AnyDTree: Anytime Algorithm for Perfect Decision Trees

Author: Iulia Hosu
Supervisor: Ir. J.G.M. van der Linden
Responsible Professor: Dr. Emir Demirović

1. Introduction

Decision Trees:

- Interpretable models that can detect non-linear relations
- **Perfect decision trees** correctly classify all training data
- **Smaller perfect trees** preferable due to: Increased **interpretability,** faster **prediction speed**, and lower **memory footprint**.

Finding the smallest perfect tree is **NP-hard**. This motivates the need of an **anytime solver**, quickly finding an initial perfect tree and refining it over time.

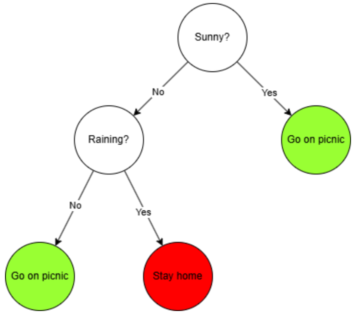

Figure 1: Example of decision tree

2. Objective

Design an anytime algorithm that:

- Always maintains **100% training accuracy**.
- Progressively reduces the **size of the decision tree**.
- Guarantees **optimality** given sufficient time.

Focus on **binary classification** with **binary features**.

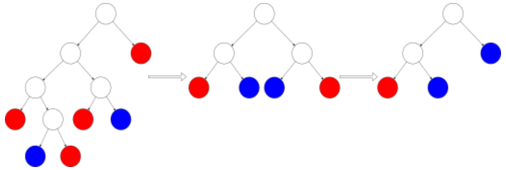

Figure 2: Example of size reduction in decision trees

3. Methodology

Base algorithm:

- Exhaustive search
- Anytime behaviour achieved through the search order: always finish growing the current tree before back-tracking to explore alternative splits.

Optimizations:

- **Memoization:** Cache results of sub-problems to avoid redundant computations.
  - Use a cache limit to reduce memory usage
- **Heuristic splitting order:** Explore promising features first to find smaller trees faster.
- **Pruning:** Use upper and lower-bound estimates to prune branches that cannot yield a smaller tree than the current best.

4. Results

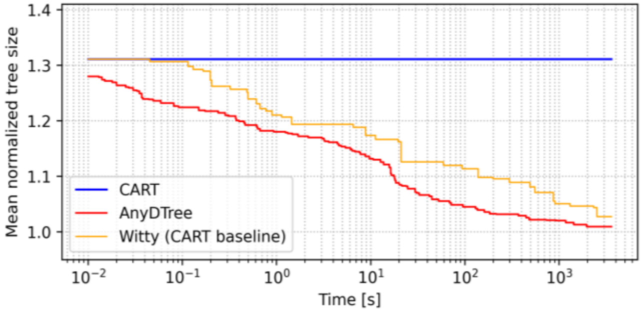

Figure 3: Mean normalized tree size over time on the 46 datasets whose optimum is known. For each dataset, we consider Witty's solution equal to CART until Witty finishes. Lower is better.

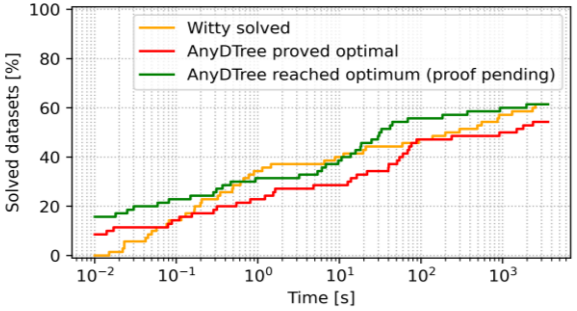

Figure 4: Cumulative percentage of the 70 datasets solved to optimality over time. Higher is better

- **Benchmark**: 70 binary-classification variants of 35 UCI datasets, part of the set shipped in Witty's archive.
- **Anytime Performance**: Median *confined primal integral* (CPI) **0 .00034**, improved over Witty **0.00059** and CART **0.20** (Wilcoxon *P* < 0 .001).
- **Reaching Optimality**: No statistically significant difference in time-to-optimality from Witty; log-rank tests in every bucket give *P* > 0 .1.

		Solved datasets in ≤1 h			AnyDTree proved vs. Witty		AnyDTree found vs. Witty	
Bucket	n	Witty	AnyDTree proved	AnyDTree found	χ^2	p	χ^2	p
All datasets	70	43	38	43	0.72	0.40	0.08	0.77
$\lvert F^*\rvert \le 189$	36	29	29	31	0.00	0.96	1.60	0.21
$\lvert F^*\rvert \ge 189$	36	16	10	14	2.72	0.10	0.29	0.59
$\lvert D\rvert \le 68$	36	32	30	32	0.39	0.53	0.45	0.50
$\lvert D\rvert \ge 68$	36	11	8	11	0.69	0.41	0.00	0.99

Table 1: Solved instances within 1h for each bucket and the log-rank statistic comparing AnyDTree with Witty. Timeouts are treated as censored observations.

5. Future Work

- **Continuous Features:** AnyDTree only supports binary features. Extend this to continuous features.
- **Multi-class Classification:** Generalize to handle more than two classes.
- **Parallel search:** Implement multi-threading to explore branches in parallel.