LEARNING OPTIMAL DECISION TREES USING CONSTRAINT PROGRAMMING

IJCAI20

Hélène Verhaeghe[1], Siegfried Nijssen[1], Gilles Pesant[2], Claude-Guy Quimper[3], and Pierre Schaus[1]

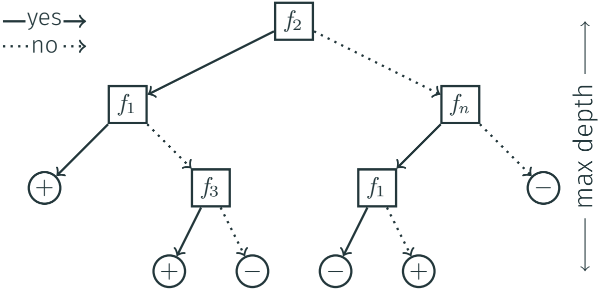
January 2021

[1] ICTEAM, UCLouvain, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium, $\{firstname.lastname\}@uclouvain.be$
[2] Polytechnique Montréal, Montréal, Canada, $gilles.pesant@polymtl.ca$
[3] Université Laval, Québec, Canada, $claude-guy.quimper@ift.ulaval.ca$

UCLouvain

Database

f_1	f_2	f_3	$\dots$	f_n	c
1	0	1	$\dots$	1	$+$
0	1	0	$\dots$	1	$-$
1	1	0	$\dots$	0	$+$
0	0	0	$\dots$	0	$+$
1	0	0	$\dots$	0	$+$
0	1	1	$\dots$	1	$-$
1	1	1	$\dots$	0	$-$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
1	1	1	$\dots$	1	$+$



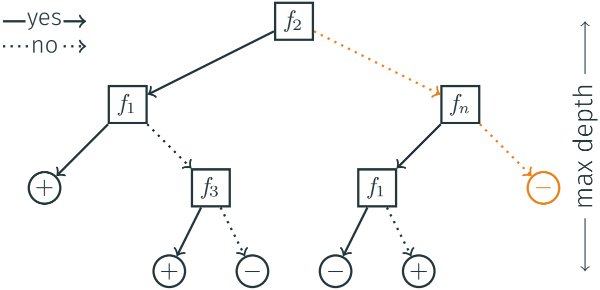
$$\min \sum (pred(i) - c(i))$$

Database

f_1	f_2	f_3	$\dots$	f_n	c
1	0	1	$\dots$	1	+
0	1	0	$\dots$	1	−
1	1	0	$\dots$	0	+
0	0	0	$\dots$	0	+
1	0	0	$\dots$	0	+
0	1	1	$\dots$	1	−
1	1	1	$\dots$	0	−
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
1	1	1	$\dots$	1	+

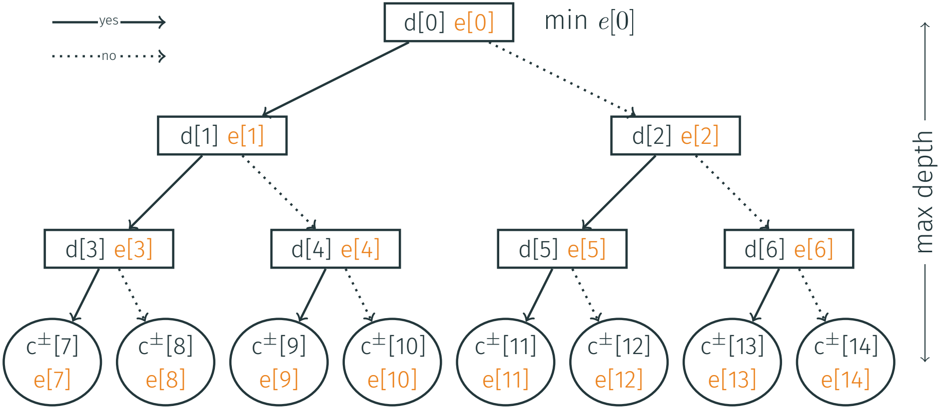
New sample

0	0	1	$\dots$	0	−



$$\min \sum (pred(i) - c(i))$$

- Mining optimal decision trees from itemset lattices, Nijssen, S., Fromont, E., 2007
- Minimising decision tree size as combinatorial optimisation, Bessiere, C., Hebrard, E., O'Sullivan, B., 2009
- Optimal constraint-based decision tree induction from itemset lattices,Nijssen, S., Fromont, É., 2010
- **Optimal classification trees**, Bertsimas, D., Dunn, J., 2017
- Learning optimal decision trees with sat, Narodytska, N., Ignatiev, A., Pereira, F., Marques-Silva, J., RAS, I., 2018
- Learning optimal and fair decision trees for non-discriminative decision-making, Aghaei, S., Azizi, M.J., Vayanos, P., 2019
- Learning optimal classification trees using a binary linear program formulation, Verwer, S., Zhang, Y., 2019



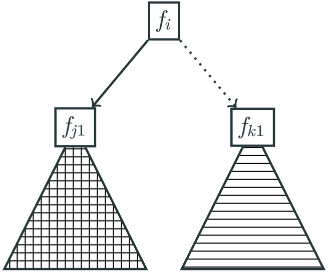
$$dom(d[i]) = \{0, 1, ..., n\}$$

$$dom(c^+[i]) = \{0, ..., N\}$$
$$dom(c^-[i]) = \{0, ..., N\}$$

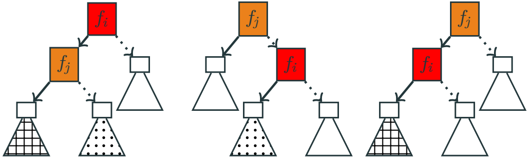
$$dom(e[i]) = \{0, ..., N\}$$

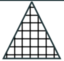
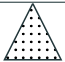
Constraints

- enforce the tree structure (AllDifferent, CoverSize,...)
- right computation of the error (allow minimization objective)
- imposing minimum at leaf (additionnal constraint)
- constraints avoiding useless decisions
- redundant constraints (speed improvement)



min T = min $left(T)$ + min $right(T)$



	yes	no	hash
	f_i f_j		$f_i, f_j -$
	f_i	f_j	$f_i - f_j$

	$N_{\min} = 1$			$N_{\min} = 5$			
	DL8	BinOCT	CP	DL8	CP	CP-c	CP-m
Proven optimality	49(64%)	13(17%)	**57**(75%)	54(71%)	56(74%)	56(74%)	**58**(76%)
Best solution found	49(64%)	21(28%)	**76**(100%)	54(71%)	**74**(97%)	**74**(97%)	70(92%)
Fastest	23(30%)	11(14%)	**49**(64%)	28(37%)	**40**(53%)	33(43%)	22(29%)
Time out	27(36%)	63(83%)	**19**(25%)	22(29%)	21(28%)	21(28%)	**19**(25%)

23 instances, depths from 2 to 5, 10 min TO

DL8: Dynamic programming approach using frequent itemsets mining

BinOCT: MIP-based approach running on CPLEX

To summarize

- efficient method
- cp based
- exploits the structure of the problem
- anytime best solution

To go further

- multi-class decision trees
- continuous features through binarization
- other sum-based cost functions
- ...

Thank you for listening!

Any questions?

Also, our extended journal paper is out!

https://link.springer.com/article/10.1007/s10601-020-09312-3