TOP
https://doi.org/10.1007/s11750-021-00594-1

ORIGINAL PAPER

Mathematical optimization in classification and regression trees

Emilio Carrizosa[1] · **Cristina Molero-Río**[1] · **Dolores Romero Morales**[2]

Received: 10 December 2020 / Accepted: 27 January 2021

Abstract
Classification and regression trees, as well as their variants, are off-the-shelf methods in Machine Learning. In this paper, we review recent contributions within the Continuous Optimization and the Mixed-Integer Linear Optimization paradigms to develop novel formulations in this research area. We compare those in terms of the nature of the decision variables and the constraints required, as well as the optimization algorithms proposed. We illustrate how these powerful formulations enhance the flexibility of tree models, being better suited to incorporate desirable properties such as cost-sensitivity, explainability, and fairness, and to deal with complex data, such as functional data.

Keywords Classification and regression trees · Tree ensembles · Mixed-integer linear optimization · Continuous nonlinear optimization · Sparsity · Explainability

Mathematics subject classification 90C11 · 90C30 · 62-07

1 Introduction

Extracting knowledge from data is a crucial task in Statistics and Machine Learning, and is at the core of many fields, such as Biomedicine (Jakaitiene et al. 2016; Pardalos et al. 2007); Business Analytics (Martens et al. 2007; Van Vlasselaer et al. 2017), Computational Optimization (Khalil et al. 2016; Lodi and Zarpellon 2017),

✉ Dolores Romero Morales
drm.eco@cbs.dk

Emilio Carrizosa
ecarrizosa@us.es

Cristina Molero-Río
mmolero@us.es

[1] Instituto de Matemáticas de la Universidad de Sevilla, Seville, Spain

[2] Department of Economics, Copenhagen Business School, Frederiksberg, Denmark

Published online: 17 March 2021

Criminal Justice (Ridgeway 2013; Zeng et al. 2017), Cybersecurity (Kaloudi and Li 2020; Martínez Torres et al. 2019), Health Care (Bertsimas et al. 2016; Souillard-Mandar et al. 2016), Policy Making (Kleinberg et al. 2018; Wager and Athey 2018), Process Monitoring (Apsemidis et al. 2020), Regulatory Benchmarking (Benítez-Peña et al. 2020a; Esteve et al. 2020). Mathematical Optimization plays an important role in building such models (Bertsimas and Shioda 2007; Fang et al. 2013; Fountoulakis and Gondzio 2016; Goodfellow et al. 2016), in interpreting their output (Carrizosa et al. 2020b; Dash et al. 2018; Rudin and Ertekin 2018; Ustun and Rudin 2016) or visualizing it (Carrizosa et al. 2017, 2018a, b, 2020a, c). See Bottou et al. (2018), Gambella et al. (2020), Liberti (2020) for surveys reviewing the use of Mathematical Optimization in Machine Learning, and Carrizosa and Romero Morales (2013), Duarte Silva (2017), Palagi (2019), and Piccialli and Sciandrone (2018) for surveys focusing on specific methodologies.

Classification and regression trees (Loh 2014) are state-of-the-art methods based on recursive partitioning (Hastie et al. 2009). They are conceptually simple, show excellent learning performance, are computationally cheap, and routines and packages to train them are available in popular languages such as Python and R, and are also appealing in terms of interpretability (Freitas 2014; Hu et al. 2019; Lin et al. 2020; Meinshausen 2010) because of their rule-based nature. This makes them popular in many applications, including, for instance, a credit scoring exercise for granting a loan, described in what follows for illustration purposes. There, we have a dataset of individuals characterized by demographic and financial predictor variables, among others, and, with this information, the model predicts whether customers will be *good* or *bad* payers. In Fig. 1, we have a stylized credit scoring tree that will help us visualize some of the concepts reviewed in this paper.

To construct a tree model, say $\mathcal{T}$, one has at hand a training sample $\mathcal{I} = \left\{\left(\boldsymbol{x}_i, y_i\right)\right\}_{1 \leq i \leq N}$, with $\boldsymbol{x}_i \in \mathbb{R}^p$ the vector of predictor variables and y_i the response variable of individual i. Note that wlog we assume categorical variables have been modeled through dummy ones, and thus replaced by a set of binary variables indicating the presence/absence of each category. The nature of the response variable depends on whether we are dealing with a classification or a regression task. For classification, $y_i \in \{1, \ldots, K\}$ is the class label associated with individual i, while for regression, $y_i \in \mathbb{R}$ is a continuous measurement.

The main goal of a classification and regression tree $\mathcal{T}$ is to predict, as accurately as possible, the response variable y using the predictor variables $\boldsymbol{x}$. On the top of this primary goal, other important characteristics may need to be considered, such as, e.g., cost-sensitivity constraints to protect risk groups (Kao and Tang 2014; Turney 1995), fairness of the method avoiding the discrimination of groups that share sensitive features such as gender and race (Aghaei et al. 2019; Miron et al. 2020; Romei and Ruggieri 2014; Zafar et al. 2017), and explainability properties, e.g., sparsity and local interpretability of the tree model (Lundberg et al. 2020; Lundberg and Lee 2017; Molnar et al. 2020; Ribeiro et al. 2016).

A tree model $\mathcal{T}$ consists of a tree decision structure and a prediction structure. The tree decision structure of $\mathcal{T}$ is defined by two elements, namely, the topology of the tree, i.e., the branch nodes τ_B and the leaf nodes τ_L, as well as the arcs between them, and the splitting rules applied at the branch nodes. The prediction structure is defined

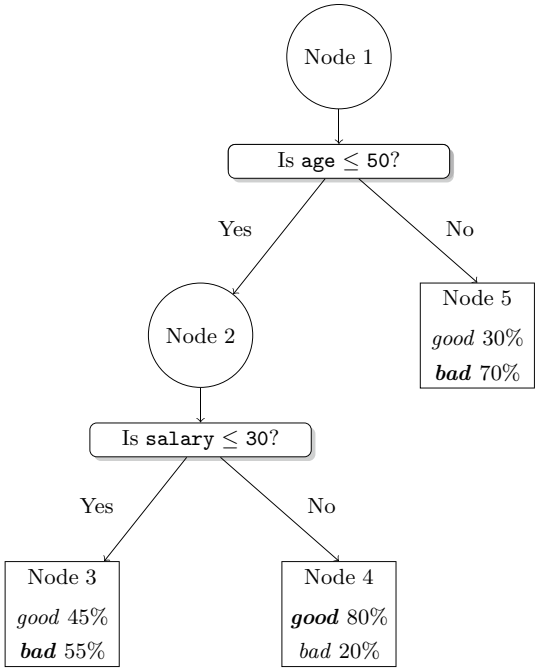

Fig. 1 A tree model $\mathcal{T}$ to predict the *good* payers class vs the *bad* payers class, with $\tau_B = \{\text{Node 1}, \text{Node 2}\}$ and $\tau_N = \{\text{Node 3}, \text{Node 4}, \text{Node 5}\}$ orthogonal cuts $\texttt{age} \leq 50$ and $\texttt{salary} \leq 30$; and prediction *good* for Node 4 and *bad* for Node 3 and Node 5

by the (statistical) prediction models attached to the leaf nodes. To illustrate these concepts, consider the topology of the tree model in Fig. 1, which consists of two branch nodes, Node 1 and Node 2, and three leaf nodes, Node 3, Node 4, and Node 5. This is a binary tree, since each branch node has two children. The root node is where all individuals of $\mathcal{I}$ start. These individuals move along the tree according to the queries asked at the branch nodes. In this way, and after partitioning the training sample $\mathcal{I}$ successively, each individual ends up reaching exactly one leaf node, yielding $\mathcal{I} = \cup_{t \in \tau_L} \mathcal{I}_t$ with $\mathcal{I}_t \cap \mathcal{I}_{t'} = \emptyset$ for $t \neq t'$. In terms of splitting rules, the query asked in this example at the root node is whether predictor variable `age` is below 30, while at Node 2, we ask whether `salary` is below 50. The purpose of this splitting process is to ensure that individuals in the same leaf node follow the same pattern (i.e., they are from the same class or their response variable can be accurately predicted by a unique model, such as, for instance, a linear or a logistic model) and such pattern is expected to be also present at new individuals falling inside this leaf node. The prediction in leaf node t is chosen fitting a model to the subsample $\mathcal{I}_t$. In Fig. 1, we can see that Node 4 predicts individuals as *good* payers, since this is the most frequent class (in bold font) in Node 4, while, following a similar argument, the other two leaf nodes predict as *bad* payers.

Once the tree model is built, the prediction of future data is done in a deterministic way. Given a new observation $\boldsymbol{x}_{\text{new}}$, starting from the root node, and applying the queries at the branch nodes, it will end up in a leaf node, say $t(\boldsymbol{x}_{\text{new}}) \in \tau_L$. The prediction made for $\boldsymbol{x}_{\text{new}}$ is that associated with leaf node $t(\boldsymbol{x}_{\text{new}})$. In our example, a new individual of `age` 43 and `salary` 28 would end up in Node 3, and therefore, it would be predicted as a *bad* payer.

Mathematical Optimization is present at the three elements that define a tree model, namely, topology design of the tree, branching, and prediction. First, we face the problem of designing the topology of the tree. This network design problem is often avoided, by, e.g., choosing a binary tree of depth D, for a given value of D. To make this decision more data-dependent, a larger tree is built and pruned afterward, collapsing existing leaf nodes into new ones containing more individuals. See, e.g., Sherali et al. (2009) for structural properties of the optimization problem associated with the pruning step. In this way, one obtains a more parsimonious tree, which is expected to perform better for future individuals. Second, we have to decide the splitting rules at each branch node. It is common to see trees implementing splitting rules that correspond to so-called orthogonal cuts, i.e., queries involving a single predictor variable, say $x_j \leq \mu$. The choice of the predictor variable x_j and the threshold value μ can be modeled with 0-1 decision variables. However, it is common to see enumerative procedures being applied independently to each of the branch nodes, thus solving the problem locally and not globally. Although orthogonal cuts are popular (easy to build and to interpret), higher efficiency can be achieved with more sophisticated splitting rules, such as, e.g., linear oblique cuts, i.e., queries of the form $\sum_{j=1}^{p} a_j x_j \leq \mu$. See, e.g., Street (2005) for the optimization of oblique cuts. Third, and last, we need to decide how predictions are made at each leaf node $t \in \tau_L$. This boils down to solving an optimization problem for each $t \in \tau_L$, the shape of which depends on the nature of the response variable. For instance, in a regression tree, predictions can be made with a linear model obtained through an Ordinary Least Squares model. See, e.g., Demirović and Stuckey (2020) for the optimization of other criteria to measure the quality of prediction.

