Aalto-yliopisto
Perustieteiden
korkeakoulu

Training decision trees using mixed-integer optimisation

Joel Vääräniemi

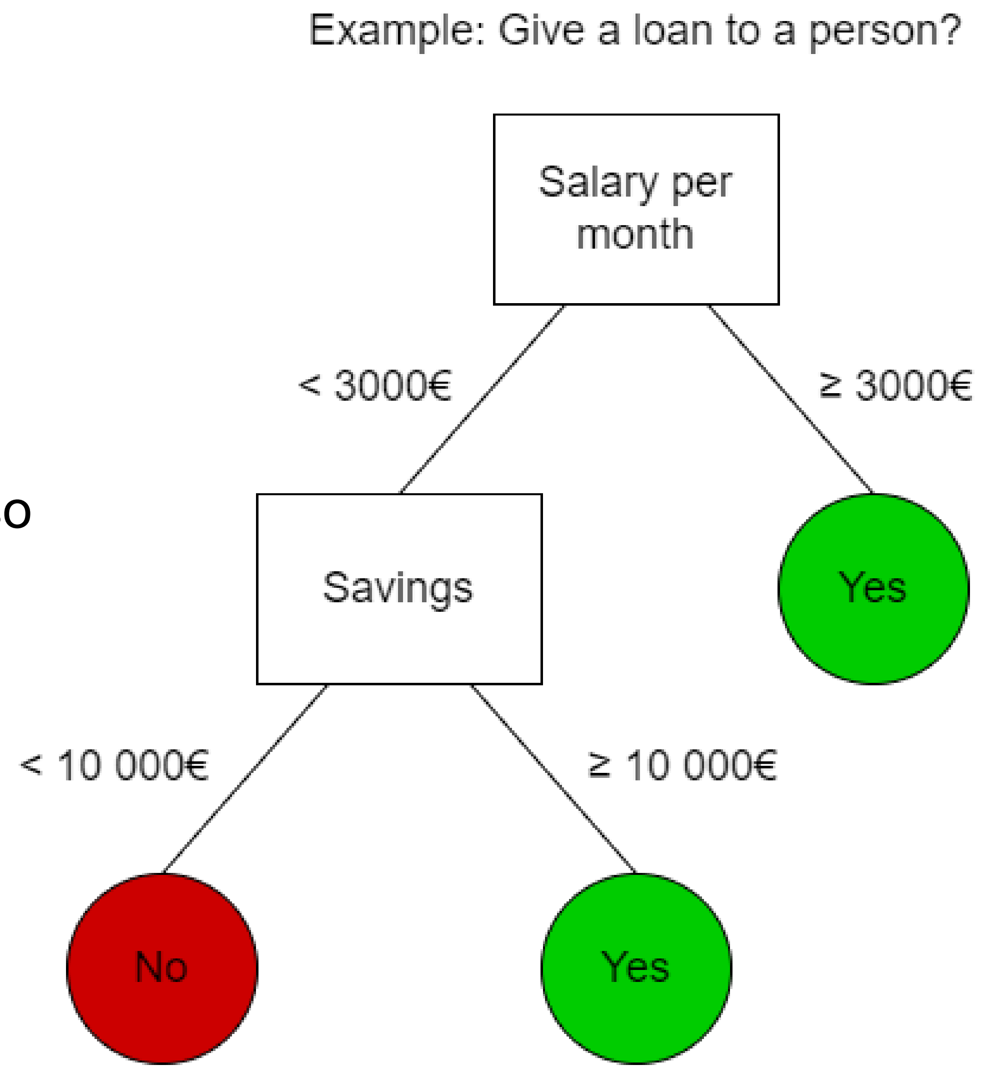
16.06.2023

Instructor: *Nikita Belyak*
Supervisor: *Fabricio Oliveira*

Systeemianalyysin
laboratorio

Decision trees

- Branch and leaf nodes
- Popular choice for classification problems
  - Usable for regression also
- Easy to interpret
- Many real-world applications



