European Journal of Operational Research 284 (2020) 255–272

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Decision Support

Sparsity in optimal randomized classification trees

Rafael Blanquero[a], Emilio Carrizosa[a], Cristina Molero-Río[a,*], Dolores Romero Morales[b]

[a] Instituto de Matemáticas de la, Universidad de Sevilla (IMUS), Seville, Spain
[b] Copenhagen Business School, Frederiksberg, Denmark

ARTICLE INFO

Article history:
Received 10 October 2018
Accepted 3 December 2019
Available online 16 December 2019

Keywords:
Data mining
Optimal classification trees
Global and local sparsity
Nonlinear programming

ABSTRACT

Decision trees are popular Classification and Regression tools and, when small-sized, easy to interpret. Traditionally, a greedy approach has been used to build the trees, yielding a very fast training process; however, controlling sparsity (a proxy for interpretability) is challenging. In recent studies, optimal decision trees, where all decisions are optimized simultaneously, have shown a better learning performance, especially when oblique cuts are implemented. In this paper, we propose a continuous optimization approach to build sparse optimal classification trees, based on oblique cuts, with the aim of using fewer predictor variables in the cuts as well as along the whole tree. Both types of sparsity, namely local and global, are modeled by means of regularizations with polyhedral norms. The computational experience reported supports the usefulness of our methodology. In all our data sets, local and global sparsity can be improved without harming classification accuracy. Unlike greedy approaches, our ability to easily trade in some of our classification accuracy for a gain in global sparsity is shown.

1. Introduction

Decision trees (Yang, Liu, Tsoka, & Papageorgiou, 2017) are a popular non-parametric tool for Classification and Regression in Statistics and Machine Learning (Hastie, Tibshirani, & Friedman, 2009). Since they are rule-based, when small-sized, they are deemed to be leaders in terms of interpretability (Athey, 2018; Baesens, Setiono, Mues, & Vanthienen, 2003; Carrizosa, Martín-Barragán, & Romero Morales, 2011; Freitas, 2014; Goodman & Flaxman, 2017; Jung, Concannon, Shroff, Goel, & Goldstein, 2017; Martens, Baesens, Van Gestel, & Vanthienen, 2007; Martín-Barragán, Lillo, & Romo, 2014; Ridgeway, 2013; Ustun & Rudin, 2016).

It is well-known that the problem of building optimal decision trees is NP-complete (Hyafil & Rivest, 1976). For this reason, classic decision trees have been traditionally designed using greedy procedures in which at each branch node of the tree, some purity criterion is (locally) optimized. For instance, CARTs (Breiman, Friedman, Stone, & Olshen, 1984) employ a greedy and recursive partitioning procedure which is computationally cheap, especially since orthogonal cuts are implemented, i.e., one single predictor variable is involved in each branching rule. These rules are of maximal sparsity at each branching node (excellent local sparsity), making classic decision trees locally easy to interpret. However, when deep, they become to be harder to interpret since many predictor variables are, in general, involved across all branching rules (not so good global sparsity).

Addressing global sparsity is a challenge in decision trees and, to the best of our knowledge, this has not been tackled appropriately in the literature. Standard CARTs or Random Forests (RFs) (Biau & Scornet, 2016; Breiman, 2001; Fernández-Delgado, Cernadas, Barro, & Amorim, 2014; Genuer, Poggi, Tuleau-Malot, & Villa-Vialaneix, 2017) cannot manage it due to the greedy construction of the trees. Nonetheless, some attempts have been made, see Deng and Runger (2012, 2013). Classic decision trees usually select their orthogonal cuts at each branch node by optimizing an information theory criterion among all possible predictor variables and thresholds. The regularization framework in Deng and Runger (2012) considers a penalty to this criterion for predictor variables that have not appeared yet in the tree. This approach is refined in Deng and Runger (2013), by also including the importance scores of the predictor variables, obtained in a preprocessing step running a preliminary RF.

The mainstream trend of using a greedy strategy in the construction of decision trees may lead to myopic decisions, which, in turn, may affect the overall learning performance. The major advances in Mathematical Optimization (Carrizosa & Romero Morales, 2013; Olafsson, Li, & Wu, 2008; Silva, 2017) have led to different approaches to build decision trees with some overall optimality criterion, called hereafter optimal classification trees. It is worth mentioning recent proposals which grow optimal classification trees of a pre-established depth, both deterministic

* Corresponding author.
E-mail addresses: rblanquero@us.es (R. Blanquero), ecarrizosa@us.es (E. Carrizosa), mmolero@us.es (C. Molero-Río), drm.eco@cbs.dk (D. Romero Morales).

https://doi.org/10.1016/j.ejor.2019.12.002
0377-2217/

(Bertsimas & Dunn, 2017; Firat, Crognier, Gabor, Hurkens, & Zhang, 2019; Günlük, Kalagnanam, Menickelly, & Scheinberg, 2019; Verwer & Zhang, 2017; Verwer, Zhang, & Ye, 2017) and randomized (Blanquero, Carrizosa, Molero-Río, & Romero Morales, 2018). The deterministic approaches formulate the problem of building the tree as a mixed-integer linear optimization problem. Such approach is the most natural, since many discrete decisions are to be made when building a decision tree. Although the results of such optimal classification trees are encouraging, the inclusion of integer decision variables makes the computing times explode, giving rise to models trained over a small subsample of the data set (Günlük et al., 2019) and, as customary, with a CPU time limit being imposed to the optimization solver. On the other hand, a continuous optimization-based approach to build optimal randomized classification trees is proposed in Blanquero et al. (2018). This is achieved by replacing the yes/no decisions in traditional trees by probabilistic decisions, i.e., instead of deciding at each branch node if an individual goes either to the left or to the right child node in the tree, the probability of going to the left is sought. The numerical results in Blanquero et al. (2018) illustrate the good performance achieved in very short time. All these optimization-based approaches are flexible enough to address critical issues that the greedy nature of classic decision trees would find it difficult, such as preferences on the classification performance in some class where misclassifying is more damaging (Blanquero et al., 2018; Verwer & Zhang, 2017; Verwer et al., 2017), or controlling the number of predictor variables used along the tree (local and global sparsity).

Optimal classification trees have been grown with both orthogonal (Bertsimas & Dunn, 2017; Firat et al., 2019; Günlük et al., 2019) and oblique cuts (Bennett & Blue, 1996; Bertsimas & Dunn, 2017; Blanquero et al., 2018; Norouzi, Collins, Johnson, Fleet, & Kohli, 2015; Verwer & Zhang, 2017; Verwer et al., 2017). Oblique cuts are more flexible than orthogonal ones since a combination of several predictor variables is allowed in the branching. Trees based on oblique cuts lead to similar or even better learning performance than those based on orthogonal cuts, and, at the same time, they exhibit a shallow depth, since several orthogonal cuts may be reduced to one single oblique cut. Apart from the flexibility that we can borrow from them, many integer decision variables associated with orthogonal cuts are not present in the oblique ones, which eases the optimization. Therefore, optimal classification trees based on oblique cuts require a lower training computing time while showing much more promising results in terms of accuracy. However, this comes at the expense of damaging interpretability, since, in principle, all the predictor variables could appear in each branching rule. In this paper, we tackle this issue.

We propose a novel optimized classification tree, based on the methodology in Blanquero et al. (2018) and, therefore, in oblique cuts, that yields rules/trees that are sparser, and thus enhance interpretability. We model this as a continuous optimization problem. As in the classic LASSO model (Tibshirani, Wainwright, & Hastie, 2015), sparsity is sought by means of regularization terms. We model local sparsity with the ℓ_1-norm, and the global sparsity with the ℓ_∞-norm. The ℓ_∞ reguralization has been applied to other classifiers, for instance, Support Vector Machines (Maldonado, Bravo, Lopez, & Perez, 2017; Maldonado & Lopez, 2017; Zou & Yuan, 2008), but the ℓ_1 is more popular. A novel continuous-based approach for building this sparse optimal randomized classification tree is provided. Theoretical results on the range of the sparsity parameters are shown. Our numerical results, where well-known real data sets are used, illustrate the efectiveness of our methodology: sparsity in optimal classification trees improves without harming learning performance. In addition, our ability to trade in some of our classification accuracy, still being superior to CART, to be comparable to CART in terms of global sparsity is shown.

The remainder of the paper is organized as follows. In Section 2 we detail the construction of the Sparse Optimal Randomized Classification Tree. Some theoretical properties are given in Section 3. In Section 4, our numerical experience is reported. Finally, conclusions and possible lines of future research are provided in Section 5.

2. Sparsity in optimal randomized classification trees

2.1. Introduction

We assume given a training sample $\{(\boldsymbol{x}_i, y_i)\}_{1 \le i \le N}$, where $\boldsymbol{x}_i$ represents the p-dimensional vector of predictor variables of individual i, and $y_i \in \{1, \ldots, K\}$ indicates the class membership. Without loss of generality, we assume $\boldsymbol{x}_i \in [0,1]^p,\ i = 1, \ldots, N$.

Sparse Optimal Randomized Classification Trees, addressed in this paper, extend the Optimal Randomized Classification Trees (ORCTs) in Blanquero et al. (2018). An ORCT is an optimal binary classification tree of a given depth D, obtained by minimizing the expected misclassification cost over the training sample Fig. 1. shows the structure of an ORCT of depth $D = 2$. Unlike classic decision trees, oblique cuts, on which more than one predictor variable takes part, are performed. ORCTs are modeled by means of a Non-Linear Continuous Optimization formulation. The usual deterministic yes/no rule at each branch node is replaced by a smoother rule: a probabilistic decision rule at each branch node, induced by a cumulative density function (CDF) F, is obtained. Therefore, the movements in ORCTs can be seen as randomized: at a given branch node of an ORCT, a random variable will be generated to indicate by which branch an individual has to continue. Since binary trees are built, the Bernoulli distribution is appropriate, whose probability of success will be determined by the value of this CDF, evaluated over the vector of predictor variables. More precisely, at a given branch node t of the tree, an individual with predictor variables $\boldsymbol{x}$ will go either to the left or to the right child nodes with probabilities $F(\frac{1}{p}\boldsymbol{a}_t^T\boldsymbol{x} - \mu_t)$ and $1 - F(\frac{1}{p}\boldsymbol{a}_t^T\boldsymbol{x} - \mu_t)$, respectively, where $\boldsymbol{a}_{.t}$ and μ_t are decision variables. For further details on the construction of ORCTs, the reader is referred to Blanquero et al. (2018). Sparse ORCT, S-ORCT, minimizes the expected misclassification cost over the training sample regularized with two polyhedral norms.

The following notation is needed:

Parameters

D	depth of the binary tree,
N	number of individuals in the training sample,
p	number of predictor variables,
K	number of classes,
$\{(\boldsymbol{x}_i, y_i)\}_{1 \le i \le N}$	training sample, where $\boldsymbol{x}_i \in [0,1]^p$ and $y_i \in \{1, \ldots, K\}$,
I_k	set of individuals in the training sample belonging to class k, $k = 1, \ldots, K$,
$W_{y_i k}$	misclassification cost incurred when classifying an individual i, whose class is y_i, in class k, $y_i, i = 1, \ldots, N,\ k = 1, \ldots, K$,
$F(\cdot)$	univariate continuous CDF centered at 0, used to define the probabilities for an individual to go to the left or the right child node in the tree. We will assume that F is the CDF of a continuous random variable with density f,
$\lambda^L \ge 0$	local sparsity regularization parameter,
$\lambda^G \ge 0$	global sparsity regularization parameter,

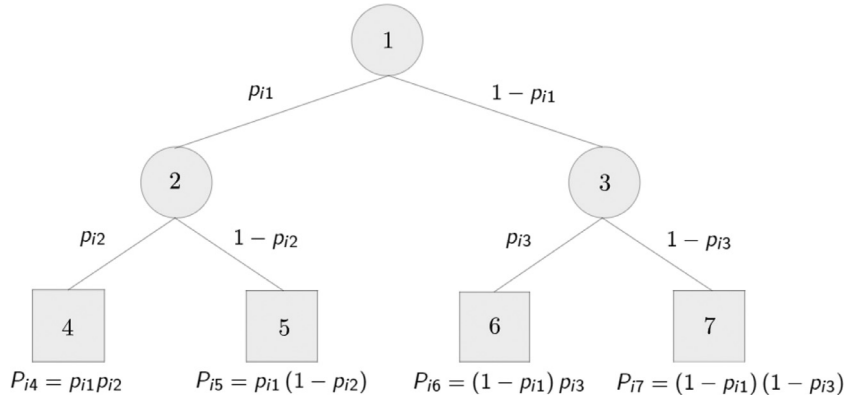


Fig. 1. Optimal randomized classification tree of depth $D=2$.

Nodes

τ_B set of branch nodes,

τ_L set of leaf nodes,

$N_L(t)$ set of ancestor nodes of leaf node t whose left branch takes part in the path from the root node to leaf node t, $t\in\tau_L$,

$N_R(t)$ set of ancestor nodes of leaf node t whose right branch takes part in the path from the root node to leaf node t, $t\in\tau_L$,

Decision variables

$a_{jt}\in[-1,1]$ coefficient of predictor variable j in the oblique cut at branch node $t\in\tau_B$, with $\boldsymbol{a}$ being the $p\times|\tau_B|$ matrix of these coefficients, $\boldsymbol{a}=\left(a_{jt}\right)_{j=1,\ldots,p,\ t\in\tau_B}$. The expressions $\boldsymbol{a}_{j\cdot}$ and $\boldsymbol{a}_{\cdot t}$ will denote the j-th row and the t-th column of $\boldsymbol{a}$, respectively,

$\mu_t\in[-1,1]$ location parameter at branch node $t\in\tau_B$, $\boldsymbol{\mu}$ being the vector that comprises every μ_t, i.e., $\boldsymbol{\mu}=(\mu_t)_{t\in\tau_B}$,

C_{kt} probability of being assigned to class label $k\in\{1,\ldots,K\}$ for an individual at leaf node t, $t\in\tau_L$, being the $K\times|\tau_L|$ matrix such that $\boldsymbol{C}=(C_{kt})_{k=1,\ldots,K,\ t\in\tau_L}$.