Because of the availability of more powerful hardware and the dramatic advances in optimization solvers over the last decades, there has been an increased interest by the Mathematical Optimization community to develop novel approaches to build classification and regression trees. In this paper, we review recent contributions within the Continuous Optimization and the Mixed-Integer Linear Optimization (MILO) paradigms, which both have shown good accuracies compared to the traditional heuristic approaches. We compare the Continuous Optimization and the MILO models in terms of the nature of the decision variables and the constraints, as well as other characteristics related to extracting explainability results to aid Data-Driven Decision-Making. Having these powerful formulations enhances the flexibility of tree models to incorporate desirable properties in data science models, stemming from different fields of application, compared to the greedy heuristic approaches.

The remainder of the paper is organized as follows. In Sect. 2, we briefly go through the simplest (greedy heuristic) approaches to construct classification and regression trees, as well as extensions such as Random Forests, to understand how the one-shot optimization of the decisions across the whole tree is overcome. In Sect. 3, we review the Continuous Optimization and the MILO paradigms to build optimal decision trees, and how they compare against each other. In Sect. 4, we describe recent progress of the mathematical optimization community to

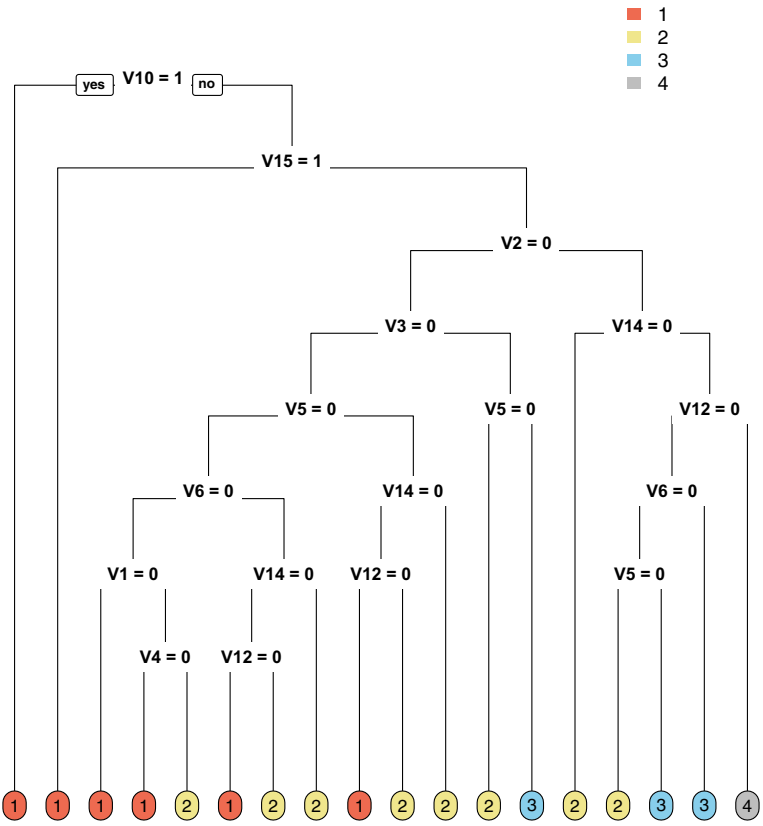

Fig. 2 Illustration of CART for `carevaluations` obtained with the R package `rpart` (Therneau et al. 2015). There are 16 leaf nodes, predicting one of the four classes, namely *unacceptable* (1), *acceptable* (2), *good* (3), or *very good* (4). The classification accuracy provided by this model is 88.1%, while 71.3% of the predictor variables are used across the tree

incorporate important desirable model properties in tree models, and pose new lines of research in this burgeoning area. Finally, Sect. 5 concludes the paper.

2 Greedy classification and regression trees

Throughout this section, we discuss optimization approaches that focus on the design of specific components of the tree model (Aglin et al. 2020; Bennett and Blue 1996; Nijssen and Fromont 2010; Savickỳ et al. 2020). Section 2.1 reviews the basic steps of greedy heuristics to build classification and regression trees with orthogonal cuts. We continue with generalizations that aim at enhancing accuracy. In Sect. 2.2, we discuss tree models with more complex cuts, while Sect. 2.3 is devoted to models that combine a collection of trees. Finally, Sect. 2.4 challenges the greedy approach

when the user needs to control other objectives in addition to the accuracy of the tree model.

2.1 Building a tree model with orthogonal cuts

Since constructing optimal binary classification and regression trees is known to be an NP-complete problem (Hyafil and Rivest 1976), early research traditionally focused on the design of greedy heuristic procedures (Yang et al. 2017) that require a low computational effort to build tree models with just orthogonal cuts. These are recursive partitioning methods that build the tree model in a forward stepwise search implementing orthogonal cuts, yielding binary trees, e.g., CART (Breiman et al. 1984) and QUEST (Loh and Shih 1997), or nonbinary trees, a.k.a. multi-way trees (Kim and Loh 2001), e.g., CHAID (Kass 1980) and C4.5 (Quinlan 1993). Figure 2 depicts the tree model for `carevaluations`, a real-world dataset from the UCI Machine Learning repository (Blake and Merz 1998) with $N = 1728$ car evaluations divided into $K = 4$ classes. This is a dataset with a small number of features, $p = 15$, used to predict whether the car evaluation is *unacceptable*, *acceptable*, *good*, or *very good*.

In these greedy heuristic approaches, a criterion is needed to guide the branching at each branch node. In our credit scoring example, at each branch node, one aims to leave (most of) the *good* payers at one branch and (most of) the *bad* payers at the other one. This has been done by optimizing some measure of the purity of a node with respect to the class split in a classification task, e.g., Gini index or entropy, or its homogeneity with respect to the response variable in a regression task, such as, e.g., mean squared error or mean absolute error. Purity, although popular, only measures classification accuracy in an indirect way, and therefore may not yield good generalization results (Fayyad and Irani 1992), and other criteria, such as Maximum Likelihood (Su et al. 2004), have been proposed. At any branch node, one searches for the splitting rule that yields the larger gain in purity or homogeneity of the children versus the parent. For orthogonal cuts, this implies examining the gain for all the predictor variables and all possible values of threshold, as many as individuals in the parent node. Although, in principle, there may be an infinite amount of values for this threshold, just taking the midpoints between consecutive observed values of the predictor variable in the training sample suffices. We refer the reader to Liu et al. (2002) for a comprehensive review on enhancing classification and regression tree methods through the discretization of continuous predictor variables (Dougherty et al. 1995). The process of partitioning finishes when a stopping criterion is satisfied, for instance, when the requirement on the minimum number of individuals at leaf nodes would be violated. Then, a prediction is chosen in each of the leaf nodes. Commonly, for classification, a leaf node is labeled with the most frequent class in the set of individuals that have fallen into the node, while for regression, the prediction equals to the average of the response variable on those individuals, which is the prediction given by a linear model with just an intercept and no predictor variables.

Trees built in this way may still overfit, and therefore, a post-pruning step is performed to remove some unnecessary splits. Pruning is usually performed in a greedy

fashion in which leaf nodes are sequentially removed. While, in the forward phase, a purity criterion was considered, now, a criterion combining accuracy and tree complexity is used. The removal of leaf nodes continues, while the value of the criterion improves. See Pangilinan and Janssens (2011) for a bi-objective approach to control both criteria.

To enhance their performance, the greedy procedures were extended in different directions, such as the use of global optimization approaches (Barros et al. 2011; Fu et al. 2003; Grubinger et al. 2014). Below, we will elaborate on two other important generalizations, namely, building trees with oblique cuts or based on combining a collection of trees.

2.2 Building a tree model with oblique cuts

The first enhancement relates to extending orthogonal splits to oblique, a.k.a. multivariate, splits, with implementations such as OC1 (Murthy et al. 1994), oblique.tree (Truong 2009), and HHCART (Wickramarachchi et al. 2016). Trees implementing oblique cuts are more versatile and tend to generate smaller trees with better performance (Brodley and Utgoff 1995; Li et al. 2003). This improvement in accuracy comes with increasing computational times, since the enumerative procedure does not apply anymore and, instead, some sort of optimization problem has to be solved at each branch node. In addition, model interpretability may also be harmed. There have been some proposals to build oblique cuts using a baseline classification method at each branch node, such as Support Vector Machines (Orsenigo and Vercellis 2003) or Logistic Regression (Truong 2009), such that the predictions obtained in this way split the parent node into children. Nevertheless, tackling the optimization of oblique cuts is already at the seminal papers of Bennett (Bennett 1992; Bennett and Blue 1996). For binary classification, she adjusts to the tree context the use of Linear Programming (LP) to build separating hyperplanes (Bennett and Mangasarian 1992). In Bennett (1992), the hyperplane that minimizes the average distance from the misclassified individuals to the hyperplane is modeled as an LP problem, while in Bennett and Blue (1996), for a fixed topology of the tree and fixed predictions at the leaf nodes, the problem of finding the optimal oblique cuts for all branch nodes is written as a set of disjunctive linear inequalities yielding a nonlinear problem. Since these approaches apply only to two-class problems, in Street (2005), multi-class problems are addressed. In Norouzi et al. (2015), and given the challenge of optimizing the empirical loss of the tree model, a convex–concave upper bound is optimized instead, using Stochastic Gradient Descent.

2.3 Building an ensemble of trees

The second enhancement of the strategy discussed in Sect. 2.1 relates to building models that combine the outputs given by a collection of trees, as opposed to a single one, by, for instance, bagging or boosting trees (González et al. 2020). The main

exponent of bagging is Random Forests (Biau and Scornet 2016; Breiman 2001; Fawagreh et al. 2014; Genuer et al. 2017), while two of the most popular approaches to boosting are AdaBoost (Freund and Schapire 1997) and Gradient Boosting Machines (Friedman 2001, 2002).

Random Forests (RFs) bag (unpruned) orthogonal trees, and more recently oblique ones (Katuwal et al. 2020; Menze et al. 2011). The trees in the RFs are built independently of each other, on bootstrapped samples of individuals, where the variable selection at each branch node is performed using a random subset of predictor variables. Hence, the trees built differ, because different samples of individuals and features are used. Once the trees are built following the greedy approach described above, RF predicts by combining the predictions of the single trees, e.g., through an average in regression or a majority vote in classification.

AdaBoost is an iterative procedure in which the so-called weak learners (trees of small depth), as well as the individuals, are assigned weights. Individuals for which the prediction was poor in the previous iterations are given a higher weight. Each iteration trains a new weak learner, calculates its error, and defines the weight of the weak learner as well as the weights of the individuals. In some variants of AdaBoost, the weights are optimized (Demiriz et al. 2002; Pfetsch and Pokutta 2020) with techniques such as column generation. While the basic version in classification is designed for two-class classification problems, there are also variants to deal with the multi-class case directly, such as in Hastie et al. (2009), where the authors show that this is equivalent to a forward stage-wise additive modeling algorithm, a.k.a. forward stage-wise boosting, that minimizes a novel exponential loss for multi-class classification.

