Optimal Decision Trees for nonlinear metrics

A geometric approach

Bogdan-Andrei Bancuta (B.A.Bancuta@student.tudelft.nl)

TU Delft – Delft University of Technology

1. Problem

Machine Learning models are becoming more complex and harder to interpret. Optimal Decision Trees are both performant and interpretable, requiring a low number of nodes to find the optimal solution. However, non-linear metrics, which are very effective when evaluating trees on imbalanced datasets, still represent a challenge in terms of runtime performance and scalability.
Our aim is to improve the merging step of the algorithm and as a result allow the construction of bigger better trees.

2. Research Questions

- Does only keeping the Convex Hull of the Pareto Front still lead to the optimal solution?
- How does the algorithm compare in run-time to the state-of-the-art?

3. Minkowski Sum Merge Algorithm

Optimal solution lies on the Pareto Front and combining them is very expensive $O(size^2)$

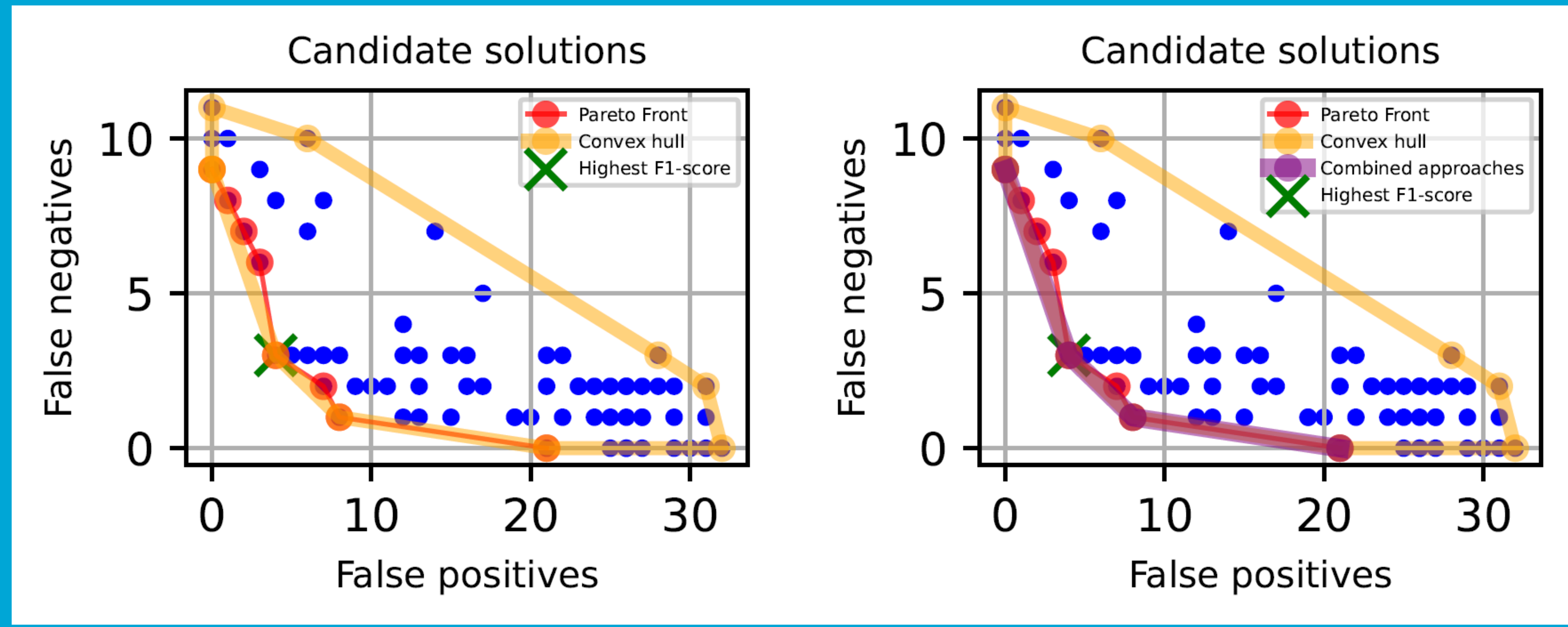
Pareto Front - a set of solution for which $\forall(x,y) \in PF, \nexists(x1,y1) \in PF$ such that $x > x1$ and $y > y1$

The intuition for our method is that the candidate points for the Pareto Front are on the Convex Hull of the whole set of points and thus we can achieve better performance by following these 3 main steps:

- Use Minkowski Sums to combine the left and right set of solutions
- Calculate the Pareto Front of the resulting set
- Compute the Convex Hull to assure the convexity of the solution set

