Tractable Explaining of Multivariate Decision Trees

Clément Carbonnel
Martin Cooper João Marques-Silva

CNRS, LIRMM, Montpellier
IRIT, CNRS, University of Toulouse III, Toulouse

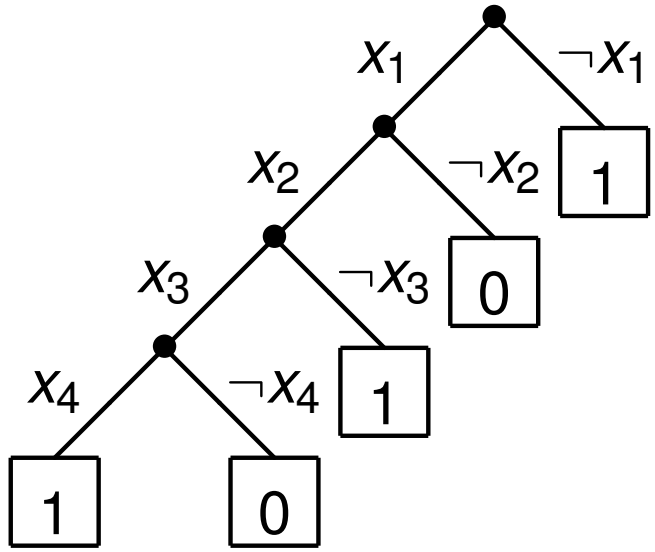
ADRIA Seminar — April 2023

Outline

- Multivariate Decision Trees are richer than DT's since they allow branching on conditions on more than one variable.
- **But** finding an abductive explanation of a decision is now NP-hard in the general case.
- Which languages of conditions allow tractable explaining?
- We give some examples over boolean/finite/infinite domains.
- We characterise all tractable languages over boolean domains.

Explaining decision of DT's

A decision tree corresponding to the classifier $\kappa(\mathbf{x}) = \neg x_1 \vee (x_2 \wedge (\neg x_3 \vee x_4))$



The decision $\kappa(1,1,1,1) = 1$ is better explained by $\{(x_2, 1), (x_4, 1)\}$ than by the whole leftmost path in the tree.

An *AXp* (abductive explanation) is a minimal subset of features sufficient to explain the decision.

Multivariate Decision Trees

Definition

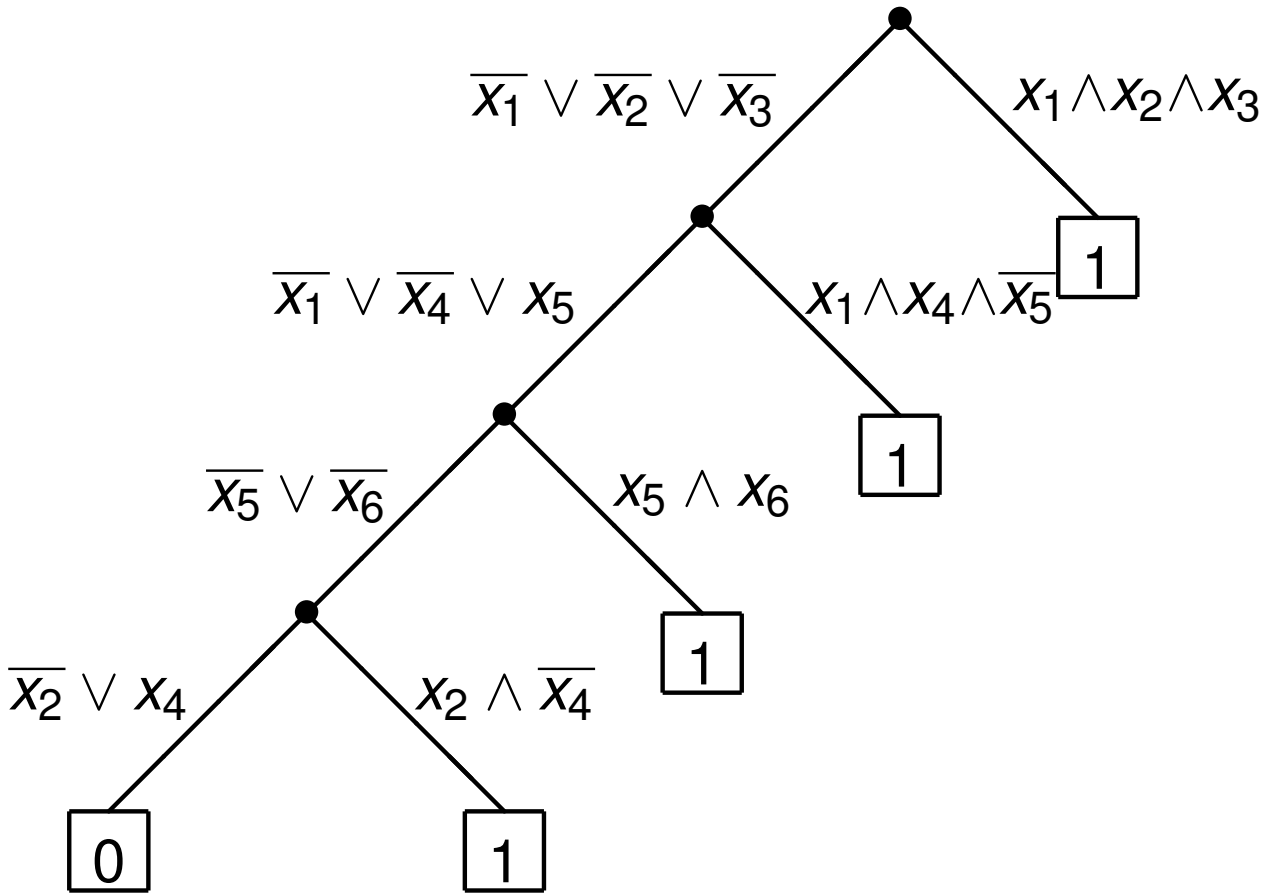
A *multivariate decision tree* is a decision tree in which the condition tested at a node is a constraint on any number of features. An $\mathcal{L}$-DT is a multivariate decision tree in which the constraint relations belong to the language $\mathcal{L}$.