Gradient Boosting Machines (GBMs) is also an iterative procedure based on combining weak tree learners but usually deeper than in AdaBoost. At the end of each iteration, the residual of the learner at hand is evaluated for each individual. These residuals become the response variable for the next iteration, and therefore, GBM can be seen as a stage-wise additive modeling algorithm. In each iteration, GBM performs a steepest descent minimization for a given loss function, such as mean squared error, mean absolute error, and huber loss functions for regression, and multi-class logistic likelihood for classification. One of the most popular implementations is XGBoost (Chen and Guestrin 2016), which is praised as highly accurate and scalable.

RFs, as well as other methods combining tree models, give, in general, better accuracies than single greedy trees (Fernández-Delgado et al. 2014). However, this is at the expense of losing interpretability and increasing running times. Indeed, these models have a highly complex decision function, being thus less appealing to novel users. The way this lack of interpretability is often addressed is by giving a measure of variable importance, which are often based on permutations of the sample values (Altmann et al. 2010; Louppe et al. 2013; Strobl et al. 2008) or on Game Theory concepts from cooperative games, such as the Shapley value (Casalicchio et al. 2019; Molnar et al. 2018). Recently, there have also been some contributions to enhance model interpretability by replacing the complex model with a simpler surrogate, say a tree model, such that the output of both models is as close as possible. This approach is suggested in Vidal et al. (2020), where the closest tree is

extracted using dynamic programming. Alternatively, one can extract a collection of rules, and techniques such as column generation may be used as in Birbil et al. (2020) or heuristics as in Bénard et al. (2019, 2020).

2.4 Shortcomings of the greedy approach

Classic classification and regression trees, as well as the extensions mentioned above, cannot easily include desirable global structural properties, such as model sparsity and cost-sensitivity, due to their greedy nature. Nonetheless, some attempts have been made to address this shortcoming. To enhance model interpretability, one wishes to perform feature selection to control the number of predictor variables used across the tree (Ruggieri 2019). The regularization framework in Deng and Runger (2012) adds to the criterion optimized in each branch node a penalty term for predictor variables that have not appeared yet in the tree, so that the process is reluctant to use too many predictor variables, yielding a sparse tree model. This approach is refined in Deng and Runger (2013), by also including the importance scores of the predictor variables (Louppe et al. 2013; Strobl et al. 2008), obtained in a preprocessing step running a preliminary RF. In the next section, we model sparsity explicitly, and thus, it can be optimized, as we do with the learning performance of the tree model. Similarly, the control on the performance of the tree model in critical/risk groups is done through cost parameters, such as penalizing with a higher cost the errors in the critical groups, as opposed to modeling the corresponding constraints explicitly as we will do in Sect. 4.

3 Optimal classification and regression trees

In recent times, and because of the dramatic improvements in hardware and optimization solvers (Bixby 2012), many papers on building optimal (in some sense) classification and regression trees have appeared. In this section, we focus on the Continuous Optimization and the Mixed-Integer Linear Optimization paradigms (Bertsimas and Dunn 2017; Blanquero et al. 2020b; Firat et al. 2020; Günlük et al. 2019). The reader is referred to, e.g., Verhaeghe et al. (2019) for a constraint programming paradigm, an SAT one in Narodytska et al. (2018), Yu et al. (2020), and a dynamic programming one in Demirović et al. (2020). This section aims at comparing the two paradigms in terms of type of decision variables and constraints required to model (1) the movement of individuals along the tree and (2) the prediction rule for new individuals.

3.1 Continuous optimization

In this section, we describe how Optimal Randomized Classification and Regression Trees work and what type of Nonlinear Continuous Optimization formulations have been provided in Blanquero et al. (2021, 2020a, b) to build them. Recall that

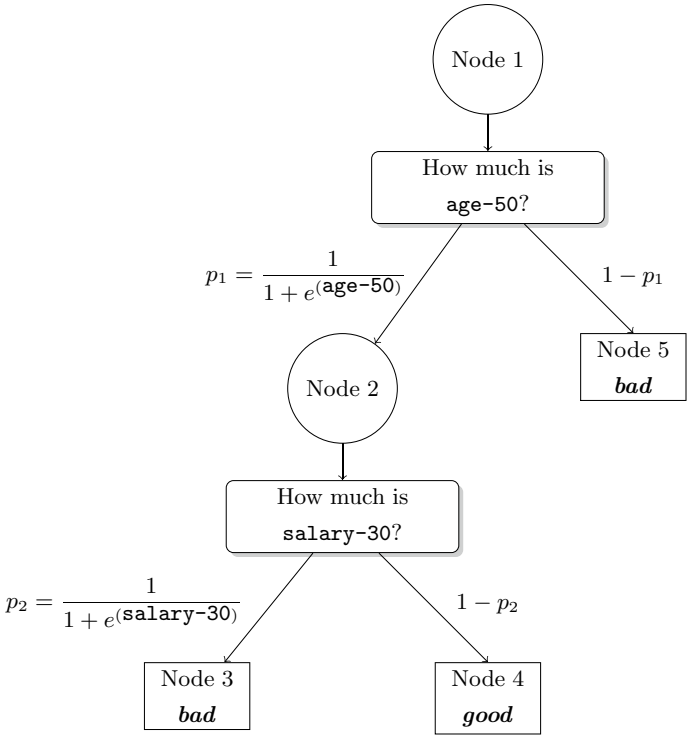

Fig. 3 A randomized tree model $\mathcal{T}$ to predict the *good* payers class vs the *bad* payers class, where F is the CDF of a logistic random variable

$\boldsymbol{x}_i$ denotes the vector of predictor variables of individual i, $i = 1, \ldots, N$. Throughout this section, we typeset other vectors and matrices of decision variables in bold font.

In Optimal Randomized Classification and Regression Trees, the splitting rule at branch node $t \in \tau_B$ is probabilistic (Irsoy et al. 2012; Yang et al. 2018), i.e., individuals move with a certain probability to the left child of t and with the complementary probability to the right one. This type of rule is modeled evaluating $F(\cdot)$, the smooth CDF of a univariate continuous random variable, at the splitting rule in node t, yielding $p_{it}(\boldsymbol{a}_{\cdot t}, \mu_t) = F\left(\frac{1}{p}\boldsymbol{a}_{\cdot t}^{\top}\boldsymbol{x}_i - \mu_t\right)$. See in Fig. 3 the probabilistic splitting rules at Nodes 1 and 2, where F is the CDF of a logistic random variable. With a probabilistic splitting rule, individual i moves along all paths in the tree. The probability distribution across the leaf nodes associated with individual i is defined by $\{P_{it}(\boldsymbol{a}, \boldsymbol{\mu})\}_{t \in \tau_L}$. With the probabilities associated with the individuals and the predictions at the leaf nodes, one can evaluate the total expected error of the randomized tree model. The goal of Optimal Randomized Classification and Regression Trees is to minimize the expected error as well as maximize the so-called local and the

global sparsity of the tree. These sparsity terms are modeled with LASSO terms and controlled with their corresponding parameters.

We present below the Continuous Optimization formulation for Optimal Randomized Classification and Regression Trees in Blanquero et al. (2021, 2020a, b) with the purpose of visualizing the conciseness of its feasible region. Before, we need to introduce some notation and decision variables:

Data

D	depth of the tree,
$\mathcal{N}_t^{\text{left}}$	set of ancestor nodes of leaf node t whose left branch takes part in the path from the root node to leaf node t, $t \in \tau_L$,
$\mathcal{N}_t^{\text{right}}$	set of ancestor nodes of leaf node t whose right branch takes part in the path from the root node to leaf node t, $t \in \tau_L$,
$W_{y_i k} \geq 0$	misclassification cost incurred when classifying an individual i, whose class is y_i, in class k, $y_i, i = 1, \ldots, N,\ k = 1, \ldots, K$,
$F(\cdot)$	smooth CDF of a univariate continuous random variable symmetric w.r.t. 0, used to define the probabilities for an individual to go to the left or the right child node in the tree,
$\lambda^{\text{local}}, \lambda^{\text{global}} \geq 0$	local and global sparsity regularization parameters.

Decisions

$a_{jt} \in \mathbb{R}$	coefficient of predictor variable j in the splitting rule at branch node $t \in \tau_B$, with $\boldsymbol{a} = \left(a_{jt}\right)_{j=1,\ldots,p,\ t\in\tau_B}$. The expressions $\boldsymbol{a}_{j\cdot}$ and $\boldsymbol{a}_{\cdot t}$ will denote the jth row and the tth column of $\boldsymbol{a}$, respectively,
$\mu_t \in \mathbb{R}$	independent term in the splitting rule at branch node $t \in \tau_B$, with $\boldsymbol{\mu} = \left(\mu_t\right)_{t\in\tau_B}$,
$C_{kt} \geq 0$	probability of being assigned to class label $k = 1, \ldots, K$, for an individual at leaf node t, $t \in \tau_L$, with $\boldsymbol{C} = \left(C_{kt}\right)_{k=1,\ldots,K,\ t\in\tau_L}$.

Probabilities

$p_{it}\left(\boldsymbol{a}_{\cdot t}, \mu_t\right)$	probability of individual i going down the left branch at branch node t. Its expression is $p_{it}\left(\boldsymbol{a}_{\cdot t}, \mu_t\right) = F\left(\frac{1}{p}\boldsymbol{a}_{\cdot t}^{\top}\boldsymbol{x}_i - \mu_t\right),\ i = 1, \ldots, N,\ t \in \tau_B$. Note that this probability is a smooth function of $\left(\boldsymbol{a}_{\cdot t}, \mu_t\right)$, due to the smoothness of the CDF F.
$P_{it}(\boldsymbol{a}, \boldsymbol{\mu})$	probability of individual i falling into leaf node t. Its expression is $P_{it}(\boldsymbol{a}, \boldsymbol{\mu}) = \prod_{t\in\mathcal{N}_t^{\text{left}}} p_{it}\left(\boldsymbol{a}_{\cdot t}, \mu_t\right) \prod_{t\in\mathcal{N}_t^{\text{right}}} \left(1 - p_{it}\left(\boldsymbol{a}_{\cdot t}, \mu_t\right)\right),\ i = 1, \ldots, N,\ t \in \tau_L$, and is also a smooth function of $(\boldsymbol{a}, \boldsymbol{\mu})$.

