Efficient Inference of Optimal Decision Trees

Florent Avellaneda

Postdoc at the Computer Research Institute of Montréal (CRIM)

AAAI-20 New York

Overview

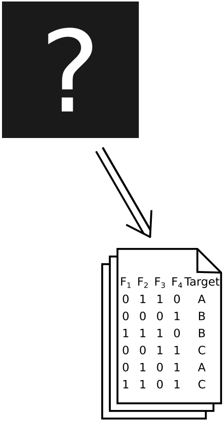
Overview

Problem

Black Box

Problem

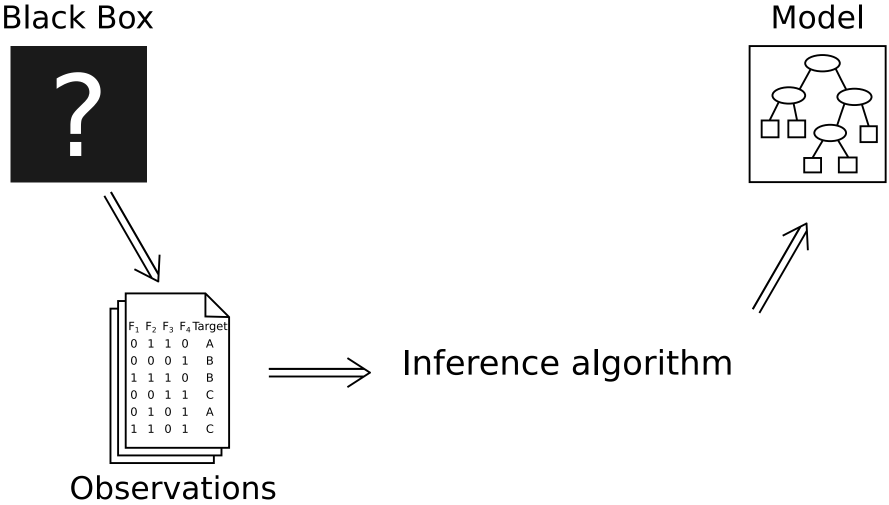
Black Box



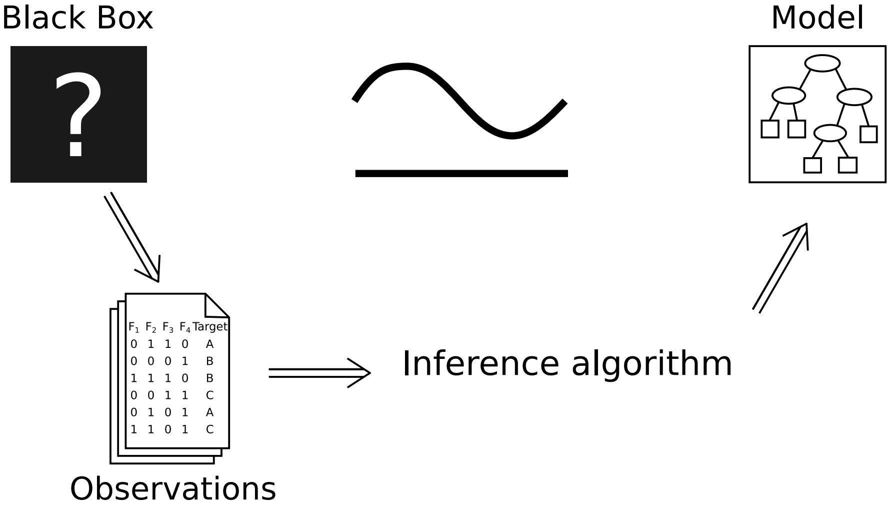
F_1	F_2	F_3	F_4	Target
0	1	1	0	A
0	0	0	1	B
1	1	1	0	B
0	0	1	1	C
0	1	0	1	A
1	1	0	1	C

Observations

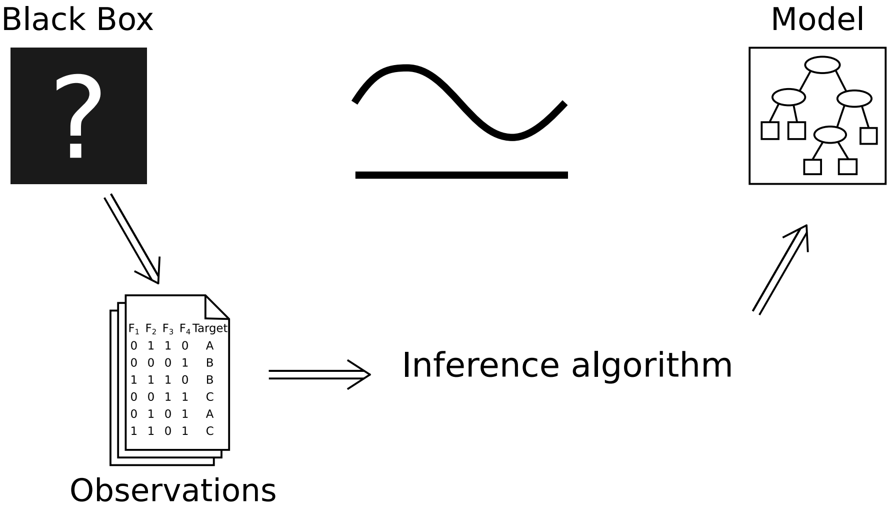
Problem


Black Box
?
Model
F1 F2 F3 F4 Target
0 1 1 0 A
0 0 0 1 B
1 1 1 0 B
0 0 1 1 C
0 1 0 1 A
1 1 0 1 C
Inference algorithm
Observations

Problem


Black Box
?
Model
F1 F2 F3 F4 Target
0 1 1 0 A
0 0 0 1 B
1 1 1 0 B
0 0 1 1 C
0 1 0 1 A
1 1 0 1 C
Observations
Inference algorithm

Problem



Question: How to choose the model?