Probabilities

$p_{it}(\boldsymbol{a}_{\cdot t},\mu_t)$ probability of individual i going down the left branch at branch node t. Its expression is $p_{it}(\boldsymbol{a}_{\cdot t},\mu_t)=F\left(\frac{1}{p}\boldsymbol{a}_t^T\boldsymbol{x}_i-\mu_t\right),\ i=1,\ldots,N,\ t\in\tau_B$,

$P_{it}(\boldsymbol{a},\boldsymbol{\mu})$ probability of individual i falling into leaf node t. Its expression is $P_{it}(\boldsymbol{a},\boldsymbol{\mu})=\prod_{t_l\in N_L(t)}p_{it_l}\left(\boldsymbol{a}_{\cdot t_l},\mu_{t_l}\right)\prod_{t_r\in N_R(t)}\left(1-p_{it_r}(\boldsymbol{a}_{\cdot t_r},\mu_{t_r})\right),\ i=1,\ldots,N,\ t\in\tau_L$,

$g(\boldsymbol{a},\boldsymbol{\mu},\boldsymbol{C})$ expected misclassification cost over the training sample. Its expression is $g(\boldsymbol{a},\boldsymbol{\mu},\boldsymbol{C})=\frac{1}{N}\sum_{i=1}^{N}\sum_{t\in\tau_L}P_{it}(\boldsymbol{a},\boldsymbol{\mu})\sum_{k=1}^{K}W_{y_ik}C_{kt}$.

2.2. The formulation

With these parameters and decision variables, the S-ORCT is formulated as follows:

$$\min\ g(\boldsymbol{a},\boldsymbol{\mu},\boldsymbol{C})+\lambda^L\sum_{j=1}^{p}\left\|\boldsymbol{a}_{j\cdot}\right\|_1+\lambda^G\sum_{j=1}^{p}\left\|\boldsymbol{a}_{j\cdot}\right\|_\infty \tag{1}$$

$$\text{s.t.}\ \sum_{k=1}^{K}C_{kt}=1,t\in\tau_L, \tag{2}$$

$$\sum_{t\in\tau_L}C_{kt}\geq 1,k=1,\ldots,K, \tag{3}$$

$$a_{jt}\in[-1,1],j=1,\ldots,p,\ t\in\tau_B, \tag{4}$$

$$\mu_t\in[-1,1],t\in\tau_B, \tag{5}$$

$$C_{kt}\in[0,1],k=1,\ldots,K,\ t\in\tau_L. \tag{6}$$

In the objective function we have three terms, the first being the expected misclassification cost in the training sample, while the second and the third are regularization terms. The second term addresses local sparsity, since it penalizes the coefficients of the predictor variables used in the cuts along the tree. Instead, the third term controls whether a given predictor variable is ever used across the whole tree, thus addressing global sparsity. The ℓ_∞-norm is used as a group penalty function, by forcing the coefficients linked to the same predictor variable to be shrunk simultaneously along all branch nodes. Note that both local and global sparsity are equivalent when dealing with depth $D=1$, as there is a single cut across the whole tree.

In terms of the feasible region, for each leaf node $t\in\tau_L$, C_{kt} represents the probability that an individual at node t is assigned to class $k\in\{1,\ldots,K\}$. Constraints (2) force that such probabilities sum to 1, while constraints (3) force the sum of the probabilities along all leaf nodes $t\in\tau_B$ assigned to class k to be at least one.

Theorem 1 guarantees the existence of an optimal deterministic solution, i.e., such probabilities C_{kt} will all be in {0, 1}, and thus (6) can be replaced by

$$C_{kt}\in\{0,1\},\ k=1,\ldots,K,\ t\in\tau_L. \tag{7}$$

Constraints (6) and (7) will be used interchangeably when needed.

Theorem 1. *There exists an optimal solution to (1)–(6) such that* $C_{kt}\in\{0,1\},\ k=1,\ldots,K,\ t\in\tau_L$.

Proof. The continuity of the objective function (1), defined over a compact set, ensures the existence of an optimal solution of the optimization problem (1)–(6), by Weierstrass Theorem. Let $\boldsymbol{a}^*=\left(a^*_{jt}\right)_{j=1,\ldots,p,\ t\in\tau_B}$, $\boldsymbol{\mu}^*=\left(\mu^*_t\right)_{t\in\tau_B}$, $\boldsymbol{C}^*=\left(C^*_{kt}\right)_{k=1,\ldots,K,\ t\in\tau_B}$ be an optimal solution. Fixed $\boldsymbol{a}^*$, $\boldsymbol{\mu}^*$, then $\boldsymbol{C}^*$ is optimal to the following problem in the decision variables C_{kt}, $k=1,\ldots,K,\ t\in\tau_L$:

$$\min\ \frac{1}{N}\sum_{i=1}^{N}\sum_{t\in\tau_L}P_{it}(\boldsymbol{a}^*,\boldsymbol{\mu}^*)\sum_{k=1}^{K}W_{y_ik}C_{kt}+\lambda^L\sum_{j=1}^{p}\left\|\boldsymbol{a}^*_{j\cdot}\right\|_1+\lambda^G\sum_{j=1}^{p}\left\|\boldsymbol{a}^*_{j\cdot}\right\|_\infty$$

$$\text{s.t.} \sum_{k=1}^{K} C_{kt} = 1,\ t \in \tau_L,$$
$$\sum_{t \in \tau_L} C_{kt} \geq 1,\ k = 1, \ldots, K,$$
$$C_{kt} \in [0,1],\ k = 1, \ldots, K,\ t \in \tau_L.$$

This is a transportation problem, to which the integrality of an optimal solution is well-known to hold, i.e., there exists $\overline{\boldsymbol{C}} = \left(\overline{C}_{kt}\right)_{k=1,\ldots,K,\ t\in\tau_L} \in \{0,1\}$ for all k, t such that $\left(\boldsymbol{a}^*, \boldsymbol{\mu}^*, \overline{\boldsymbol{C}}\right)$ is also optimal for (1)–(6). □

Theorem 1 gives a new interpretation of constraints (2) and (3): if (7) is used instead of (6), when C_{kt} takes the value 1, then all the individuals at node $t \in \tau_L$ are labelled as k; and 0, otherwise. Constraints (2) state that any leaf node $t \in \tau_L$ must be labelled with exactly one class label, and constraints (3) state that each class k has at least one node t with such label.

Once the optimization problem is solved, the S-ORCT predicts the class of a new unlabeled observation with predictor vector $\boldsymbol{x}$ with a probabilistic rule, namely, we estimate the probability of being in class k as $\sum_{t\in\tau_L} C_{kt} \cdot P_{\boldsymbol{x}t}(\boldsymbol{a}, \boldsymbol{\mu})$. If a deterministic classification rule is sought, we allocate to the most probable class. Moreover, if prior probabilities $\Pi_k(\boldsymbol{x})$ are given, one can also use the Bayes rule.

ORCTs were also shown to deal effectively with controlling the correct classification rate on different classes. This idea can also be applied to S-ORCTs. Hence, given the classes $k = 1, \ldots, K$ to be controlled and their corresponding desired performances ρ_k, the expectation of achieving each performance guarantee can be computed with the ORCT parameters, provided that the following set of constraints is added to the model:

$$\sum_{i\in I_k} \sum_{t\in\tau_L} P_{it}(\boldsymbol{a}, \boldsymbol{\mu}) C_{kt} \geq \rho_k |I_k|,\ k = 1, \ldots, K. \tag{8}$$

With these constraints we have a direct control on the classification performance in each class separately. This is useful when dealing with imbalanced data sets.

2.3. A smooth reformulation

Problem (1)–(6) is non-smooth due to the norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$ appearing in the objective function. A smooth version is easily obtained by rewritting both regularization terms using new decision variables. Since the first regularization term includes absolute values,

$$\left\|\boldsymbol{a}_{j\cdot}\right\|_1 = \sum_{t\in\tau_B} \left|a_{jt}\right|,\ j = 1, \ldots, p,$$

decision variables $a_{jt} \in [-1,1],\ j = 1, \ldots, p,\ t \in \tau_B$, are split into their positive and negative counterparts $a^+_{jt},\ a^-_{jt} \in [0,1],\ j = 1, \ldots, p,\ t \in \tau_B$, respectively, holding $a_{jt} = a^+_{jt} - a^-_{jt}$ and $\left|a_{jt}\right| = a^+_{jt} + a^-_{jt}$. Similarly, we denote $\boldsymbol{a}^+ = \left(a^+_{jt}\right)_{j=1,\ldots,p,\ t\in\tau_B}$ and $\boldsymbol{a}^- = \left(a^-_{jt}\right)_{j=1,\ldots,p,\ t\in\tau_B}$. Regarding the second regularization term, new decision variables $\beta_j \in [0,1]$ are needed:

$$\left\|\boldsymbol{a}_{j\cdot}\right\|_\infty = \max_{t\in\tau_B} \left|a_{jt}\right| = \beta_j \in [0,1],\ j = 1, \ldots, p,$$

and have to force $\beta_j \geq \left|a_{jt}\right| = a^+_{jt} + a^-_{jt},\ j = 1, \ldots, p,\ t \in \tau_B$.

We can now formulate S-ORCT as a smooth problem, thus solvable with standard continuous optimization solvers, as done in our computational section. Indeed, we have that (1)–(6) is equivalent to

$$\min\ g\left(\boldsymbol{a}^+ - \boldsymbol{a}^-, \boldsymbol{\mu}, \boldsymbol{C}\right) + \lambda^L \sum_{j=1}^{p} \sum_{t\in\tau_B} \left(a^+_{jt} + a^-_{jt}\right) + \lambda^G \sum_{j=1}^{p} \beta_j \tag{9}$$

$$\text{s.t.} \sum_{k=1}^{K} C_{kt} = 1,\ t \in \tau_L, \tag{10}$$

$$\sum_{t\in\tau_L} C_{kt} \geq 1,\ k = 1, \ldots, K, \tag{11}$$

$$\beta_j \geq a^+_{jt} + a^-_{jt},\ j = 1, \ldots, p, \tag{12}$$

$$a^+_{jt},\ a^-_{jt} \in [0,1],\ j = 1, \ldots, p,\ t \in \tau_B, \tag{13}$$

$$\beta_j \in [0,1],\ j = 1, \ldots, p, \tag{14}$$

$$\mu_t \in [-1,1],\ t \in \tau_B, \tag{15}$$

$$C_{kt} \in [0,1],\ k = 1, \ldots, K,\ t \in \tau_L. \tag{16}$$

Observe that, if we are only concerned about global sparsity, and thus we set $\lambda^L = 0$, the rewriting of the decision variables $a_{jt},\ j = 1, \ldots, p,\ t \in \tau_B$ is no longer necessary and (4) replaces (13), and (12) turns into

$$\beta_j \geq a_{jt},\ j = 1, \ldots, p,\ t \in \tau_B, \tag{17}$$

$$\beta_j \geq -a_{jt},\ j = 1, \ldots, p,\ t \in \tau_B. \tag{18}$$

3. Theoretical properties

This section discusses some theoretical properties enjoyed by the S-ORCT. Let us consider the objective function of (1)–(6). When taking λ^L and λ^G large enough, the first term related to the performance of the classifier becomes negligible and therefore $\boldsymbol{a}$ will shrink to $\boldsymbol{0}$. The tree with $\boldsymbol{a} = \boldsymbol{0}$ is the sparsest possible tree though not the best promising one from the accuracy point of view, since none of the predictor variables is used to classify. In this case, the probability of an individual with predictor variables $\boldsymbol{x}$ being assigned to class k is independent of $\boldsymbol{x}$, and nothing more than the distribution of classes is available. In this section, we derive upper bounds for the sparsity parameters, λ^L and λ^G, in the sense that above these bounds the sparsest tree (with $\boldsymbol{a}^* = \boldsymbol{0}$) is a stationary point of the S-ORCT, that is, there exists $(\boldsymbol{a}^* = \boldsymbol{0}, \boldsymbol{\mu}^*, \boldsymbol{C}^*)$ such that the necessary optimality condition with respect to $\boldsymbol{a}$ is satisfied. This is done in Theorems 2 and 3.

Theorem 2. *Let $\sigma \in [0,1]$. For*

$$\lambda^L \geq (1-\sigma) \max_{\substack{\boldsymbol{\mu}\in[-1,1]^{|\tau_B|} \\ \boldsymbol{C}\in\{0,1\}^{K\times|\tau_L|}}} \max_{j=1,\ldots,p} \left\|\nabla_{\boldsymbol{a}_{j\cdot}} g(\boldsymbol{0}, \boldsymbol{\mu}, \boldsymbol{C})\right\|_\infty \text{and}$$

$$\lambda^G \geq \sigma \max_{\substack{\boldsymbol{\mu}\in[-1,1]^{|\tau_B|} \\ \boldsymbol{C}\in\{0,1\}^{K\times|\tau_L|}}} \max_{j=1,\ldots,p} \left\|\nabla_{\boldsymbol{a}_{j\cdot}} g(\boldsymbol{0}, \boldsymbol{\mu}, \boldsymbol{C})\right\|_1,$$

$\boldsymbol{a}^* = \boldsymbol{0}$ is a stationary point of the S-ORCT.

Proof. Let σ, λ^L, λ^G be such that they satisfy the assumptions.

By Theorem 1, there exists $(\boldsymbol{a}^*, \boldsymbol{\mu}^*, \boldsymbol{C}^*)$ optimal solution to (1)–(6) satisfying $C^*_{kt} \in \{0,1\} \forall k = 1, \ldots, K,\ t \in \tau_L$. In the following we

will show that $(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*)$ is a stationary point of the S-ORCT, i.e.,

$$-\nabla_{\boldsymbol{a}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) \in \partial_{\boldsymbol{a}} \left(\lambda^L \sum_{j=1}^{p} \left\| \boldsymbol{a}_{j\cdot} \right\|_1 + \lambda^G \sum_{j=1}^{p} \left\| \boldsymbol{a}_{j\cdot} \right\|_\infty \right)(\mathbf{0}) \tag{19}$$

where $\partial_{\boldsymbol{a}}$ is the subdifferential operator.