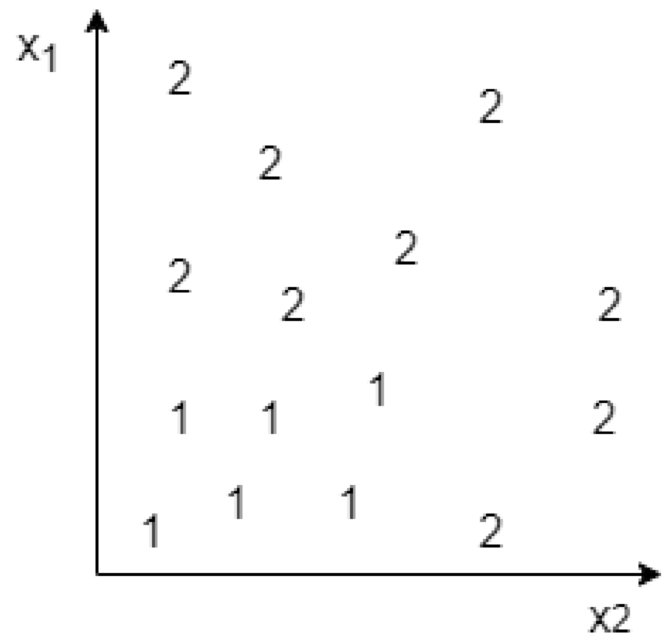
Top-down method: Classification and regression trees (CART)

- Top-down approach, starting from the root node
- Every split is made in isolation without information about future splits
- Split determined by optimisation problem (minimising impurity score)
- Creates a binary tree (splits into two groups in every branch node)

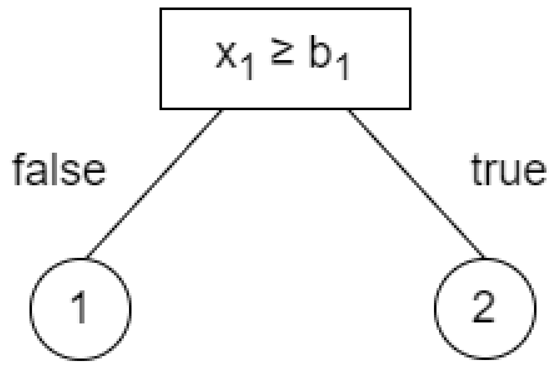
(Breiman et al., 1984)