With this notation, the Continuous Optimization formulation to build a randomized classification tree model reads as follows:

$$\underset{(\boldsymbol{a},\boldsymbol{\mu},\boldsymbol{C})\in\mathcal{F}}{\text{minimize}} \ \frac{1}{N}\sum_{i=1}^{N}\sum_{t\in\tau_L} P_{it}(\boldsymbol{a},\boldsymbol{\mu})\sum_{k=1}^{K} W_{y_i k} C_{kt} + \lambda^{\text{local}}\sum_{j=1}^{p}\left\|\boldsymbol{a}_{j\cdot}\right\|_1 + \lambda^{\text{global}}\sum_{j=1}^{p}\left\|\boldsymbol{a}_{j\cdot}\right\|_\infty \tag{1}$$

with

$$\mathcal{F} = \left\{(\boldsymbol{a},\boldsymbol{\mu},\boldsymbol{C}) \in \mathbb{R}^{p|\tau_B|}\times\mathbb{R}^{|\tau_B|}\times\mathbb{R}^{K|\tau_L|} \ : \ \sum_{k=1}^{K} C_{kt} = 1, C_{kt}\geq 0, \ \forall k = 1,\ldots,K \ \forall t\in\tau_L\right\}. \tag{2}$$

The objective function of Problem (1) has three terms. The first one is equal to the average misclassification cost in the training sample, while the second and the third ones are regularization terms. The second term addresses the local sparsity of the tree model, penalizing the ℓ_1-norm of the coefficients of the predictor variables in each of the splitting rules along the tree. The larger the parameter λ^{local}, the fewer predictor variables in the splitting rules, and thus, the tree model would be more similar to a tree with orthogonal cuts, which is much more interpretable. However, it is also important to control the global sparsity of the tree model, i.e., whether a given predictor variable is ever used by the tree model, and thus, the (hopefully few) predictor variables which really affect the classification/regression are identified. This is done in the third term of the objective function through summing for the different predictor variables the ℓ_∞-norm of the vector of coefficients associated with such variable, which forces these coefficients to be shrunk simultaneously across all branch nodes.

In terms of the feasible region $\mathcal{F}$, for each $t\in\tau_L$, we need to impose semi-assignment constraints as well as nonnegativity of C_{kt} to ensure that $\{C_{kt}\}_{k=1}^{K}$ is a probability distribution for the class assignment in leaf node $t\in\tau_L$. Note that, since the technology matrix satisfies the total unimodularity property and the objective function is linear

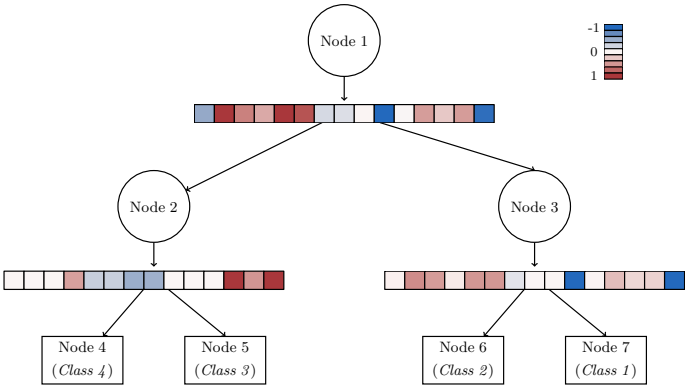

Fig. 4 Illustration of Optimal Randomized Classification Tree for `carevaluations`, with $\lambda^{\text{global}} = \lambda^{\text{local}} = 0$. The classification accuracy of this model is 92.7%, while 100% of the predictor variables are used across the tree as well as in each of the three branch nodes. The magnitude of the coefficients of the splitting rule in each branch node is visualized with a heatmap. The heatmap transitions from blue for negative coefficients, to red for positive ones, while white is chosen for values close to 0

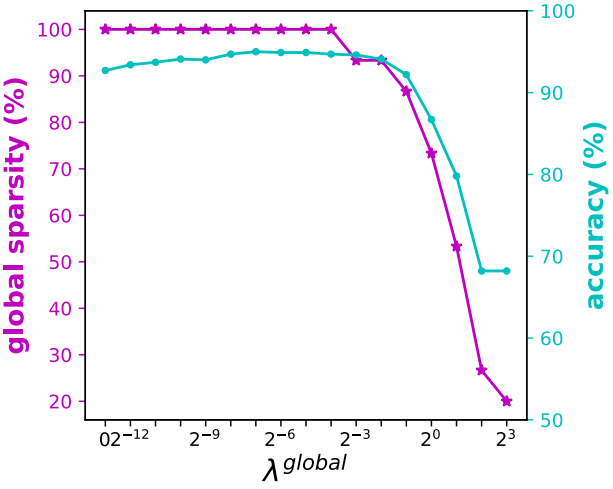

Fig. 5 Trade-off between accuracy and global sparsity in Optimal Randomized Classification Trees for `carevaluations` where λ^{global} is varied

for fixed $(\boldsymbol{a}, \boldsymbol{\mu})$, there exists an optimal solution to Problem (1), such that $C_{kt} \in \{0, 1\}$, meaning that each leaf node predicts exactly one class. Figure 4 plots the Optimal Randomized Classification Tree for `carevaluations`, with $\lambda^{\text{global}} = \lambda^{\text{local}} = 0$, while Fig. 5 illustrates the trade-off between accuracy and global sparsity when λ^{global} is varied.

Once the tree model is built, the prediction of future data is done as follows. Let $(\boldsymbol{a}^*, \boldsymbol{\mu}^*, \boldsymbol{C}^*)$ be the optimal solution to Problem (1). The probability of individual $i \in \mathcal{I}$ being assigned to class k is equal to $\sum_{t \in \tau_L} P_{it}(\boldsymbol{a}^*, \boldsymbol{\mu}^*) C^*_{kt}$, for each $k = 1, \ldots, K$. For an incoming individual with predictor vector $\mathbf{x}$, the probability of belonging to class k returned by the randomized tree is equal to:

$$\mathbf{x} \to \Pi_k(\mathbf{x}) := \sum_{t \in \tau_L} P_{\mathbf{x}\, t}(\boldsymbol{a}^*, \boldsymbol{\mu}^*) C^*_{kt}, \tag{3}$$

where $P_{\mathbf{x}\, t}(\cdot, \cdot)$ is defined similarly to $P_{it}(\cdot, \cdot)$ where $\mathbf{x}$ replaces $\mathbf{x}_i$. Note that $\Pi_k(\cdot)$ is smooth in the continuous predictor variables, since the CDF F is assumed to be a smooth function. This means that even small changes in these variables will produce changes in $\Pi_k(\cdot)$. This is not the case for deterministic tree models such as CART and RF, where there are no changes at all in the class membership probabilities when there are small changes in the continuous predictor variables. The output associated with each $\mathbf{x}$ is probabilistic, namely, the vector of probabilities $\big(\Pi_1(\mathbf{x}), \ldots, \Pi_K(\mathbf{x})\big)$. If a deterministic classification is sought, the class predicted for $\mathbf{x}$ is $k(\mathbf{x}) \in \arg\max\{\Pi_k(\mathbf{x}), k = 1, \ldots, K\}$.

Problem (1) has $(p+1)|\tau_B| + K|\tau_L|$ continuous decision variables, associated with the coefficients of the predictor variables, including the independent terms, as well as with the class assignment, and $|\tau_L|$ linear constraints relating to the class

assignment too. The first term in the objective function is smooth, while the other two terms are not, due to the ℓ_1 and ℓ_∞ norms. With standard techniques, we can find an equivalent smooth formulation, which can be given to any nonlinear solver that can deal with constrained problems. The number of nodes in the tree, and thus the number of decision variables in Problem (1), grows exponentially with the depth of the tree, D. Hence, solving Problem (1) may be time demanding for large or even moderate values of D. Fortunately, the computational experiments in Blanquero et al. (2021, 2020a, b) illustrate that good accuracies can be achieved with small values of D, namely, $D \leq 4$.

Several important remarks on Problem (1) follow. First, the feasible region $\mathcal{F}$ in (2) speaks favorably toward the scalability of Problem (1) with respect to the size of the training sample. Indeed, when the number of individuals N grows, the feasible region remains of the same size, since there are no decision variables directly relating to the individuals. Hence, although the evaluation of the objective function becomes more time demanding with larger N, the dimensionality of the problem to be solved remains the same. Second, there are two regularization terms that can help with feature selection, i.e., identify a subset of predictor variables with a good trade-off between accuracy and sparsity. Third, we can perform with standard techniques a full sensitivity analysis to study the impact that predictor variables have on the class prediction for each individual. Recall that the function $\Pi_k(\cdot)$ in (3) is smooth in the continuous predictor variables. Therefore, we have that small changes in the continuous predictor variables in a given individual lead to small changes in the values of the probabilities of class membership, since $\Pi_k(\cdot)$ can be approximated by its first order Taylor expansion. This means that, for any individual, we can perform a full sensitivity analysis to study the impact that each continuous predictor variable has on the probability of class membership. This is a step forward toward local explainability of tree models, addressed in Sect. 4.

To end this section, we note that Problem (1) can easily be modified for regression. Indeed, one needs to replace the information relating to the prediction of the K classes and the loss incurred, by the prediction of the (continuous) response at each leaf node and a suitable loss. In terms of prediction, one can use any regression model that is compatible with an optimization approach to learning, such as, e.g., a linear, a generalized linear, or an LASSO model. For each individual, the prediction is the expected value of the predictions made at the different leaf nodes, using the probability distribution $\{P_{it}(\boldsymbol{a}, \boldsymbol{\mu})\}_{t \in \tau_L}$. If we take, for instance, the mean squared error, we would have the following unconstrained problem:

$$\begin{aligned} \underset{(\boldsymbol{a},\boldsymbol{\mu},\tilde{\boldsymbol{a}},\tilde{\boldsymbol{\mu}}) \in \mathbb{R}^{(p+1)(|\tau_B|+|\tau_L|)}}{\text{minimize}} \quad & \frac{1}{N} \sum_{i=1}^{N} \Big(\sum_{t \in \tau_L} P_{it}(\boldsymbol{a}, \boldsymbol{\mu}) \, (\tilde{\boldsymbol{a}}_{\cdot t}^{\top} \boldsymbol{x}_i + \tilde{\mu}_t) - y_i \Big)^2 \\ + \lambda^{\text{local}} \sum_{j=1}^{p} \left\| (\boldsymbol{a}_{j\cdot}, \tilde{\boldsymbol{a}}_{j\cdot}) \right\|_1 & + \lambda^{\text{global}} \sum_{j=1}^{p} \left\| (\boldsymbol{a}_{j\cdot}, \tilde{\boldsymbol{a}}_{j\cdot}) \right\|_\infty . \end{aligned}$$

As for Problem (1), and by rewriting the regularization terms, this can be reformulated as a smooth problem with linear constraints. Other losses can be easily modeled too, such as the mean absolute error or the quantile regression (Koenker and

Hallock 2001; Kriegler and Berk 2010). These losses though need to be rewritten, in a similar fashion as for regularization terms, to ensure the smoothness of the objective function. For these losses, none of the decision variables is directly associated with the individuals, and therefore, the dimension of the optimization problem behind regression still does not depend on the size of the training sample N.

