\emptyset$
- Relevant: $\{1,2,3\}$
- Irrelevant:

Another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must consider only $x_1 = 1$
  - **Hint:** Can construct restricted truth-table
- All AXps: $\{\{1,2\},\{1,3\}\}$
- All CXps: $\{\{1\},\{2,3\}\}$
- AXp-necessary: $\{1\}$ (singleton CXp)
- CXp-necessary: $\varnothing$
- Relevant: $\{1,2,3\}$
- Irrelevant: $\{4,5\}$

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1, 2, 3, 4, 5\}$; $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 5$; $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

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

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables
- All AXps:

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1, 2, 3, 4, 5\}$; $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 5$; $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1, 1, 1, 1, 1), 1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables
- All AXps: $\{\{1\}, \{2, 3\}\}$
- All CXps:

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1, 2, 3, 4, 5\}$; $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 5$; $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1, 1, 1, 1, 1), 1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables
- All AXps: $\{\{1\}, \{2, 3\}\}$
- All CXps: $\{\{1, 2\}, \{1, 3\}\}$
- AXp-necessary:

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1, 2, 3, 4, 5\}$; $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 5$; $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1, 1, 1, 1, 1), 1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables
- All AXps: $\{\{1\}, \{2, 3\}\}$
- All CXps: $\{\{1, 2\}, \{1, 3\}\}$
- AXp-necessary: $\emptyset$
- CXp-necessary:

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables
- All AXps: $\{\{1\},\{2,3\}\}$
- All CXps: $\{\{1,2\},\{1,3\}\}$
- AXp-necessary: $\emptyset$
- CXp-necessary: $\{1\}$ (singleton AXp)
- Relevant:

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables
- All AXps: $\{\{1\},\{2,3\}\}$
- All CXps: $\{\{1,2\},\{1,3\}\}$
- AXp-necessary: $\emptyset$
- CXp-necessary: $\{1\}$ (singleton AXp)
- Relevant: $\{1,2,3\}$
- Irrelevant:

Yet another example – feature necessity & relevancy

- Classifier: $\mathcal{F} = \{1,2,3,4,5\}$; $\mathcal{D}_i = \{0,1\}$, $i = 1,\ldots,5$; $\mathcal{K} = \{0,1\}$

$$\kappa(x_1,x_2,x_3,x_4,x_5) \quad := \quad \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geqslant 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $((1,1,1,1,1),1)$
- **Obs:** If $x_1 = 0$ and $x_2 = x_3 = 0$, then $\kappa(\mathbf{x}) = 0$; i.e. must either set $x_1 = 1$ or $x_2 = x_3 = 1$
  - **Hint:** Can construct restricted truth-tables
- All AXps: $\{\{1\},\{2,3\}\}$
- All CXps: $\{\{1,2\},\{1,3\}\}$
- AXp-necessary: $\emptyset$
- CXp-necessary: $\{1\}$ (singleton AXp)
- Relevant: $\{1,2,3\}$
- Irrelevant: $\{4,5\}$

Some use cases

Some use cases

Q: How to decide whether some protected feature occurs in some explanation?

Some use cases

Q: How to decide whether some protected feature occurs in some explanation?

- Decide feature relevancy

Some use cases

Q: How to decide whether some protected feature occurs in some explanation?

- Decide feature relevancy

Q: How to decide whether some protected feature occurs in all explanations?

Some use cases

Q: How to decide whether some protected feature occurs in some explanation?

- Decide feature relevancy

Q: How to decide whether some protected feature occurs in all explanations?

- Decide feature necessity

Some use cases

Q: How to decide whether some protected feature occurs in some explanation?

- Decide feature relevancy

Q: How to decide whether some protected feature occurs in all explanations?

- Decide feature necessity

Q: What can we do if human decision maker finds computed AXp/CXp to be unsatisfactory?

Some use cases

Q: How to decide whether some protected feature occurs in some explanation?

- Decide feature relevancy

Q: How to decide whether some protected feature occurs in all explanations?

- Decide feature necessity

Q: What can we do if human decision maker finds computed AXp/CXp to be unsatisfactory?

- Partially enumerate AXps/CXps, exploiting bias in enumeration

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

Detour: Monotonic Classification & Voting Power

Monotonically increasing boolean classifiers

Monotonically increasing boolean classifiers

- Monotonic classifier $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$, such that each $\mathcal{D}_i = \{0, 1\}$ and $\mathcal{K} = \{0, 1\}$ are ordered (i.e. $0 < 1$), and
  - $\kappa(\mathbf{1}) = 1$;
  - Non-constant classifier, i.e. $\kappa(\mathbf{0}) = 0$; and
  - $\kappa(\mathbf{x}_1) \leqslant \kappa(\mathbf{x}_2)$ when $\mathbf{x}_1 \leqslant \mathbf{x}_2$

Monotonically increasing boolean classifiers

- Monotonic classifier $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$, such that each $\mathcal{D}_i = \{0, 1\}$ and $\mathcal{K} = \{0, 1\}$ are ordered (i.e. $0 < 1$), and
  - $\kappa(\mathbf{1}) = 1$;
  - Non-constant classifier, i.e. $\kappa(\mathbf{0}) = 0$; and
  - $\kappa(\mathbf{x}_1) \leqslant \kappa(\mathbf{x}_2)$ when $\mathbf{x}_1 \leqslant \mathbf{x}_2$
- Let $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}$ be such that $\kappa(\mathbf{v}_1) = \kappa(\mathbf{v}_2) = 1$, and $\mathbf{v}_1 \leqslant \mathbf{v}_2$
  Define the explanation problems:
  - $\mathcal{E}_1 = (\mathcal{M}, (\mathbf{v}_1, 1))$
  - $\mathcal{E}_2 = (\mathcal{M}, (\mathbf{v}_2, 1))$
  - $\mathcal{E}_{\mathbb{1}} = (\mathcal{M}, ((1, \ldots, 1), 1)) = (\mathcal{M}, (\mathbb{1}, 1))$

Monotonically increasing boolean classifiers

- Monotonic classifier $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$, such that each $\mathcal{D}_i = \{0, 1\}$ and $\mathcal{K} = \{0, 1\}$ are ordered (i.e. $0 < 1$), and
  - $\kappa(\mathbf{1}) = 1$;
  - Non-constant classifier, i.e. $\kappa(\mathbf{0}) = 0$; and
  - $\kappa(\mathbf{x}_1) \leqslant \kappa(\mathbf{x}_2)$ when $\mathbf{x}_1 \leqslant \mathbf{x}_2$
- Let $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}$ be such that $\kappa(\mathbf{v}_1) = \kappa(\mathbf{v}_2) = 1$, and $\mathbf{v}_1 \leqslant \mathbf{v}_2$
  Define the explanation problems:
  - $\mathcal{E}_1 = (\mathcal{M}, (\mathbf{v}_1, 1))$
  - $\mathcal{E}_2 = (\mathcal{M}, (\mathbf{v}_2, 1))$
  - $\mathcal{E}_{\mathbb{1}} = (\mathcal{M}, ((1, \ldots, 1), 1)) = (\mathcal{M}, (\mathbb{1}, 1))$
- Then,
  - If WAXp$(\mathcal{S}; \mathcal{E}_1)$ holds, then WAXp$(\mathcal{S}; \mathcal{E}_2)$ holds; in particular:
  - $\mathbb{A}(\mathcal{E}_{\mathbb{1}})$ contains **all** the AXps of **any** instance of the form $(\mathbf{v}_r, 1)$
    - Why?
      - Pick any explanation problem $\mathcal{E}_r$ with instance $(\mathbf{v}_r, 1)$
      - Start from $\mathbb{1} = (1, 1, \ldots, 1)$
      - Remove features that take value 0 in $\mathbf{v}_r$; we still have an WAXp
      - Then compute any AXp starting from features taking value 1 in $\mathbf{v}_r$

      $\therefore$ **Suffices to find explanations for $\mathcal{E}_{\mathbb{1}}$** (or alternatively, the global explanations for prediction 1)

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function

- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function

- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function

- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function

- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function

- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function

- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function
- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep
    - Feature 6: can be dropped

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function
- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep
    - Feature 6: can be dropped
    - AXp: $\{2, 3, 4, 5\}$

An example

- ML model $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$:
  - Boolean classifier: $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features: $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$
    - I.e. $\mathcal{D}_i = \{0, 1\}$, $i = 1, \ldots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

    - κ is a monotonically increasing boolean function
- We are interested in identifying the AXps of $\mathcal{M}$, given the instance $((1, 1, 1, 1, 1, 1), 1)$
  - Or alternatively, the global AXps for prediction 1
  - For example, with order $\langle 1, 2, 3, 4, 5, 6 \rangle$:
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep
    - Feature 6: can be dropped
    - AXp: $\{2, 3, 4, 5\}$; Q: Is feature 6 relevant?

All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $(\mathbb{1}, 1)$

All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $(\mathbb{1}, 1)$
- Computing the AXps:
  - Must pick 2 out of features $\{1, 2, 3\}$
  - If only 2 out of features $\{1, 2, 3\}$ picked, then we must pick both features 4 and 5
  - Feature 6 is never matters, i.e. it is irrelevant...

All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $(\mathbb{1}, 1)$
- Computing the AXps:
  - Must pick 2 out of features $\{1, 2, 3\}$
  - If only 2 out of features $\{1, 2, 3\}$ picked, then we must pick both features 4 and 5
  - Feature 6 is never matters, i.e. it is irrelevant...
- AXps:

All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $(\mathbb{1}, 1)$
- Computing the AXps:
  - Must pick 2 out of features $\{1, 2, 3\}$
  - If only 2 out of features $\{1, 2, 3\}$ picked, then we must pick both features 4 and 5
  - Feature 6 is never matters, i.e. it is irrelevant...
- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}\}$$

All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $(\mathbb{1}, 1)$
- Computing the AXps:
  - Must pick 2 out of features $\{1, 2, 3\}$
  - If only 2 out of features $\{1, 2, 3\}$ picked, then we must pick both features 4 and 5
  - Feature 6 is never matters, i.e. it is irrelevant...
- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}\}$$

- CXps:

All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance: $(\mathbb{1}, 1)$
- Computing the AXps:
  - Must pick 2 out of features $\{1, 2, 3\}$
  - If only 2 out of features $\{1, 2, 3\}$ picked, then we must pick both features 4 and 5
  - Feature 6 is never matters, i.e. it is irrelevant...
- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}\}$$

- CXps:

$$\mathbb{C} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 4\}, \{1, 5\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}\}$$

What is a priori voting power?