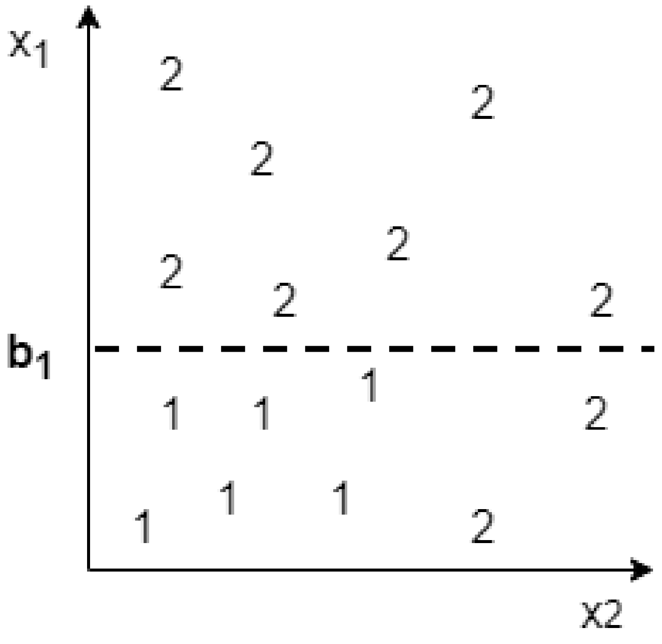
CART algorithm

- Example: two-dimensional data, two class labels

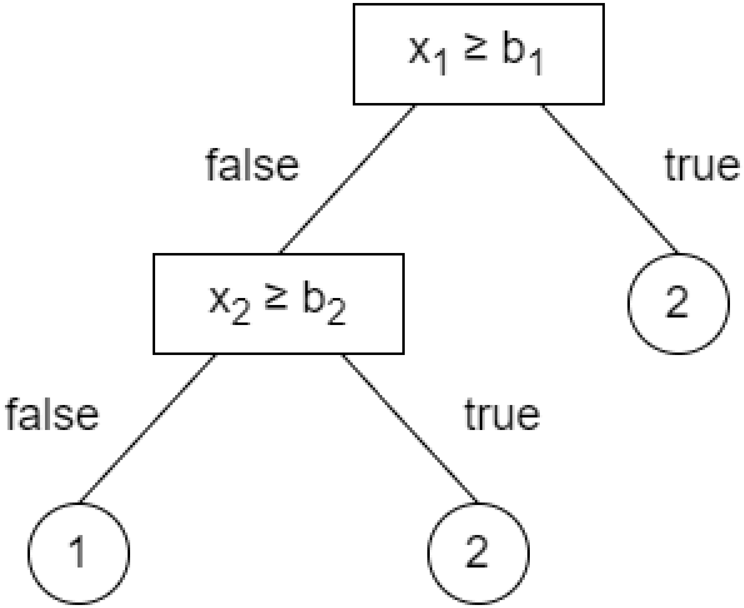


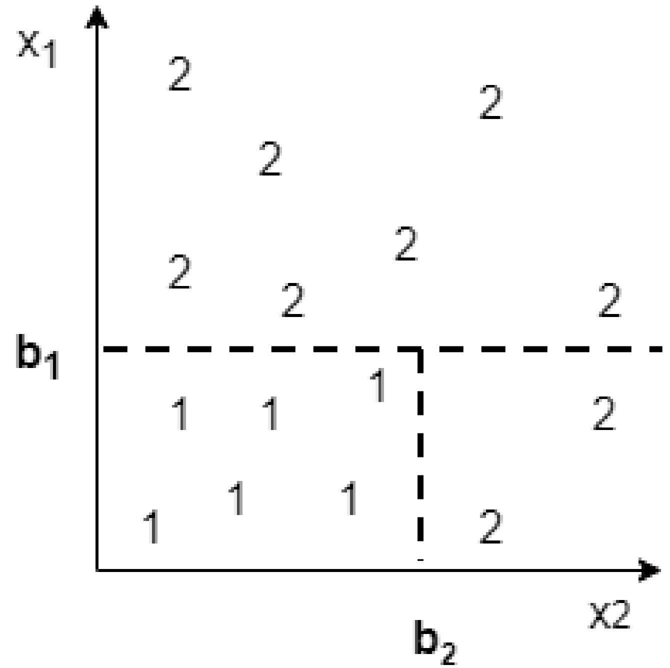
CART algorithm





CART algorithm



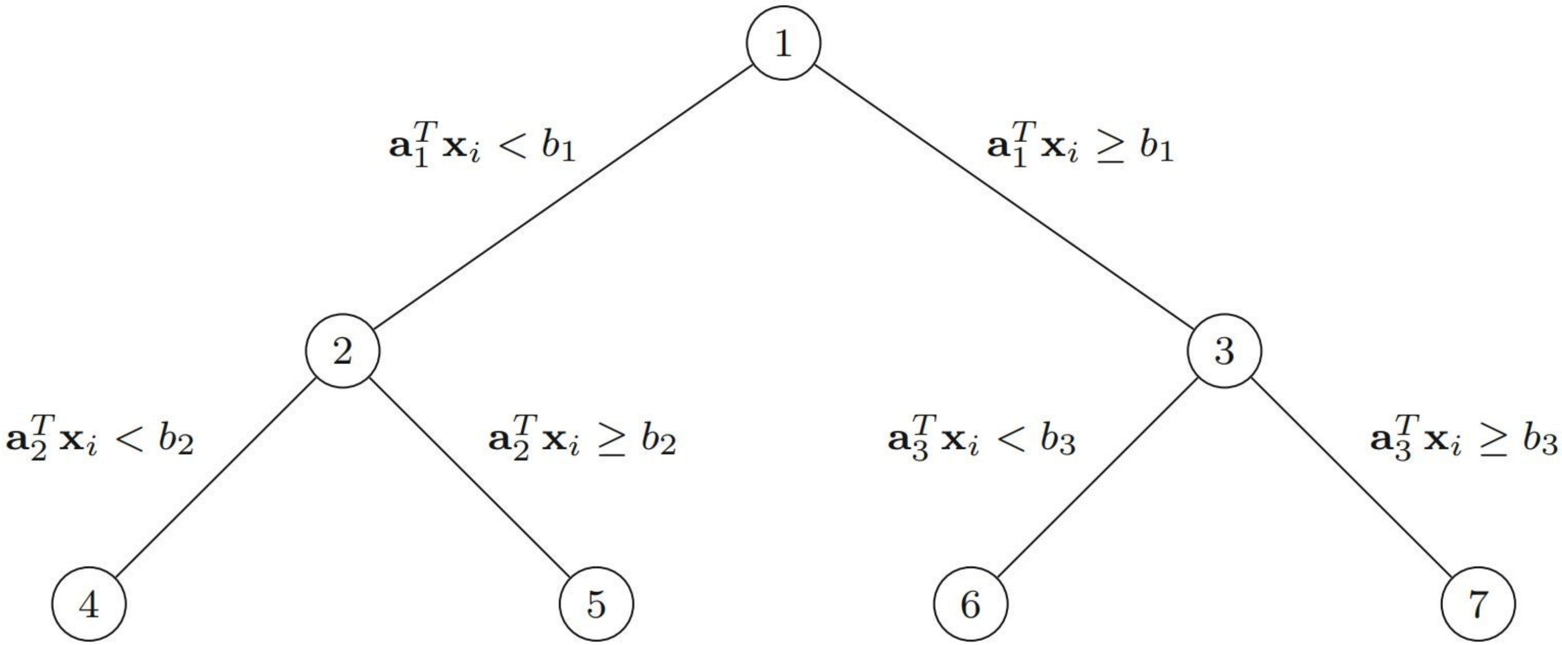


MIO method: Optimal classification trees (OCT) model

- Creates the whole tree at once with full knowledge of all possible splits
- Results in a globally optimal decision tree
  - Objective function: minimise misclassification error given the preference on the complexity of the tree structure
- Construction is an NP-hard problem
  - However, reasonable with modern hardware and solvers (D. Bertsimas and J. Dunn., 2017)

MIO formulation for OCT model

- Framework: binary tree, univariate splits, maximal tree
- 3 hyper-parameters: maximum tree depth, minimum leaf size, complexity parameter



Aims

- Implement the OCT as MIO formulation in Julia, using Gurobi as the optimiser
- Model testing: train the OCT for given data with different combinations of hyper-parameters
  - Focus on the training time, as no distinguished literature thereof
  - Misclassification error (in-sample accuracy)
- Results will be compared to the top-down approach (CART)
  - Trade-off between training time and classification accuracy

Design of the experiments

- We train the trees on the Iris data (n = 150, p = 4, K = 3) https://archive.ics.uci.edu/ml/datasets/Iris