For every $\boldsymbol{a}_{j\cdot}$, $j = 1, \ldots, p$, we have that

$$\partial_{\boldsymbol{a}_{j\cdot}} \left(\left\| \boldsymbol{a}_{j\cdot} \right\|_1 \right)(\mathbf{0}) = \mathbb{B}_\infty = \left\{ \boldsymbol{q} \in \mathbb{R}^{|\tau_B|} : \|\boldsymbol{q}\|_\infty \leq 1 \right\}$$
$$\partial_{\boldsymbol{a}_{j\cdot}} \left(\left\| \boldsymbol{a}_{j\cdot} \right\|_\infty \right)(\mathbf{0}) = \mathbb{B}_1 = \left\{ \boldsymbol{q} \in \mathbb{R}^{|\tau_B|} : \|\boldsymbol{q}\|_1 \leq 1 \right\}.$$

Hence,

$$-\nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) \in \lambda^L \partial_{\boldsymbol{a}_{j\cdot}} \left(\left\| \boldsymbol{a}_{j\cdot} \right\|_1 \right)(\mathbf{0}) + \lambda^G \partial_{\boldsymbol{a}_{j\cdot}} \left(\left\| \boldsymbol{a}_{j\cdot} \right\|_\infty \right)(\mathbf{0}),$$

if, and only if,

$$-\nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) \in \lambda^L \mathbb{B}_\infty + \lambda^G \mathbb{B}_1,$$

if, and only if, there exist $\boldsymbol{q}_j^L$, $\boldsymbol{q}_j^G \in \mathbb{R}^{|\tau_B|}$ such that

$$\left\| \boldsymbol{q}_j^L \right\|_\infty \leq 1,$$
$$\left\| \boldsymbol{q}_j^G \right\|_1 \leq 1,$$
$$-\nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) = \lambda^L \boldsymbol{q}_j^L + \lambda^G \boldsymbol{q}_j^G,$$

if, and only if, there exist $\tilde{\boldsymbol{q}}_j^L$, $\tilde{\boldsymbol{q}}_j^G \in \mathbb{R}^{|\tau_B|}$ such that

$$\left\| \tilde{\boldsymbol{q}}_j^L \right\|_\infty \leq \lambda^L,$$
$$\left\| \tilde{\boldsymbol{q}}_j^G \right\|_1 \leq \lambda^G,$$
$$-\nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) = \tilde{\boldsymbol{q}}_j^L + \tilde{\boldsymbol{q}}_j^G.$$

Let us consider

$$\tilde{\boldsymbol{q}}_j^L = -(1-\sigma) \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*),$$
$$\tilde{\boldsymbol{q}}_j^G = -\quad \sigma \quad \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*),$$

and check that the conditions are satisfied:

$$\begin{aligned}
\left\| \tilde{\boldsymbol{q}}_j^L \right\|_\infty &= (1-\sigma) \left\| \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) \right\|_\infty \leq (1-\sigma) \\
&\quad \max_{\substack{\boldsymbol{\mu} \in [-1,1]^{|\tau_B|} \\ \mathbf{C} \in \{0,1\}^{K \times |\tau_L|}}} \max_{j=1,\ldots,p} \left\| \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}, \mathbf{C}) \right\|_\infty \leq \lambda^L, \\
\left\| \tilde{\boldsymbol{q}}_j^G \right\|_1 &= \sigma \left\| \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) \right\|_1 \leq \sigma \\
&\quad \max_{\substack{\boldsymbol{\mu} \in [-1,1]^{|\tau_B|} \\ \mathbf{C} \in \{0,1\}^{K \times |\tau_L|}}} \max_{j=1,\ldots,p} \left\| \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}, \mathbf{C}) \right\|_1 \leq \lambda^G, \\
\tilde{\boldsymbol{q}}_j^L + \tilde{\boldsymbol{q}}_j^G &= -(1-\sigma) \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) - \sigma \nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*) \\
&= -\nabla_{\boldsymbol{a}_{j\cdot}} g(\mathbf{0}, \boldsymbol{\mu}^*, \mathbf{C}^*).
\end{aligned}$$

Therefore, the desired result follows. □

A stronger result is proven for the S-ORCT of depth $D = 1$ and $K = 2$. Since local and global sparsity are equivalent for the S-ORCT of depth $D = 1$, without loss of generality, we can assume that $\lambda^G = 0$. Therefore, the objective function of the S-ORCT of depth $D = 1$ can be written as:

$$g_1(\boldsymbol{a}_{\cdot 1}, \mu_1, \mathbf{C}) = g(\boldsymbol{a}_{\cdot 1}, \mu_1, \mathbf{C}) + \lambda^L \|\boldsymbol{a}_{\cdot 1}\|_1,$$

where

$$\begin{aligned}
g(\boldsymbol{a}_{\cdot 1}, \mu_1, \mathbf{C}) &= \frac{1}{N} \sum_{i=1}^{N} \left[p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1) \sum_{k=1}^{2} W_{y_i k} C_{k2} \right. \\
&\quad \left. + (1 - p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)) \sum_{k=1}^{2} W_{y_i k} C_{k3} \right] \\
&= \frac{1}{N} \sum_{k=1}^{2} \sum_{i \in I_k} \left[p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1) \sum_{k' \neq k} W_{kk'} C_{k'2} \right. \\
&\quad \left. + (1 - p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)) \sum_{k' \neq k} W_{kk'} C_{k'3} \right]
\end{aligned} \tag{20}$$

and

$$p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1) = F\left(\frac{1}{p} \boldsymbol{a}_{\cdot 1}^T \boldsymbol{x}_i - \mu_1 \right),\ i = 1, \ldots, N.$$

A technical lemma is needed to prove the desired result.

Lemma 1. *For any allocation rule $\mathbf{C}$, the objective function of the S-ORCT of depth $D = 1$, g_1, is monotonic in μ_1 when $\boldsymbol{a}_{\cdot 1} = \mathbf{0}$.*

Proof. Fixed $\boldsymbol{a}_{\cdot 1} = \left(a_{j1}\right)_{j=1,\ldots,p}$, and $\mathbf{C} = (C_{kt})_{k=1,2,\ t=2,3}$,

$$\begin{aligned}
\left. \frac{\partial g_1}{\partial \mu_1} \right|_{\boldsymbol{a}_{\cdot 1} = \mathbf{0}} &= \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in I_k} \left(\sum_{k' \neq k} W_{kk'} C_{k'2} \right. \\
&\quad \left. - \sum_{k' \neq k} W_{kk'} C_{k'3} \right) \left. \frac{\partial p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)}{\partial \mu_1} \right|_{\boldsymbol{a}_{\cdot 1} = \mathbf{0}},
\end{aligned}$$

where

$$\begin{aligned}
\frac{\partial p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)}{\partial \mu_1} &= \frac{\partial F\left(\frac{1}{p} \boldsymbol{a}_{\cdot 1}^T \boldsymbol{x}_i - \mu_1 \right)}{\partial \left(\frac{1}{p} \boldsymbol{a}_{\cdot 1}^T \boldsymbol{x}_i - \mu_1 \right)} \frac{\partial \left(\frac{1}{p} \boldsymbol{a}_{\cdot 1}^T \boldsymbol{x}_i - \mu_1 \right)}{\partial \mu_1} \\
&= -f\left(\frac{1}{p} \boldsymbol{a}_{\cdot 1}^T \boldsymbol{x}_i - \mu_1 \right),\ i = 1, \ldots, N,
\end{aligned}$$

and

$$\left. \frac{\partial p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)}{\partial \mu_1} \right|_{\boldsymbol{a}_{\cdot 1} = \mathbf{0}} = -f(-\mu_1),\ i = 1, \ldots, N.$$

Thus,

$$\begin{aligned}
\left. \frac{\partial g_1(\boldsymbol{a}_{\cdot 1}, \mu_1, \mathbf{C})}{\partial \mu_1} \right|_{\boldsymbol{a}_{\cdot 1} = \mathbf{0}} &= \frac{1}{N} f(-\mu_1) \left(\sum_{i \in I_1} W_{12}(C_{23} - C_{22}) \right. \\
&\quad \left. + \sum_{i \in I_2} W_{21}(C_{13} - C_{12}) \right) \\
&= \frac{1}{N} f(-\mu_1)(W_{12}(C_{23} - C_{22})|I_1| \\
&\quad + W_{21}(1 - C_{23} - 1 + C_{22})|I_2|) \\
&= \frac{1}{N} f(-\mu_1)(C_{23} - C_{22})(W_{12}|I_1| - W_{21}|I_2|).
\end{aligned}$$

Since f is a probability density function, the expression $\left. \frac{\partial g_1(\boldsymbol{a}_{\cdot 1}, \mu_1, \mathbf{C})}{\partial \mu_1} \right|_{\boldsymbol{a}_{\cdot 1} = \mathbf{0}}$ will always have the same sign for any value of μ_1 and the desired result follows. □

Theorem 3. *For*

$$\lambda^L \geq \frac{1}{N} \max_{j=1,\ldots,p} \left| -W_{21} \sum_{i \in I_2} x_{ij} + W_{12} \sum_{i \in I_1} x_{ij} \right| \max_{\mu_1 \in \{-1,1\}} f(\mu_1), \tag{21}$$

$\boldsymbol{a}_{\cdot 1}^* = \mathbf{0}$ is a stationary point of the S-ORCT of depth $D = 1$.

Proof. Using the monotonicity of μ_1 proven in Lemma 1 and Theorem 2 with $\sigma = 0$, we have that for

$$\lambda^L \geq \max_{\substack{\mu_1 \in \{-1,1\} \\ \mathbf{C} \in \{0,1\}^{2 \times 2}}} \max_{j=1,\ldots,p} \left| \nabla_{\boldsymbol{a}_{j1}} g(\mathbf{0}, \mu_1, \mathbf{C}) \right|$$

$$= \max_{\substack{\mu_1 \in \{-1,1\} \\ \mathbf{C} \in \{0,1\}^{2\times 2}}} \|\nabla_{\boldsymbol{a}_{\cdot 1}} g(\mathbf{0}, \mu_1, \mathbf{C})\|_\infty, \tag{22}$$

where g is as in (20), $\boldsymbol{a}^*_{\cdot 1} = \mathbf{0}$ is a stationary point of thr S-ORCT. The remainder of the proof is devoted to rewriting (22) as in (21).

We proceed with the calculation of the gradient.

For $j = 1, \ldots, p$:

$$\frac{\partial g(\mathbf{0}, \mu_1, \mathbf{C})}{\partial a_{j1}} = \left.\frac{\partial g(\boldsymbol{a}_{\cdot 1}, \mu_1, \mathbf{C})}{\partial a_{j1}}\right|_{\boldsymbol{a}_{\cdot 1}=\mathbf{0}} = \frac{1}{N}\sum_{k=1}^{2}\sum_{i\in I_k}\left(\sum_{k'\neq k} W_{kk'}C_{k'2} - \sum_{k'\neq k} W_{kk'}C_{k'3}\right)\left.\frac{\partial p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)}{\partial a_{j1}}\right|_{\boldsymbol{a}_{\cdot 1}=\mathbf{0}},$$

where

$$\frac{\partial p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)}{\partial a_{j1}} = \frac{\partial F\left(\frac{1}{p}\boldsymbol{a}_1^T\boldsymbol{x}_i - \mu_1\right)}{\partial\left(\frac{1}{p}\boldsymbol{a}_1^T\boldsymbol{x}_i - \mu_1\right)}\frac{\partial\left(\frac{1}{p}\boldsymbol{a}_1^T\boldsymbol{x}_i - \mu_1\right)}{\partial a_{j1}} = \frac{x_{ij}}{p} f\left(\frac{1}{p}\boldsymbol{a}_1^T\boldsymbol{x}_i - \mu_1\right),\ i = 1, \ldots, N.$$

and

$$\left.\frac{\partial p_{i1}(\boldsymbol{a}_{\cdot 1}, \mu_1)}{\partial a_{j1}}\right|_{\boldsymbol{a}_{\cdot 1}=\mathbf{0}} = \frac{x_{ij}}{p} f(-\mu_1),\ i = 1, \ldots, N.$$

Thus,

$$\frac{\partial g(\mathbf{0}, \mu_1, \mathbf{C})}{\partial a_{j1}} = \frac{1}{Np} f(-\mu_1)\left(W_{12}\sum_{i\in I_1} x_{ij}(C_{22} - C_{23}) + W_{21}\sum_{i\in I_2} x_{ij}(C_{12} - C_{13})\right).$$

Now, we look for the maximum λ^L among every possible allocation of the decision variables $\mathbf{C}$, i.e.:

$$\lambda^L_{\mu_1} = \max_{\mathbf{C}\in\{0,1\}^{2\times 2}} \|\nabla_{\boldsymbol{a}_{\cdot 1}} g(\mathbf{0}, \mu_1, \mathbf{C})\|_\infty = \max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} \|D\overline{\mathbf{C}}\|_\infty,$$

where

$$D = \frac{1}{Np} f(-\mu_1) \times \begin{pmatrix} -W_{21}\sum_{i\in I_2} x_{i1} & W_{21}\sum_{i\in I_2} x_{i1} & -W_{12}\sum_{i\in I_1} x_{i1} & W_{12}\sum_{i\in I_1} x_{i1} \\ \vdots & \vdots & \vdots & \vdots \\ -W_{21}\sum_{i\in I_2} x_{ip} & W_{21}\sum_{i\in I_2} x_{ip} & -W_{12}\sum_{i\in I_1} x_{ip} & W_{12}\sum_{i\in I_1} x_{ip} \end{pmatrix}$$

and $\overline{\mathbf{C}} = (C_{12}, C_{13}, C_{22}, C_{23})^T$.

$$\begin{aligned}\max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} \|D\overline{\mathbf{C}}\|_\infty &= \max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} \max\left\{|d_1^T\overline{\mathbf{C}}|, \ldots, |d_p^T\overline{\mathbf{C}}|\right\} \\ &= \max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} \max\left\{d_1^T\overline{\mathbf{C}}, -d_1^T\overline{\mathbf{C}}, \ldots, d_p^T\overline{\mathbf{C}}, -d_p^T\overline{\mathbf{C}}\right\} \\ &= \max\left\{\max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} d_1^T\overline{\mathbf{C}}, \max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} -d_1^T\overline{\mathbf{C}}, \ldots, \max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} d_p^T\overline{\mathbf{C}}, \max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} -d_p^T\overline{\mathbf{C}}\right\}.\end{aligned}$$

