Learning Optimal Classification Trees: Strong Max-Flow Formulations

Sina Aghaei[1], Andrés Gómez[2], and Phebe. Vayanos[1]

[1] Center for Artificial Intelligence in Society
[2] Department of Industrial and Systems Engineering, Viterbi School of Engineering
University of Southern California
{saghaei,gomezand,phebe.vayanos}@usc.edu

Decision trees are among the most popular techniques for interpretable machine learning [1]. In this work we consider the problem of learning optimal binary classification trees using mixed-integer programming (MIP). Literature on the topic has burgeoned in recent years, see e.g. [2,3,5], motivated both by the empirical suboptimality of heuristic approaches and the tremendous improvements in mixed-integer programming technology. Yet, existing approaches from the literature do not leverage the power of MIP to its full extent. Indeed, they rely on weak formulations, resulting in slow convergence and large optimality gaps. To fill this gap in the literature, we propose a flow-based MIP formulation for optimal binary classification trees.

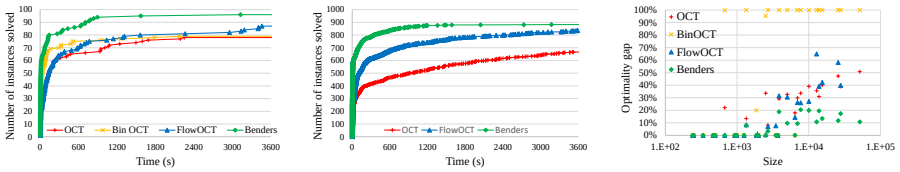
Our approach and main contributions in this paper are:

(a) We propose an intuitive flow-based MIP formulation for the problem of learning optimal classification trees with binary data. We model the correct classification of a point as a max flow problem, where the datapoint flows from source to sink through a single path and only reaches the sink if it is correctly classified (incorrectly classified datapoints face a "road block" that prevents them from traversing the graph at all). Similar to traditional algorithms for learning decision trees, we allow labels to be assigned to internal nodes of the tree. In that case, correctly classified datapoints that reach such nodes are directly routed to the sink node (as if we had a "short circuit"). To get sparser trees we penalize the number of splitting nodes in the objective function with a regularization parameter λ.

(b) Unlike most approaches in the literature, our proposed formulation does not use big-M constraints, and benefits from a stronger LP relaxation as a result. We provide an intuitive proof to justify that our LP relaxation is stronger than existing alternatives. In addition, our formulation achieves this additional strength without a substantial increase in the size of the formulation.

(c) Our proposed formulation is amenable to Benders' decomposition. In particular, if the binary tests to be performed are fixed, the max flow problems to determine which points are correctly classified decompose into independent subproblems.

(d) We conduct extensive computational studies on benchmark datasets, demonstrating that our formulations improve upon the state-of-the-art MIP algorithms, both in terms of in-sample solution quality (and speed) and out-of-sample performance.To the best of the knowledge, this is the first comparison of several MIP methods on the same platform. Specifically, we compare the OCT method [2], BinOCT [5], the proposed flow based formulation (FlowOCT) and its Benders' decomposition (Benders).



(a) Performance profile with $\lambda = 0$.

(b) Performance profile with $\lambda > 0$.

(c) Optimality gaps as a function of the size.

Fig. 1: Summary of optimization performance.

Figure 1 summarizes the in-sample performance, i.e., how good the methods are at solving the optimization problems. From Figure 1(a), we observe that for $\lambda = 0$, BinOCT is able to solve 79 instances within the time limit (and outperforms OCT), but Benders solves the same quantity of instances in only 140 seconds, resulting in a $30\times$ speedup. Similarly, from Figure 1(b), it can be seen that for $\lambda > 0$, OCT is able to solve 666 instances within the time limit, while Benders requires only 70 seconds to do so, resulting in a $50\times$ speedup. BinOCT does not include the option to have a regularization parameter, and is omitted. Finally, Figure 1(c) shows the optimality gaps proven as a function of the dimension. We observe that all methods result in a gap of 0% in small instances. As the dimension increases, BinOCT (which relies on weak formulations but fast enumeration) yields 100% optimality gaps in most cases. OCT and BinOCT prove better gaps, but the performance degrades substantially as the dimension increases. Benders results in the best performance, proving optimality gaps of 20% or less regardless of dimension.

With regards to the out-of-sample accuracy after cross-validation, we observe that the better optimization performance translates to superior statistical properties as well: Among 32 instances (8 datasets and 4 depths) OCT is the best method in two instances (excluding ties), BinOCT in six, while the new formulations FlowOCT and Benders are better in 13 (of which Benders accounts for 10, and is second after FlowOCT in an additional two).

References

1. Breiman, L.: Classification and regression trees. Technical report (1984)
2. Bertsimas, D., Dunn, J.: Optimal classification trees. Machine Learning 106(7), 1039–1082 (2017)
3. Günlük, O., Kalagnanam, J., Menickelly, M., Scheinberg, K. : Optimal decision trees for categorical data via integer programming. arXiv preprint arXiv:1612.03225 (2018)
4. Aghaei, S., Azizi, M., Vayanos, P. : Learning optimal and fair decision trees for non-discriminative decision-making. In: 33rd AAAI Conference on Artificial Intelligence (2019)
5. Verwer, S., Zhang, Y.: Learning optimal classification trees using a binary linear. In: 33rd AAAI Conference on Artificial Intelligence (2019)