- General set-up of weighted voting games:

What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly $\mathcal{A}$ of voters, with $m = |\mathcal{A}|$
  - Each voter $i \in \mathcal{A}$ votes Yes with n_i votes; otherwise no votes are counted (and he/she votes No)

What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly $\mathcal{A}$ of voters, with $m = |\mathcal{A}|$
  - Each voter $i \in \mathcal{A}$ votes Yes with n_i votes; otherwise no votes are counted (and he/she votes No)
  - A coalition is a subset of voters, $\mathcal{C} \subseteq \mathcal{A}$
  - Quota q is the sum of votes required for a proposal to be approved
    - Coalitions leading to sums not less than q are **winning** coalitions

What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly $\mathcal{A}$ of voters, with $m = |\mathcal{A}|$
  - Each voter $i \in \mathcal{A}$ votes Yes with n_i votes; otherwise no votes are counted (and he/she votes No)
  - A coalition is a subset of voters, $\mathcal{C} \subseteq \mathcal{A}$
  - Quota q is the sum of votes required for a proposal to be approved
    - Coalitions leading to sums not less than q are **winning** coalitions
  - A weighted voting game (WVG) is a tuple $[q; n_1, \ldots, n_m]$
    - Example: $[12; 4, 4, 4, 2, 2, 1]$

What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly $\mathcal{A}$ of voters, with $m = |\mathcal{A}|$
  - Each voter $i \in \mathcal{A}$ votes Yes with n_i votes; otherwise no votes are counted (and he/she votes No)
  - A coalition is a subset of voters, $\mathcal{C} \subseteq \mathcal{A}$
  - Quota q is the sum of votes required for a proposal to be approved
    - Coalitions leading to sums not less than q are **winning** coalitions
  - A weighted voting game (WVG) is a tuple $[q; n_1, \ldots, n_m]$
    - Example: $[12; 4, 4, 4, 2, 2, 1]$
  - Problem: find a measure of importance of each voter !
    - I.e. measure the a priori voting power of each voter

An example – EEC (EU) members voting power in 1958

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:	12	

An example – EEC (EU) members voting power in 1958

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:	12	

- WVG: $[12; 4, 4, 4, 2, 2, 1]$

An example – EEC (EU) members voting power in 1958

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:	12	

- WVG: $[12; 4, 4, 4, 2, 2, 1]$
- **Q**: What should be the voting power of Luxembourg?

An example – EEC (EU) members voting power in 1958

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:	12	

- WVG: $[12; 4, 4, 4, 2, 2, 1]$
- **Q**: What should be the voting power of Luxembourg?
- Can Luxembourg (L) ***matter*** for some winning coalition?

An example – EEC (EU) members voting power in 1958

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:	12	

- WVG: $[12; 4, 4, 4, 2, 2, 1]$
- **Q**: What should be the voting power of Luxembourg?
- Can Luxembourg (L) ***matter*** for some winning coalition?
- Perhaps surprisingly, answer is **No**!
  - In 1958, Luxembourg was a **dummy** voter/player

Understanding weighted voting games

- Obs: A WVG is a monotonically increasing boolean classifier
- Each subset-minimal winning coalition is an AXp of the instance $(\mathbb{1}, 1)$

Understanding weighted voting games

- **Obs:** A WVG is a monotonically increasing boolean classifier
- Each subset-minimal winning coalition is an AXp of the instance $(\mathbb{1}, 1)$
- Recall EEC voting example:

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
	Quota:	12

Understanding weighted voting games

- Obs: A WVG is a monotonically increasing boolean classifier
- Each subset-minimal winning coalition is an AXp of the instance $(\mathbb{1}, 1)$
- Recall EEC voting example:

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:	12	

- The corresponding classifier is:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) \quad := \quad \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geqslant 12) \\ 0 & \text{otherwise} \end{cases}$$

which we have seen before! E.g. $\{2, 3, 4, 5\}$ is an AXp & feature 6 (L) is irrelevant

Another example

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]

Another example

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7

Another example

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

Another example

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

$$\mathbb{A} = \{\{1,2\},\{1,3,4,5\},\{1,3,4,6\},\{1,3,4,7\}\}$$

Another example

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

$$\mathbb{A} = \{\{1,2\},\{1,3,4,5\},\{1,3,4,6\},\{1,3,4,7\}\}$$

- CXps:

Another example

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

$$\mathbb{A} = \{\{1,2\},\{1,3,4,5\},\{1,3,4,6\},\{1,3,4,7\}\}$$

- CXps:

$$\mathbb{C} = \{\{1\},\{2,3\},\{2,4\},\{2,5,6,7\}\}$$

Another example

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

$$\mathbb{A} = \{\{1,2\},\{1,3,4,5\},\{1,3,4,6\},\{1,3,4,7\}\}$$

- CXps:

$$\mathbb{C} = \{\{1\},\{2,3\},\{2,4\},\{2,5,6,7\}\}$$

- **Q:** How should features be ranked in terms of importance?

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1,2\},\{1,3\},\{2,3\}\}$$

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

- CXps:

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1,2\},\{1,3\},\{2,3\}\}$$

- CXps:

$$\mathbb{C} = \{\{1,2\},\{1,3\},\{2,3\}\}$$

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

- CXps:

$$\mathbb{C} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

- **Obs:** features (resp. voters) 4, 5 and 6 are irrelevant (resp. dummy)

Yet another example

- WVG: [16; 9, 9, 7, 3, 1, 1]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1,2\},\{1,3\},\{2,3\}\}$$

- CXps:

$$\mathbb{C} = \{\{1,2\},\{1,3\},\{2,3\}\}$$

- **Obs:** features (resp. voters) 4, 5 and 6 are irrelevant (resp. dummy)
- **Q:** How should features be ranked in terms of importance?

Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws (more tomorrow) [MSH24, HMS24, HM23b]

Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws (more tomorrow) [MSH24, HMS24, HM23b]
- Recently, we have devised ways of correcting SHAP scores [LHMS24]

Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws (more tomorrow) [MSH24, HMS24, HM23b]
- Recently, we have devised ways of correcting SHAP scores [LHMS24]
- In turn, this revealed novel connections between logic-based XAI and a priori voting power [LHAMS24]

Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws (more tomorrow) [MSH24, HMS24, HM23b]
- Recently, we have devised ways of correcting SHAP scores [LHMS24]
- In turn, this revealed novel connections between logic-based XAI and a priori voting power [LHAMS24]
- Homework:
  - Create your own weighted voting games;
  - Compute the sets of AXps and CXps; and
  - Assess the importance of features and how they compare to each other

Unit #06

Advanced Topics

General definition of prediction sufficiency

- Instance $(\mathbf{v}, c)$
- Let $\mathcal{S} \subseteq \mathcal{F}$:
  - Recall,

$$\Upsilon(\mathcal{S};\mathbf{v}) = \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

- $\mathcal{S} \subseteq \mathcal{F}$ suffices for prediction c if:

$$\forall(\mathbf{x} \in \mathbb{F}).(\mathbf{x} \in \Upsilon(\mathcal{S};\mathbf{v})) \rightarrow (\sigma(\mathbf{x}))$$

  - **Obs:** a WAXp is just one possible example
- But there are other ways to study prediction sufficiency:
  - One can envision defining other sets of points Γ, parameterized by $\mathcal{E} = (\mathcal{M}, (\mathbf{v}, c))$; $\mathcal{S} \subseteq \mathcal{F}$ suffices for prediction c if:

$$\forall(\mathbf{x} \in \mathbb{F}).(\mathbf{x} \in \Gamma(\mathcal{S};\mathcal{E})) \rightarrow (\sigma(\mathbf{x}))$$

    - And one can also envision generalizations of σ!

Outline – Unit #06

[IISM24]

- Recall:

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

  - For non-boolean features, use of $=$ may convey little information, e.g. with real-valued features, having $x_1 = 1.157$ does not help in understanding what values of feature 1 are also acceptable

Towards more expressive explanations – inflated explanations

[IISM24]

- Recall:

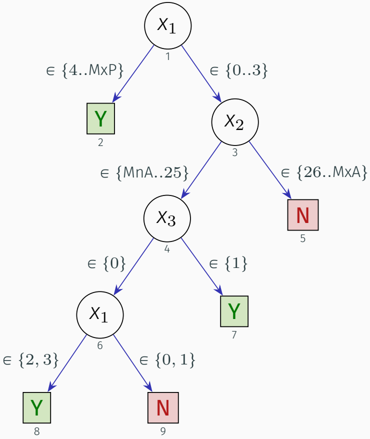
$$\mathrm{WAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

  - For non-boolean features, use of $=$ may convey little information, e.g. with real-valued features, having $x_1 = 1.157$ does not help in understanding what values of feature 1 are also acceptable

- Inflated explanations allow for more expressive literals, i.e. $=$ replaced with $\in$, and individual values replaced by ranges of values
  - Operational definition: Given an AXp, expand set of values of each feature, in some chosen order, such that the set of picked features remains unchanged

Inflated explanations – an example

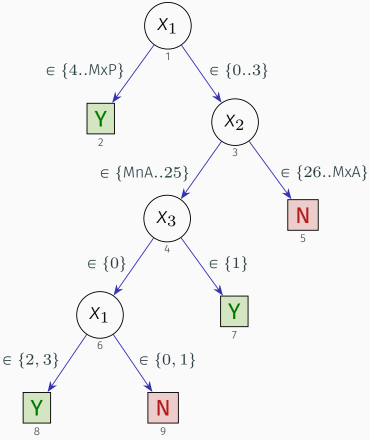
[IIM22]