3.2 Mixed-integer linear optimization

In this section, we review Mixed-Integer Linear Optimization (MILO) approaches to build Optimal Classification and Regression Trees. The key issue in this paradigm is that one controls the path each individual takes and thus calls for the modeling of (many) binary decision variables. We start with the approach in Bertsimas and Dunn (2017), Dunn (2018), and discuss how it compares to the continuous formulation in the previous section. We then continue by reviewing other relevant literature, which involves different decision variables and/or more sophisticated solution approaches.

In Bertsimas and Dunn (2017), the aim is to build a deterministic optimal binary tree of depth D guided by two objectives, namely, the misclassification error and the complexity of the tree, where the latter is measured as the summation across all branch nodes of the number of predictor variables used in the splitting rules. The MILO formulation in Bertsimas and Dunn (2017), OCT-H in Eq. (28) on pages 1054–1055, requires the following notation and decision variables:

Data

D	depth of the tree,
$\alpha \geq 0$	complexity parameter in the objective function,

Decisions

$d_t \in \{0,1\}$	1 if a cut is applied at branch node t, $t \in \tau_B$,
$l_t \in \{0,1\}$	1 if leaf node t contains individuals, $t \in \tau_L$,
$a_{jt} \in [-1,1]$	coefficient of predictor variable j in the splitting rule at branch node t, $j = 1,\ldots,p,\ t \in \tau_B$,
$\mu_t \in \mathbb{R}$	independent term in the splitting rule at branch node t, $t \in \tau_B$,
$z_{it} \in \{0,1\}$	1 if individual i is in leaf node t, $i = 1,\ldots,N,\ t \in \tau_L$,
$C_{kt} \in \{0,1\}$	1 if leaf node t is labeled with class k, $k = 1,\ldots,K,\ t \in \tau_L$,
$s_{jt} \in \{0,1\}$	1 if predictor variable j is used at branch node t, $j = 1,\ldots,p,\ t \in \tau_B$.

As in Sect. 3.1, $\boldsymbol{d}$, $\boldsymbol{l}$, $\boldsymbol{a}$, $\boldsymbol{\mu}$, z, $\boldsymbol{C}$, and $\boldsymbol{s}$ denote the corresponding vector/matrix of decision variables. This formulation requires binary decision variables to define the topology of the tree, namely, $\boldsymbol{d}$ and $\boldsymbol{l}$; continuous decision variables $\boldsymbol{a}$ and $\boldsymbol{\mu}$ to define the splitting rules; binary decision variables $\boldsymbol{z}$ to control in which leaf node the individuals are placed by the tree model; binary decision variables $\boldsymbol{C}$ associated with the class prediction; and binary decision variables $\boldsymbol{s}$ to control the local sparsity of the tree model. This formulation requires more decision variables than Problem (1), many of them are binary, and the number of some of them, z, linearly depends on

the size of the training sample N. Therefore, as noted in Dunn (2018), this approach is only feasible for moderate values of N.

With regards to the objective function, it consists of two terms, and their linear combination through parameter α is to be minimized. The first term in the objective function represents the total misclassification error across all leaf nodes, assuming that individuals are assigned to the majority class in the leaf node they have been assigned to. Auxiliary decision variables are required to linearize the maximization in the majority class as well as big-M constraints linking them with variables C_{kt}. Thus, we cannot use the same arguments as in the previous section, to prove that the integrality constraints of C_{kt} can be relaxed without loss of optimality. The second term of the objective function measures the local sparsity of the tree, by counting the number of predictor variables used at each branch node and summing these up, which can be done through decision variables $\boldsymbol{s}$. This formulation has a nonlinear objective function as in Problem (1), but as mentioned above, one can define additional variables and additional constraints to linearize it. In addition, the regularization terms in Problem (1) are replaced by a term that fully controls the local sparsity with binary decision variables. Global sparsity, although not modeled in Bertsimas and Dunn (2017), can be included, e.g., with an LASSO term as in Sect. 3.1, or by adding new binary decision variables and additional constraints, linking them to the existing ones $\boldsymbol{s}$.

In terms of the feasible region, and as in Problem (1), we have the semi-assignment constraints associated with $\boldsymbol{C}$. In addition, there are other constraints that need to be included in the MILO formulation. Indeed, since this approach is deterministic, we need to impose that each individual reaches exactly one leaf node, and that $\boldsymbol{z}$ are well-defined, i.e., they are compatible with splitting rules applied at the branch nodes. There are well-defined constraints between $\boldsymbol{s}$ and $\boldsymbol{d}$. We have big-M constraints to ensure that $a_{jt} = 0$ if predictor variable j is not used in branch node t, i.e., $s_{jt} = 0$. There are also big-M constraints to ensure that $z_{it} = 0$ if node t does not contain individuals, i.e., $l_t = 0$, as well as to force that the corresponding coefficients $\boldsymbol{a}$ and $\boldsymbol{\mu}$ to be zero if no split is applied at a branch node. Finally, we have to forbid that a branch node splits if its parent did not, except for the root node.

This formulation can be given to any MILO solver. As in Sect. 3.1, the computational experiments in Bertsimas and Dunn (2017) illustrate that good accuracies can be achieved with small values of D, but at a considerable computational cost for small and medium problem instances. To reduce this computational burden, a local search approach is proposed in Dunn (2018), where the MILO formulation is solved for the subproblems associated with branch nodes, thus yielding smaller formulations that are solved repeatedly. With this local search procedure, it is possible to deal with deeper trees, $D \leq 10$, more efficiently. However, it is harder to directly control, for instance, the global sparsity of the tree, a crucial issue if, on the top of having a procedure yielding high accuracies, identification of the relevant predictor variables is sought. Moreover, contrary to the randomized trees, we cannot perform a proper sensitivity analysis to explain how small perturbations on a given feature affect the prediction. This means that it is not easy

to identify the relevant variables for a given individual, while this is obtained as a byproduct for randomized trees.

The MILO formulation we have just described can be modified to implement orthogonal cuts by making $\boldsymbol{a}$ binary, while the feasible region and the objective function require some small changes. The formulation can also be modified to address a regression task (Dunn 2018), as we have done for Problem (1).

There are other approaches in the literature within the MILO paradigm. In Aghaei et al. (2020), a flow-based MILO formulation is proposed for binary predictor variables. A sink node is added to the tree, yielding a directed acyclic graph (Ahuja et al. 1993). Only individuals ending up in the sink are correctly classified, while flow conservation constraints are imposed at the other nodes of the tree. In Firat et al. (2020), an alternative formulation is proposed with new decisions variables associated with the paths from the root node to the leaf nodes and their splits, which is solved with a column generation-based heuristic. In Günlük et al. (2019), an MILO formulation for combinatorial splitting rules for categorical variables is proposed, i.e., rules defined by a subset of categories that move individuals to the left child node if the rule is satisfied and to the right child node, otherwise, yielding a binary representation of them (Carrizosa et al. 2019).

With the MILO approach, we face the curse of dimensionality, since the number of binary decision variables grows linearly with the size of the training sample N. Recent attempts to address this can be found in the literature. An alternative formulation is proposed in Verwer and Zhang (2017), Verwer et al. (2017, 2019) with a more compact feasible region that aggregates some of the constraints described above. In Zhu et al. (2020), a subset of the training sample is selected in a preprocessing step using an LP problem, while, in Zantedeschi et al. (2020), a continuous relaxation is developed.

4 Challenges for the future

Throughout this paper, we have illustrated how powerful optimization is to construct classification and regression tree models $\mathcal{T}$ that show a good trade-off between accuracy and sparsity. This section is devoted to discuss new challenges posed by desirable properties we may want to seek or by the complexity of the data at hand, and the first steps that the Mathematical Optimization community has taken to give answers to them. The first research avenue we touch on consists of expanding the family of criteria under consideration: on the top of accuracy and sparsity, there may be other important requirements on $\mathcal{T}$, such as fairness, to ensure that $\mathcal{T}$ protects sensitive groups (Romei and Ruggieri 2014), or explainability, to ensure that the knowledge gained is actionable (Aouad et al. 2019; Bertsimas et al. 2019; Cui et al. 2015; Höppner et al. 2020) in, for instance, the design of drug therapies (Mišić 2020). The second research avenue consists of designing tree models for more complex data: we will discuss the new challenges that arise when some of the predictor variables available to construct $\mathcal{T}$, or even the response, may not be continuous or categorical, and novel tree models are required with these new types of data. Although of interest, this section does not

address the asymptotic behavior of tree models: assuming data to be a random sample from a given distribution, an important question is to identify the statistical convergence of the random sequence of optimal trees and optimal values (e.g., optimal expected squared error in a regression tree) when the size of the training set goes to infinity. Very limited results are available in the literature, making strong assumptions on the structure of the tree models. The reader is referred to Biau et al. (2008), Denil et al. (2013), Scornet (2016), and Scornet et al. (2015) for some results in this line.

While there has been a paramount increase in the use of Machine Learning in Decision-Making, it is less well understood how models arrive to decisions. Yet, transparent (a.k.a. interpretable, comprehensible, understandable) models, (Cerquitelli et al. 2017; Hofman et al. 2017), are desirable in Medical Diagnosis (Freitas 2014; Ustun and Rudin 2016), or Criminal Justice (Jung et al. 2017, 2020), to name a few. While a black-box model could be extremely good at predicting who would benefit from a policy intervention, policy makers should be able to explain why decisions are taken, as is evident, e.g., in the COVID-19 crisis. Moreover, transparency (Chen et al. 2017) is a must when, for instance, benchmarking the providers of utilities (Benítez-Peña et al. 2020a) or in credit scoring in consumer lending (Baesens et al. 2003), the reason being the so-called right-to-explanation in algorithmic decision-making, imposed by the European Union since 2018 (European Commission 2020; Goodman and Flaxman 2017; Wachter et al. 2017). Although the term Explainable Artificial Intelligence (*XAI*) was coined a while ago, it is now tracking a lot of attention from different communities, see, e.g., Barredo Arrieta et al. (2020), Gunning and Aha (2019), Holter et al. (2018), Miller (2019).