Parsimony Principle: Definitions

William of Ockham (1287–1347)
Plurality must never be posited without necessity

Parsimony Principle: Definitions

William of Ockham (1287–1347)
Plurality must never be posited without necessity

Modern formulation
Among competing hypotheses, the one with the fewest assumptions should be selected

Parsimony Principle: Definitions

William of Ockham (1287–1347)
Plurality must never be posited without necessity

Modern formulation
Among competing hypotheses, the one with the fewest assumptions should be selected

Another modern formulation
The simplest explanation for some phenomenon is more likely to be accurate than more complicated explanations

Parsimony Principle: Definitions

William of Ockham (1287–1347)
Plurality must never be posited without necessity

Modern formulation
Among competing hypotheses, the one with the fewest assumptions should be selected

Another modern formulation
The simplest explanation for some phenomenon is more likely to be accurate than more complicated explanations

Keep things simple!

Illustration of the Parsimony Principle

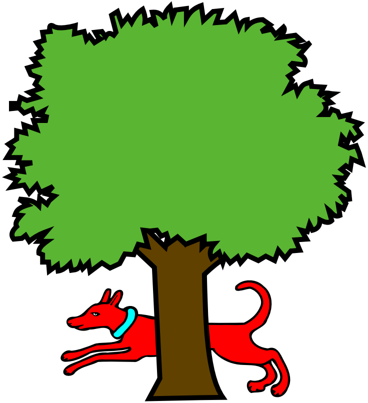


Illustration of the Parsimony Principle

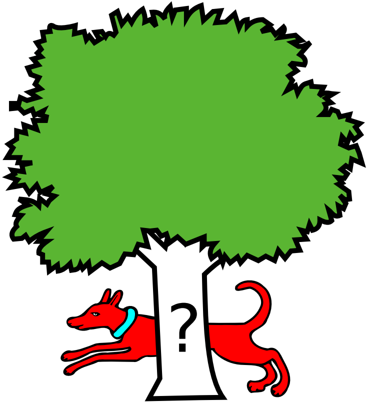


Illustration of the Parsimony Principle

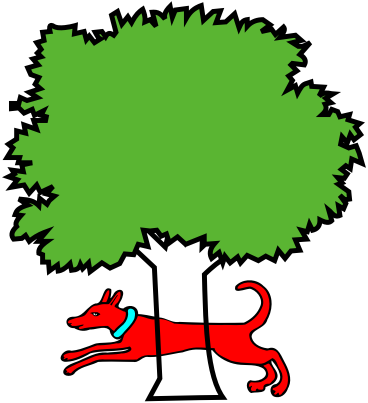
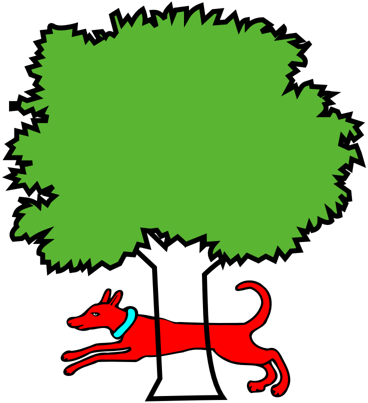


Illustration of the Parsimony Principle



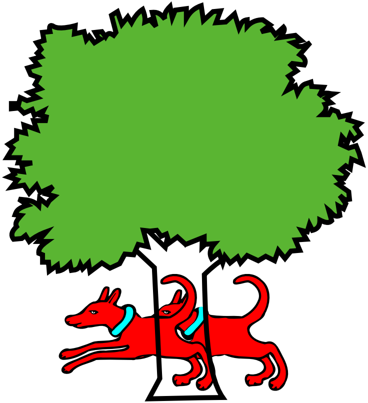
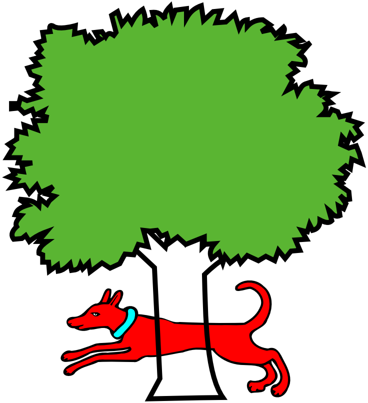


Illustration of the Parsimony Principle



Parsimony Principle for Decision Tree

Question: How to choose the model?

$\Rightarrow$ Choose the "simplest" decision tree

Parsimony Principle for Decision Tree

Question: How to choose the model?