A finite number of transportation problems is to be solved, with the form:

$$\begin{aligned} z = \max_{\overline{\mathbf{C}}\in\{0,1\}^{4\times 1}} \ & \left\{\pm d_j^T\overline{\mathbf{C}}\right\} \\ \text{s.t.}\ \ & C_{12} + C_{22} = 1 \\ & C_{13} + C_{23} = 1 \\ & C_{12} + C_{13} \geq 1 \\ & C_{22} + C_{23} \geq 1, \end{aligned}$$

for which the integrality property holds. Then, we only have as possible solutions: $\overline{\mathbf{C}} = (1,0,0,1)^T$ or $\overline{\mathbf{C}} = (0,1,1,0)^T$. Thus, the optimal objective is obtained as follows:

$$\begin{aligned} z_{opt} &= \max\left\{\pm d_j^T\overline{\mathbf{C}}\big|_{\overline{\mathbf{C}}=(1,0,0,1)^T}, \pm d_j^T\overline{\mathbf{C}}\big|_{\overline{\mathbf{C}}=(0,1,1,0)^T}\right\} \\ &= \max\left\{\frac{1}{Np} f(-\mu_1)\left(-W_{21}\sum_{i\in I_2} x_{ij} + W_{12}\sum_{i\in I_1} x_{ij}\right), \frac{1}{Np} f(-\mu_1)\left(W_{21}\sum_{i\in I_2} x_{ij} - W_{12}\sum_{i\in I_1} x_{ij}\right)\right\} \\ &= \frac{1}{Np} f(-\mu_1)\left|-W_{21}\sum_{i\in I_2} x_{ij} + W_{12}\sum_{i\in I_1} x_{ij}\right|. \end{aligned}$$

Let us define

$$\lambda^L_{\mu_1} = \frac{1}{Np} f(-\mu_1) \max_{j=1,\ldots,p}\left|-W_{21}\sum_{i\in I_2} x_{ij} + W_{12}\sum_{i\in I_1} x_{ij}\right|,$$

and the result holds when

$$\lambda^L \geq \max\left\{\lambda^L_{\mu_1=-1}, \lambda^L_{\mu_1=1}\right\}.$$

□

4. Computational experience

4.1. Introduction

The aim of this section is to illustrate the performance of our sparse optimal randomized classification trees S-ORCT's. We have run our model for a grid of values of the sparsity regularization parameters λ^L and λ^G. The message that can be drawn from our experimental experience is twofold. First, we show empirically that our S-ORCT can gain in both local and global sparsity, without harming classification accuracy. Second, we benchmark our approach against CART, the classic approach to build decision trees, which considers orthogonal cuts and therefore has the best possible local sparsity. We show that we are able to trade in some of our classification accuracy, still being superior to CART, to be comparable to CART in terms of global sparsity.

The S-ORCT smooth formulation (9)–(16) has been implemented using Pyomo optimization modeling language (Hart et al., 2017; Hart, Watson, & Woodruff, 2011) in Python 3.5 (Python Core Team, 2015). As solver, we have used IPOPT 3.11.1 (Wächter & Biegler, 2006), and have followed a multistart approach, where the process is repeated 20 times starting from different random initial solutions. For CART, the implementation in the `rpart` R package (Therneau & Atkinson, 2019) is used. Our experiments have been conducted on a PC, with an Intel®Core™ i7-2600 CPU 3.40 gigahertz processor and 16 gigabytes RAM. The operating system is 64 bits.

The remainder of the section is structured as follows. Section 4.2 gives details on the procedure followed to test S-ORCT. In Sections 4.3 and 4.4, respectively, we discuss the results for local and global sparsities separately, while in Section 4.5 we present results when both sparsities are simultaneously taken into account. Finally, Section 4.6 statistically compares S-ORCT versus CART in terms of classification accuracy and global sparsity.

4.2. Setup

An assorted collection of well-known real data sets from the UCI Machine Learning Repository (Lichman, 2013) has been chosen for the computational experiments. Table 1 lists their names together with their number of observations, number of predictor

Table 1
Information about the data sets considered.

Data set	Abbrev.	N	p	K	Class distribution
Monks-problems-3	Monks-3	122	11	2	51–49%
Monks-problems-1	Monks-1	124	11	2	50–50%
Monks-problems-2	Monks-2	169	11	2	62–38%
Connectionist-bench-sonar	Sonar	208	60	2	55–45%
Ionosphere	Ionosphere	351	34	2	64–36%
Breast-cancer-Wisconsin	Wisconsin	569	30	2	63–37%
Credit-approval	Creditapproval	653	37	2	55–45%
Pima-indians-diabetes	Pima	768	8	2	65–35%
Statlog-project-German-credit	Germancredit	1000	48	2	70–30%
Banknote-authentification	Banknote	1372	4	2	56–44%
Ozone-level-detection-one	Ozone	1848	72	2	97–3%
Spambase	Spam	4601	57	2	61–39%
Iris	Iris	150	4	3	33.3–33.3–33.3%
Wine	Wine	178	13	3	40–33–27%
Seeds	Seeds	210	7	3	33.3–33.3–33.3%
Balance-scale	Balance	625	16	3	46–46–8%
Thyroid-disease-ann-thyroid	Thyroid	3772	21	3	92.5–5–2.5%
Car-evaluation	Car	1728	15	4	70–22–4–4%

variables and number of classes with the corresponding class distribution. In our pursuit of building small and, therefore, less complex trees, the construction of S-ORCTs has been restricted to depth $D = 1$ for two-class problems and depth $D = 2$ for three- and four-class problems.

Each data set has been split into two subsets: the training subset (75%) and the test subset (25%). The corresponding S-ORCT is built on the training subset and, then, accuracy, local and global sparsities are measured. The out-of-sample accuracy over the test subset is denoted by *acc*. Local sparsity is denoted by δ^L and reads as the average percentage of predictor variables not used per branch node:

$$\delta^L = \frac{1}{|\tau_B|} \sum_{t \in \tau_B} \frac{\left|\left\{a_{jt} = 0,\ \ j = 1, \ldots, p\right\}\right|}{p} \times 100.$$

Global sparsity, δ^G, is measured as the percentage of predictor variables not used at any of the branch nodes, i.e., across the whole tree:

$$\delta^G = \frac{\left|\left\{\boldsymbol{a}_{j\cdot} = \mathbf{0},\ \ j = 1, \ldots, p\right\}\right|}{p} \times 100.$$

Note that when $D = 1$, local and global sparsity are measuring the same since there is a single cut across the whole tree. The training/testing procedure has been repeated ten times in order to avoid the effect of the initial split of the data. The results shown in the tables represent the average of such ten runs to each of the three performance criteria.

In what follows, we describe the choices made for the parameters in S-ORCT. Equal misclassification weights, $W_{y_i k} = 0.5,\ k = 1, \ldots, K,\ k \neq y_i$, have been used for the experiments. We have added the set of constraints (8) with $\rho_k = 0.1,\ k = 1, \ldots, K$. The logistic CDF has been chosen for our experiments:

$$F(\cdot) = \frac{1}{1 + \exp\left(-(\cdot)\gamma\right)},$$

with a large value of γ, namely, $\gamma = 512$. The larger the value of γ, the closer the decision rule defined by F is to a deterministic rule. We will illustrate that a small level of randomization is enough for obtaining good results. We have trained S-ORCT, as formulated in (9)–(16), for 17×17 pairs of values for (λ^L, λ^G) starting from $\lambda^L = 0$

Table 2
Results for the local S-ORCT of depth $D = 1$ as a function of λ^L, where δ^L represents the average percentage of predictor variables not used per branch node in the tree over the ten runs and *acc*, the average out-of-sample accuracy.

λ^L	Monks-3		Monks-1		Monks-2		Sonar		Ionosphere		Wisconsin		Creditapproval		Pima		Germancredit		Banknote		Ozone		Spam	
	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc
0	0	89.7	1	77.7	3	74.3	0	75.8	0	84.1	0	96.2	1	84.1	0	75.8	0	73.5	0	99.0	0	96.5	0	89.8
2^{-12}	1	91.0	21	80.6	28	77.1	0	75.8	4	84.2	1	96.4	9	84.0	0	75.8	0	72.8	0	99.0	4	96.6	0	89.8
2^{-11}	0	91.0	21	79.0	28	77.1	0	77.5	3	84.7	4	96.1	9	83.7	0	75.6	0	72.9	0	99.0	10	96.5	0	89.8
2^{-10}	0	90.0	28	80.0	28	77.1	1	77.5	4	84.5	7	96.0	11	83.9	0	75.6	1	73.3	0	99.1	18	96.4	1	89.8
2^{-9}	0	89.3	27	82.9	28	77.1	2	77.5	4	84.4	10	96.1	13	83.7	0	76	1	73.2	3	99.1	29	96.5	2	89.8
2^{-8}	2	90.0	30	81.6	28	77.1	2	77.7	4	85.2	16	96.3	15	84.2	0	75.7	1	73.2	3	99.1	44	96.4	3	89.8
2^{-7}	0	90.7	23	78.4	28	77.1	5	76.9	8	84.6	28	96.3	16	84.2	0	75.9	2	73.4	3	99.0	62	96.4	5	89.6
2^{-6}	7	90.3	34	80.6	28	77.1	9	77.1	10	85.3	39	96.3	20	84.1	1	75.8	3	73.8	3	98.7	78	96.6	9	89.4
2^{-5}	2	90.3	32	78.4	28	77.1	18	75.4	19	85.9	50	96.3	29	84.6	9	76.2	0	74.2	23	98.5	83	96.6	25	88.8
2^{-4}	3	92.0	29	81.3	28	77.1	28	76.3	32	86.3	59	96.5	44	85.1	20	76.1	31	73.9	15	98.3	87	96.6	44	88.5
2^{-3}	15	92.7	30	83.5	28	77.1	40	77.1	49	86.2	67	96.3	62	86.1	25	75.9	37	73.4	10	98.0	90	96.7	52	86.1
2^{-2}	45	94.3	38	81.6	28	77.1	56	76.9	57	86.1	74	95.8	75	85.4	44	75.3	50	73.8	25	98.0	92	96.7	71	83
2^{-1}	54	94.7	39	81.0	32	78.6	72	78.6	74	85.6	85	95.7	95	86.3	61	74.9	69	71.8	25	97.5	95	96.7	82	78.6
2^{0}	54	94.7	62	81.0	39	76.7	85	78.1	87	86.8	87	94.7	97	86.7	81	73.7	93	69.6	25	96.7	96	96.7	97	64.4
2^{1}	54	94.7	71	78.4	95	63.3	90	78.3	91	84.7	91	92.7	97	86.7	94	65.8	98	69.5	50	85.8	97	96.7	100	60.4
2^{2}	77	74.3	84	72.6	100	64.3	98	62.9	94	75.1	95	91.2	97	86.7	100	63.4	100	69.5	50	84.0	100	96.7	100	60.4
2^{3}	93	55.7	91	72.2	100	64.3	100	51.5	100	61.1	99	64.1	97	86.7	100	63.4	100	69.5	100	56.3	100	96.7	100	60.4

Table 3
Results for the local S-ORCT of depth $D = 2$ as a function of λ^L, where δ^L represents the average percentage of predictor variables not used per branch node in the tree over the ten runs and *acc*, the average out-of-sample accuracy.

λ^L	Iris		Wine		Seeds		Balance		Thyroid		Car	
	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc	δ^L	acc
0	8	95.9	15	96.6	10	94.4	33	96.6	57	92.8	20	92.7
2^{-12}	42	95.9	51	98.6	33	93.8	58	92.0	61	92.7	36	91.5
2^{-11}	42	95.9	54	98.4	38	93.8	60	91.1	59	92.9	33	91.9
2^{-10}	42	96.2	54	97.3	38	94.0	65	91.0	64	92.6	36	91.5
2^{-9}	42	95.9	56	97.5	43	93.8	67	91.2	62	92.7	36	91.4
2^{-8}	42	95.9	56	96.8	48	93.2	60	91.9	65	92.5	36	91.4
2^{-7}	42	95.9	59	96.8	48	91.3	60	91.7	70	92.1	36	91.3
2^{-6}	42	95.9	59	96.8	52	94.0	65	92.2	72	92.1	38	91.6
2^{-5}	42	95.4	59	96.8	52	94.4	58	92.6	74	92.2	40	91.3
2^{-4}	42	95.9	59	97.3	57	93.8	58	92.4	79	92.2	42	91.1
2^{-3}	42	93.2	62	97.5	67	94.6	63	91.1	83	92.1	40	91.7
2^{-2}	50	89.7	62	97.7	67	94.4	65	90.6	87	92.3	47	90.4
2^{-1}	50	92.7	64	98.2	71	93.6	67	89.2	90	92.0	51	90.2
2^{0}	58	90.0	69	96.8	76	93.6	71	88.1	91	91.9	64	87.6
2^{1}	67	90.5	77	95.2	81	90.2	75	87.2	92	92.0	71	85.4
2^{2}	75	91.1	82	89.5	81	88.5	77	82.6	95	91.8	80	80.8
2^{3}	83	88.6	90	76.4	91	73.6	83	77.3	100	92.2	91	68.2

followed by the grid $\left\{\frac{2^r}{p|\tau_B|},\ -12 \le r \le 3,\ r \in \mathbb{Z}\right\}$, and, similarly, $\lambda^G = 0$ followed by the grid $\left\{\frac{2^r}{p},\ -12 \le r \le 3,\ r \in \mathbb{Z}\right\}$. We start solving the optimization problem with $\left(\lambda^L, \lambda^G\right) = (0, 0)$, where the multistart approach uses 20 random initial solutions. We continue solving the optimization problem for $\lambda^L = 0$ but with larger values of λ^G. Once all values of λ^G are executed, we start the process all over again with the next value of λ^L in the grid. For pair (λ^L, λ^G), we feed the corresponding optimization problem with the 20 solutions resulting from the problem solved for the previous pair. For a given initial solution, the computing time taken by the S-ORCT typically ranges from 0.33 seconds (in Monks-1) to 22.27 seconds (in Thyroid).

For CART, the default parameter setting in `rpart` is used.