- Explanation for $((2, 20, 0), \text{Y})$? (Obs: $\text{MnA} = 18; \text{MxP} > 4$)

Inflated explanations – an example

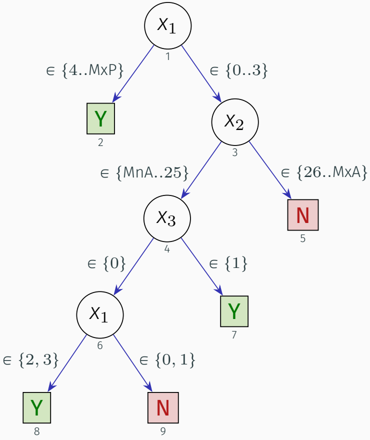
[IIM22]



- Explanation for $((2, 20, 0), \text{Y})$? (Obs: $\text{MnA} = 18; \text{MxP} > 4$)
  - AXp: $\{1, 2\}$

Inflated explanations – an example

[IIM22]

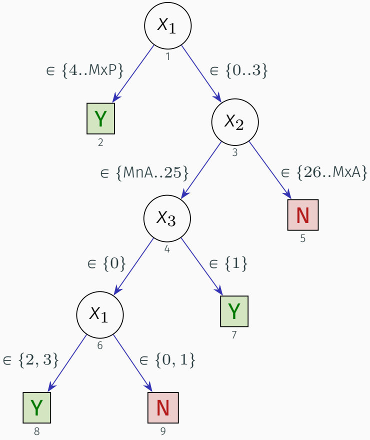


- Explanation for $((2,20,0),\mathsf{Y})$? (Obs: $\mathsf{MnA}=18; \mathsf{MxP}>4$)
  - AXp: $\{1,2\}$
  - Default interpretation:

$$\forall(\mathbf{x}\in\mathbb{F}).(x_1=2 \land x_2=20)\rightarrow(\kappa(\mathbf{x})=\mathsf{Y})$$

Inflated explanations – an example

[IIM22]



- Explanation for $((2, 20, 0), \mathsf{Y})$? (Obs: $\mathsf{MnA} = 18; \mathsf{MxP} > 4$)
  - AXp: $\{1, 2\}$
  - Default interpretation:

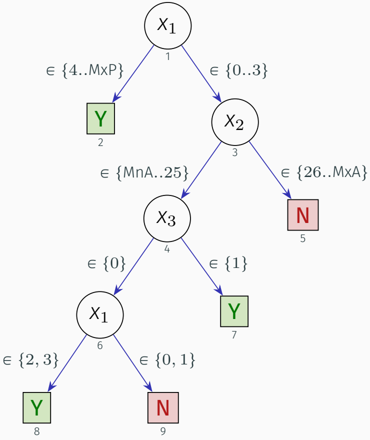
$$\forall(\mathbf{x} \in \mathbb{F}).(x_1 = 2 \land x_2 = 20) \rightarrow (\kappa(\mathbf{x}) = \mathsf{Y})$$

  - Corresponding rule:

$$\text{IF } (x_1 = 2 \land x_2 = 20) \text{ THEN } (\kappa(\mathbf{x}) = \mathsf{Y})$$

Inflated explanations – an example

[IIM22]



- Explanation for $((2, 20, 0), \text{Y})$? (Obs: $\text{MnA} = 18; \text{MxP} > 4$)
  - AXp: $\{1, 2\}$
  - Default interpretation:

$$\forall(\mathbf{x} \in \mathbb{F}).(x_1 = 2 \land x_2 = 20) \rightarrow (\kappa(\mathbf{x}) = \text{Y})$$

  - Corresponding rule:

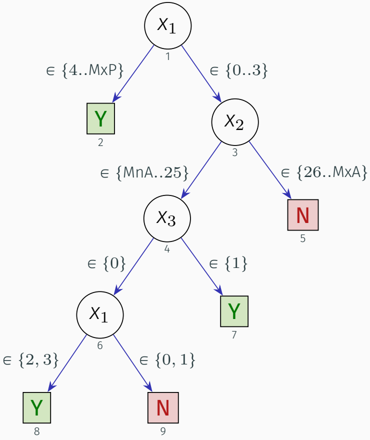
$$\text{IF } (x_1 = 2 \land x_2 = 20) \text{ THEN } (\kappa(\mathbf{x}) = \text{Y})$$

  - With inflated explanations:

$$\forall(\mathbf{x} \in \mathbb{F}).(x_1 \in \{2..\text{MxP}\} \land x_2 \in \{\text{MnA}..25\}) \rightarrow (\kappa(\mathbf{x}) = \text{Y})$$

Inflated explanations – an example

[IIM22]



- Explanation for $((2,20,0),\text{Y})$? (Obs: $\text{MnA} = 18; \text{MxP} > 4$)
  - AXp: $\{1,2\}$
  - Default interpretation:

$$\forall(\mathbf{x} \in \mathbb{F}).(x_1 = 2 \land x_2 = 20) \rightarrow (\kappa(\mathbf{x}) = \text{Y})$$

  - Corresponding rule:

$$\text{IF } (x_1 = 2 \land x_2 = 20) \text{ THEN } (\kappa(\mathbf{x}) = \text{Y})$$

  - With inflated explanations:

$$\forall(\mathbf{x} \in \mathbb{F}).(x_1 \in \{2..\text{MxP}\} \land x_2 \in \{\text{MnA}..25\}) \rightarrow (\kappa(\mathbf{x}) = \text{Y})$$

  - Corresponding rule:

$$\text{IF } (x_1 \in \{2..\text{MxP}\} \land x_2 \in \{\text{MnA}..25\}) \text{ THEN } (\kappa(\mathbf{x}) = \text{Y})$$

Approach

- Compute AXp $\mathcal{X}$
- For each feature:
  - Categorical: iteratively add elements to literal
  - Ordinal:
    - Expand literal for larger values;
    - Expand literal for smaller values

Approach

- Compute AXp $\mathcal{X}$
- For each feature:
    - Categorical: iteratively add elements to literal
    - Ordinal:
        - Expand literal for larger values;
        - Expand literal for smaller values
- **Obs:** More complex alternative is to find AXp and expand domains simultaneously
    - This is conjectured to change the complexity class of finding one explanation

Outline – Unit #06

Probabilistic (formal) explanations

[WMHK21, IIN+22, IHI+22, ABOS22, IHI+23, IMM24]

- Explanation size is critical for human understanding [Mil56]
- Probabilistic explanations provide smaller explanations, by trading off rigor of explanation by explanation size

Probabilistic (formal) explanations

[WMHK21, IIN+22, IHI+22, ABOS22, IHI+23, IMM24]

- Explanation size is critical for human understanding [Mil56]
- Probabilistic explanations provide smaller explanations, by trading off rigor of explanation by explanation size
- Definition of weak probabilistic AXp $\mathcal{X} \subseteq \mathcal{F}$:

$$\text{WPAXp}(\mathcal{X}) \quad := \quad \mathbf{Pr}(\kappa(\mathbf{x}) = c) \,|\, \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geqslant \delta$$

  - Obs: $\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}$ requires points $\mathbf{x} \in \mathbb{F}$ to match the values of $\mathbf{v}$ for the features dictated by $\mathcal{X}$
  - Obs: for $\delta = 1$ we obtain a WAXp

Definitions

Definitions

- Weak probabilistic AXp (WPAXp):

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$$

$$\mathrm{Pr}_{\mathbf{x}}(\kappa(\mathbf{x}) = c \,|\, \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geqslant \delta \; := \; \frac{|\{\mathbf{x} \in \mathbb{F} : \kappa(\mathbf{x}) = c \land (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x} \in \mathbb{F} : (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|} \geqslant \delta$$

Definitions

- Weak probabilistic AXp (WPAXp):

$$\mathsf{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) :=$$
$$\mathsf{Pr}_{\mathbf{x}}(\kappa(\mathbf{x})=c\,|\,\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}}) \geqslant \delta \;:=\; \frac{|\{\mathbf{x}\in\mathbb{F}:\kappa(\mathbf{x})=c\land(\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x}\in\mathbb{F}:(\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}})\}|} \geqslant \delta$$

- Probabilistic AXp (PAXp):

$$\mathsf{PAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) \;:=$$
$$\mathsf{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta)\land\forall(\mathcal{X}'\subsetneq\mathcal{X}).\neg\mathsf{WeakPAXp}(\mathcal{X}';\mathbb{F},\kappa,\mathbf{v},c,\delta)$$

Definitions

- Weak probabilistic AXp (WPAXp):

$$\mathsf{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) :=$$
$$\mathrm{Pr}_{\mathbf{x}}(\kappa(\mathbf{x})=c\,|\,\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}}) \geqslant \delta \;:=\; \frac{|\{\mathbf{x}\in\mathbb{F}:\kappa(\mathbf{x})=c\land(\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x}\in\mathbb{F}:(\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}})\}|} \geqslant \delta$$

- Probabilistic AXp (PAXp):

$$\mathsf{PAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) \;:=$$
$$\mathsf{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) \land \forall(\mathcal{X}'\subsetneq\mathcal{X}).\neg\mathsf{WeakPAXp}(\mathcal{X}';\mathbb{F},\kappa,\mathbf{v},c,\delta)$$

- Locally-minimal PAXp (LmPAXp):

$$\mathsf{LmPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) \;:=$$
$$\mathsf{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) \land \forall(j\in\mathcal{X}).\neg\mathsf{WeakPAXp}(\mathcal{X}\setminus\{j\};\mathbb{F},\kappa,\mathbf{v},c,\delta)$$

Definitions

- Weak probabilistic AXp (WPAXp): – definition is non-monotonic

$$\text{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) :=$$

$$\text{Pr}_{\mathbf{x}}(\kappa(\mathbf{x})=c\,|\,\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}}) \geqslant \delta \;:=\; \frac{|\{\mathbf{x}\in\mathbb{F}:\kappa(\mathbf{x})=c \land (\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x}\in\mathbb{F}:(\mathbf{x}_{\mathcal{X}}=\mathbf{v}_{\mathcal{X}})\}|} \geqslant \delta$$

- Probabilistic AXp (PAXp):

$$\text{PAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) :=$$

$$\text{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) \land \forall(\mathcal{X}'\subsetneq\mathcal{X}).\neg\text{WeakPAXp}(\mathcal{X}';\mathbb{F},\kappa,\mathbf{v},c,\delta)$$

- Locally-minimal PAXp (LmPAXp): – may differ from PAXp due to non-monotonicity

$$\text{LmPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) :=$$

$$\text{WeakPAXp}(\mathcal{X};\mathbb{F},\kappa,\mathbf{v},c,\delta) \land \forall(j\in\mathcal{X}).\neg\text{WeakPAXp}(\mathcal{X}\backslash\{j\};\mathbb{F},\kappa,\mathbf{v},c,\delta)$$

What is known about PAXps?