⇒ Choose the "simplest" decision tree

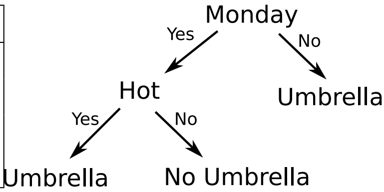
Rain?	Monday?	Hot?	Umbrella?
Yes	No	No	Yes
Yes	Yes	Yes	Yes
Yes	No	Yes	Yes
No	Yes	No	No

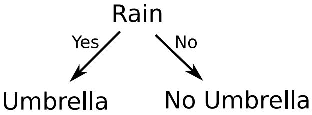
Parsimony Principle for Decision Tree

Question: How to choose the model?

$\Rightarrow$ Choose the "simplest" decision tree

Rain?	Monday?	Hot?	Umbrella?
Yes	No	No	Yes
Yes	Yes	Yes	Yes
Yes	No	Yes	Yes
No	Yes	No	No





Related Works

Learning an optimal decision tree is an NP-complete problem [Hyafil & Rivest '76, Hancock et al. '96]

Related Works

Learning an optimal decision tree is an NP-complete problem [Hyafil & Rivest '76, Hancock et al. '96]

The majority of decision tree inference algorithms are greedy algorithms (C4.5, ID3, CART, ...)

Related Works

Learning an optimal decision tree is an NP-complete problem [Hyafil & Rivest '76, Hancock et al. '96]

The majority of decision tree inference algorithms are greedy algorithms (C4.5, ID3, CART, ...)

Recently, some algorithms have been proposed to infer optimal decision trees:

- [Narodytska et al. IJCAI-18] Inferring decision trees with a **minimum number of nodes** and consistent with the training dataset
- [Verwer et al. AAAI-19] Inferring decision trees with a given depth and with a **minimum number of classification error** on training dataset

Related Works

Learning an optimal decision tree is an NP-complete problem [Hyafil & Rivest '76, Hancock et al. '96]

The majority of decision tree inference algorithms are greedy algorithms (C4.5, ID3, CART, ...)

Recently, some algorithms have been proposed to infer optimal decision trees:

- [Narodytska et al. IJCAI-18] Inferring decision trees with a **minimum number of nodes** and consistent with the training dataset
- [Verwer et al. AAAI-19] Inferring decision trees with a given depth and with a **minimum number of classification error** on training dataset

In this work: Inferring decision trees with a **minimum depth** and consistent with the training dataset

Overview

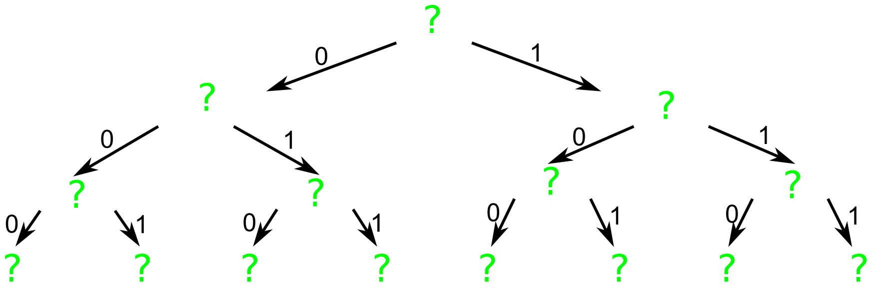
General Idea

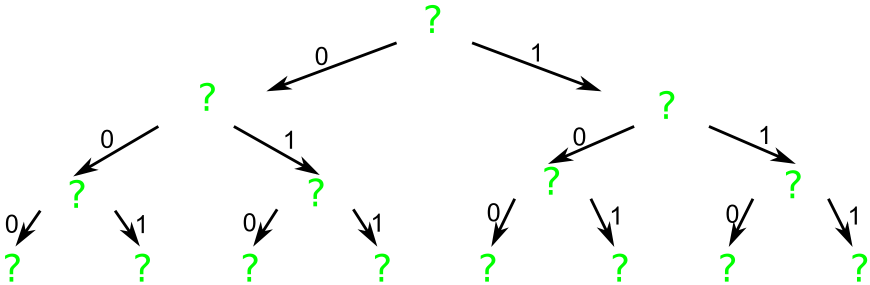
Idea:

- Set the tree depth
- Assume the tree is balanced
- Use a SAT solver to assign each example to a leaf without creating conflict

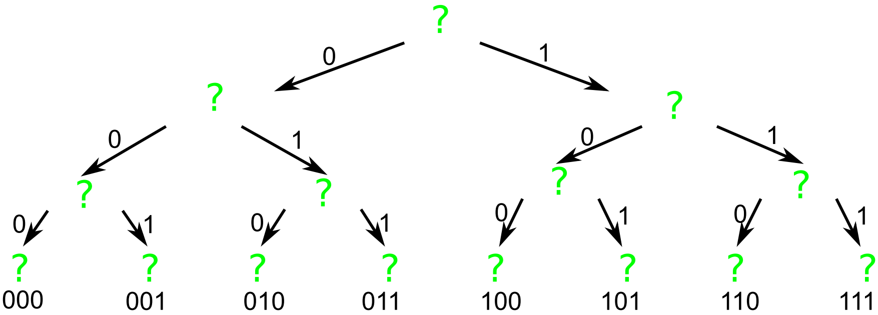
Advantages:

- Since the structure of the tree is fixed, there is no need to learn it
- A binary coding can be assigned to each node, the semantics of which indicate the position of the node in the tree