- Three different approaches:
  - Normal OCT
  - OCT with warm starts (feasible initial solutions created with CART)
  - OCT without complexity penalty, predetermined number of maximum splits
- Cut-off time for a single MIO optimisation problem: 30 min

Hyper-parameters

- Maximum depth of tree, D:
  - 2, 3, 4
- Minimum leaf size, N_{min}:
  - 8, 15
- Complexity parameter, α:
  - 0.0, 0.1, 0.5, 0.9

Results (1): Normal OCT vs. CART

- α = 0 and D = 3 or 4: No result in 30 minutes
- α value heavily influences the training time and accuracy
- Little to no improvement in training accuracy when using OCT vs. CART

D	N_{min}	α	Training time	In-sample accuracy	CART training time	CART in-sample accuracy
2	8	0.000	41.260	0.960	0.000	0.953
		0.100	39.070	0.960		
		0.500	15.880	0.960		
		0.900	18.660	0.667		
	15	0.000	55.870	0.960	0.000	0.953
		0.100	30.050	0.960		
		0.500	17.150	0.960		
		0.900	14.220	0.667		
3	8	0.000	1800.000	–	0.000	0.960
		0.100	65.070	0.960		
		0.500	8.990	0.960		
		0.900	9.650	0.667		
	15	0.000	1800.000	–	0.000	0.953
		0.100	42.830	0.960		
		0.500	5.050	0.960		
		0.900	20.330	0.667		
4	8	0.000	1800.000	–	0.000	0.960
		0.100	234.120	0.960		
		0.500	36.490	0.960		
		0.900	14.870	0.667		
	15	0.000	1800.000	–	0.000	0.953
		0.100	371.820	0.960		
		0.500	16.090	0.960		
		0.900	24.940	0.667		