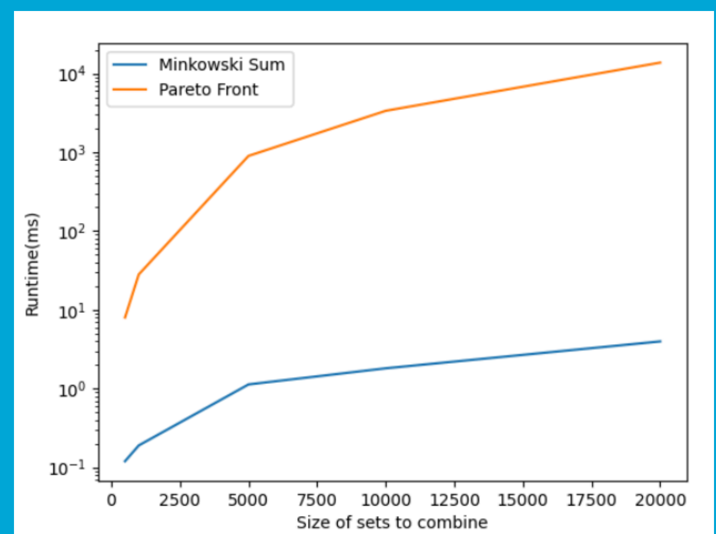
Convex Hull - convex polygon enclosing all points in given space

Minkowski Sum - the polygon resulting from translating the second polygon by all the points of the first

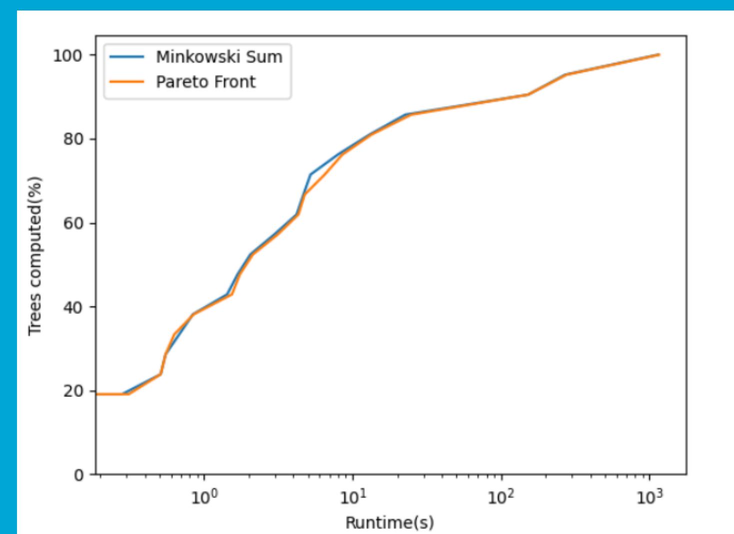


4. Experiments and conclusions

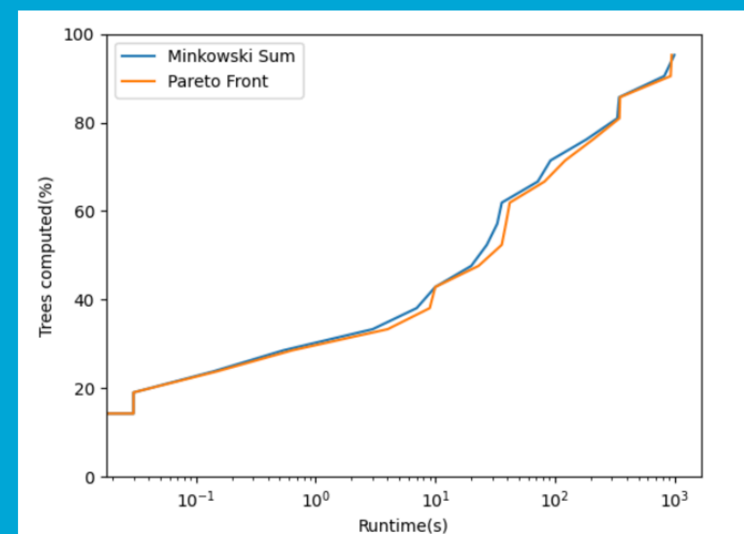
To evaluate our algorithm, we compared it against the Pareto Front approach on artificial datasets as well as against STreeD [1] using Pareto Fronts. Our experiments show a close to 10% speed-up in our approach.



In isolation we can observe the huge impact of switching from a $O(n^2)$ approach to $O(n * \log(n))$ one when n grows.



We can already start to notice some improvements when constructing trees of depth 4.



As we reach trees of depth 5, the improvements become much more noticeable, achieving a close to 12% speed-up.

References:

[1] J. G. M. van der Linden, M. M. de Weerdt, and E. Demirović, "Necessary and sufficient conditions for optimal decision trees using dynamic programming," in Advances in NeurIPS-23, pp. 9173–9212, 2023.

Supervisors:
Dr. Emir Demirović
Jacobus G. M. van der Linden