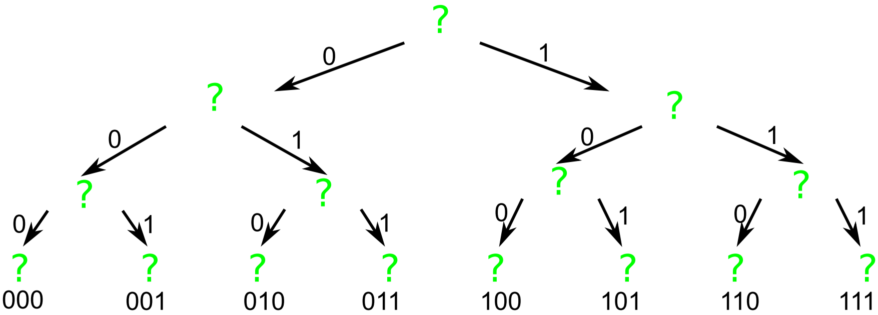

?
0
1
?
?
0
1
0
1
?
?
?
?
0
1
0
1
0
1
0
1
?
?
?
?
?
?
?
?



f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

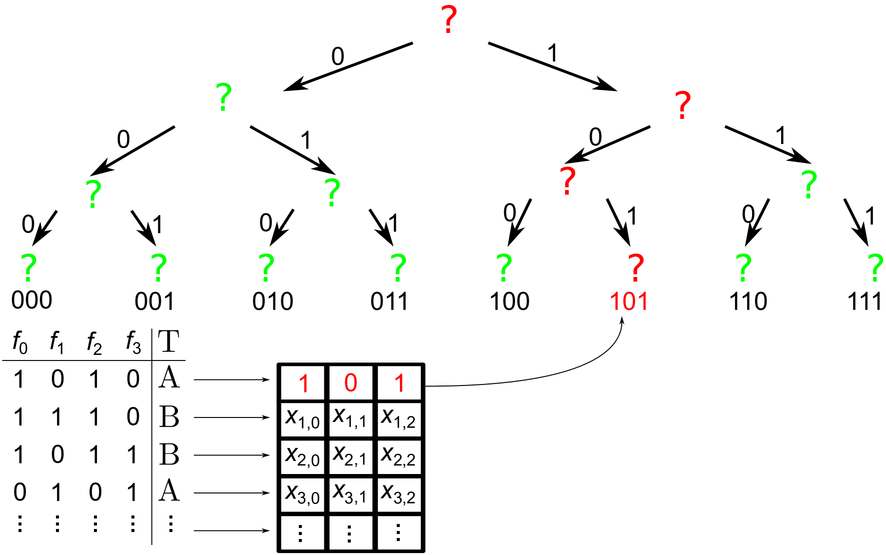


f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

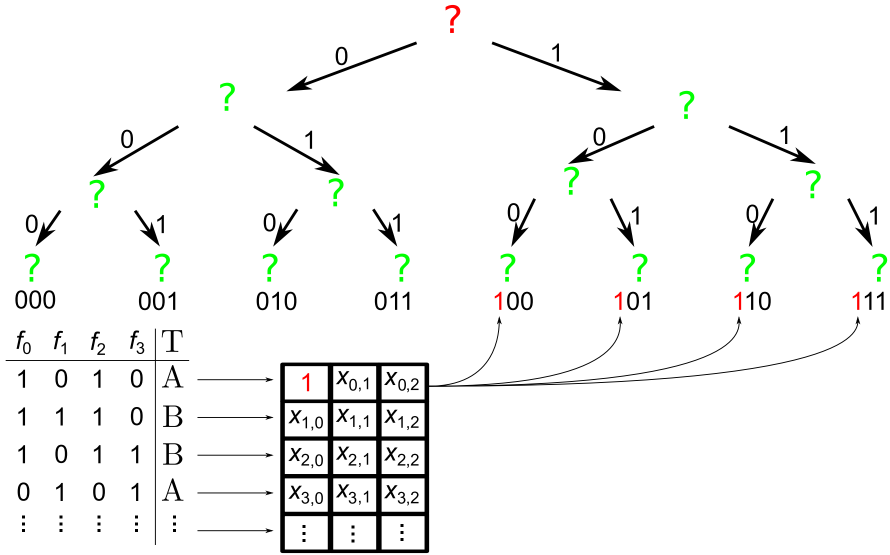


f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

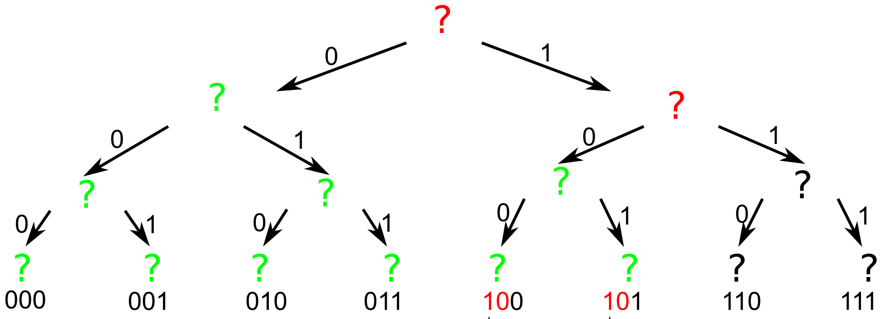
$$\begin{array}{|c|c|c|}
\hline
x_{0,0} & x_{0,1} & x_{0,2} \\ \hline
x_{1,0} & x_{1,1} & x_{1,2} \\ \hline
x_{2,0} & x_{2,1} & x_{2,2} \\ \hline
x_{3,0} & x_{3,1} & x_{3,2} \\ \hline
\vdots & \vdots & \vdots \\ \hline
\end{array}$$