There is a big body of literature relating to a common surrogate for explainability, namely, model sparsity. The aim there is to perform feature selection to work with a smaller number of predictor variables as a first step toward explaining the behavior of the model globally. This can be done after the model has been built, through variable importance measures, deleting those variables with a small importance (Cohen et al. 2007; Guyon and Elisseeff 2003). Examples of these measures were given in Sect. 2, and are based, for instance, on calculating the impact on accuracy by permuting the values of the feature under investigation, or, inspired by cooperative games, calculating the contribution toward the accuracy of the feature to any coalition of features. Alternatively, we can embed the sparsity in the optimization model solved to train the model, as we have seen in the optimal trees reviewed in the previous section, either with LASSO terms (Hastie et al. 2015) or zero-norm terms (Weston et al. 2003).

Equally important is to give local explanations, i.e., at the individual level, say $\boldsymbol{x}^0$. Take, for instance, the stylized credit scoring tree in Fig. 1 and the customer discussed in the introduction of `age` 43 and `salary` 28. Recall this individual was assigned to the *bad* payers class, and therefore was denied the credit. The tree model in Fig. 1 has made this decision, because `age` is below 50 and `salary` is also below 30. This explanation, however, is of limited use to the individual who would still not understand how to improve the credit score to be labeled as *good*, and thus get the credit granted. Moreover, this type of explanations, i.e., the ones given by the

path from the root to the corresponding leaf node, can be arbitrarily long (Izza et al. 2020). Instead, one can offer to individuals counterfactual explanations (Fernández et al. 2020; Lucic et al. 2020; Mothilal et al. 2020). See Karimi et al. (2020), Sokol and Flach (2019), Verma et al. (2020) for recent surveys on counterfactual explanations. This is an explanation about how features need to change to obtain a different class prediction. For our individual at hand, with $\boldsymbol{x}^0 = (43, 28)$, we can say that we have labeled him/her as *bad*, but that with the same `age` and `salary` above 30, the label would have been *good* instead. This is an example of *local explainability*, (Molnar et al. 2020), in which the aim is to identify a small set of features and their corresponding value to make the prediction change.

The goal of local explainability is, therefore, to understand which are the predictor variables that have the largest impact on the individual prediction. In case of a linear regression model, this analysis can naturally be performed using the coefficients β_j, $j = 1, \ldots, p$, of the predictor variables. If variable j changes by Δ_j units, then the response variable changes by $\beta_j \Delta_j$ units, which is clearly independent of the individual. For nonlinear models, one can make use of model-agnostic approaches to build local explanations, such as the so-called Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al. 2016). The latter relies on building linear approximations to the model at $\boldsymbol{x}^0$, using predictor vectors generated in the neighborhood of $\boldsymbol{x}^0$ and predictions obtained with the original model. Although popular, there are a number of shortcomings to this approach around, for instance, the generation of data or the loss of accuracy, which have been addressed with enhancements (Botari et al. 2020; Visani et al. 2020; Zhang et al. 2019), or with alternative approaches (Lundberg et al. 2020, 2018; Lundberg and Lee 2017).

Instead, and as advocated by Rudin (2019), it is better to work with models that can derive local explanations directly, as is the case for Neural Networks (Gevrey et al. 2003) but also for the Optimal Randomized Classification and Regression Trees in Sect. 3.1. To illustrate this, and for simplicity, we consider a classification problem where all predictor variables are continuous. For an individual with predictor variables $\boldsymbol{x}^0$, we analyze how an infinitesimal change $\boldsymbol{\Delta} \in \mathbb{R}^p$ in the predictor variables affects the probability Π_k of being in class $k, k = 1, \ldots, K$. By linearizing Π_k close to $\boldsymbol{x}^0$, we have:

$$\Pi_k(\boldsymbol{x}^0 + \boldsymbol{\Delta}) \approx \Pi_k(\boldsymbol{x}^0) + \sum_{j=1}^{p} \frac{\partial \Pi_k}{\partial x_j}(\boldsymbol{x}^0) \cdot \Delta_j.$$

Thus, the matrix of partial derivatives

$$\left(\frac{\partial \Pi_k}{\partial x_j}(\boldsymbol{x}^0) \right)_{\substack{k = 1, \ldots K \\ j = 1, \ldots, p}}$$

gives full information on the sensitivity of the class membership probabilities Π_k around $\boldsymbol{x}^0$.

Even more, we can provide counterfactual explanations to an individual with $\boldsymbol{x} = \boldsymbol{x}^0$ on what are the minimum changes to the predictor variables, such that the

individual with $\boldsymbol{x} = \boldsymbol{x}^0 + \boldsymbol{\Delta}$ is predicted to be in class k^*. Indeed, given a norm $\|\cdot\|$ and a set $\mathcal{A} \subset \mathbb{R}^p$ of allowed movements from $\boldsymbol{x}^0$, one can solve a nonlinear problem of the form:

$$\begin{array}{ll} \text{minimize}_{\boldsymbol{\Delta}} & \|\boldsymbol{\Delta}\| \\ s.t. & \Pi_{k^*}(\boldsymbol{x}^0 + \boldsymbol{\Delta}) \geq \Pi_k(\boldsymbol{x}^0 + \boldsymbol{\Delta}) \ \forall k = 1, \ldots, K \\ & \boldsymbol{\Delta} \in \mathcal{A}. \end{array}$$

The use of Machine Learning models in socially sensitive decision-making calls for analyzing their fairness (Iosifidis and Ntoutsi 2019; Miron et al. 2020; Zafar et al. 2017). The fairness can refer to the accuracy achieved in risk groups, for which the consequences of a wrong prediction are much more severe than for the rest (Kao and Tang 2014; Turney 1995). These examples abound, for instance, in medical diagnosis. The most natural way to handle this cost-sensitivity is to add the so-called performance constraints, one for each risk group, to ensure that the accuracy achieved in an independent sample is acceptable, i.e., above a threshold. Second, fairness can also refer to avoiding that the outcome of the model discriminates groups of people sharing sensitive features, such as gender or race, (Miron et al. 2020). This has gained attention due to the increase of automatization in decision-making, but also concerns that existing biases in data may be amplified by, not carefully built, data-driven tools.

The goal is to build a model with high accuracy, but at the same time prevent any type of discrimination, either direct because it uses sensitive data, or indirect because the prediction is disproportionally negatively impacted in those individuals, although no sensitive features have been used. New criteria have been defined to achieve this, namely disparate treatment and disparate impact (Barocas and Selbst 2016; Zafar et al. 2017), and in recent works, optimal trees have been extended to include them (Aghaei et al. 2019). The extension of the models in Sect. 3.1 to address this issue is straightforward. To illustrate this, and again wlog, we consider a classification problem. Suppose that we have a group $\mathcal{S} \subset \{1, \ldots, N\}$ of individuals to be protected against discrimination by Problem (1). We may impose that the average probability of being assigned to class k for individuals in $\mathcal{S}$ does not differ much from the average in the whole training sample $\mathcal{I}$. This can be model through the following constraint:

$$\left| \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \sum_{t \in \tau_L} P_{it}(\boldsymbol{a}, \boldsymbol{\mu}) C_{kt} - \frac{1}{N} \sum_{i=1}^{N} \sum_{t \in \tau_L} P_{it}(\boldsymbol{a}, \boldsymbol{\mu}) C_{kt} \right| \leq \varepsilon,$$

for $\varepsilon > 0$ sufficiently small.

We end the section with some considerations on other types of data, apart from continuous and binary, available to construct tree models that show a good trade-off between accuracy and sparsity. When building classification and regression models, there may be characteristics which are recorded as, for example, time-series data (Barrow and Crone 2016; Carrizosa et al. 2013; Saha et al. 2020), spatial data (Georganos et al. 2019), functional data (Balakrishnan and Madigan 2006; Möller et al. 2016; Pospisil and Lee 2019; Rahman et al. 2019), text data (Martens and Provost

2014; Ramon et al. 2020), or network data (Óskarsdóttir et al. 2020), which are not captured appropriately by standard implementations of these models. This calls for new mathematical optimization formulations and/or numerical solution approaches to address these complexities adequately. The changes may stem from the functions we use to measure accuracy or sparsity. The typical losses such as the mean squared error or the expected misclassification cost may not be suitable to measure the accuracy for more complex response variables. In terms of sparsity, take, for instance, the case of time-series data, where we have an observation for each time period in the series, the response for this observation is the measurement in that time period and the features are the measurements in previous time periods, as in Benítez-Peña et al. (2020b) for the short-term predictions of the evolution of COVID-19. In this way, we have that individuals are characterized by p lags, but possibly other predictor variables. In addition to the ones described in Sect. 3, one may wish other types of sparsity, ensuring that only recent lags are used, or that as few as possible series are used in multivariate time-series (Tuncel and Baydogan 2018), or even more sophisticated versions of sparsity for hierarchical ones (Athanasopoulos et al. 2017; Karmy and Maldonado 2019; Wickramasuriya et al. 2019). For classification and regression problems with functional data, i.e., functions $x_i : [0, 1] \to \mathbb{R}$, one can easily adapt the models in Sect. 3.1 by replacing the definition of the probabilities $p_{it}(\boldsymbol{a}_{\cdot t}, \mu_t)$ at branch node t by, for instance:

$$F\left(\gamma_{1t}\int_0^{c_1} x_i(s)\,ds + \gamma_{2t}\int_{c_1}^{c_2} x_i(s)\,ds + \ldots \gamma_{rt}\int_{c_{r-1}}^{1} x_i(s)\,ds\right),$$

where the thresholds $c_1, c_2, \ldots, c_{r-1}$ and the weights $\gamma_{1t}, \gamma_{2t}, \ldots, \gamma_{rt}$ are decision variables. See, for instance, (Blanquero et al. 2019, 2020) for a related approach in Support Vector Machines. The examples above show that new forms of losses and/or sparsity can be incorporated in both the Continuous Optimization and the MILO paradigms, by making changes to the objective function, but new decisions as well as new constraints may be required, yielding challenging Mixed-Integer Nonlinear Optimization formulations.

5 Conclusions

The impressive advances in hardware and software in the last decades have allowed the development of more powerful versions of classification and regression trees than classic ones. In this paper, we have reviewed recent Continuous Optimization and Mixed-Integer Linear Optimization formulations to build optimal classification and regression trees that trade off accuracy and sparsity, the latter understood as a proxy for interpretability. Contrary to standard classification and regression trees built in a greedy heuristic manner, formulating the design of the tree model as an optimization problem allows the inclusion, either as hard or soft constraints, of other important criteria. We have illustrated this flexibility for an important social criterion, the fairness of the model, which aims to avoid predictions that discriminate

against race, or other sensitive data, and/or ensures an acceptable accuracy performance for groups at risk. We have also shown how optimization provides in a natural way counterfactual explanations for individuals, enhancing the local explainability of tree models. In the future, we foresee new optimization models which will be needed to tailor optimal trees to complex data arising in decision-making, yielding large-scale global optimization problems, usually with integer variables, and sophisticated numerical optimization strategies are to be devised to address these challenging problems.

