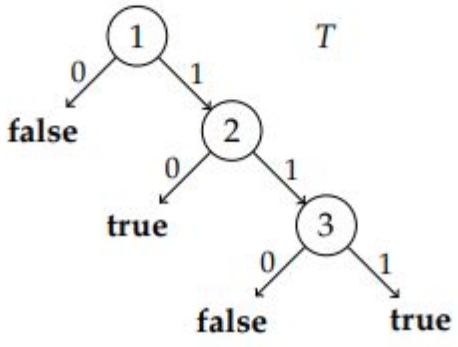
On Computing Probabilistic Explanations for Decision Trees

Marcelo Arenas, Pablo Barceló, Miguel Romero,
Bernardo Subercaseaux

Presented by: Mateusz Błajda, Maciej Nadolski

Background

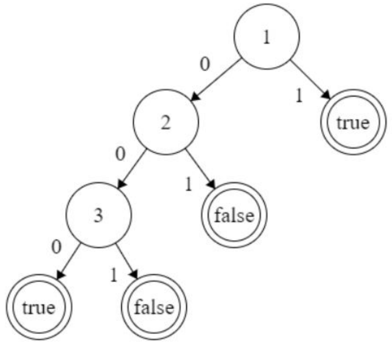
- Model
  - A decision tree
  - Binary features and classification
- Sufficient reason
  - a form of local explanation
  - a set of features which are sufficient for a particular classification



- x = (1, 1, 1); M(x) = true
- Sufficient reasons:
  - y = {1, 2, 3}
  - y = {1, 3}

Minimal/minimum sufficient reason

- sufficient reason is minimal if it's minimal under subset partial ordering
  - requiring y to be subset minimal
- sufficient reason is minimum if it's minimal under reason size ordering
  - requiring |y| to be minimal



- x = (1, 0, 0); M(x) = true
- Minimal reasons:
  - y = {1}; y = {2, 3}
- Minimum reason:
  - y = {1}

Minimal sufficient reason algorithm

A fairly straightforward polynomial time algorithm:

Algorithm 1: Minimal Sufficient Reason

Input: Decision tree T and instance $\boldsymbol{x}$, both of dimension n
Output: A minimal sufficient reason $\boldsymbol{y}$ for $\boldsymbol{x}$ under T.

$\boldsymbol{y} \leftarrow \boldsymbol{x}$
while *true* **do**
    reduced $\leftarrow$ **false**
    for $i \in \{1, \ldots, n\}$ **do**
        $\hat{\boldsymbol{y}} \leftarrow \boldsymbol{y}$
        $\hat{\boldsymbol{y}}[i] \leftarrow \bot$
        if *CheckSufficientReason*$(T, \hat{\boldsymbol{y}}, \boldsymbol{x})$ **then**
            $\boldsymbol{y} \leftarrow \hat{\boldsymbol{y}}$
            reduced $\leftarrow$ **true**
            break
    if *(¬reduced) or* $|\boldsymbol{y}|_\bot = n$ **then**
        return $\boldsymbol{y}$

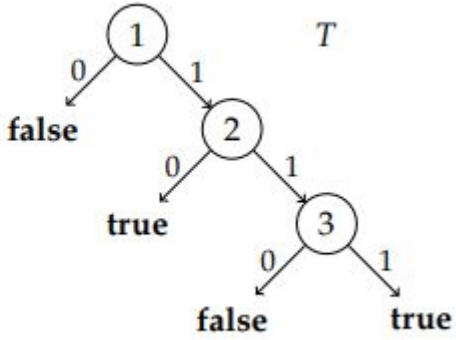
Minimum sufficient reason algorithm

- can’t be computed in polynomial time
  (assuming P != NP)
- proof in previous work (2020)

δ-sufficient reasons

- treat all unspecified features as random with uniform distribution
- y is a δ-sufficient reason for x if
  - $P(M(x) = M(Y)) > \delta$
  - Y is an instance that is equal to x on features in y, uniformly distributed otherwise
- we define minimal/minimum δ-sufficient reasons analogously
- finding both is an NP-hard problem