f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

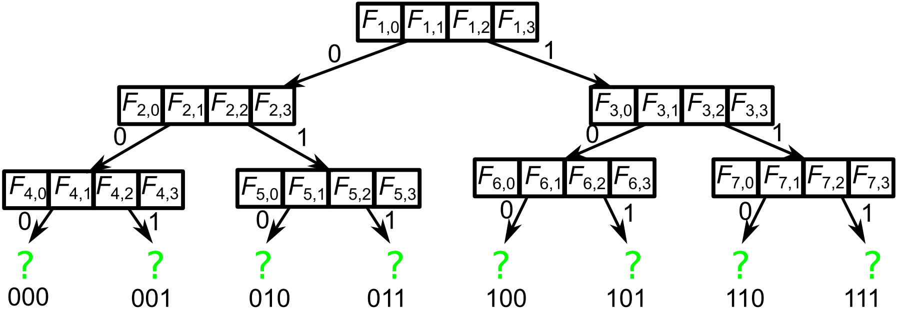


f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮



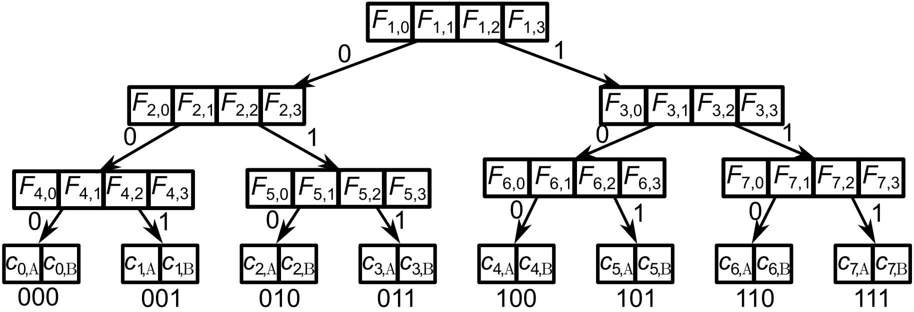
f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

$$\begin{array}{|c|c|c|}
\hline
1 & 0 & x_{0,2} \\
\hline
x_{1,0} & x_{1,1} & x_{1,2} \\
\hline
x_{2,0} & x_{2,1} & x_{2,2} \\
\hline
x_{3,0} & x_{3,1} & x_{3,2} \\
\hline
\vdots & \vdots & \vdots \\
\hline
\end{array}$$

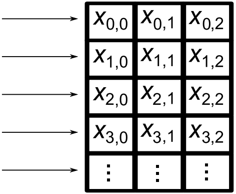


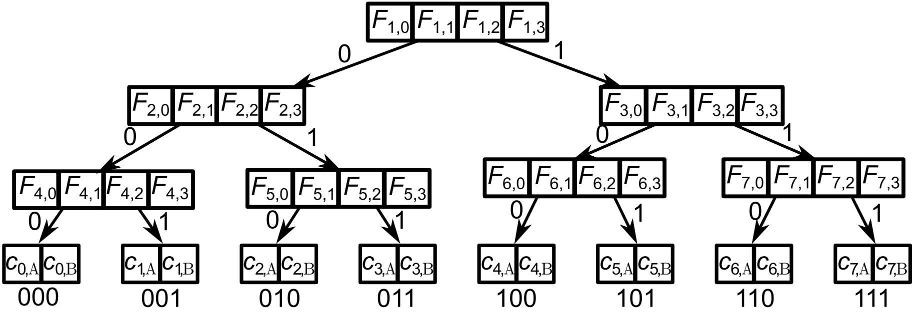
f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

$x_{0,0}$	$x_{0,1}$	$x_{0,2}$
$x_{1,0}$	$x_{1,1}$	$x_{1,2}$
$x_{2,0}$	$x_{2,1}$	$x_{2,2}$
$x_{3,0}$	$x_{3,1}$	$x_{3,2}$
⋮	⋮	⋮