- **Obs:** Definition of WPAXp is non-monotonic (from previous slide)

What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp cannot be used

What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp cannot be used
  - For DTs, finding on PAXp is computationally hard [ABOS22]

What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp cannot be used
  - For DTs, finding on PAXp is computationally hard [ABOS22]
  - In general, complexity is unwiedly [WMHK21]

What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp cannot be used
  - For DTs, finding on PAXp is computationally hard [ABOS22]
  - In general, complexity is unwiedly [WMHK21]
  - Recent dedicated algorithms for simple ML models [IHI+23]

What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp cannot be used
  - For DTs, finding on PAXp is computationally hard [ABOS22]
  - In general, complexity is unwiedly [WMHK21]
  - Recent dedicated algorithms for simple ML models [IHI+23]
  - Recent approximate algorithms for complex ML models [IMM24]

Results for decision trees

Dataset	DT N	DT A	Path M	Path m	Path avg	δ	MinPAXp Length M	MinPAXp Length m	MinPAXp Length avg	MinPAXp Prec avg	MinPAXp Time avg	LmPAXp Length M	LmPAXp Length m	LmPAXp Length avg	LmPAXp Prec avg	LmPAXp $m_\subseteq$	LmPAXp Time avg	D	Anchor Length M	Anchor Length m	Anchor Length avg	Anchor Length $F_{\notin P}$	Anchor Prec avg	Anchor Time avg
						100	11	3	6.8	100	2.34	11	3	6.9	100	100	0.00	d	12	2	7.0	26.8	76.8	0.96
adult	1241	89	14	3	10.7	95	11	3	6.2	98.4	5.36	11	3	6.3	98.6	99.0	0.01	u	12	3	10.0	29.4	93.7	2.20
						90	11	2	5.6	94.6	4.64	11	2	5.8	95.2	96.4	0.01							
						100	12	1	4.4	100	0.35	12	1	4.4	100	100	0.00	d	31	1	4.8	58.1	32.9	3.10
dermatology	71	100	13	1	5.1	95	12	1	4.1	99.7	0.37	12	1	4.1	99.7	99.3	0.00	u	34	1	13.1	43.2	87.2	25.13
						90	11	1	4.0	98.8	0.35	11	1	4.0	98.8	100	0.00							
						100	12	2	4.8	100	0.93	12	2	4.9	100	100	0.00	d	36	2	7.9	44.8	69.4	1.94
kr-vs-kp	231	100	14	3	6.6	95	11	2	3.9	98.1	0.97	11	2	4.0	98.1	100	0.00	u	12	2	3.6	16.6	97.3	1.81
						90	10	2	3.2	95.4	0.92	10	2	3.3	95.4	99.0	0.00							
						100	12	4	8.2	100	16.06	11	4	8.2	100	100	0.00	d	16	3	13.2	43.1	71.3	12.22
letter	3261	93	14	4	11.8	95	12	4	8.0	99.6	18.28	11	4	8.0	99.5	100	0.00	u	16	3	13.7	47.3	66.3	10.15
						90	12	4	7.7	97.7	16.35	10	4	7.8	97.8	100	0.00							
						100	14	3	6.4	100	0.92	14	3	6.5	100	100	0.00	d	35	2	8.6	55.4	33.6	5.43
soybean	219	100	16	3	7.3	95	14	3	6.4	99.8	0.95	14	3	6.4	99.8	100	0.00	u	35	3	19.2	66.0	75.0	38.96
						90	14	3	6.1	98.1	0.94	14	3	6.1	98.2	98.5	0.00							
						0	12	3	7.4	100	1.23	12	3	7.5	100	100	0.01	d	38	2	6.3	65.3	63.3	24.12
spambase	141	99	14	3	8.5	95	9	1	3.7	96.1	2.16	9	1	3.8	96.5	100	0.01	u	57	3	28.0	86.2	65.3	834.70
						90	6	1	2.4	92.4	2.15	8	1	2.4	92.2	100	0.01							

Results for naive Bayes classifiers

Dataset	(#F	#I)	NBC	AXp		LmPAXp$_{\leqslant 9}$				LmPAXp$_{\leqslant 7}$				LmPAXp$_{\leqslant 4}$			
			A%	Length	δ	Length	Precision	W%	Time	Length	Precision	W%	Time	Length	Precision	W%	Time
adult	(13	200)	81.37	6.8± 1.2	98	6.8± 1.1	100± 0.0	100	0.003	6.3± 0.9	99.61± 0.6	96	0.023	4.8± 1.3	98.73± 0.5	48	0.059
					95	6.8± 1.1	99.99± 0.2	100	0.074	5.9± 1.0	98.87± 1.8	99	0.058	3.9± 1.0	96.93± 1.1	80	0.071
					93	6.8± 1.1	99.97± 0.4	100	0.104	5.7± 1.3	98.34± 2.6	100	0.086	3.4± 0.9	95.21± 1.6	90	0.093
					90	6.8± 1.1	99.95± 0.6	100	0.164	5.5± 1.4	97.86± 3.4	100	0.100	3.0± 0.8	93.46± 1.5	94	0.103
agaricus	(23	200)	95.41	10.3± 2.5	98	7.7± 2.7	99.12± 0.8	92	0.593	6.4± 3.0	98.75± 0.6	87	0.763	6.0± 3.1	98.67± 0.5	29	0.870
					95	6.9± 3.1	97.62± 2.1	95	0.954	5.3± 3.2	96.59± 1.6	92	1.273	4.8± 3.3	96.24± 1.2	55	1.217
					93	6.5± 3.1	96.65± 2.8	95	1.112	4.8± 3.1	95.38± 1.9	93	1.309	4.3± 3.1	94.92± 1.3	64	1.390
					90	5.9± 3.3	94.95± 4.1	96	1.332	4.0± 3.0	92.60± 2.8	95	1.598	3.6± 2.8	92.08± 1.7	76	1.830
chess	(37	200)	88.34	12.1± 3.7	98	8.1± 4.1	99.27± 0.6	64	0.383	5.9± 4.9	98.70± 0.4	64	0.454	5.7± 5.0	98.65± 0.4	46	0.457
					95	7.7± 3.8	98.51± 1.4	68	0.404	5.5± 4.4	97.90± 0.9	64	0.483	5.3± 4.5	97.85± 0.8	46	0.478
					93	7.3± 3.5	97.56± 2.4	68	0.419	5.0± 4.1	96.26± 2.2	64	0.485	4.8± 4.1	96.21± 2.1	64	0.493
					90	7.3± 3.5	97.29± 2.9	70	0.413	4.9± 4.0	95.99± 2.6	64	0.483	4.8± 4.0	95.93± 2.5	64	0.543
vote	(17	81)	89.66	5.3± 1.4	98	5.3± 1.4	100± 0.0	100	0.000	5.3± 1.3	99.95± 0.2	100	0.007	4.6± 1.1	99.60± 0.4	64	0.014
					95	5.3± 1.4	100± 0.0	100	0.000	5.3± 1.3	99.93± 0.3	100	0.008	4.1± 1.0	98.25± 1.7	64	0.018
					93	5.3± 1.4	100± 0.0	100	0.000	5.2± 1.3	99.78± 1.1	100	0.012	4.1± 0.9	98.10± 1.9	64	0.018
					90	5.3± 1.4	100± 0.0	100	0.000	5.2± 1.3	99.78± 1.1	100	0.012	4.0± 1.2	97.24± 3.1	64	0.022
kr-vs-kp	(37	200)	88.07	12.2± 3.9	98	7.8± 4.2	99.19± 0.5	64	0.387	6.5± 4.7	98.99± 0.4	64	0.427	6.1± 4.9	98.88± 0.3	43	0.457
					95	7.3± 3.9	98.29± 1.4	64	0.416	6.0± 4.3	97.89± 1.1	64	0.453	5.5± 4.5	97.79± 0.9	43	0.462
					93	6.9± 3.5	97.21± 2.5	69	0.422	5.6± 3.8	96.82± 2.2	64	0.448	5.2± 4.0	96.71± 2.1	43	0.468
					90	6.8± 3.5	96.65± 3.1	69	0.418	5.4± 3.8	95.69± 3.0	64	0.468	5.0± 4.0	95.59± 2.8	61	0.487
mushroom	(23	200)	95.51	10.7± 2.3	98	7.5± 2.4	98.99± 0.7	90	0.641	6.5± 2.6	98.74± 0.5	83	0.751	6.3± 2.7	98.70± 0.4	18	0.828
					95	6.5± 2.6	97.35± 1.8	96	1.011	5.1± 2.5	96.52± 1.0	90	1.130	5.0± 2.5	96.39± 0.8	54	1.113
					93	5.8± 2.8	95.77± 2.7	96	1.257	4.4± 2.5	94.67± 1.6	94	1.297	4.2± 2.4	94.48± 1.3	65	1.324