Assuming binary branching, for each branch corresponding to condition C there is a branch corresponding to $\neg C$.

$\Longrightarrow$ we are only interested in languages $\mathcal{L}$ **closed under complement**

Example of a Multivariate Decision Tree

$$\kappa(\mathbf{x}) = (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_4 \wedge \overline{x_5}) \vee (x_5 \wedge x_6) \vee (x_2 \wedge \overline{x_4})$$



An MDT can be *exponentially* smaller than an equivalent DT

Tractable explaining

A *weak* AXp X is a set of features that is sufficient to explain the decision, but is not necessarily minimal. Let wAXpDT($\mathcal{L}$) be the problem of deciding whether a set of features is a weak AXp (of a decision taken by an MDT in $\mathcal{L}$-DT).

Theorem

If wAXpDT($\mathcal{L}$) $\in$ P, then there is a polynomial-time algorithm to find an AXp of a decision taken by a MDT in $\mathcal{L}$-DT.

This follows from the classic 'deletion' algorithm:

$S \leftarrow \{1, \ldots, n\}$ (i.e. all features)
for each feature i :
if $S \setminus \{i\}$ is a weak AXp then $S \leftarrow S \setminus \{i\}$

Tractable explaining

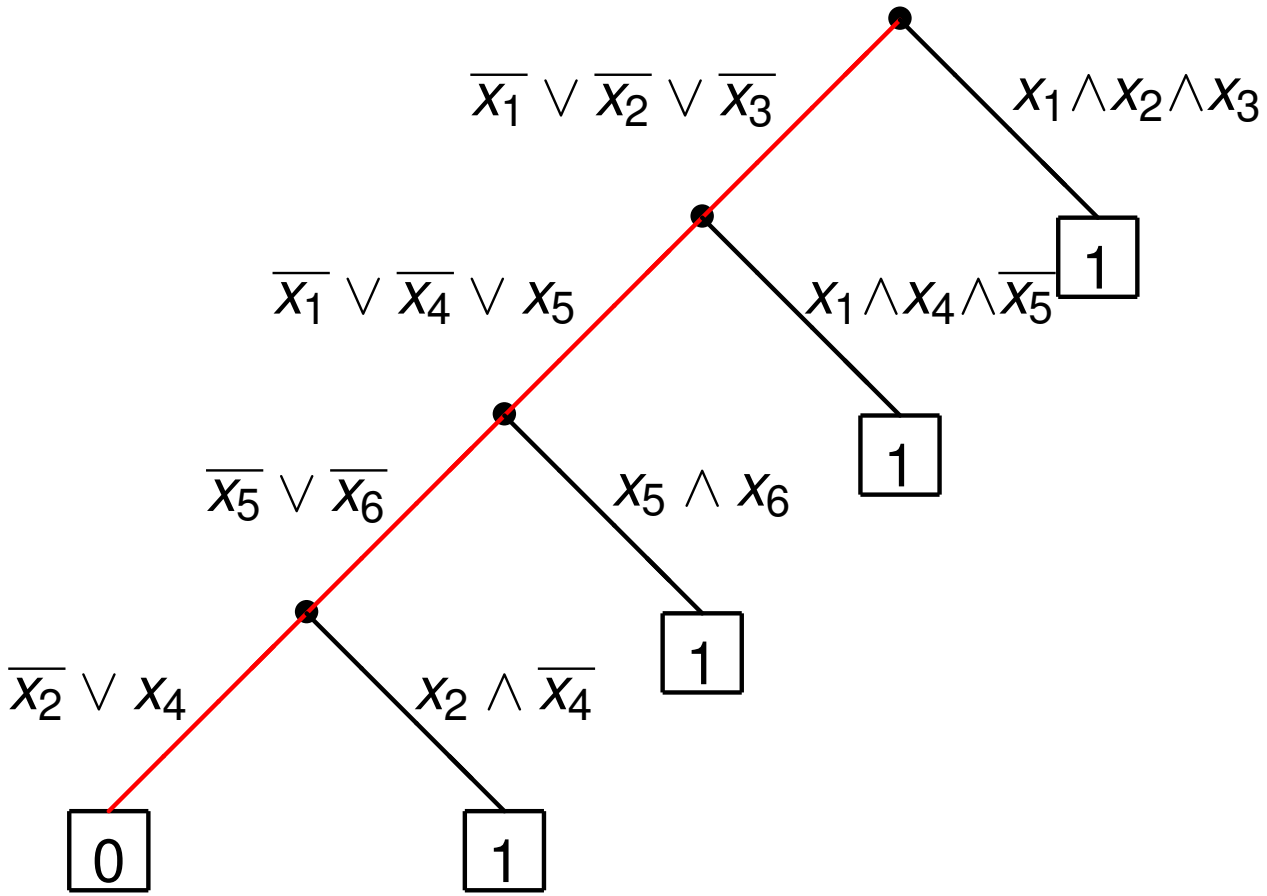
CSP($\mathcal{L}$) is the Constraint Satisfaction Problem with the restriction that constrains belong to the language $\mathcal{L}$. Let *Asst* be the set of unary assignment constraints (i.e. constraints of the form $x_i = c$ where c is a constant).

Theorem

$\mathcal{L}$ closed under complement $\wedge$ CSP($\mathcal{L} \cup$ Asst) $\in P \Rightarrow$ wAXpDT($\mathcal{L}$) $\in P$ (and hence finding an AXp is polytime)

proof: To determine whether X is a weak AXp of a classifier defined by an MDT in $\mathcal{L}$-DT, we only need to test whether each leaf corresponding to a different outcome is incompatible with the partial assignment corresponding to X. In other words, we have to solve a linear number of instances of CSP($\mathcal{L} \cup$ *Asst*).

Explaining a decision $\Rightarrow$ solving CSPs



To know whether $\{2, 4, 6\}$ is a weak AXp of the decision $\kappa(1, \ldots, 1) = 1$, we need to solve the CSP:
$x_2{=}1$, $x_4{=}1$, $x_6{=}1$, $\overline{x_1} \vee \overline{x_2} \vee \overline{x_3}$, $\overline{x_1} \vee \overline{x_4} \vee x_5$, $\overline{x_5} \vee \overline{x_6}$, $\overline{x_2} \vee x_4$.