4.3. Results for local sparsity

Tables 2 and 3 present the results of the so-called local S-ORCT, i.e., when $\lambda^G = 0$ and thus only local sparsity is taken into account. Figs. 2 and 3 depict these results per data set, by showing simultaneously δ^L (blue solid line) and *acc* (red dashed line) as a function of the grid of the λ^L's considered. As expected, the larger the λ^L, the larger the δ^L. The sparsest tree is shown in most of the data sets for large values of the parameter λ^L, where the best solution in terms of sparsity is obtained but the worst possible one in terms of accuracy. In terms of accuracy, the best rates are sometimes achieved when not all the predictor variables are included in the model. For instance, best performance is reached when sparsity is about 9–25% for Pima, the 30% for Monks-1, the 32% for Monks-2, the 44% for Germancredit, the 47% for Car, the 52–56% for Thyroid, the 54% for Monks-3, the 55–60% for Iris, the 72–90% for Sonar, the 81% for both Wine and Seeds and the 87% for Ionosphere. We highlight the Creditapproval data set, on which one single predictor variable can already guarantee very good accuracy. For Ozone, accuracy remains over the 96% for the grid of λ^L's considered. Accuracy might be slightly damaged but a great gain in sparsity is obtained. This is the case for Banknote, Spam, Balance or Wisconsin, which present a loss of accuracy lower than the 1 percentage point (p.p.), 4 p.p., 6 p.p. and 1 p.p. but 25%, 52%, 63% and 85% of local sparsity is reached, respectively.

Table 4
Results for the global S-ORCT of depth $D = 2$ as a function of λ^G, where δ^G represents the average percentage of predictor variables not used per tree over ten runs and *acc*, the average out-of-sample accuracy.

λ^G	Iris		Wine		Seeds		Balance		Thyroid		Car	
	δ^G	acc	δ^G	acc	δ^G	acc	δ^G	acc	δ^G	acc	δ^G	acc
0	0	95.9	0	96.6	0	94.4	0	96.6	1	92.8	0	92.7
2^{-12}	0	96.2	18	97.7	0	94.0	0	96.7	3	93.0	0	93.4
2^{-11}	0	96.2	15	97.5	0	93.8	0	95.4	5	93.9	0	93.7
2^{-10}	0	96.2	15	97.5	0	94.0	0	95.9	5	93.9	0	94.1
2^{-9}	0	95.9	15	97.3	0	93.8	0	96.7	7	94.0	0	94.0
2^{-8}	0	95.9	15	97.7	0	93.8	0	96.2	12	94.1	0	94.7
2^{-7}	0	95.9	15	97.9	14	94.6	0	95.8	17	94.0	0	95.0
2^{-6}	0	95.4	15	98.2	14	95.4	0	96.1	26	94.0	0	94.9
2^{-5}	2	95.7	15	98.2	14	95.4	0	96.7	40	93.9	0	94.9
2^{-4}	0	95.4	15	98.4	14	94.6	0	96.5	57	93.8	0	94.7
2^{-3}	0	95.7	23	98.4	29	93.6	0	94.7	65	93.5	7	94.6
2^{-2}	25	95.4	23	97.9	29	95.2	0	91.1	73	91.5	7	94.1
2^{-1}	25	95.7	31	96.6	29	94.2	19	87.4	81	90.6	13	92.2
2^{0}	50	96.2	39	95.7	43	92.5	25	87.0	83	90.0	27	86.7
2^{1}	50	96.2	46	94.3	57	90.2	44	80.5	87	92.4	47	79.8
2^{2}	50	96.5	62	93.6	71	85.8	56	71.3	95	91.7	73	68.2
2^{3}	75	96.2	85	71.1	86	72.5	94	48.8	100	92.2	80	68.2

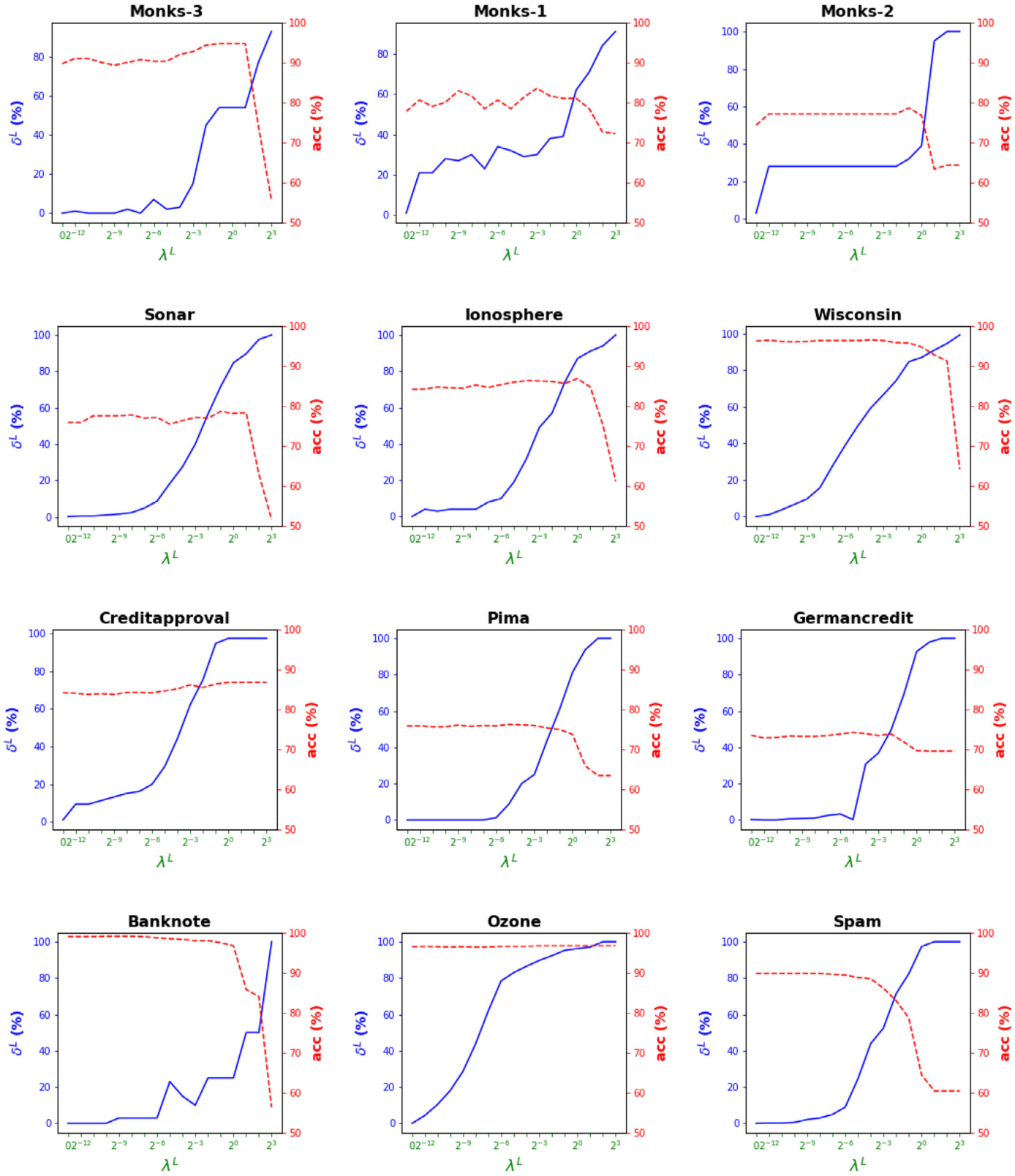


Fig. 2. Graphical representation, for each data set, of the average percentage of predictor variables per branch node, δ^L, together with the average out-of-sample accuracy obtained, *acc*, as a function of the values of λ^L considered in the local S-ORCT construction. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

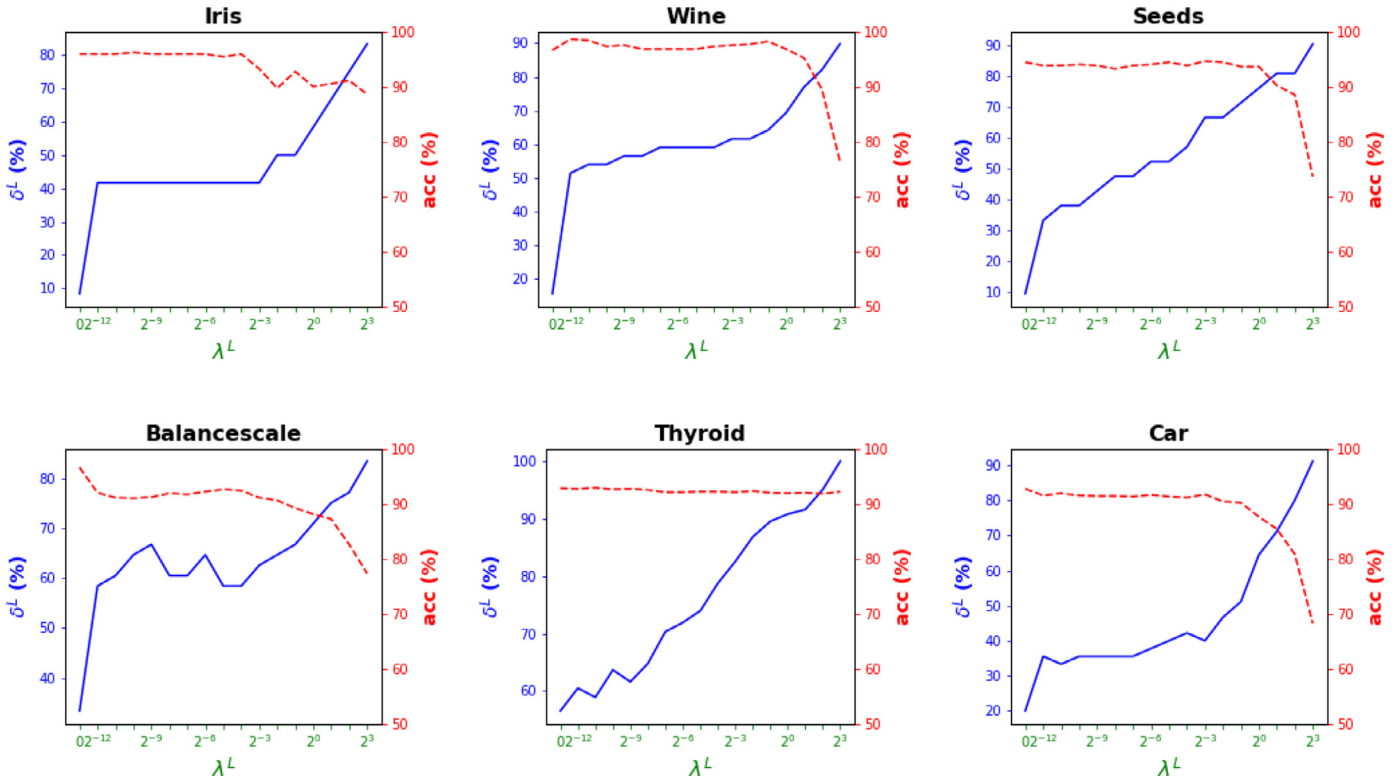


Fig. 3. Graphical representation, for each data set, of the average percentage of predictor variables per branch node, δ^L, together with the average out-of-sample accuracy obtained, *acc*, as a function of the values of λ^L considered in the local S-ORCT construction. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

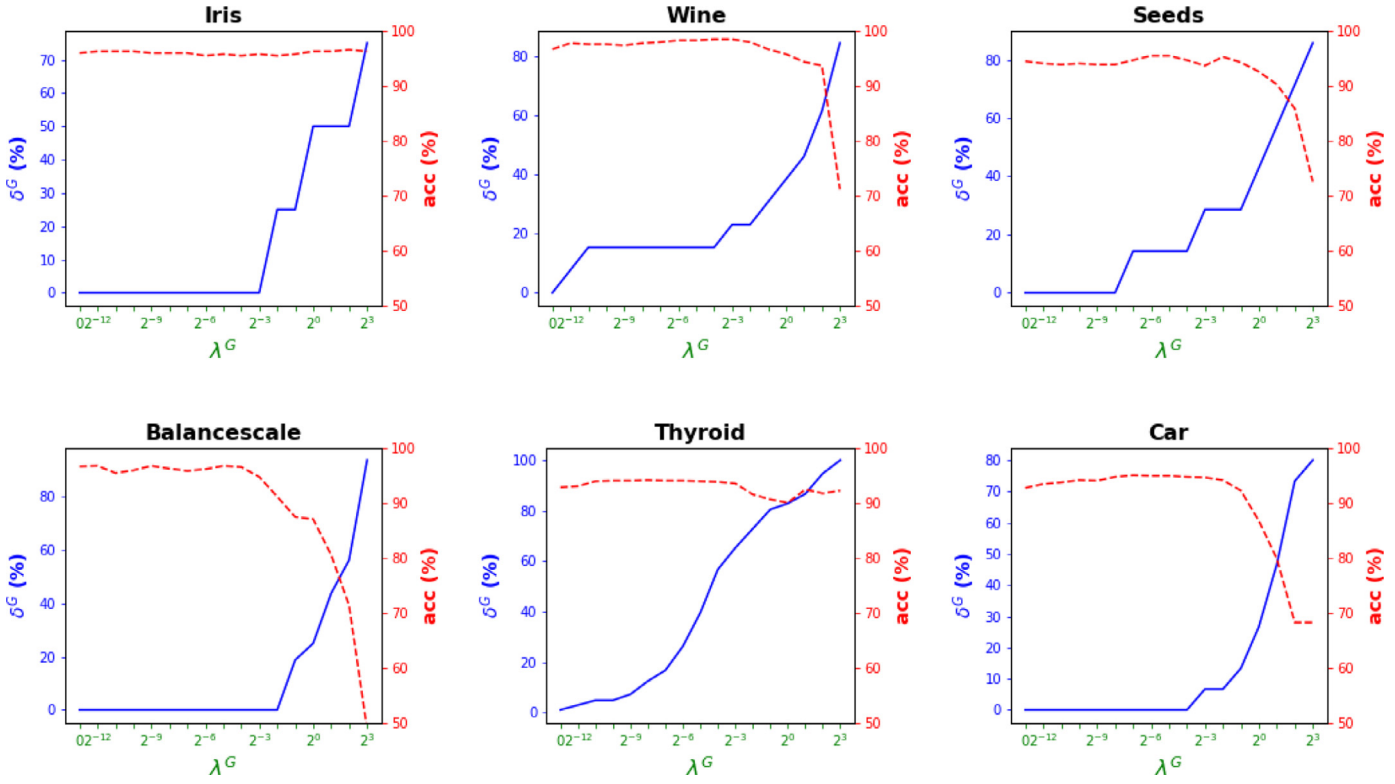
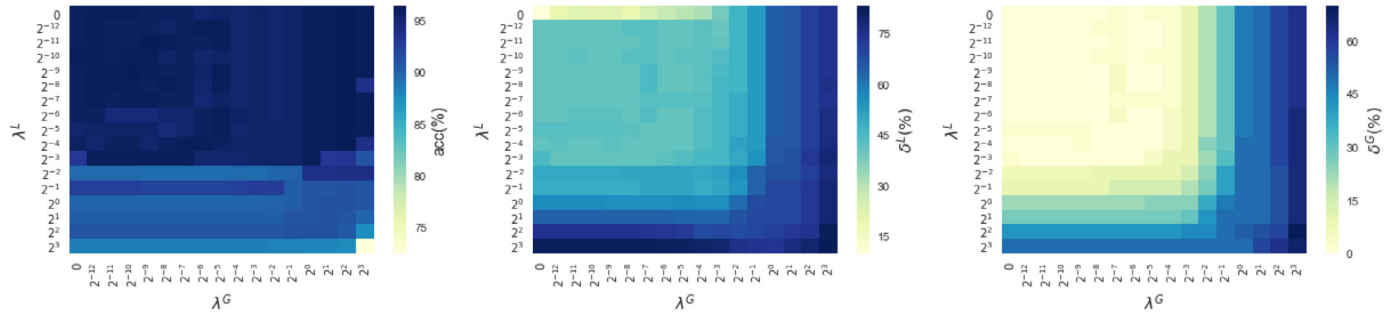
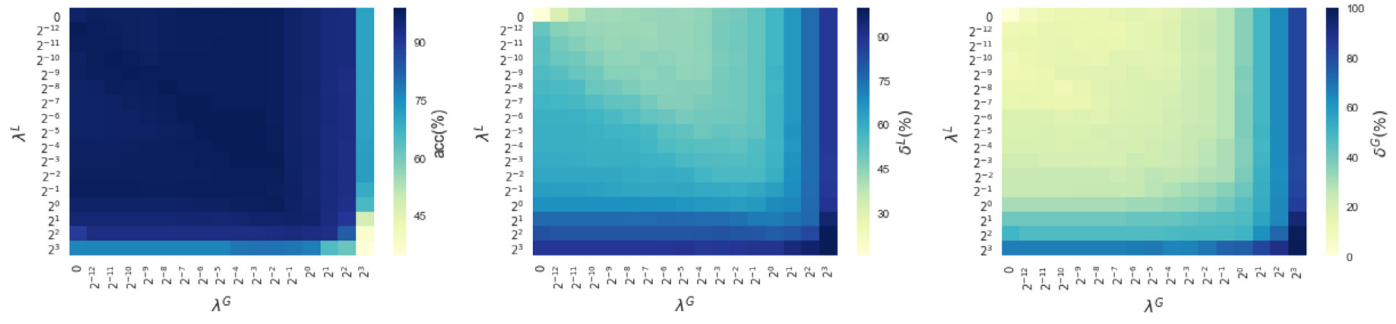