Results for decision diagrams

Dataset	#I	#F	OMDD		δ	MinPAXp					LmPAXp					
						Length			Prec	Time	Length			Prec	$m_\subseteq$	Time
			#N	A%		M	m	avg	avg	avg	M	m	avg	avg		avg
lending	100	9	1103	81.7	100	9	6	8.0	100	24.24	9	6	7.9	100	100	1.57
					95	9	5	7.8	99.7	21.48	9	6	7.8	99.8	100	1.49
					90	9	4	7.2	96	24.65	9	5	7.4	97.0	100	1.48
monk2	100	6	70	79.3	100	6	4	5.1	100	0.10	6	4	5.1	100	100	0.03
					95	6	4	5.1	100	0.09	6	4	5.1	100	100	0.03
					90	6	3	4.8	98.1	0.09	6	3	4.8	98.1	100	0.03
postoperative	74	8	109	80	100	8	4	6.1	100	0.26	8	4	6.2	100	100	0.04
					95	8	2	6.0	99.3	0.25	8	2	6.0	99.3	100	0.04
					90	8	2	5.3	95.9	0.23	8	2	5.4	96.6	94.6	0.04
tic_tac_toe	100	9	424	70.3	100	9	5	7.7	100	3.60	9	5	7.8	100	100	0.38
					95	9	5	7.5	99.5	3.24	9	5	7.7	99.6	99.0	0.38
					90	9	3	7.3	98.3	4.06	9	3	7.5	98.6	98.0	0.38
xd6	100	9	76	83.1	100	9	4	4.6	100	0.10	9	4	4.6	100	100	0.03
					95	9	3	3.8	97	0.09	9	3	3.8	97.0	99.0	0.03
					90	9	3	3.3	94.8	0.10	9	3	3.4	94.6	100	0.03

Remarks on LmPAXps

[IHI+23]

- LmPAXps ignore non-monotonicity, and so overapproximate PAXps
  - Theoretical guarantees, but may be reducible
- For DTs, computation of LmPAXps is in P
- Experimental results confirm LmPAXps match PAXps in most cases
- Recent results on approximating LmPAXps for RFs [IMM24]

Outline – Unit #06

Not all inputs may be possible – input constraints

[GR22, YIS[+]23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed

Not all inputs may be possible – input constraints

[GR22, YIS+23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed
- Infer constraints on the inputs
  - Learn simple rules relating inputs
  - Represent rules as a constraint set, e.g. $\mathcal{C}(\mathbf{x})$

Not all inputs may be possible – input constraints

[GR22, YIS+23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed
- Infer constraints on the inputs
  - Learn simple rules relating inputs
  - Represent rules as a constraint set, e.g. $\mathcal{C}(\mathbf{x})$
- Redefine WAXps/WCXps to account for input constraints:

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

$$\exists(\mathbf{x} \in \mathbb{F}).\left[\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j) \land \mathcal{C}(\mathbf{x})\right] \land (\kappa(\mathbf{x}) \neq c)$$

  - Compute AXps/CXps given new definitions

Not all inputs may be possible – input constraints

[GR22, YIS+23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed
- Infer constraints on the inputs
  - Learn simple rules relating inputs
  - Represent rules as a constraint set, e.g. $\mathcal{C}(\mathbf{x})$
- Redefine WAXps/WCXps to account for input constraints:

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

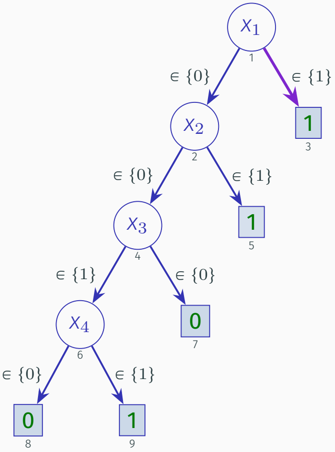
$$\exists(\mathbf{x} \in \mathbb{F}).\left[\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j) \land \mathcal{C}(\mathbf{x})\right] \land (\kappa(\mathbf{x}) \neq c)$$

  - Compute AXps/CXps given new definitions
- Constrained AXps/CXps find other applications!

An example

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

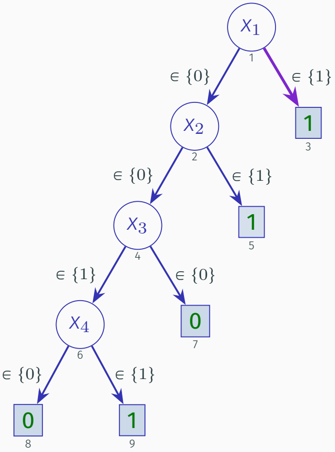
- Unconstrained AXps:



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

An example

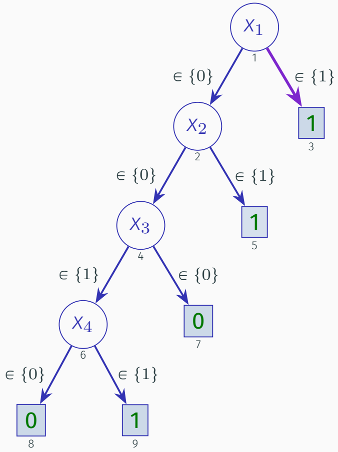
- Instance: $((1,1,1,1),1)$
- Unconstrained AXps:
  - AXps:



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

An example

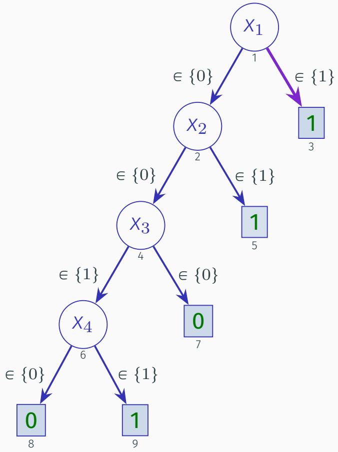
- Instance: $((1,1,1,1),1)$
- Unconstrained AXps:
  - AXps: $\{\{1\},\{2\},\{3,4\}\}$



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

An example

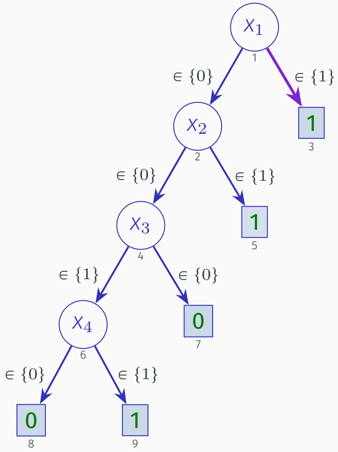
- Instance: $((1,1,1,1),1)$
- Unconstrained AXps:
  - AXps: $\{\{1\},\{2\},\{3,4\}\}$
- Constrained AXps:



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

An example

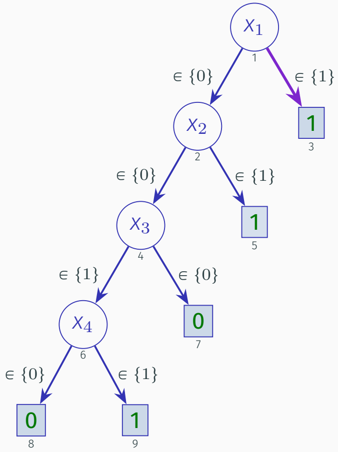
- Instance: $((1,1,1,1),1)$
- Unconstrained AXps:
  - AXps: $\{\{1\},\{2\},\{3,4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

An example

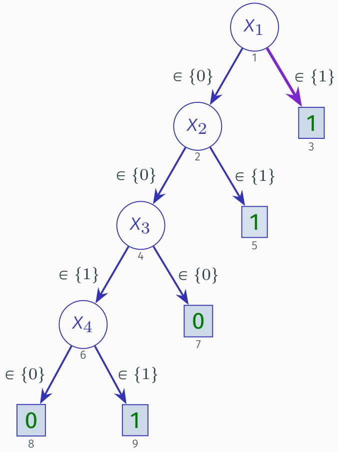
- Instance: $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps: $\{\{1\}, \{2\}, \{3, 4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1
  - If feature 4 is fixed (with value 1), then feature 3 must be assigned value 1



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

An example

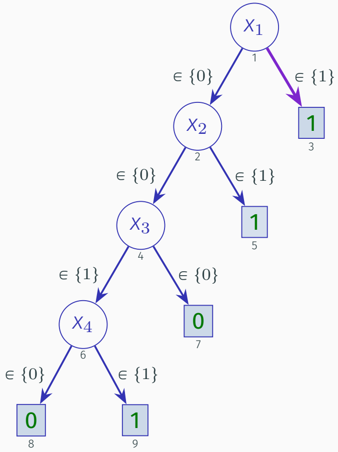
- Instance: $((1,1,1,1),1)$
- Unconstrained AXps:
  - AXps: $\{\{1\},\{2\},\{3,4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1
  - If feature 4 is fixed (with value 1), then feature 3 must be assigned value 1
  - AXps:



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

An example

- Instance: $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps: $\{\{1\}, \{2\}, \{3, 4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1
  - If feature 4 is fixed (with value 1), then feature 3 must be assigned value 1
  - AXps: $\{\{1\}, \{2\}, \{3\}, \{4\}\}$



- Constraint: $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

Outline – Unit #06

How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons [INM19a]

How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons [INM19a]
- But, robustness tools scale for much larger NNs

How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons [INM19a]
- But, robustness tools scale for much larger NNs
  - Q: can we relate AXps with adversarial examples?

How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons [INM19a]
- But, robustness tools scale for much larger NNs
  - Q: can we relate AXps with adversarial examples?
  - Obs: we already proved some basic (duality) properties for global explanations [INM19b]

How to tackle poor performance on NNs?