Examples of tractable *boolean* languages $\mathcal{L}$

In the following examples, $\mathcal{L}$ is closed under complement and CSP($\mathcal{L} \cup \mathit{Asst}$) $\in$ P:

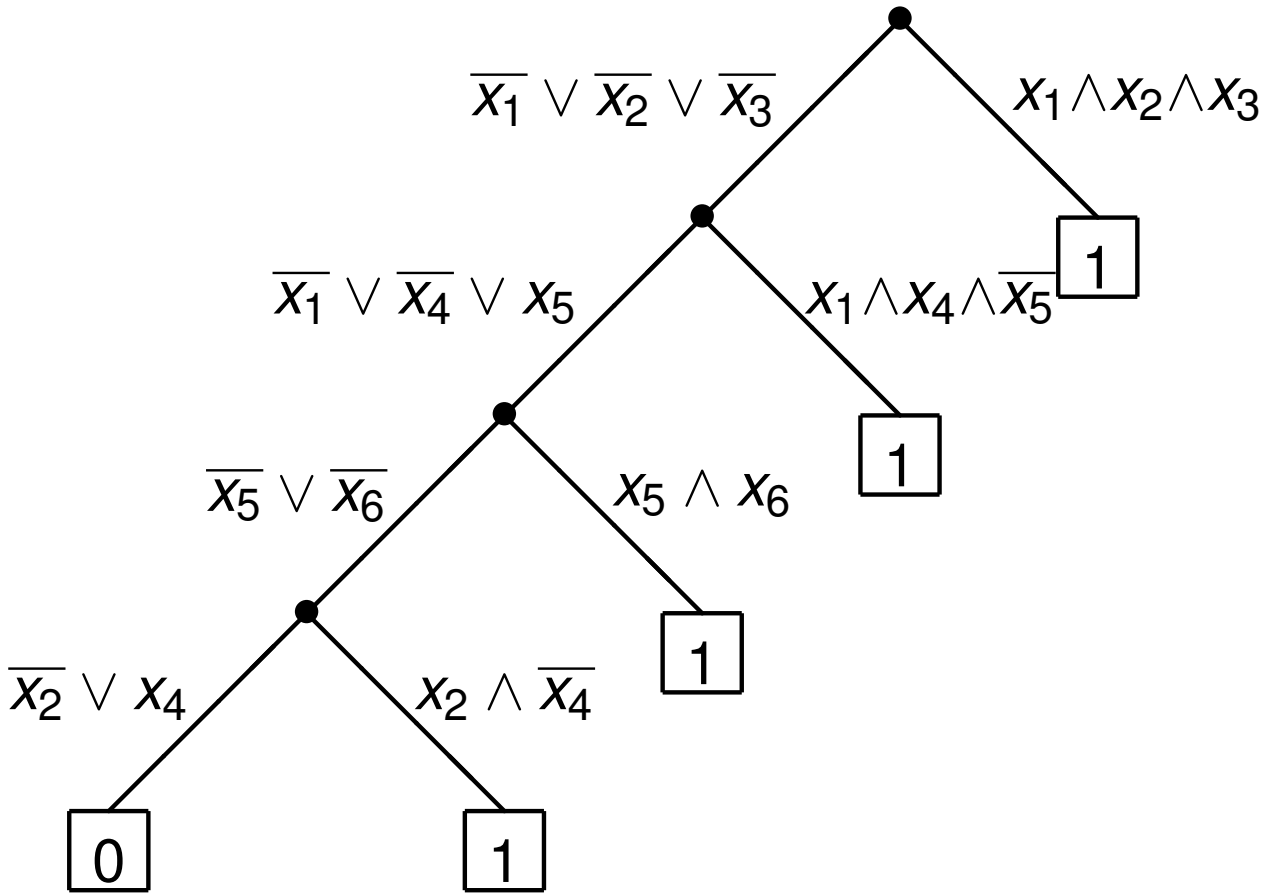
1. Horn clauses (and their negations)
2. 2-conjunctions of 2-clauses (and their negations). The complement of a 2-conjunction of 2-clauses is also the conjunction of 2-clauses, since $\neg((a \vee b) \wedge (c \vee d)) \equiv (\neg a \vee \neg c) \wedge (\neg a \vee \neg d) \wedge (\neg b \vee \neg c) \wedge (\neg b \vee \neg d)$.

Theorem

A k-term DNF in which each term contains at most 1 negative literal can be expressed as a size-$(2k+1)$ MDT with Horn-clause conditions.