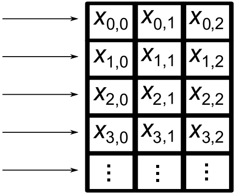


f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

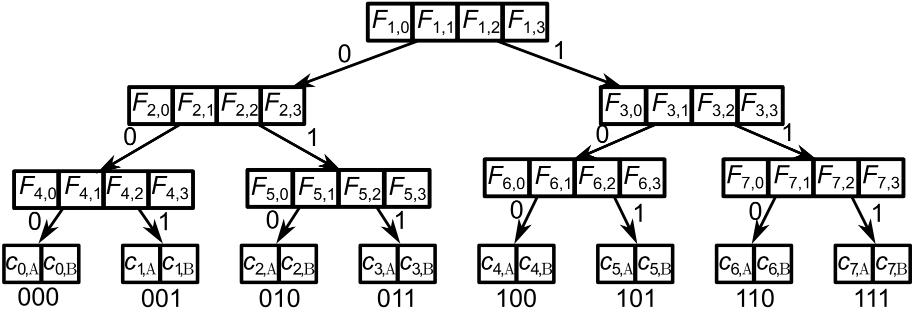




f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮



$$\forall i \in [1, 2^k - 1] : F_{i,0} \vee ... \vee F_{i,m-1}$$

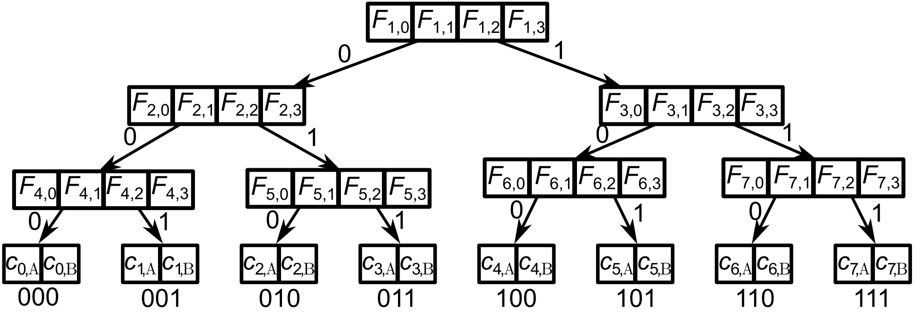


f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

$x_{0,0}$	$x_{0,1}$	$x_{0,2}$
$x_{1,0}$	$x_{1,1}$	$x_{1,2}$
$x_{2,0}$	$x_{2,1}$	$x_{2,2}$
$x_{3,0}$	$x_{3,1}$	$x_{3,2}$
⋮	⋮	⋮

$$\forall i \in [1, 2^k - 1] : F_{i,0} \vee ... \vee F_{i,m-1}$$

$$\forall i \in [1, 2^k - 1], 0 \leq f_1 \leq f_2 < m : F_{i,f_1} \vee F_{i,f_2}$$



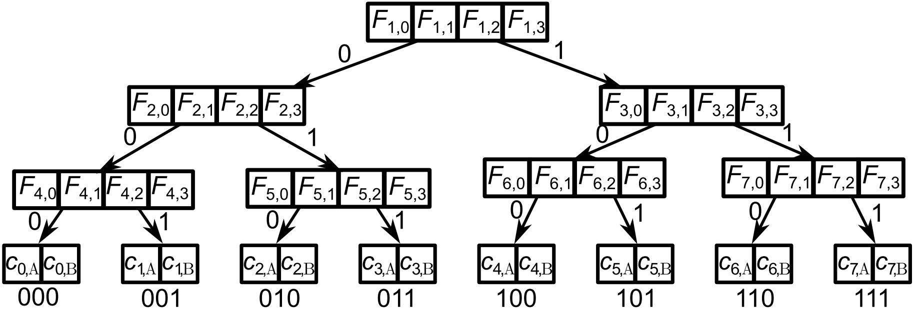
f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮

$$\begin{array}{|c|c|c|}\hline x_{0,0} & x_{0,1} & x_{0,2} \\ \hline x_{1,0} & x_{1,1} & x_{1,2} \\ \hline x_{2,0} & x_{2,1} & x_{2,2} \\ \hline x_{3,0} & x_{3,1} & x_{3,2} \\ \hline \vdots & \vdots & \vdots \\ \hline \end{array}$$

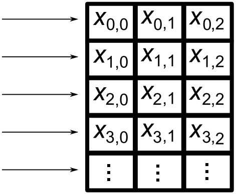
$$\forall i \in [1, 2^k - 1] : F_{i,0} \vee ... \vee F_{i,m-1}$$

$$\forall i \in [1, 2^k - 1], 0 \leq f_1 \leq f_2 < m : F_{i,f_1} \vee F_{i,f_2}$$

$$\forall i, f | e_i[f] = 0 : X_{i,j} \Rightarrow \neg F_{X_i[..j],f}$$
$$\forall i, f | e_i[f] = 1 : \neg X_{i,j} \Rightarrow \neg F_{X_i[..j],f}$$



f_0	f_1	f_2	f_3	T
1	0	1	0	A
1	1	1	0	B
1	0	1	1	B
0	1	0	1	A
⋮	⋮	⋮	⋮	⋮



$$\forall i \in [1, 2^k - 1] : F_{i,0} \vee ... \vee F_{i,m-1}$$

$$\forall i \in [1, 2^k - 1], 0 \leq f_1 \leq f_2 < m : F_{i,f_1} \vee F_{i,f_2}$$

$$\forall i, f | e_i[f] = 0 : X_{i,j} \Rightarrow \neg F_{X_i[..j],f}$$

$$\forall i, f | e_i[f] = 1 : \neg X_{i,j} \Rightarrow \neg F_{X_i[..j],f}$$

$$\forall v \in [0, 2^k - 1] : (X_i = v) \Rightarrow C_{v,label(e_i)} \wedge \neg C_{v,\neg label(e_i)}$$

Iterative Algorithm

Input: The maximum depth k of the tree to infer and the set of training examples $\mathcal{E} = \{\mathcal{E}_0, \mathcal{E}_1, ..., \mathcal{E}_{c-1}\}$