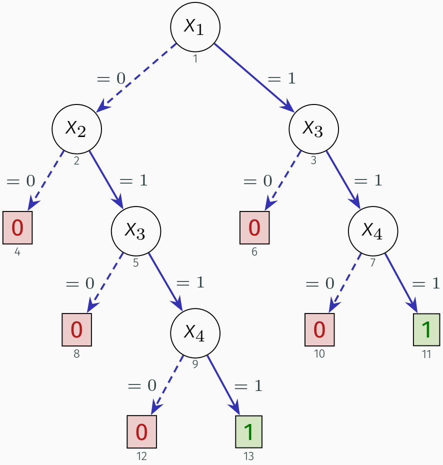
- For NNs, computation of plain AXps scales to a few tens of neurons [INM19a]
- But, robustness tools scale for much larger NNs
  - Q: can we relate AXps with adversarial examples?
  - Obs: we already proved some basic (duality) properties for global explanations [INM19b]
- Change definition of WAXp/WCXp to account for l_p distance to $\mathbf{v}$:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j) \land \left(\|\mathbf{x}-\mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j) \land \left(\|\mathbf{x}-\mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \land (\neg\sigma(\mathbf{x}))$$

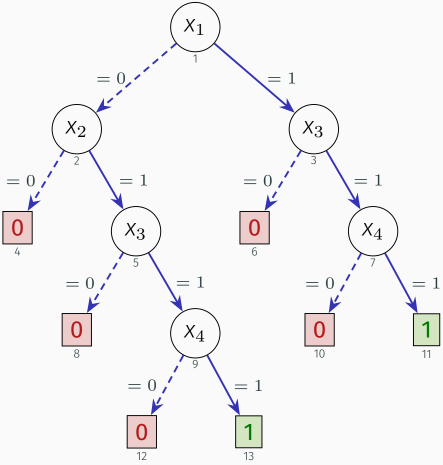
  - Norm l_p is arbitrary, e.g. Hamming, Manhattan, Euclidean, etc.
  - **Distance-restricted explanations**: $\mathfrak{d}$AXp/$\mathfrak{d}$CXp

An example – DT & instance $((1,1,1,1),1)$



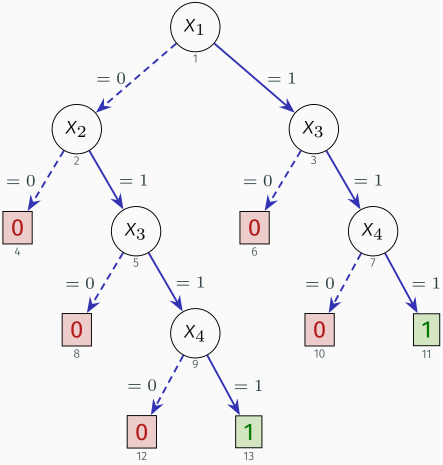
An example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:



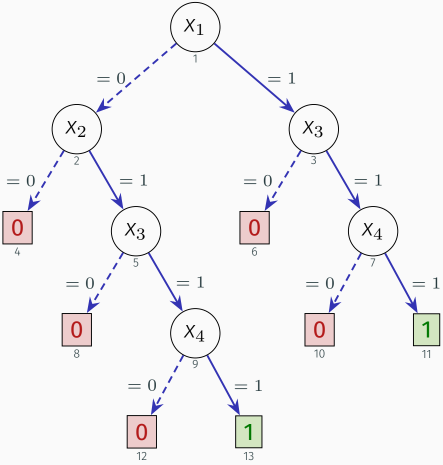
An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?



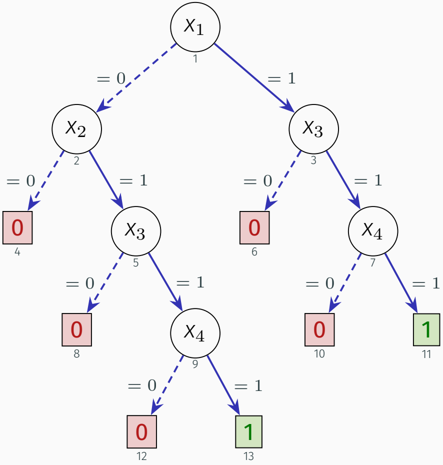
An example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps?



An example – DT & instance $((1,1,1,1),1)$

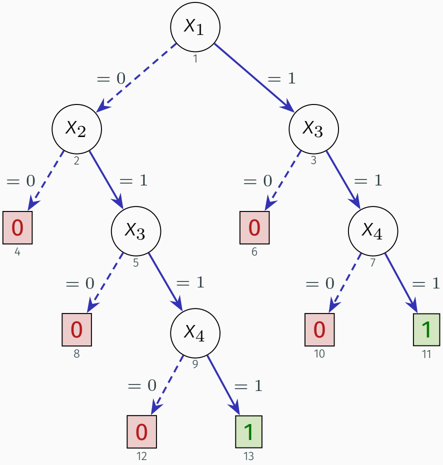
- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps? $\{\{1,2\},\{3\},\{4\}\}$



An example – DT & instance $((1,1,1,1),1)$

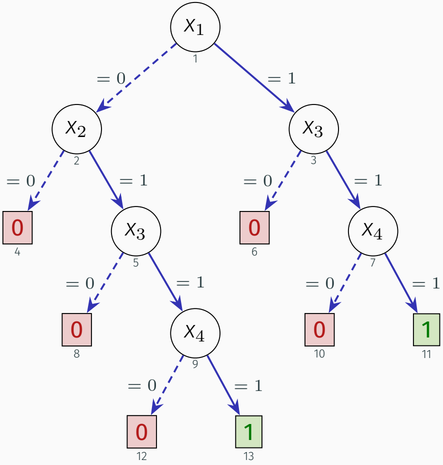
- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps? $\{\{1,2\},\{3\},\{4\}\}$

- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:



An example – DT & instance $((1,1,1,1),1)$

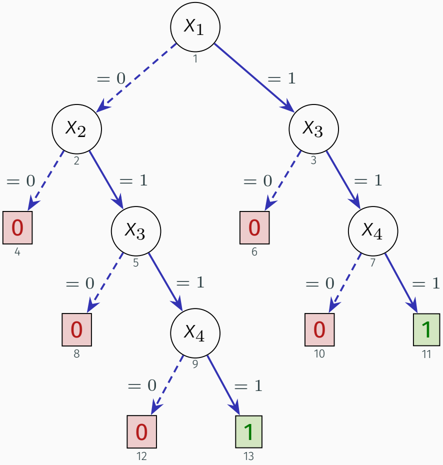
- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps? $\{\{1,2\},\{3\},\{4\}\}$
- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$



An example – DT & instance $((1,1,1,1),1)$

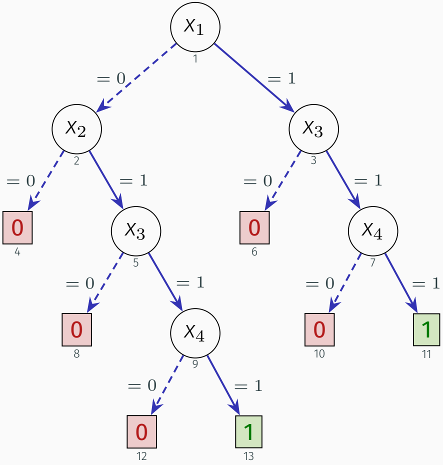
- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps? $\{\{1,2\},\{3\},\{4\}\}$

- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$
  - $\mathfrak{d}$AXps?



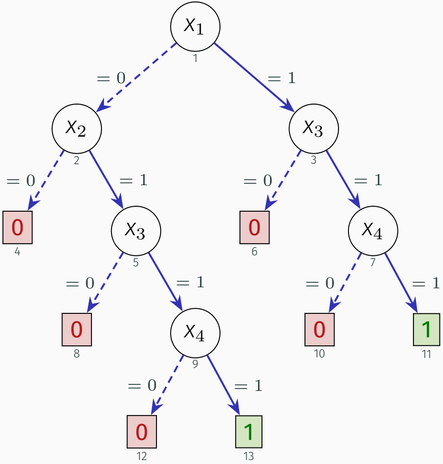
An example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps? $\{\{1,2\},\{3\},\{4\}\}$
- Distance-restricted AXps/CXps, ∂AXp/∂CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$
  - ∂AXps? $\{\{3,4\}\}$
  - ∂CXps?



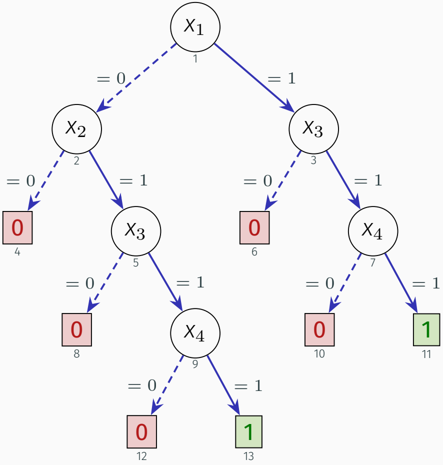
An example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps? $\{\{1,2\},\{3\},\{4\}\}$
- Distance-restricted AXps/CXps, ∂AXp/∂CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$
  - ∂AXps? $\{\{3,4\}\}$
  - ∂CXps? $\{\{3\},\{4\}\}$

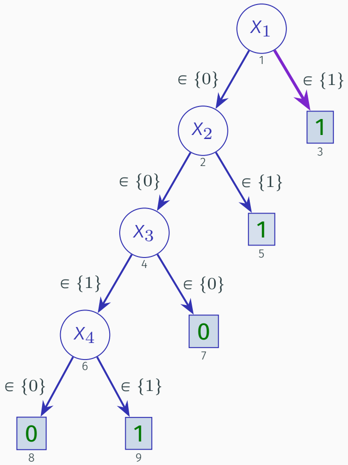


An example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1,3,4\},\{2,3,4\}\}$
  - CXps? $\{\{1,2\},\{3\},\{4\}\}$
- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$
  - $\mathfrak{d}$AXps? $\{\{3,4\}\}$
  - $\mathfrak{d}$CXps? $\{\{3\},\{4\}\}$
- Given ϵ, larger adversarial examples are excluded