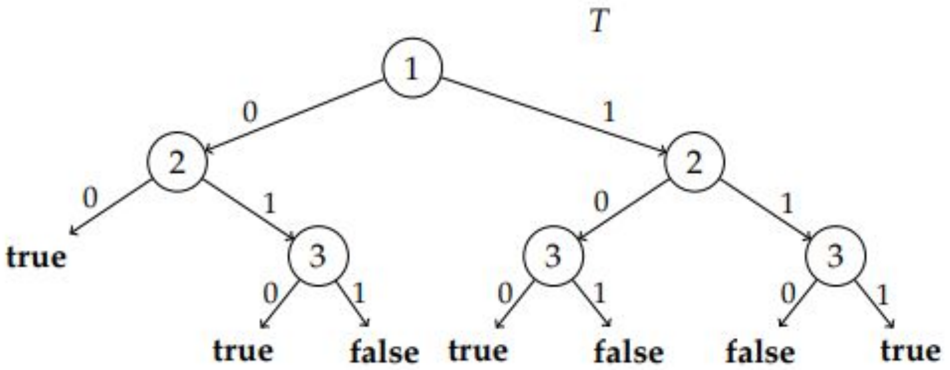
δ-sufficient reasons



$p(\emptyset) = \frac{3}{8}$	$p(\{1,2\}) = \frac{1}{2}$
$p(\{1\}) = \frac{3}{4}$	$p(\{1,3\}) = 1$
$p(\{2\}) = \frac{1}{4}$	$p(\{2,3\}) = \frac{1}{2}$
$p(\{3\}) = \frac{1}{2}$	$p(\{1,2,3\}) = 1$

Figure 1: The decision tree T and the values $p(X)$ from Example 1.

δ-sufficient reasons



$$p(\emptyset) = \tfrac{5}{8} \qquad p(\{1,2\}) = \tfrac{1}{2}$$
$$p(\{1\}) = \tfrac{1}{2} \qquad p(\{1,3\}) = \tfrac{1}{2}$$
$$p(\{2\}) = \tfrac{1}{2} \qquad p(\{2,3\}) = \tfrac{1}{2}$$
$$p(\{3\}) = \tfrac{1}{2} \qquad p(\{1,2,3\}) = 1$$

Figure 2: The decision tree T and the values $p(X)$ from Example 2.

Time complexity of δ-sufficient reasons

We consider the problem of computing minimum/minimal δ-SR for a fixed $\delta \in (0,1]$

Assuming that PTIME ≠ NP

- There is no polynomial-time algorithm for δ-Compute-Minimum-SR.
- There is no polynomial-time algorithm for δ-Compute-Minimal-SR.

(Proofs in a paper)

Trackable Cases

We can find polynomial algorithms for both minimum and minimal δ-SR if we apply some restrictions:

- Bounded split number
- Monotonicity

Bounded split number

T - decision tree

U - set of nodes

$N_u^{\downarrow}$- set of nodes in a subtree rooted in u

$N_u^{\uparrow}$- all the other nodes

F(U) - set of features in nodes U

Split number =

$$\max_{\text{node } u \text{ in } T} |F(N_u^{\downarrow}) \cap F(N_u^{\uparrow})|$$

The measure of interaction (number of common features) between the subtrees of the form T_u and their exterior

Bounded split number

Let $c \geq 1$ be a fixed integer. Both
Compute-Minimum-SR and Compute-Minimal-SR

can be solved in polynomial time for decision trees with split number at most c

Proof in the paper

Monotonicity

Lets define an order:

$x \preccurlyeq y$ iff $x[i] \leq y[i]$, for all $i \in \{1, . . . , n\}$

Model M is monotone if:

$x \preccurlyeq y \Rightarrow M(x) \leq M(y)$.

So basically if M(x) = 1 than if we replace any 0 in x with 1 and get x’: M(x’) = 1

For monotone models we can find both minimal and minimum δ-sufficient reasons in polynomial time

Again, proof in a paper

SAT to the Rescue!

We can encode minimal/minimum δ-sufficient reasons problem as CNF satisfiability (SAT) problem. (like any NP problem)

We can then use SAT Solvers to solve it.

Such encryptions are described in the paper.

Thank you for your attention