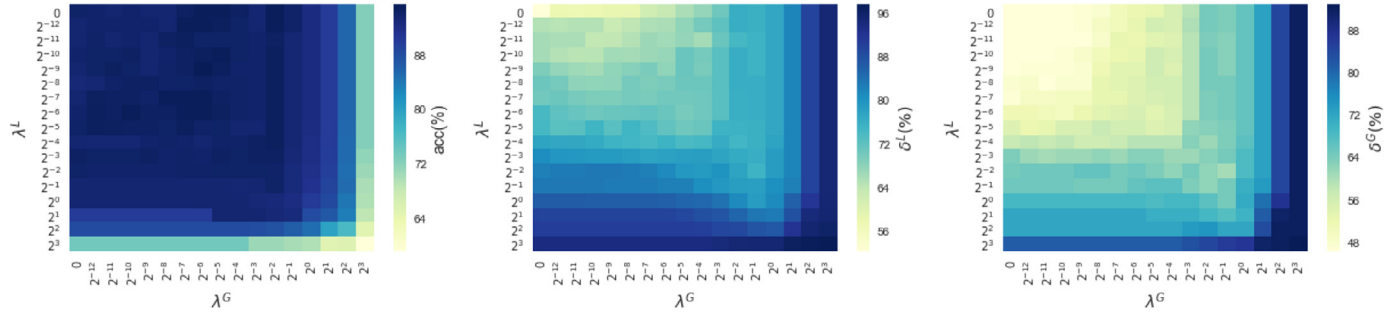
Fig. 4. Graphical representation, for each data set, of the average percentage of predictor variables per tree, δ^G, together with the average out-of-sample accuracy obtained, *acc*, as a function of the values of λ^G considered in the global S-ORCT construction. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.



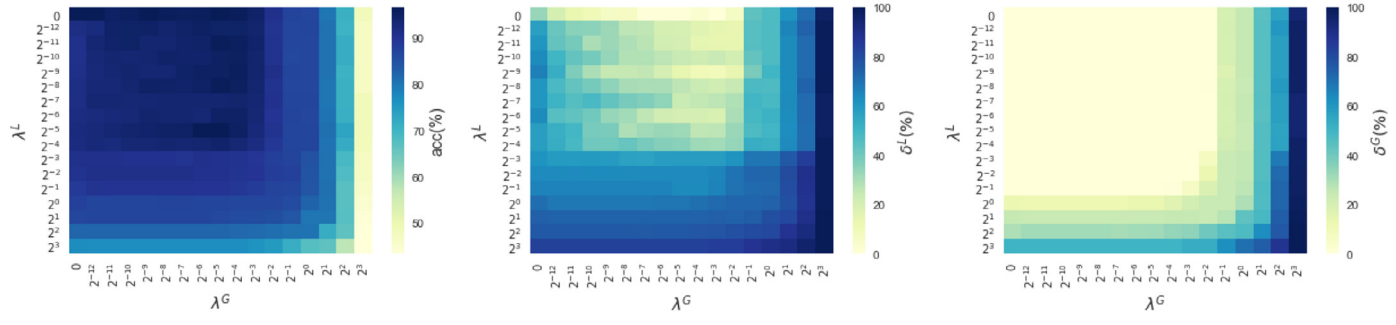
(a) Iris



(b) Wine

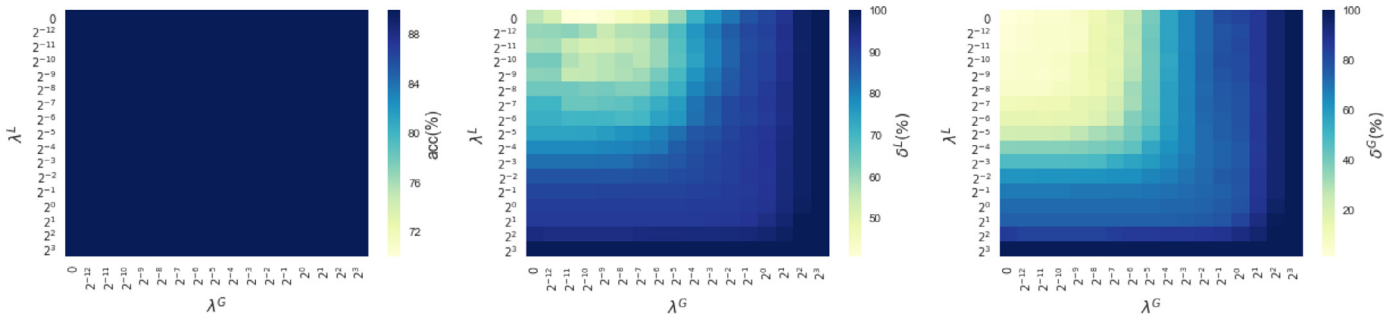


(c) Seeds

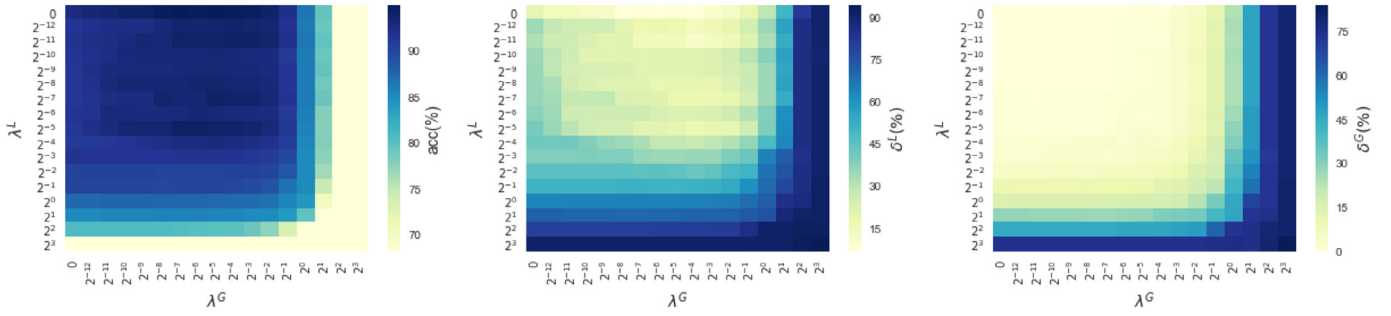


(d) Balance

Fig. 5. Heatmaps representation, for each data set, of the average out-of-sample accuracy, *acc*, the average percentage of predictor variables not used per branch node, δ^L, and the average percentage of predictor variables not used per tree, δ^G, respectively, as a funcion of the grid of the sparsity parameters, λ^L and λ^G, considered in the S-ORCT of depth $D = 2$ construction. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article..



(e) Thyroid



(f) Car

Fig. 5. Continued

4.4. Results for global sparsity

This section is devoted to the global S-ORCT, i.e., when $\lambda^L = 0$ and thus only global sparsity is taken into account. We focus on depth $D = 2$, since for $D = 1$ global sparsity is equal to local sparsity. Similarly to Section 4.3 and Table 4 presents the results of the global S-ORCT, while Fig. 4 visualizes these results by showing simultaneously, per data set, δ^G (blue solid line) and *acc* (red dashed line) as a function of the grid of the λ^G's considered. As for local sparsity, as λ^G grows, δ^G increases. For Iris and Seeds, a similar classification accuracy to that with all of the predictor variables is obtained while removing the 75% and 29% of them, respectively. For Wine, the best rates of accuracy are obtained with 15–23% of global sparsity. A loss of less than 10 p.p. of accuracy is observed for Balance but 25% of predictor variables are not being used, respectively. Car remains around the accuracy rate of 80% while using half of the predictor variables. Thyroid, an imbalanced data set, is over the 90% of accuracy for the whole grid of λ^G's considered.

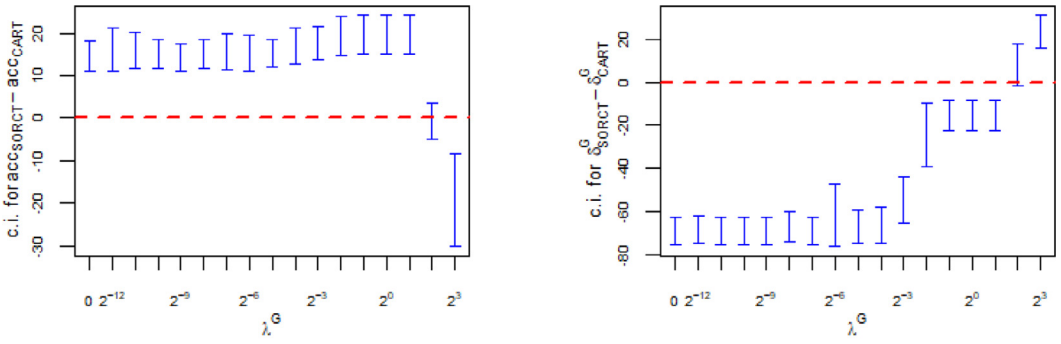
4.5. Results for local and global sparsity

In this section, results enforcing local and global sparsity are presented by means of heatmaps, as seen in Fig. 5. The experiment has been conducted on data sets of $K = 3$ and 4 classes, for which S-ORCTs of depth $D = 2$ are built. For each dataset, three heatmaps are depicted as a function of the grid of the sparsity regularization parameters, λ^L and λ^G: the average out-of-sample accuracy, *acc*, and the local and global sparsities, δ^L and δ^G, respectively, obtained over the ten runs performed. The color bar of each heatmap goes from light green to dark blue, being the latter the maximum accuracy, local sparsity or global sparsity achieved, respectively. As a general behavior, the best rates of accuracy are not always achieved only for $(\lambda^L, \lambda^G) = (0, 0)$, but also for other pairs of the chosen grid, i.e., the data set remains equally well explained while needing less information. As before, according to local sparsity, for a fixed λ^G, δ^L has a growing trend. A similar behavior is observed for δ^G when λ^L is fixed. It is also worth mentioning that small changes of λ^L quickly lead to a gain in δ^L. Nevertheless, as expected, the gain in δ^G is slower for the same range in λ^G.

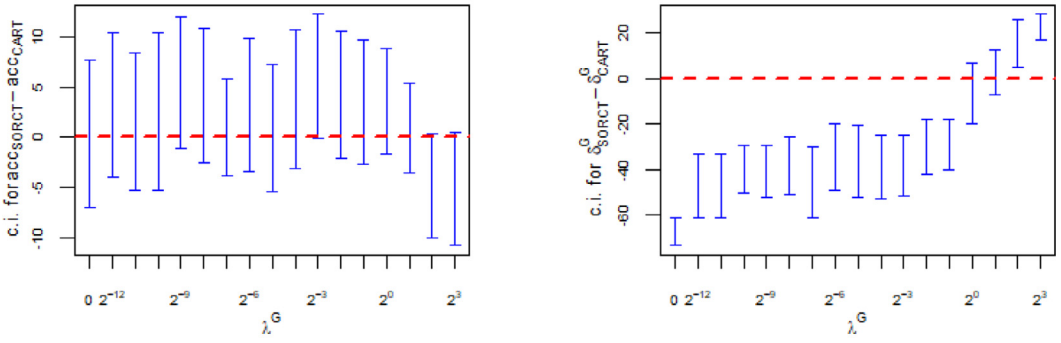
4.6. Comparison S-ORCT versus CART

A statistical comparison between the proposed S-ORCT and CART, the classic approach to build decision trees, is provided in this section. As stated in the introduction of the paper, CARTs, as many other approaches that implement orthogonal cuts (Bertsimas & Dunn, 2017; Firat et al., 2019; Günlük et al., 2019), are leaders in terms of local sparsity. Thus, the comparison S-ORCT versus CART is performed in terms of accuracy and global sparsity. Tables 2 and 4 for S-ORCT have been considered for the experiment.