$C := \mathit{StructureConstraints}()$
while C *is satisfiable* **do**
    Let T be a decision tree of a solution of C
    if $\mathcal{E} \subseteq T$ **then**
        return T
    end
    Let $e \in \mathcal{E}_a$ be an example mislabeled by T
    $C := C \wedge \mathit{FeatureConstraints}(e) \wedge \mathit{ClassConstraints}(e, a)$
end
return "No solution"

Minimizing the Number of Nodes

For each $i \in [0, 2^k - 1]$ and each class $a \in [0, c - 1]$, we have the clauses:

$$\neg C_{i,a} \vee U_i \tag{1}$$

For each $i \in [0, 2^k - 1]$ and each class $j \in [0, MaxNodes + 1]$, we have the clauses:

$$\neg H_{i,j} \vee H_{i+1,j} \tag{2}$$

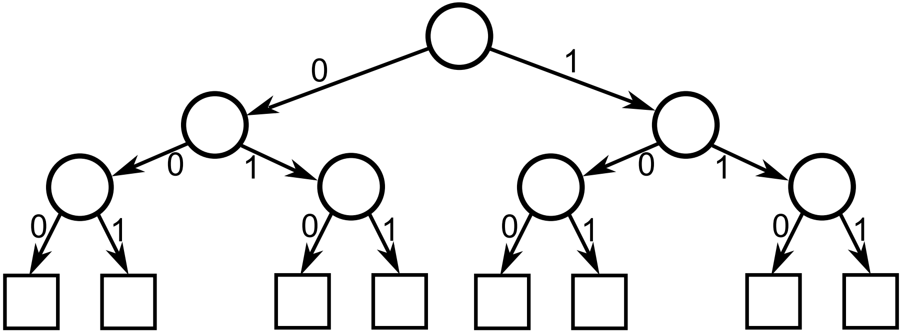
For each $i \in [0, 2^k - 1]$ and each class $j \in [0, MaxNodes + 1]$, we have the clauses:

$$\neg U_i \vee \neg H_{i,j} \vee H_{i+1,j+1} \tag{3}$$

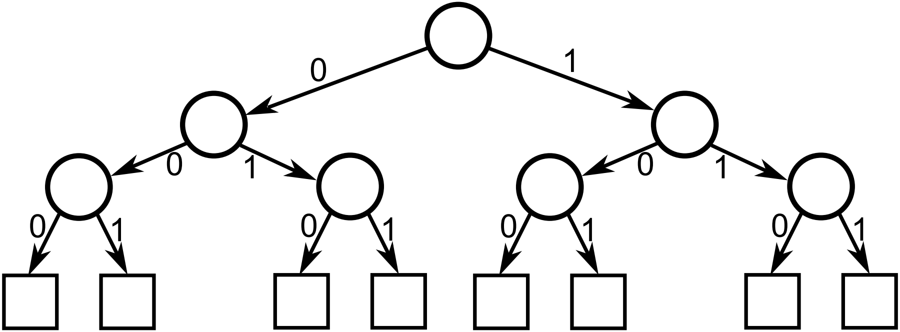
Finally, we assign the start and end of the counter H :

$$\neg H_{2^{k+1}, \lfloor MaxNodes/2 \rfloor + 2} \wedge H_{0,0} \tag{4}$$

Minimizing the Number of Nodes

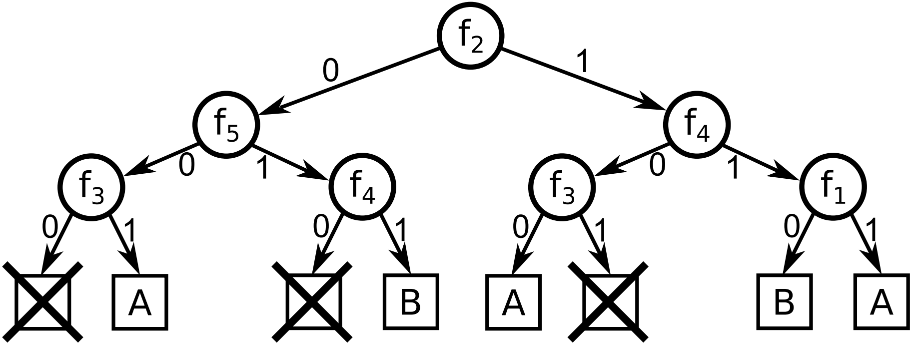


Minimizing the Number of Nodes



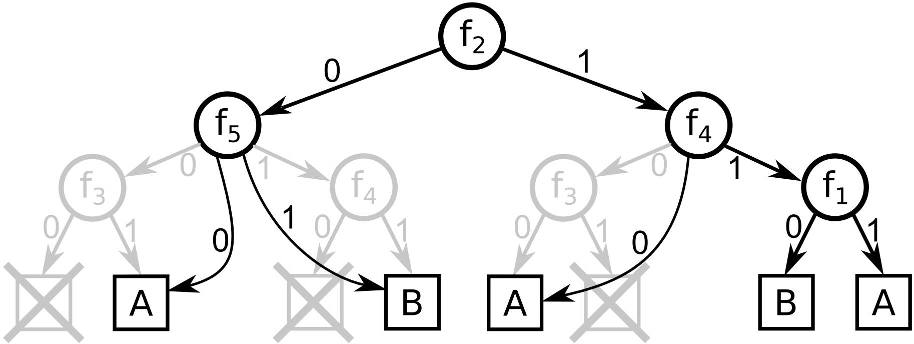
$$\#leaves = \#nodes + 1$$

Minimizing the Number of Nodes



$$\#leaves = \#nodes + 1$$

Minimizing the Number of Nodes



$$\#leaves = \#nodes + 1$$

Overview

Related Works 1

Problem: Inferring decision trees with a **minimal number of nodes** without minimizing the depth