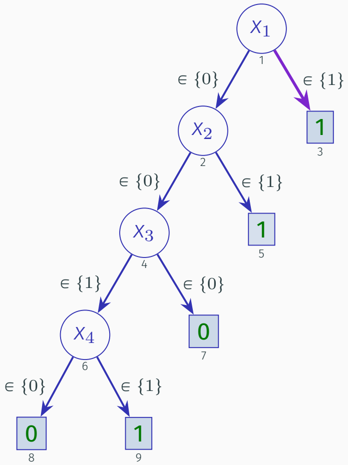


Another example – DT & instance $((1,1,1,1),1)$



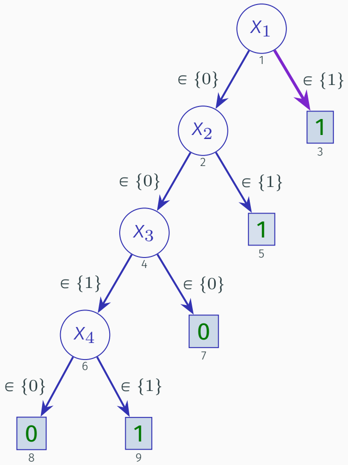
Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:

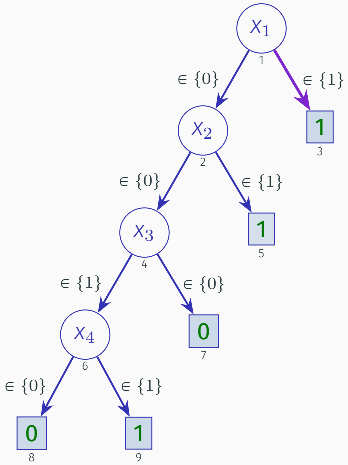


Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?

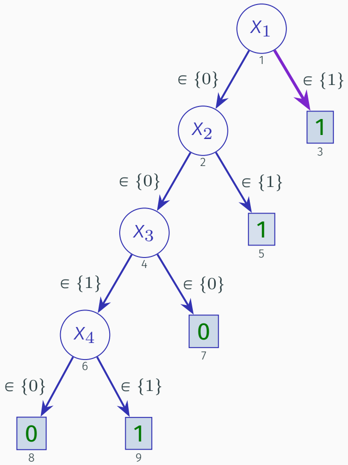


- Plain AXps/CXps:
    - AXps? $\{\{1\},\{2\}\{3,4\}\}$
    - CXps?



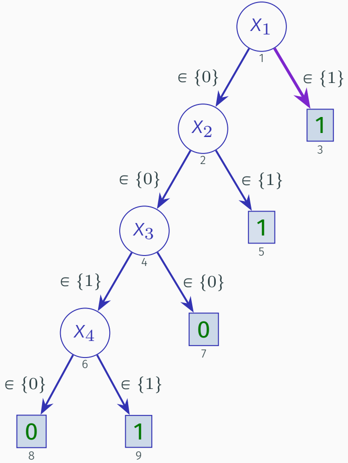
Another example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1\},\{2\}\{3,4\}\}$
  - CXps? $\{\{1,2,3\},\{1,2,4\}\}$



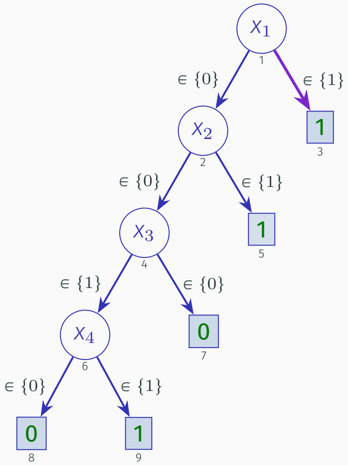
- Plain AXps/CXps:
  - AXps? $\{\{1\},\{2\}\{3,4\}\}$
  - CXps? $\{\{1,2,3\},\{1,2,4\}\}$

- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$



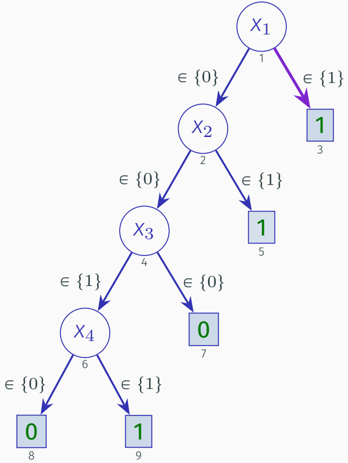
Another example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1\},\{2\}\{3,4\}\}$
  - CXps? $\{\{1,2,3\},\{1,2,4\}\}$
- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$
  - Constant function...



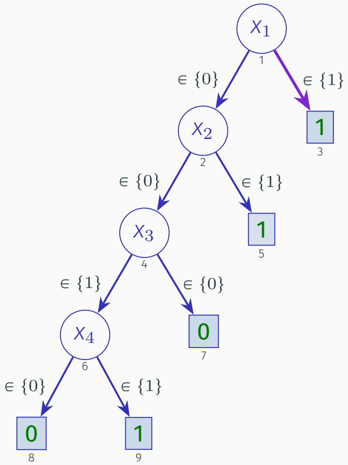
Another example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1\},\{2\}\{3,4\}\}$
  - CXps? $\{\{1,2,3\},\{1,2,4\}\}$
- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$
  - Constant function...
  - $\mathfrak{d}$AXps?



Another example – DT & instance $((1,1,1,1),1)$

- Plain AXps/CXps:
  - AXps? $\{\{1\},\{2\}\{3,4\}\}$
  - CXps? $\{\{1,2,3\},\{1,2,4\}\}$
- Distance-restricted AXps/CXps, $\mathfrak{d}$AXp/$\mathfrak{d}$CXp, with Hamming distance (l_0) and $\epsilon = 1$:
  - Points of interest: $\{(1,1,1,1),(0,1,1,1),(1,0,1,1),(1,1,0,1),(1,1,1,0)\}$
  - Constant function...
  - $\mathfrak{d}$AXps? $\{\emptyset\}$



Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{j\in\mathcal{X}}(x_j = v_j) \wedge \left(\|\mathbf{x}-\mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{j\in\mathcal{X}}(x_j = v_j) \wedge \left(\|\mathbf{x}-\mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \wedge (\neg\sigma(\mathbf{x}))$$

Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j) \land \left(\|\mathbf{x} - \mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge\nolimits_{j\in\mathcal{X}}(x_j = v_j) \land \left(\|\mathbf{x} - \mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \land (\neg\sigma(\mathbf{x}))$$

- Given norm l_p and distance ϵ, there exists a (distance-restricted) WCXp iff there exists an adversarial example
  - Use robustness tool to decide existence of WCXp
  - But, WAXp decided given non existence of CXp!

Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \land \left(\|\mathbf{x} - \mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{j \in \mathcal{X}}(x_j = v_j) \land \left(\|\mathbf{x} - \mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \land (\neg\sigma(\mathbf{x}))$$

- Given norm l_p and distance ϵ, there exists a (distance-restricted) WCXp iff there exists an adversarial example
  - Use robustness tool to decide existence of WCXp
  - But, WAXp decided given non existence of CXp!
- Efficiency of distance-restricted explanations correlates with efficiency of finding adversarial examples
  - One can use most complete robustness tools, e.g. VNN-COMP [BMB+23]

Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge \left(\|\mathbf{x} - \mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists(\mathbf{x} \in \mathbb{F}). \left[\bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge \left(\|\mathbf{x} - \mathbf{v}\|_{l_p} \leqslant \epsilon\right)\right] \wedge (\neg\sigma(\mathbf{x}))$$

- Given norm l_p and distance ϵ, there exists a (distance-restricted) WCXp iff there exists an adversarial example
  - **Use robustness tool to decide existence of WCXp**
  - But, WAXp decided given non existence of CXp!
- Efficiency of distance-restricted explanations correlates with efficiency of finding adversarial examples
  - One can use most complete robustness tools, e.g. VNN-COMP [BMB+23]
- Clear scalability improvements for explaining NNs (see next) [HM23a, WWB23, IHM+24a, IHM+24b]

Basic algorithm

[HM23a]

Input: Arguments: ϵ; Parameters: $\mathcal{E}$, p
Output: One $\mathfrak{d}$AXp $\mathcal{S}$

```
function FindAXpDel(ε; ℰ, p)
    𝒮 ← ℱ                              ▷ Initially, no feature is allowed to change
    for i ∈ ℱ do                        ▷ Invariant: ∂WAXp(𝒮)
        𝒮 ← 𝒮\{i}
        outc ← FindAdvEx(ε, 𝒮; ℰ, p)
        if outc then
            𝒮 ← 𝒮 ∪ {i}
    return 𝒮                           ▷ ∂WAXp(𝒮) ∧ minimal(𝒮) → ∂AXp(𝒮)
```

Basic algorithm

[HM23a]

Input: Arguments: ϵ; Parameters: $\mathcal{E}$, p
Output: One $\mathfrak{d}$AXp $\mathcal{S}$

function FindAXpDel(ϵ; $\mathcal{E}, p$)
    $\mathcal{S} \leftarrow \mathcal{F}$ ▷ Initially, no feature is allowed to change
    for $i \in \mathcal{F}$ **do** ▷ Invariant: $\mathfrak{d}$WAXp($\mathcal{S}$)
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$
        outc $\leftarrow$ FindAdvEx($\epsilon, \mathcal{S}; \mathcal{E}, p$)
        if outc **then**
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$
    return $\mathcal{S}$ ▷ $\mathfrak{d}$WAXp($\mathcal{S}$) $\wedge$ minimal($\mathcal{S}$) $\rightarrow$ $\mathfrak{d}$AXp($\mathcal{S}$)