Results (2): OCT with warm starts

- Using warm starts reduces the training time to some extent
- *D* = 3 and 4: No result in 30 minutes

Training times:

D	N_{min}	Normal OCT	With CART warm start
2	8	41.260	25.410
	15	55.870	40.510
3	8	1800.000	1800.000
	15	1800.000	1800.000
4	8	1800.000	1800.000
	15	1800.000	1800.000

Results (3): OCT, maximum number of splits approach

- *C* = 4 and *D* = 3 or 4: No result in 30 minutes
- Shows potential of OCT: in-sample accuracy of 0.973 in some cases

D	N_{min}	C	Training time	In-sample accuracy
2	8	3	40.020	0.960
		2	10.200	0.960
	15	3	55.870	0.960
		2	20.630	0.960
3	8	4	1800.000	–
		3	583.900	0.973
		2	17.340	0.960
	15	4	1800.000	–
		3	655.310	0.973
		2	15.710	0.960
4	8	4	1800.000	–
		3	677.540	0.973
		2	10.090	0.960
	15	4	1800.000	–
		3	577.740	0.973
		2	20.190	0.960

References

- D. Bertsimas and J. Dunn. Optimal classification trees. Machine Learning, 106: 1039–1082, 2017.
- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. Wadsworth and Brooks, Monterey, CA, 1984.
- D. Dua and C. Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Appendix: MIO formulation of OCT

$$\min \quad \frac{1}{\hat{L}} \sum_{t \in \mathcal{T}_L} L_t + \alpha \cdot C$$

s.t.

$$\begin{aligned}
& L_t \geq N_t - N_{kt} - n(1 - c_{kt}), && \forall t \in \mathcal{T}_L, k \in [K], \\
& L_t \leq N_t - N_{kt} + n c_{kt}, && \forall t \in \mathcal{T}_L, k \in [K], \\
& L_t \geq 0, && \forall t \in \mathcal{T}_L, \\
& N_{kt} = \sum_{i: y_i = k} z_{it}, && \forall t \in \mathcal{T}_L, k \in [K], \\
& N_t = \sum_{i=1}^{n} z_{it}, && \forall t \in \mathcal{T}_L, \\
& \sum_{k=1}^{K} c_{kt} = l_t, && \forall t \in \mathcal{T}_L, \\
& C = \sum_{t \in \mathcal{T}_B} d_t,
\end{aligned}$$

$$\begin{aligned}
& \mathbf{a}_m^{\mathsf{T}} \mathbf{x}_i \geq b_m - (1 - z_{it}), && \forall i \in [n], t \in \mathcal{T}_L, m \in A_R(t), \\
& \mathbf{a}_m^{\mathsf{T}} (\mathbf{x}_i + \boldsymbol{\epsilon} - \epsilon_{min}) + \epsilon_{min} \\
& \qquad \leq b_m + (1 + \epsilon_{max})(1 - z_{it}), && \forall i \in [n], t \in \mathcal{T}_L, m \in A_L(t), \\
& \sum_{t \in \mathcal{T}_L} z_{it} = 1, && \forall i \in [n], \\
& z_{it} \leq l_t, && \forall t \in \mathcal{T}_L, \\
& \sum_{i=1}^{n} z_{it} \geq N_{min} l_t, && \forall t \in \mathcal{T}_L, \\
& \sum_{j=1}^{p} a_{jt} = d_t, && \forall t \in \mathcal{T}_B, \\
& 0 \leq b_t \leq d_t, && \forall t \in \mathcal{T}_B, \\
& d_t \leq d_{p(t)}, && \forall t \in \mathcal{T}_B \setminus \{1\}, \\
& z_{it}, l_t, c_{kt} \in \{0, 1\}, && \forall i \in [n], k \in [K], t \in \mathcal{T}_L, \\
& a_{jt}, d_t \in \{0, 1\}, && \forall j \in [p], t \in \mathcal{T}_B,
\end{aligned}$$