Acknowledgements This research has been financed in part by research projects EC H2020 MSCA RISE NeEDS (Grant agreement ID: 822214), FQM-329 and P18-FR-2369 (Junta de Andalucía), and PID2019-110886RB-I00 (Ministerio de Ciencia, Innovación y Universidades, Spain). This support is gratefully acknowledged.

References

Aghaei S, Azizi MJ, Vayanos P (2019) Learning optimal and fair decision trees for non-discriminative decision-making. Proc AAAI Conf Artif Intell 33:1418–1426

Aghaei S, Gomez A, Vayanos P (2020) Learning optimal classification trees: strong max-flow formulations. arXiv:2002.09142

Aglin G, Nijssen S, Schaus P (2020) Learning optimal decision trees using caching branch-and-bound search. In: Thirty-Fourth AAAI Conference on Artificial Intelligence

Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows: theory, algorithms, and applications. Prentice Hall, New Jersey

Altmann A, Toloşi L, Sander O, Lengauer T (2010) Permutation importance: a corrected feature importance measure. Bioinformatics 26(10):1340–1347

Aouad A, Elmachtoub AN, Ferreira KJ, McNellis R (2019) Market segmentation trees. arXiv:1906.01174

Apsemidis A, Psarakis S, Moguerza JM (2020) A review of machine learning kernel methods in statistical process monitoring. Comput Ind Eng 142:106376

Athanasopoulos G, Hyndman RJ, Kourentzes N, Petropoulos F (2017) Forecasting with temporal hierarchies. Eur J Oper Res 262(1):60–74

Baesens B, Setiono R, Mues C, Vanthienen J (2003) Using neural network rule extraction and decision tables for credit-risk evaluation. Manage Sci 49(3):312–329

Balakrishnan S, Madigan D (2006) Decision trees for functional variables. In: Sixth international conference on data mining (ICDM'06), pp 798–802

Barocas S, Selbst AD (2016) Big data's disparate impact. California Law Rev 104:671

Barredo Arrieta A, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, García G, Gil-López S, Molina D, Benjamins R, Chatila R, Herrera F (2020) Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 58:82–115

Barros RC , Basgalupp MP, De Carvalho ACPLF, Freitas AA (2011) A survey of evolutionary algorithms for decision-tree induction. IEEE Trans Syst Man Cybern Part C (Appl Rev) 42(3):291–312

Barrow DK, Crone SF (2016) A comparison of Adaboost algorithms for time series forecast combination. Int J Forecast 32(4):1103–1119

Bénard C, Biau G, Da Veiga S, Scornet E (2019) SIRUS: making random forests interpretable. arXiv: 1908.06852

Bénard C, Biau G, Da Veiga S, Scornet E (2020) Interpretable random forests via rule extraction. arXiv: 2004.14841

Benítez-Peña S, Bogetoft P, Romero Morales D (2020a) Feature selection in data envelopment analysis: a mathematical optimization approach. Omega 96:102068

Benítez-Peña S, Carrizosa E, Guerrero V, Jiménez-Gamero MD, Martín-Barragán B, Molero-Río C, Ramírez-Cobo P, Romero Morales D, Sillero-Denamiel MR (2020b) On sparse ensemble methods: an application to short-term predictions of the evolution of covid-19. Technical report, IMUS, Sevilla, Spain. https://www.researchgate.net/publication/341608874_On_Sparse_Ensemble_Methods_An_Application_to_Short-Term_Predictions_of_the_Evolution_of_COVID-19

Bennett KP (1992) Decision tree construction via linear programming. In: Computer Sciences Department, University of Wisconsin, Center for Parallel Optimization

Bennett KP, Blue J (1996) Optimal decision trees. In: Rensselaer Polytechnic Institute Math Report, p 214

Bennett KP, Mangasarian OL (1992) Robust linear programming discrimination of two linearly inseparable sets. Optim Methods Softw 1:23–24

Bertsimas D, Dunn J (2017) Optimal classification trees. Mach Learn 106(7):1039–1082

Bertsimas D, Dunn J, Mundru N (2019) Optimal prescriptive trees. INFORMS J Optim 1(2):164–183

Bertsimas D, O'Hair A, Relyea S, Silberholz J (2016) An analytics approach to designing combination chemotherapy regimens for cancer. Manage Sci 62(5):1511–1531

Bertsimas D, Shioda R (2007) Classification and regression via integer optimization. Oper Res 55(2):252–271

Biau G, Devroye L, Lugosi G (2008) Consistency of random forests and other averaging classifiers. J Mach Learn Res 9:2015–2033

Biau G, Scornet E (2016) A random forest guided tour. TEST 25(2):197–227

Birbil SI, Edali M, Yüceoğlu B (2020) Rule covering for interpretation and boosting. arXiv:2007.06379

Bixby RE (2012) A brief history of linear and mixed-integer programming computation. Documenta Math 2012:107–121

Blake CL, Merz CJ (1998) UCI repository of machine learning databases. http://www.ics.uci.edu/~mlearn/MLRepository.html, University of California, Irvine, Department of Information and Computer Sciences

Blanquero R, Carrizosa E, Jiménez-Cordero A, Martín-Barragán B (2019) Functional-bandwidth kernel for support vector machine with functional data: an alternating optimization algorithm. Eur J Oper Res 275(1):195–207

Blanquero R, Carrizosa E, Jiménez-Cordero A, Martín-Barragán B (2020) Selection of time instants and intervals with support vector regression for multivariate functional data. Comput Oper Res 123:105050

Blanquero R, Carrizosa E, Molero-Río C, Romero Morales D (2021) Optimal Randomized classification trees. Forthcoming Compu Oper Res. https://doi.org/10.1016/j.cor.2021.105281

Blanquero R, Carrizosa E, Molero-Río C, Romero Morales D (2020a) On sparse optimal regression trees. In: Technical report, IMUS, Sevilla, Spain. https://www.researchgate.net/publication/326901224_Optimal_Randomized_Classification_Trees

Blanquero R, Carrizosa E, Molero-Río C, Romero Morales D (2020b) Sparsity in optimal randomized classification trees. Eur J Oper Res 284(1):255 – 272

Botari T, Hvilshøj F, Izbicki R, de Carvalho ACPLF (2020) MeLIME: Meaningful local explanation for machine learning models. arXiv:2009.05818

Bottou L, Curtis F, Nocedal J (2018) Optimization methods for large-scale machine learning. SIAM Rev 60(2):223–311

Breiman L (2001) Random forests. Mach Learn 45(1):5–32

Breiman L, Friedmann JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth, Belmont

Brodley CE, Utgoff PE (1995) Multivariate decision trees. Mach Learn 19(1):45–77

Carrizosa E, Galvis Restrepo M, Romero Morales D (2019) On clustering categories of categorical predictors in generalized linear models. Technical report, Copenhagen Business School, Denmark. https://www.researchgate.net/publication/349179679_On_Clustering_Categories_of_Categorical_Predictors_in_Generalized_Linear_Models

Carrizosa E, Guerrero V, Hardt D, Romero Morales D (2018a) On building online visualization maps for news data streams by means of mathematical optimization. Big Data 6(2):139–158
Carrizosa E, Guerrero V, Romero Morales D (2018b) Visualizing data as objects by DC (difference of convex) optimization. Math Program Ser B 169:119–140
Carrizosa E, Guerrero V, Romero Morales D, Satorra A (2020a) Enhancing interpretability in factor analysis by means of mathematical optimization. Multivariate Behav Res 55(5):748–762
Carrizosa E, Kurishchenko K, Marin A, Romero Morales D (2020b) Interpreting clusters by prototype optimization. Technical report, Copenhagen Business School, Denmark. https://www.researchgate.net/publication/349287282_Interpreting_Clusters_via_Prototype_Optimization
Carrizosa E, Mortensen LH, Romero Morales D, Sillero-Denamiel MR (2020c) On linear regression models with hierarchical categorical variables. Technical report, IMUS, Sevilla, Spain. https://www.researchgate.net/publication/341042405_On_linear_regression_models_with_hierarchical_categorical_variables
Carrizosa E, Nogales-Gómez A, Romero Morales D (2017) Clustering categories in support vector machines. Omega 66:28–37
Carrizosa E, Olivares-Nadal AV, Ramírez-Cobo P (2013) Time series interpolation via global optimization of moments fitting. Eur J Oper Res 230(1):97–112
Carrizosa E, Romero Morales D (2013) Supervised classification and mathematical optimization. Comput Oper Res 40(1):150–165
Casalicchio G, Molnar C, Bischl B (2019) Visualizing the feature importance for black box models. In: Berlingerio M, Bonchi F, Gärtner T, Hurley N, Ifrim G (eds) Machine Learning and Knowledge Discovery in Databases, pp 655–670, Cham. Springer International Publishing
Cerquitelli T, Quercia D, Pasquale F (2017) Transparent data mining for Big and small data. Springer, Berlin
Chen D, Fraiberger SP, Moakler R, Provost F (2017) Enhancing transparency and control when drawing data-driven inferences about individuals. Big Data 5(3):197–212
Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 785–794
Cohen S, Dror G, Ruppin E (2007) Feature selection via coalitional game theory. Neural Comput 19(7):1939–1961
Cui Z, Chen W, He Y, Chen Y (2015) Optimal action extraction for random forests and boosted trees. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 179–188
Dash S, Günlük O, Wei D (2018) Boolean decision rules via column generation. In: Advances in neural information processing systems, pp 4655–4665
Demiriz A, Bennett KP, Shawe-Taylor J (2002) Linear programming boosting via column generation. Mach Learn 46:225–254
Demirović E, Lukina A, Hebrard E, Chan J, Bailey J, Leckie C, Ramamohanarao K, Stuckey PJ (2020) MurTree: optimal classification trees via dynamic programming and search. arXiv:2007.12652
Demirović E, Stuckey PJ (2020) Optimal decision trees for nonlinear metrics. arXiv:2009.06921
Deng H, Runger G (2012) Feature selection via regularized trees. In: 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 Recogn 46(12):3483–3489
Denil M, Matheson D, Freitas N (2013) Consistency of online random forests. In: International Conference on Machine Learning, pp 1256–1264
Dougherty J, Kohavi R, Sahami M (1995) Supervised and unsupervised discretization of continuous features. In: Machine Learning Proceedings 1995, pp 194–202, Elsevier
Duarte Silva AP (2017) Optimization approaches to supervised classification. Eur J Oper Res 261(2):772–788
Dunn J (2018) Optimal trees for prediction and prescription. In: PhD thesis, Massachusetts Institute of Technology
Esteve M, Aparicio J, Rabasa A, Rodriguez-Sala JJ (2020) Efficiency analysis trees: a new methodology for estimating production frontiers through decision trees. Expert Syst Appl 162:113783
European Commission (2020) White Paper on Artificial Intelligence: a European approach to excellence and trust. https://ec.europa.eu/info/sites/info/files/commission-white-paper-artificial-intelligence-feb2020_en.pdf