Algorithms evaluated:

- Algorithm **DT1** from Naradytska et al. [IJCAI 2018]
- Algorithm **DT2** from Bessiere et co. [CP 2009]
- First version of our algorithm **DT_depth**[1]
- Second version of our algorithm **DT_size**[1]

[1] Code available at: `https://github.com/FlorentAvellaneda/InferDT/`

Benchmark 1

Table 1: Benchmark for "Mouse" dataset (70 examples, 45 features, 2 classes)

Algo	Time (s)	Examples used	k	Nodes	acc.
$DT2$	577	70	4	15	83.5%
$DT1$	12.9	70	4	15	83.5%
DT_depth	**0.015**	33	4	31	85.8%
DT_size	0.075	37	4	15	83.5%

Table 2: Benchmark for "Car" dataset (1727 examples, 21 features, 2 classes)

Algo	Time (s)	Examples used	k	Nodes	acc.
$DT1$	684	173	7	23.67	55%
DT_depth	**170**	635	8	511	**98.8%**
DT_size	260	758	8	136	**98.8%**

Related Works 2

Problem: Inferring decision trees with a given depth and with a **minimum number of classification error** on training dataset

Algorithms evaluated:

- Algorithm **BinOCT**∗ from Verwer and Zhang [AAAI 2019]
- Algorithm **CART** from sciki-learn with its default parameter setting
- Algorithm **OCT** from Bertsimas and Dunn [Machine Learning 2017]
- First version of our algorithm **DT_depth**
- Second version of our algorithm **DT_size**

Benchmark 2

Dataset	Examples	Features	Classes
iris	150	114	3
Monk1	124	17	2
Monk2	169	17	2
Monk3	122	17	2
wine	178	1276	3
balance	625	20	3

Dataset	*DT_depth*			*DT_size*			*BinOCT**	*CART*	*OCT*
	time (sec.)	acc.	k	time (sec.)	acc.	n	acc.	acc.	acc.
iris	0.018	92.9%	3	0.03	93.2%	10.6	**98.4%**	92.4%	93.5%
Monk1	0.024	90.3%	4.4	0.08	**95.5%**	17	87.1%	76.8%	74.2%
Monk2	0.19	70.2%	5.8	9.1	**74.0%**	47.8	63.3%	63.3%	54.0%
Monk3	0.03	78.1%	4.8	0.21	82.6%	23.4	93.5%	**94.2%**	**94.2%**
wine	0.6	89.3%	3	1.2	92.0%	7.8	92.0%	88.9%	**94.2%**
balance	50	**93.0%**	8	183	92.6%	268	78.9%	77.5%	71.6%
Average		85.6%			**88.3%**		85.5%	82.18%	81.1%

Table 3: Benchmark comparing algorithms *DT_depth*, *DT_size*, *BinOCT**, *CART* and *OCT*.

Benchmark on Artificial Dataset

Problem: We randomly generated 1000 decision trees of depth 5, with 10 features and 2 classes and we used them to randomly generate learning examples. We check if the models we inferred are equivalent to the these generated decision trees.

Algorithms evaluated:

- Algorithm *C4.5* implemented in the Weka tool under the name of *J48*
- First version of our algorithm **DT_depth**
- Second version of our algorithm **DT_size**

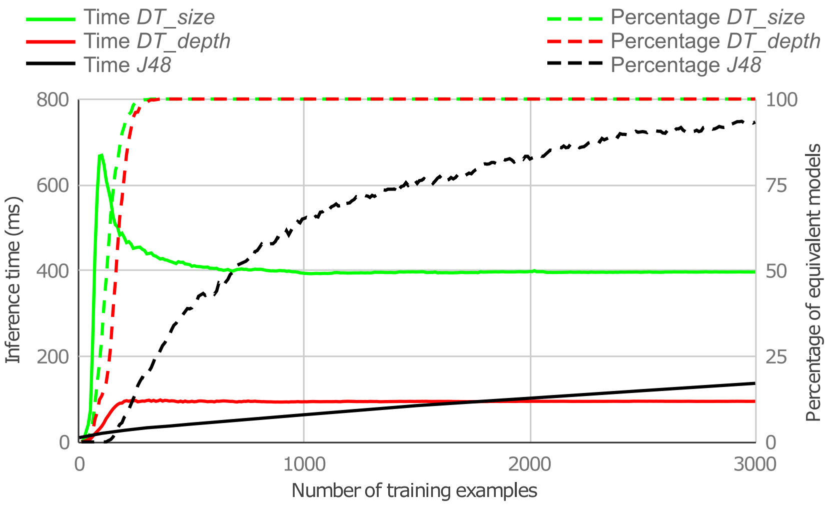


Figure 1: Chart of the average time and accuracy percentage

Overview

Conclusion

Result:

- Efficient algorithm for inferring optimal decision trees
- Incremental algorithm whose performance does not deteriorate with the number of observations
- Optimal decision trees on popular datasets

Future Works:

- Using alternative definitions of optimality
- Considering noisy data
- Improving performance

Thank you