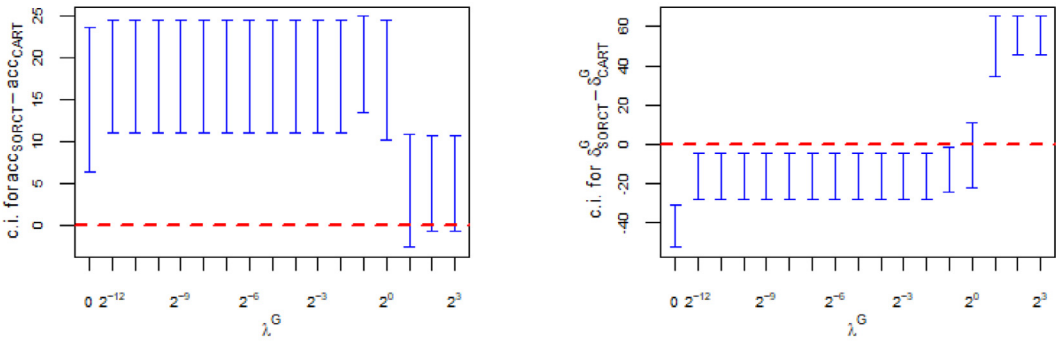
CART has been trained and tested over the same ten runs as S-ORCT. For each pair S-ORCT(λ^G) versus CART, two hypothesis tests for the equality of means of paired samples were carried out, one for accuracy and another for global sparsity, assuming normality, at a 5% significance level. For this task, the `t.test` function in R has been used. Fig. 6 depicts, for each data set, the resulting confidence intervals (blue solid line) at the 95% confidence level for the difference in average accuracy (on the left) and global sparsity (on the right) between S-ORCT(λ^G) and CART. The red dashed horizontal line represents the null hypothesis in each case. Except for Creditapproval and Thyroid, for the smaller values of λ^G, our approach is significantly better than, or at least comparable to, CART in terms of accuracy, while CART is significantly better than, or at least comparable to, in terms of global sparsity. For the larger values of λ^G, our approach starts to be comparable and



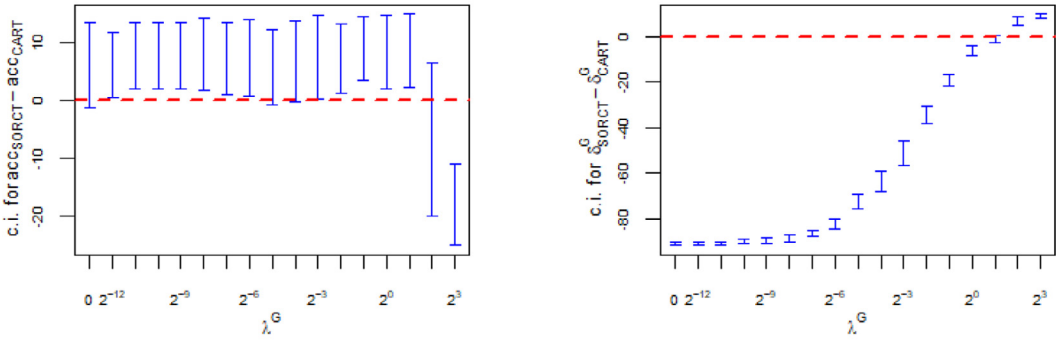
(a) Monks-3



(b) Monks-1

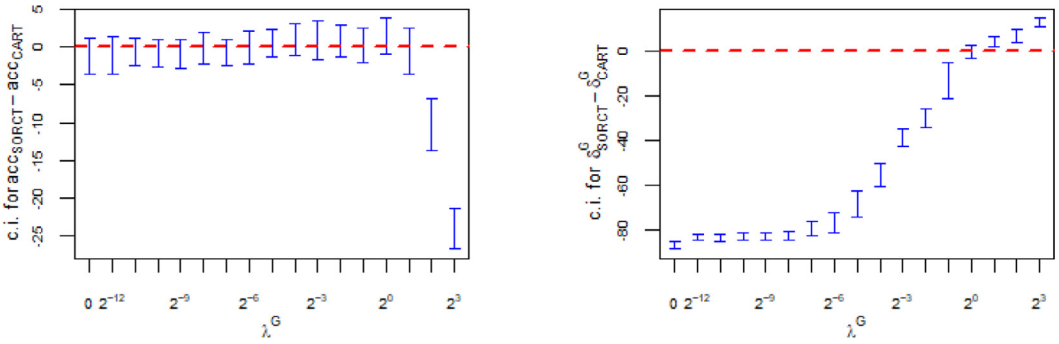


(c) Monks-2

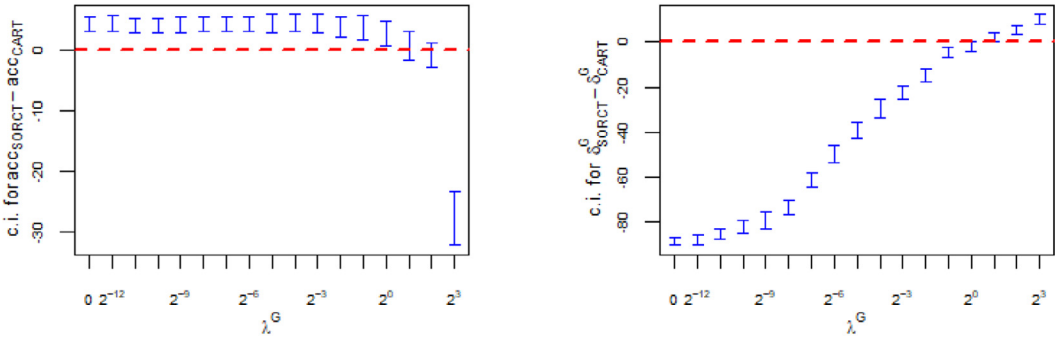


(d) Sonar

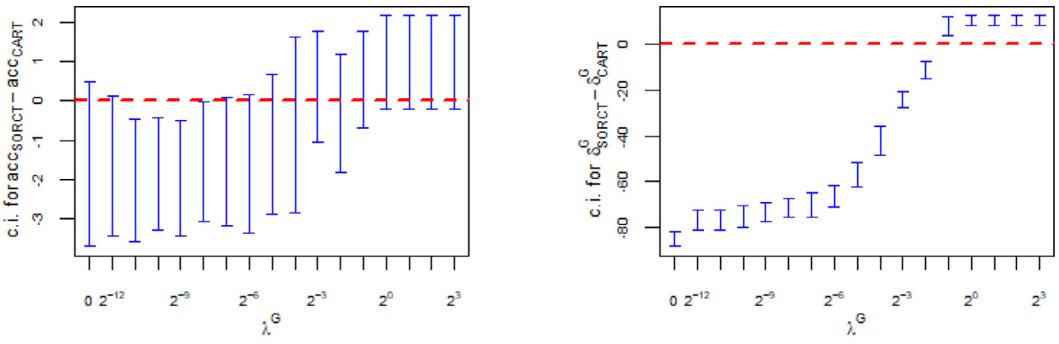
Fig. 6. Graphical representation, for each data set, of the confidence intervals (blue solid line) at the 95% for the difference in average accuracy (on the left) and global sparsity (on the right) between S-ORCT(λ^G) and CART. The red dashed horizontal line represents the null hypothesis in each case. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article..