- **Obs:** Efficiency of logic-based XAI tracks efficiency of robustness tools

Basic algorithm

[HM23a]

Input: Arguments: ϵ; Parameters: $\mathcal{E}$, p
Output: One $\mathfrak{d}$AXp $\mathcal{S}$

```
function FindAXpDel(ϵ; ℰ, p)
    𝒮 ← ℱ                                  ▷ Initially, no feature is allowed to change
    for i ∈ ℱ do                           ▷ Invariant: 𝔡WAXp(𝒮)
        𝒮 ← 𝒮\{i}
        outc ← FindAdvEx(ϵ, 𝒮; ℰ, p)
        if outc then
            𝒮 ← 𝒮 ∪ {i}
    return 𝒮                               ▷ 𝔡WAXp(𝒮) ∧ minimal(𝒮) → 𝔡AXp(𝒮)
```

- **Obs:** Efficiency of logic-based XAI tracks efficiency of robustness tools
- **Limitation:** Running time grows with number of features

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

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

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

[HM23a]

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

Scales to a few **hundred** neurons

Recent improvements

Input: Arguments: ϵ; Parameters: $\mathcal{E}$, p
Output: One $\mathfrak{d}$AXp $\mathcal{S}$

function FindAXpDel($\epsilon; \mathcal{E}, p$)
    $\mathcal{S} \leftarrow \mathcal{F}$ ▷ Initially, no feature is allowed to change
    for $i \in \mathcal{F}$ **do**   ▷ Invariant: $\mathfrak{d}$WAXp($\mathcal{S}$)
        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$
        outc $\leftarrow$ FindAdvEx($\epsilon, \mathcal{S}; \mathcal{E}, p$)
        if outc **then**
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$
    return $\mathcal{S}$ ▷ $\mathfrak{d}$WAXp($\mathcal{S}$) $\wedge$ minimal($\mathcal{S}$) $\rightarrow$ $\mathfrak{d}$AXp($\mathcal{S}$)

Recent improvements

Input: Arguments: ϵ; Parameters: $\mathcal{E}$, p
Output: One ∂AXp $\mathcal{S}$

```
function FindAXpDel(ϵ; ℰ, p)
    𝒮 ← ℱ                              ▷ Initially, no feature is allowed to change
    for i ∈ ℱ do                        ▷ Invariant: ∂WAXp(𝒮)
        𝒮 ← 𝒮\{i}
        outc ← FindAdvEx(ϵ, 𝒮; ℰ, p)
        if outc then
            𝒮 ← 𝒮 ∪ {i}
    return 𝒮                            ▷ ∂WAXp(𝒮) ∧ minimal(𝒮) → ∂AXp(𝒮)
```

- To drop features from $\mathcal{S} \subseteq \mathcal{F}$, it is open whether paralellization might be applicable
  - Algorithm FindAXpDel is mostly sequential (see above)
  - Exploit parallelization for other algorithms, e.g. dichotomic search [IHM+24b]

Input: Arguments: ϵ; Parameters: $\mathcal{E}$, p
Output: One ∂AXp $\mathcal{S}$

```
function FindAXpDel(ϵ; ℰ, p)
    𝒮 ← ℱ                                  ▷ Initially, no feature is allowed to change
    for i ∈ ℱ do                           ▷ Invariant: ∂WAXp(𝒮)
        𝒮 ← 𝒮\{i}
        outc ← FindAdvEx(ϵ, 𝒮; ℰ, p)
        if outc then
            𝒮 ← 𝒮 ∪ {i}
    return 𝒮                               ▷ ∂WAXp(𝒮) ∧ minimal(𝒮) → ∂AXp(𝒮)
```

- To drop features from $\mathcal{S} \subseteq \mathcal{F}$, it is open whether paralellization might be applicable
  - Algorithm FindAXpDel is mostly sequential (see above)
  - Exploit parallelization for other algorithms, e.g. dichotomic search [IHM+24b]
- However, to decide whether $\mathcal{S}$ is an AXp, we can exploit parallelization:
  - Recall: $\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \wedge \forall(t \in \mathcal{X}).\neg\text{WAXp}(\mathcal{X} \setminus \{t\})$
  - Each $\neg$WAXp($\cdot$) (and also WAXp($\cdot$)) check can be run in parallel!
  - Do this opportunistically, i.e. when set $\mathcal{S}$ is expected to be AXp [IHM+24b]

More recent results (from 2024)...

[IHM+24a, IHM+24b]

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

More recent results (from 2024)...

[IHM+24a, IHM+24b]

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

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

More recent results (from 2024)...

[IHM+24a, IHM+24b]

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

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

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

Outline – Unit #06

Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models

Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models

- Approach:
  - Train a surrogate model, e.g. DT, RF/TE, small(er) NN, etc.
  - Target high accuracy of surrogate model

Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models
- Approach:
  - Train a surrogate model, e.g. DT, RF/TE, small(er) NN, etc.
  - Target high accuracy of surrogate model
- Explain the surrogate model
  - Compute rigorous explanation: plain AXp, probabilistic AXp,

Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models
- Approach:
  - Train a surrogate model, e.g. DT, RF/TE, small(er) NN, etc.
  - Target high accuracy of surrogate model
- Explain the surrogate model
  - Compute rigorous explanation: plain AXp, probabilistic AXp,
- Report computed explanation as explanation for the complex ML model

Certified explainer (for monotonic classification)

[HM23c]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness

Certified explainer (for monotonic classification)

[HM23c]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness

- Certification of implementations is one possible alternative
  - Formalize algorithm, e.g. explanations for monotonic classifiers, e.g. using Coq
  - Prove that formalized algorithm is correct
  - Extract certified algorithm from proof of correctness

Certified explainer (for monotonic classification)

[HM23c]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness

- Certification of implementations is one possible alternative
  - Formalize algorithm, e.g. explanations for monotonic classifiers, e.g. using Coq
  - Prove that formalized algorithm is correct
  - Extract certified algorithm from proof of correctness

- Downsides:
  - Efficiency of certified algorithm
  - Dedicated algorithm for each explainer

Certified explainer (for monotonic classification)

[HM23c]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness

- Certification of implementations is one possible alternative
  - Formalize algorithm, e.g. explanations for monotonic classifiers, e.g. using Coq
  - Prove that formalized algorithm is correct
  - Extract certified algorithm from proof of correctness

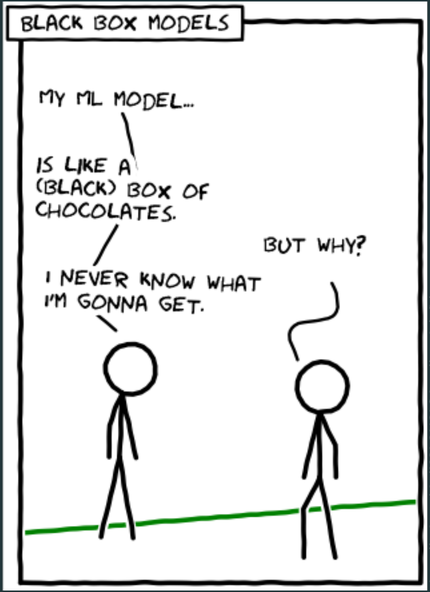
- Downsides:
  - Efficiency of certified algorithm
  - Dedicated algorithm for each explainer

- Certification envisioned for **any** explainability algorithm

Plan for this course – light at the end of the tunnel...

- 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

Questions?



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

References i

[ABOS22] Marcelo Arenas, Pablo Barceló, Miguel A. Romero Orth, and Bernardo Subercaseaux.
On computing probabilistic explanations for decision trees.
In *NeurIPS*, 2022.

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

[BMB+23] Christopher Brix, Mark Niklas Müller, Stanley Bak, Taylor T. Johnson, and Changliu Liu.
First three years of the international verification of neural networks competition (VNN-COMP).
Int. J. Softw. Tools Technol. Transf., 25(3):329–339, 2023.

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

[HM23a] Xuanxiang Huang and João Marques-Silva.
From robustness to explainability and back again.
CoRR, abs/2306.03048, 2023.

[HM23b] Xuanxiang Huang and João Marques-Silva.
The inadequacy of Shapley values for explainability.
CoRR, abs/2302.08160, 2023.

[HM23c] Aurélie Hurault and João Marques-Silva.
Certified logic-based explainable AI - the case of monotonic classifiers.
In *TAP*, pages 51–67, 2023.

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

[IHI+22] Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.
On computing probabilistic abductive explanations.
CoRR, abs/2212.05990, 2022.

[IHI+23] Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.
On computing probabilistic abductive explanations.
Int. J. Approx. Reason., 159:108939, 2023.

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

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

[IIM22] Yacine Izza, Alexey Ignatiev, and João Marques-Silva.
On tackling explanation redundancy in decision trees.
J. Artif. Intell. Res., 75:261–321, 2022.

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

[IISM24] Yacine Izza, Alexey Ignatiev, Peter J. Stuckey, and João Marques-Silva.
Delivering inflated explanations.
In *AAAI*, pages 12744–12753, 2024.

[IMM24] Yacine Izza, Kuldeep Meel, and João Marques-Silva.
Locally-minimal probabilistic explanations.
In *ECAI*, 2024.

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

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

References iv

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

[LHAMS24] Olivier Létoffé, Xuanxiang Huang, Nicholas Asher, and Joao Marques-Silva.
From SHAP scores to feature importance scores.
CoRR, abs/2405.11766, 2024.

[LHMS24] Olivier Létoffé, Xuanxiang Huang, and Joao Marques-Silva.
On correcting SHAP scores.
CoRR, abs/2405.00076, 2024.

[Mil56] George A Miller.
The magical number seven, plus or minus two: Some limits on our capacity for processing information.
Psychological review, 63(2):81–97, 1956.

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

References v

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

[WWB23] Min Wu, Haoze Wu, and Clark W. Barrett.
VeriX: Towards verified explainability of deep neural networks.
In *NeurIPS*, 2023.

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