Example of an MDT with Horn-clause conditions

$$\kappa(\mathbf{x}) = (x_1 \wedge x_2 \wedge x_3) \vee (x_1 \wedge x_4 \wedge \overline{x_5}) \vee (x_5 \wedge x_6) \vee (x_2 \wedge \overline{x_4})$$



The conditions are Horn clauses (or their negations)

Examples of tractable *finite-domain* languages $\mathcal{L}$

In the following examples, $\mathcal{L}$ is closed under complement and CSP($\mathcal{L} \cup Asst$) $\in$ P:

1. Two-fan constraints (i.e. constraints of the form $X_i = a \vee X_j = b$, where a, b are constants) and their complements, together with all unary constraints $X_i \in S$ where S is any subset. The complement of the two-fan $X_i = a \vee X_j = b$ is the constraint $X_i \neq a \wedge X_j \neq b$ which is the conjunction of two unary constraints.
2. Generalised Interval Constraints (GIC's) (i.e. constraint of the form $X_i \leq a \ \vee \ X_j \geq b$, where a, b are constants) and their complements, together with all unary constraints. The complement of the GIC $X_i \leq a \ \vee \ X_j \geq b$ is the constraint $X_i > a \ \wedge \ X_j < b$, which is the conjunction of unary constraints.

Examples of tractable *infinite-domain* languages $\mathcal{L}$

In the following examples, $\mathcal{L}$ is closed under complement and CSP($\mathcal{L} \cup Asst$) $\in$ P:

1. Unit two variable per inequality (UTVPI) constraints (i.e. constraints of the form $aX_i + bX_j \leq d$ where X_i and X_j are integer variables, the coefficients $a, b \in \{-1, 0, 1\}$ and the bound d is an integer constant). The negation of such a constraint is $-aX_i - bX_j \leq -(d+1)$ and is hence also an UTVPI constraint.
2. Linear inequalities over the reals. The complement of a linear inequality is again a linear inequality. Such MDT's are known as *oblique decision trees*.

Characterising tractable languages

Over finite domains, we know for which finite languages $\text{CSP}(\mathcal{L}) \in \text{P}$.

The problem is to characterisie the sublanguages of these tractable langauges which are also *closed under complement*.

In the case of boolean domains, this means finding *sub-languages closed under complement* of the four following languages:

1. Horn formulas (i.e. conjunctions of Horn clauses)
2. anti-Horn formuals (i.e. conjunctions of anti-Horn clauses)
3. conjunctions of linear equations over $\{0, 1\}$
4. 2CNFs (i.e. conjunctions of 2-clauses)

Characterisation of tractable boolean languages

For each of these four languages, we show that there is a unique maximal sub-language that is closed under complement:

1. Horn formulas $\longrightarrow$ star-nested Horn formulas
2. anti-Horn formulas $\longrightarrow$ star-nested anti-Horn formulas
3. conjunctions of linear equations $\longrightarrow$ a single equation
4. 2CNFs $\longrightarrow$ square 2CNFs

Star-nested Horn formulas

Definition

A *star-nested Horn formula* is a conjunction $C_1 \wedge \ldots \wedge C_k$ where for all pairs of clauses C_i, C_j:

1. all clauses C_i are Horn, and
2. negative literals of $C_i \subseteq$ negative literals of C_j , or
   negative literals of $C_j \subseteq$ negative literals of C_i

Examples of formulas that are equivalent to a star-nested Horn formula:

- a Horn clause
- $x_1 \wedge \ldots \wedge x_k \wedge (\overline{y_1} \vee \ldots \vee \overline{y_m})$
- $\overline{x_1} \vee \ldots \vee \overline{x_k} \vee (y_1 \wedge \ldots \wedge y_m)$

Square 2CNFs

Definition

A non-trivial *square 2CNF* is expressible in one of the three following forms (in which the four literals a, b, c, d are not necessarily distinct):

1. $(a \vee b) \wedge (c \vee d)$,
2. $(a \vee b) \wedge (b \vee c) \wedge (c \vee d)$,
3. $(a \vee b) \wedge (b \vee c) \wedge (c \vee d) \wedge (d \vee a)$.

Examples of formulas that are equivalent to a square 2CNF:

- a 2-conjunction of 2-clauses
- any boolean function on 2 variables
- $x_1 + x_2 + x_3 \geq 2$

Conclusion

- We can extend DT's to multivariate $\mathcal{L}$-DT's for certain languages $\mathcal{L}$ while preserving tractability of finding an abductive explanation.
- For boolean domains, we have a P/NP-hard language dichotomy, but the problem is still open for finite/infinite domains.
- Tractable explainability applies to certain known MDT's such as oblique DT's (linear constraints over the reals) but also to some novel types of MDT's.
- There are many interesting open questions concerning, for example, algorithms for *learning* MDT's corresponding to the tractable languages we have identified.