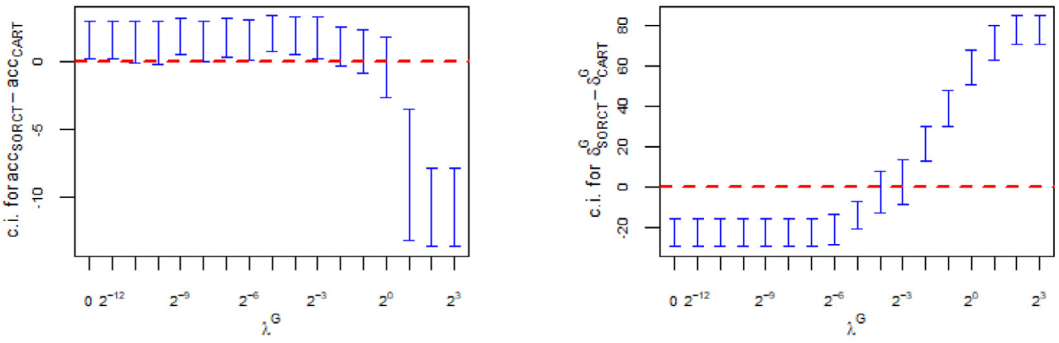
(e) Ionosphere



(f) Wisconsin

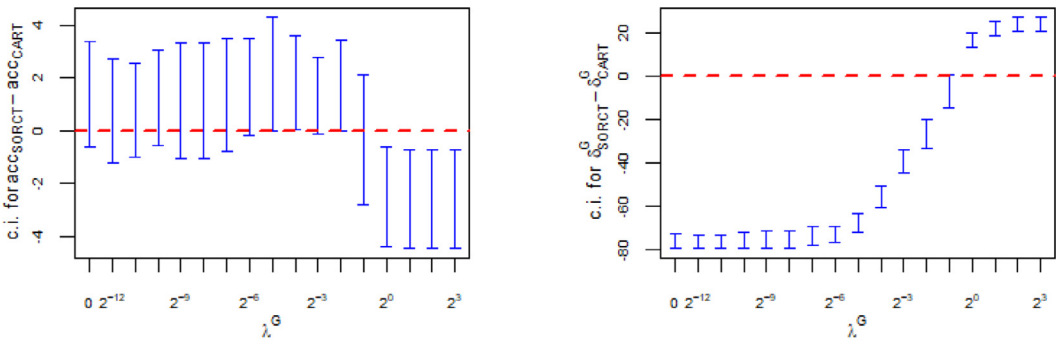


(g) Creditapproval

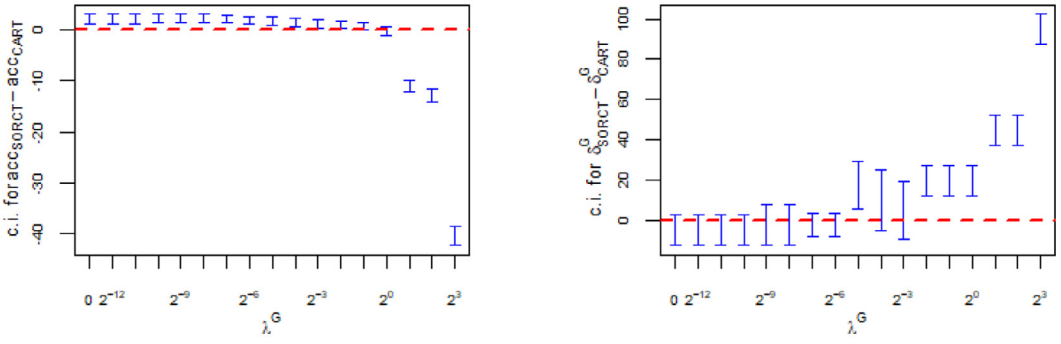


(h) Pima

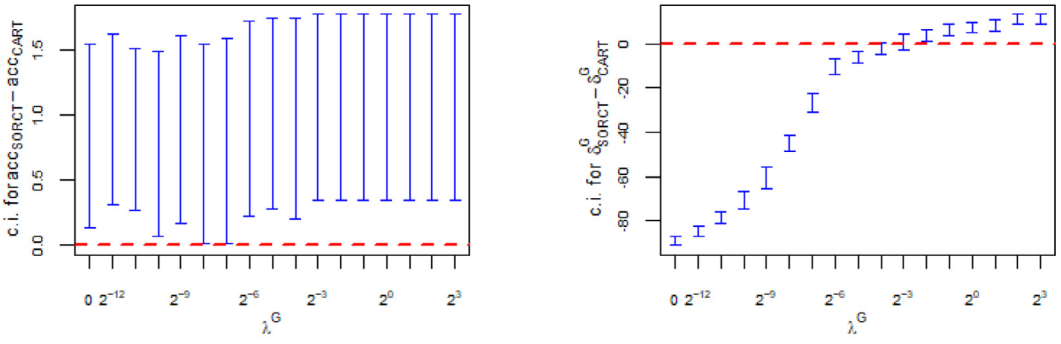
Fig. 6. Continued



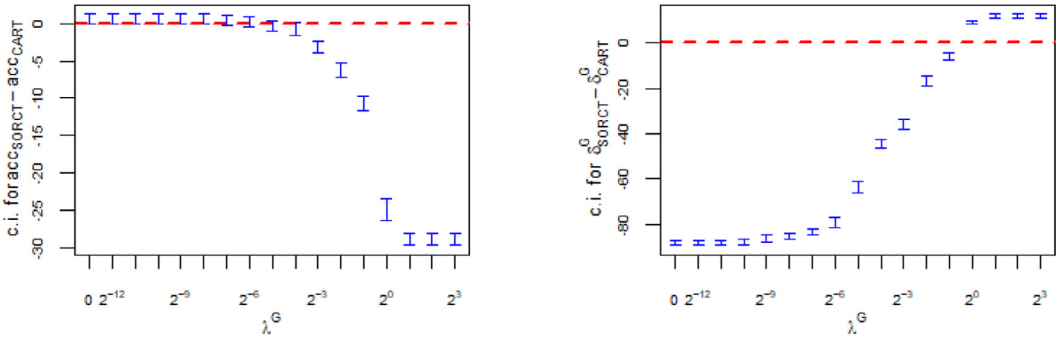
(i) Germancredit



(j) Banknote

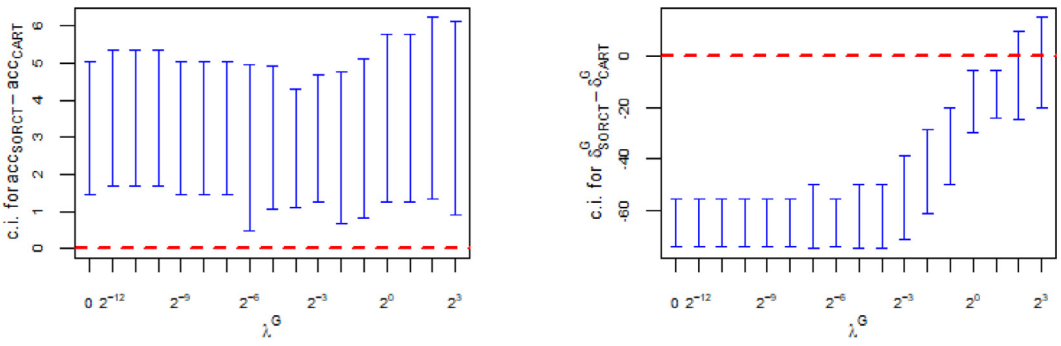


(k) Ozone

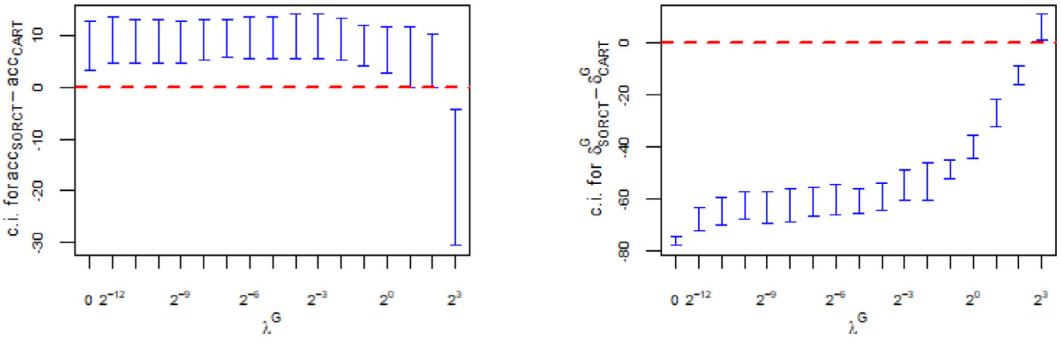


(l) Spam

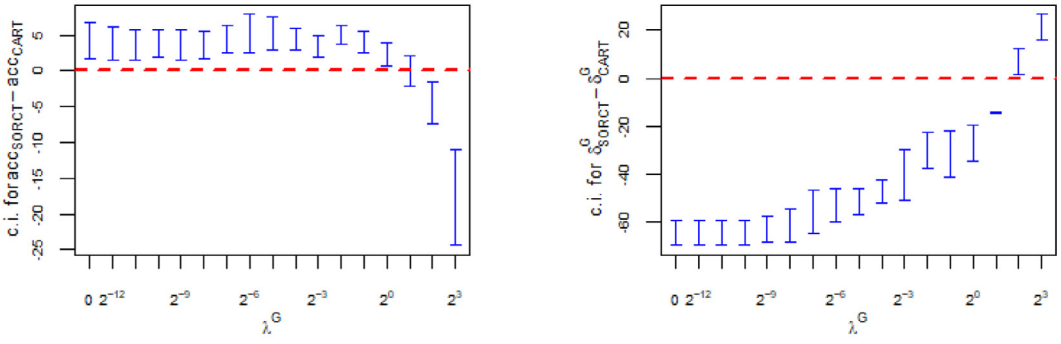
Fig. 6. Continued



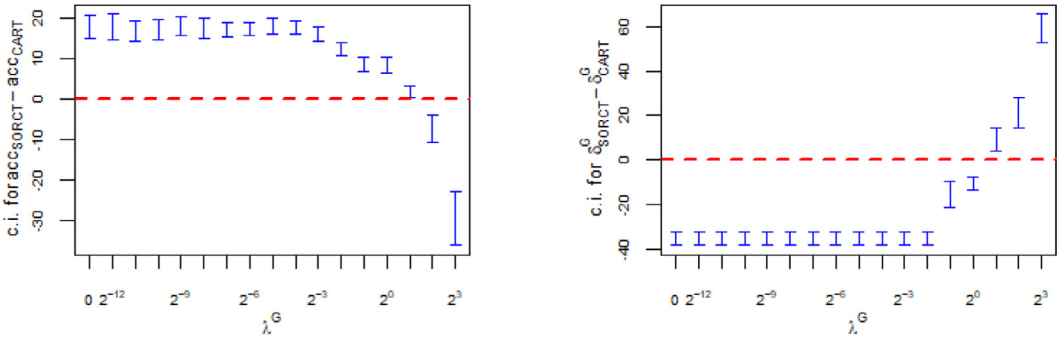
(m) Iris



(n) Wine

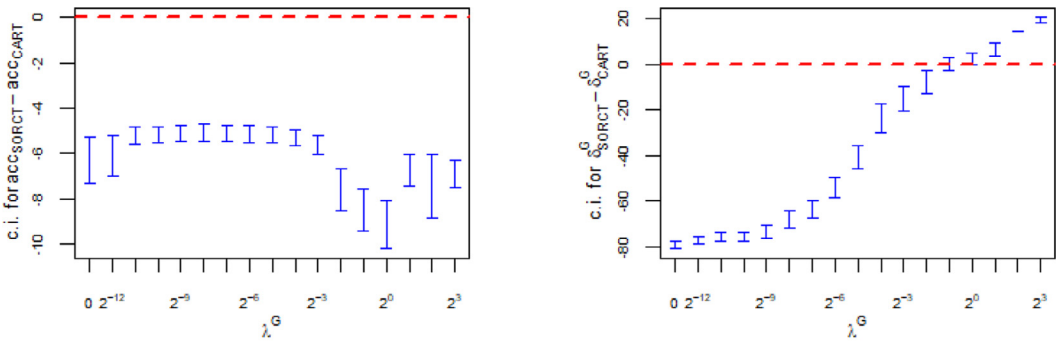


(o) Seeds

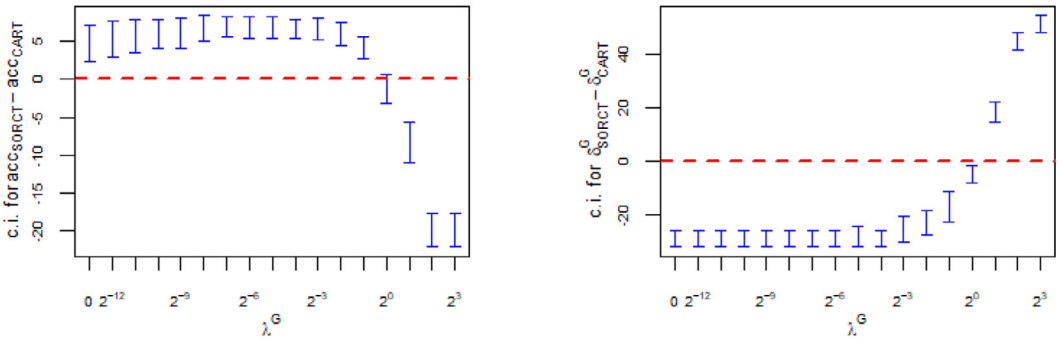


(p) Balance

Fig. 6. Continued



(q) Thyroid



(r) Car

Fig. 6. Continued

then dominate CART in terms of global sparsity at the cost of accuracy.

5. Conclusions and future research

Recently, several proposals focused on building optimal classification trees are found in the literature to address the shortcomings of the classic greedy approaches. In this paper, we have proposed a novel continuous optimization-based approach, the Sparse Optimal Randomized Classification Tree (S-ORCT), in which a compromise between good classification accuracy and sparsity is pursued. Local and global sparsity in the tree are modeled by including in the objective function norm-like regularizations, namely, ℓ_1 and ℓ_∞, respectively. Our numerical results illustrate that our approach can improve both sparsities without harming classification accuracy. Unlike CART, we are able to easily trade in some of our classification accuracy for a gain in global sparsity.

Some extensions of our approach are of interest. First, this methodology can be extended straightaway to a regression tree counterpart, where the response variable is continuous. Second, categorical data is addressed in this paper through the inclusion of dummy predictor variables. For a given categorical predictor variable, and by means of an ℓ_∞-norm regularization, one can link all its dummies across all the branch nodes in the tree, with the aim of better modeling its contribution to the classifier. Third, it is known that bagging trees tends to enhance accuracy. An appropiate bagging scheme of our approach, where sparsity is a key point, is a nontrivial design question.

Acknowledgments

This research has been financed in part by research projects EC H2020 MSCA RISE NeEDS (Grant agreement ID: 822214), COSECLA - Fundación BBVA, MTM2015-65915R, Spain, P11-FQM-7603 and FQM-329, Junta de Andalucía, the last three with EU ERF funds. This support is gratefully acknowledged.

References

Athey, S. (2018). The impact of machine learning on economics. In *The economics of artificial intelligence: An agenda*. University of Chicago Press.
Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science, 49*(3), 312–329.
Bennett, K. P., & Blue, J. (1996). Optimal decision trees. *Rensselaer Polytechnic Institute Math Report 214*.
Bertsimas, D., & Dunn, J. (2017). Optimal classification trees. *Machine Learning, 106*(7), 1039–1082.
Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test, 25*(2), 197–227.
Blanquero, R., Carrizosa, E., Molero-Río, C., & Romero Morales, D. (2018). Optimal Randomized Classification Trees. https://www.researchgate.net/publication/326901224_Optimal_Randomized_Classification_Trees.
Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.
Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC Press.
Carrizosa, E., Martín-Barragán, B., & Romero Morales, D. (2011). Detecting relevant variables and interactions in supervised classification. *European Journal of Operational Research, 213*(1), 260–269.
Carrizosa, E., & Romero Morales, D. (2013). Supervised classification and mathematical optimization. *Computers & Operations Research, 40*(1), 150–165.
Deng, H., & Runger, G. (2012). Feature selection via regularized trees. In *Proceedings of the 2012 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.
Deng, H., & Runger, G. (2013). Gene selection with guided regularized random forest. *Pattern Recognition, 46*(12), 3483–3489.
Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research, 15*(1), 3133–3181.
Firat, M., Crognier, G., Gabor, A., Hurkens, C. A. J., & Zhang, Y. (2019). Column generation based math-heuristic for classification trees. *Computers & Operations Research*.
Freitas, A. (2014). Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter, 15*(1), 1–10.
Genuer, R., Poggi, J.-M., Tuleau-Malot, C., & Villa-Vialaneix, N. (2017). Random Forests for Big Data. *Big Data Research, 9*, 28–46.

Goodman, B., & Flaxman, S. (2017). European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine, 38*(3), 50–57.

Günlük, O., Kalagnanam, J., Menickelly, M., & Scheinberg, K. (2019). Optimal generalized decision trees via integer programming. arXiv:1612.03225v3.

Hart, W. E., Laird, C. D., Watson, J.-P., Woodruff, D. L., Hackebeil, G. A., Nicholson, B. L., & Siirola, J. D. (2017). *Pyomo-optimization modeling in python*: 67 (2nd). Springer Science & Business Media.

Hart, W. E., Watson, J.-P., & Woodruff, D. L. (2011). Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation, 3*(3), 219–260.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning* (2nd). New York: Springer.

Hyafil, L., & Rivest, R. L. (1976). Constructing optimal binary decision trees is NP–complete. *Information Processing Letters, 5*(1), 15–17.

Jung, J., Concannon, C., Shroff, R., Goel, S., & Goldstein, D.G. (2017). Simple rules for complex decisions. arXiv:1702.04690v3.

Lichman, M. (2013). UCI Machine Learning Repository.http://archive.ics.uci.edu/ml. University of California, Irvine, School of Information and Computer Sciences.

Maldonado, S., Bravo, C., Lopez, J., & Perez, J. (2017). Integrated framework for profit-based feature selection and SVM classification in credit scoring. *Decision Support Systems, 104*, 113–121.

Maldonado, S., & Lopez, J. (2017). Synchronized feature selection for support vector machines with twin hyperplanes. *Knowledge-Based Systems, 132*, 119–128.

Martens, D., Baesens, B., Van Gestel, T., & Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research, 183*(3), 1466–1476.

Martín-Barragán, B., Lillo, R., & Romo, J. (2014). Interpretable support vector machines for functional data. *European Journal of Operational Research, 232*(1), 146–155.

Norouzi, M., Collins, M., Johnson, M. A., Fleet, D. J., & Kohli, P. (2015). Efficient non–greedy optimization of decision trees. In *Proceedings of the Advances in neural information processing systems* (pp. 1729–1737).

Olafsson, S., Li, X., & Wu, S. (2008). Operations research and data mining. *European Journal of Operational Research, 187*(3), 1429–1448.

Python Core Team (2015). *Python: A dynamic, open source programming language* https://www.python.org.

Ridgeway, G. (2013). The pitfalls of prediction. *National Institute of Justice Journal, 271*, 34–40.

Silva, A. P. D. (2017). Optimization approaches to supervised classification. *European Journal of Operational Research, 261*(2), 772–788.

Therneau, T. & Atkinson, B. (2019). rpart: Recursive partitioning and regression trees. R package version 4.1–15, https://CRAN.R-project.org/package=rpart.

Tibshirani, R., Wainwright, M., & Hastie, T. (2015). *Statistical learning with sparsity. The lasso and generalizations*. Chapman and Hall/CRC.

Ustun, B., & Rudin, C. (2016). Supersparse linear integer models for optimized medical scoring systems. *Machine Learning, 102*(3), 349–391.

Verwer, S., & Zhang, Y. (2017). Learning decision trees with flexible constraints and objectives using integer optimization. In *Proceedings of the International conference on AI and OR techniques in constraint programming for combinatorial optimization problems* (pp. 94–103). Springer.

Verwer, S., Zhang, Y., & Ye, Q. C. (2017). Auction optimization using regression trees and linear models as integer programs. *Artificial Intelligence, 244*, 368–395.

Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming, 106*(1), 25–57.

Yang, L., Liu, S., Tsoka, S., & Papageorgiou, L. G. (2017). A regression tree approach using mathematical programming. *Expert Systems with Applications, 78*, 347–357.

Zou, H., & Yuan, M. (2008). The F-infinity norm support vector machine. *Statistica Sinica, 18*, 379–398.