Partitioning data for the optimal classification tree problem

Zacharie Ales[1,2], Valentine Huré[1], Amélie Lambert[1]

[1] CNAM, Paris, France
valentine.hure@lecnam.net,amelie.lambert@cnam.fr
[2] UMA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France.
zacharie.ales@ensta-paris.fr

Key words: *supervised learning, classification tree, clustering, mixed-integer linear programming.*

A binary classification tree is a supervised learning model whose internal nodes split up data based on their features, and external nodes assign labels to the data points that reach them. Once overlooked for their low accuracy compared with other types of classifiers, classification trees have seen a resurgence in popularity in recent years due to their inherent interpretability. In particular, numerous combinatorial optimization methods have recently been developed for computing optimal classification trees since the work of Bertsimas and Dunn [1]. These methods outperform simpler heuristics such as CART [2] and are competitive with more advanced heuristics while providing simpler trees [3]. The main issue with mathematical programming approaches is their scalability since the size of their models grow with the size of datasets.

In this paper we present an iterative approach based on resolution of smaller MILPs to build optimal classification trees. The size reduction is obtained by an initial clustering of the data points that we call *data-partitions*.

1 Reducing the problem size with data-partitions

To reduce the size of MILPs, our idea is to group data points that are likely to end up in the same leaf, i.e. to be assigned the same label. Doing this, allows us to only consider one data point per cluster, called its *representative*, and thus to reduce the number of variables in the formulation. More formally, given a dataset $\{(X_i, Y_i)\}_{i\in\mathcal{I}}$, we build a *data-partition* $(\mathcal{P}, \widetilde{X}, \widetilde{Y})$ where $\mathcal{P}$ is a partition of $\mathcal{I}$ and $(\widetilde{X}, \widetilde{Y})$ is the dataset composed of the representative of the clusters of $\mathcal{P}$. We present in Figure 1-left a dataset and an associated data-partition Figure 1-right where the colored data points are the clusters' representatives.

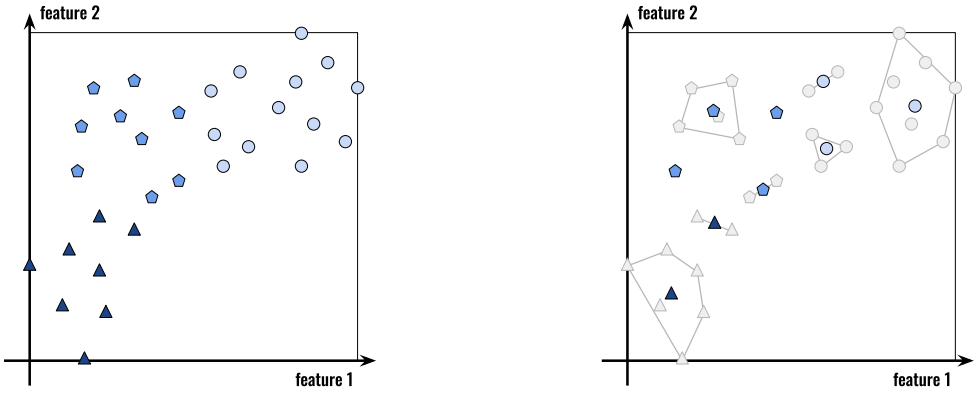

FIG. 1 – Initial dataset (left) : 30 points, 3 labels ; data-partition (right) : 9 clusters

From $(\mathcal{P}, \widetilde{X}, \widetilde{Y})$ we build a new model where the misclassifications are weighted by the size of clusters. Building a relevant partition $\mathcal{P}$ is challenging. We propose three different algorithms for this task.

2 Building optimal classifications trees with data-partitions

An optimal solution for $(\mathcal{P}, \widetilde{X}, \widetilde{Y})$ is not always optimal for the original dataset (X, Y). The number of good predictions is often overestimated because data points do not always follow the path of their representative. In this case, we say that the tree *intersects* $\mathcal{P}$. Otherwise, the number of misclassifications in both problems is the same.

For solving the problem of computing an optimal classification tree, we introduce an iterative algorithm called `ItOCT`. As sketched-up in Algorithm 1, starting from a data-partition, we first solve (M) on $(\mathcal{P}, \widetilde{X}, \widetilde{Y})$. If the obtained tree does not intersects $\mathcal{P}$, we are done. Otherwise, we update $\mathcal{P}$ by splitting the intersected clusters and proceed to the next iteration.

Algorithm 1: `ItOCT` $((X, Y),(M),(\mathcal{P}, \widetilde{X}, \widetilde{Y}))$

$T_0 \leftarrow$ solution of (M) for $(\mathcal{P}, \widetilde{X}, \widetilde{Y})$
$k \leftarrow 0$
while T_k *intersects* $(\mathcal{P}, \widetilde{X}, \widetilde{Y})_k$ **do**
  Split clusters of $(\mathcal{P}, \widetilde{X}, \widetilde{Y})_k$ to create $(\mathcal{P}, \widetilde{X}, \widetilde{Y})_{k+1}$ that is not intersected by T_k
  $T_{k+1} \leftarrow$ solution of (M) for $(\mathcal{P}, \widetilde{X}, \widetilde{Y})_{k+1}$
  $k \leftarrow k + 1$
end
Return T

In our experiments, we consider two different MILP formulations (M). In the first one, the representatives of the clusters are fixed in a pre-processing step. This leads to smaller models but heuristic solutions. In the second one, the representatives are determined during the MILP resolution which ensures to get an optimal solution at the cost of a larger number of constraints. Anyway, for both variants, a key step of Algorithm 1 is how to split the clusters, since it impacts the number of iterations and thus the total CPU time.

References

[1] Dimitris Bertsimas, and Jack Dunn. Optimal classification tree. *Machine Learning*, 106(7) :1039–1082, 2017.

[2] L. Breiman, J. Friedman, C. J. Stone, and R. A Olshen. Classification and Regression Trees. Wadsworth, Belmont, Calif., 1984.

[3] Zacharie Alès, Valentine Huré, and Amélie Lambert. New optimization models for optimal classification trees. 2022. hal-03865931v2.