Fang X, Liu Sheng OR, Goes P (2013) When is the right time to refresh knowledge discovered from data? Oper Res 61(1):32–44
Fawagreh K, Medhat Gaber M, Elyan E (2014) Random forests: from early developments to recent advancements. Syst Sci Control Eng 2(1):602–609
Fayyad UM, Irani KB (1992) The attribute selection problem in decision tree generation. In: AAAI, pp 104–110
Fernández-Delgado M, Cernadas E, Barro S, Amorim D (2014) Do we need hundreds of classifiers to solve real world classification problems? J Mach Learn Res 15:3133–3181
Fernández RR, Martín de Diego I, Aceña V, Fernández-Isabel A, Moguerza JM (2020) Random forest explainability using counterfactual sets. Inf Fusion 63:196–207
Firat M, Crognier G, Gabor AF, Hurkens CAJ, Zhang Y (2020) Column generation based heuristic for learning classification trees. Comput Oper Res 116:104866
Fountoulakis K, Gondzio J (2016) A second-order method for strongly convex ℓ_1-regularization problems. Math Program 156(1):189–219
Freitas AA (2014) Comprehensible classification models: a position paper. ACM SIGKDD Explor Newsl 15(1):1–10
Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139
Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Stat 29(5):1189–1232
Friedman JH (2002) Stochastic gradient boosting. Comput Stat Data Anal 38(4):367–378
Fu Z, Golden BL, Lele S, Raghavan S, Wasil EA (2003) A genetic algorithm-based approach for building accurate decision trees. INFORMS J Comput 15(1):3–22
Gambella C, Ghaddar B, Naoum-Sawaya J (2020) Optimization models for machine learning: a survey. Eur J of Oper Res 290(3):807–828
Genuer R, Poggi J-M, Tuleau-Malot C, Villa-Vialaneix N (2017) Random forests for big data. Big Data Res 9:28–46
Georganos S, Grippa T, Gadiaga AN, Linard C, Lennert M, Vanhuysse S, Mboga N, Wolff E, Kalogirou S (2019) Geographical random forests: a spatial extension of the random forest algorithm to address spatial heterogeneity in remote sensing and population modelling. Geocarto International 36(2):121–136
Gevrey M, Dimopoulos I, Lek S (2003) Review and comparison of methods to study the contribution of variables in artificial neural network models. Ecol Model 160(3):249–264
González S, García S, Del Ser J, Rokach L, Herrera F (2020) A practical tutorial on bagging and boosting based ensembles for machine learning: algorithms, software tools, performance study, practical perspectives and opportunities. Inf Fusion 64:205–237
Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Hoboken
Goodman B, Flaxman S (2017) European Union regulations on algorithmic decision-making and a "right to explanation". AI Mag 38(3):50–57
Grubinger T, Zeileis A, Pfeiffer K-P (2014) evtree: evolutionary learning of globally optimal classification and regression trees in R. J Stat Softw Articles 61(1):1–29
Günlük O, Kalagnanam J, Menickelly M, Scheinberg K (2019) Optimal decision trees for categorical data via integer programming. arXiv:1612.03225v3
Gunning D, Aha DW (2019) DARPA's explainable artificial intelligence program. AI Mag 40(2):44–58
Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
Hastie T, Rosset S, Zhu J, Zou H (2009) Multi-class AdaBoost. Stat Interface 2(3):349–360
Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, 2nd edn. Springer, New York
Hastie T, Tibshirani R, Wainwright M (2015) Statistical learning with sparsity: the lasso and generalizations. CRC Press, Hoboken
Hofman JM, Sharma A, Watts DJ (2017) Prediction and explanation in social systems. Science 355(6324):486–488
Holter S, Gomez O, Bertini E (2018) FICO Explainable Machine Learning Challenge. https://community.fico.com/s/explainable-machine-learning-challenge
Höppner S, Stripling E, Baesens B, vanden Broucke S, Verdonck T (2020) Profit driven decision trees for churn prediction. Eur J Oper Res 284(3):920–933

Hu X, Rudin C, Seltzer M (2019) Optimal sparse decision trees. Adv Neural Inf Process Syst 32:7265–7273
Hyafil L, Rivest RL (1976) Constructing optimal binary decision trees is NP-complete. Inf Process Lett 5(1):15–17
Iosifidis V, Ntoutsi E (2019) Adafair: cumulative fairness adaptive boosting. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM '19, pp 781–790, New York, NY, USA, Association for Computing Machinery
Irsoy O, Yıldız OT, Alpaydın E (2012) Soft decision trees. In: Proceedings of the 21st international conference on pattern recognition (ICPR2012), pp 1819–1822
Izza Y, Ignatiev A, Marques-Silva J (2020) On explaining decision trees. arXiv:2010.11034
Jakaitiene A, Sangiovanni M, Guarracino MR, Pardalos PM (2016) Multidimensional scaling for genomic data, pp 129–139. Springer International Publishing, Cham
Jung J, Concannon C, Shroff R, Goel S, Goldstein DG (2017) Creating simple rules for complex decisions. Harvard Business Rev 2017:1
Jung J, Concannon C, Shroff R, Goel S, Goldstein DG (2020) Simple rules to guide expert classifications. J R Stat Soc Ser A (Stat Soc) 183(3):771–800
Kaloudi N, Li J (2020) The AI-based cyber threat landscape: a survey. ACM Comput Surv (CSUR) 53(1):1–34
Kao H-P, Tang K (2014) Cost-sensitive decision tree induction with label-dependent late constraints. INFORMS J Comput 26(2):238–252
Karimi A-H, Barthe G, Schölkopf B, Valera I (2020) A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. arXiv:2010.04050
Karmy JP, Maldonado S (2019) Hierarchical time series forecasting via support vector regression in the European travel retail industry. Expert Syst Appl 137:59–73
Kass GV (1980) An exploratory technique for investigating large quantities of categorical data. J R Stat Soc Ser C (Appl Stat) 29(2):119–127
Katuwal R, Suganthan PN, Zhang L (2020) Heterogeneous oblique random forest. Pattern Recogn 99:107078
Khalil EB, Le Bodic P, Song L, Nemhauser GL, Dilkina BN (2016) Learning to branch in mixed integer programming. In: AAAI, pp 724–731
Kim H, Loh W-Y (2001) Classification trees with unbiased multiway splits. J Am Stat Assoc 96(454):589–604
Kleinberg J, Lakkaraju H, Leskovec J, Ludwig J, Mullainathan S (2018) Human decisions and machine predictions. Q J Econ 133(1):237–293
Koenker R, Hallock KF (2001) Quantile regression. J Econ Perspect 15(4):143–156
Kriegler B, Berk R (2010) Small area estimation of the homeless in Los Angeles: an application of cost-sensitive stochastic gradient boosting. Ann Appl Stat 2010:1234–1255
Li X-B, Sweigart JR, Teng JTC, Donohue JM, Thombs LA, Wang SM (2003) Multivariate decision trees using linear discriminants and tabu search. IEEE Trans Syst Man Cybern-Part A Syst Hum 33(2):194–205
Liberti L (2020) Distance geometry and data science. TOP 28:271–339
Lin J, Zhong C, Hu D, Rudin C, Seltzer M (2020) Generalized and scalable optimal sparse decision trees. arXiv:2006.08690
Liu H, Hussain F, Tan C, Dash M (2002) Discretization: an enabling technique. Data Min Knowl Disc 6(4):393–423
Lodi A, Zarpellon G (2017) On learning and branching: a survey. TOP 25(2):207–236
Loh W-Y (2014) Fifty years of classification and regression trees. Int Stat Rev 82(3):329–348
Loh W-Y, Shih Y-S (1997) Split selection methods for classification trees. Stat Sin 7(4):815–840
Louppe G, Wehenkel L, Sutera A, Geurts P (2013) Understanding variable importances in forests of randomized trees. Adv Neural Inf Process Syst 2013:431–439
Lucic A, Oosterhuis H, Haned H, de Rijke M (2020) FOCUS: Flexible optimizable counterfactual explanations for tree ensembles. arXiv:1911.12199
Lundberg SM, Erion G, Chen H, DeGrave A, Prutkin JM, Nair B, Katz R, Himmelfarb J, Bansal N, Lee S-I (2020) From local explanations to global understanding with explainable AI for trees. Nature Mach Intell 2(1):2522–5839
Lundberg SM, Erion G, Lee S-I (2018) Consistent individualized feature attribution for tree ensembles. arXiv:1802.03888

Lundberg SM, Lee S-I (2017) A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 2017:4765–4774
Martens D, Baesens B, Gestel TV, Vanthienen J (2007) Comprehensible credit scoring models using rule extraction from support vector machines. Eur J Oper Res 183(3):1466–1476
Martens D, Provost F (2014) Explaining data-driven document classifications. MIS Q 38(1):73–99
Martínez Torres J, Iglesias Comesaña C, García-Nieto PJ (2019) Machine learning techniques applied to cybersecurity. Int J Mach Learn Cybern 10(10):2823–2836
Meinshausen N (2010) Node harvest. Ann Appl Stat 4(4):2049–2072
Menze BH, Kelm BM, Splitthoff DN, Koethe U, Hamprecht FA (2011) On oblique random forests. In: Gunopulos D, Hofmann T, Malerba D, Vazirgiannis M (eds) Machine Learning and Knowledge Discovery in Databases, pp 453–469
Miller T (2019) Explanation in artificial intelligence: insights from the social sciences. Artif Intell 267:1–38
Miron M, Tolan S, Gómez E, Castillo C (2020) Addressing multiple metrics of group fairness in data-driven decision making. arXiv:2003.04794
Mišić VV (2020) Optimization of Tree Ensembles. Oper Res 68(5):1605–1624
Möller A, Tutz G, Gertheiss J (2016) Random forests for functional covariates. J Chemom 30(12):715–725
Molnar C, Casalicchio G, Bischl B (2018) iml: an R package for interpretable machine learning. J Open Sourc Softw 3(26):786
Molnar C, Casalicchio G, Bischl B (2020) Interpretable machine learning – a brief history, state-of-the-art and challenges. arXiv:2010.09337
Mothilal RK, Sharma A, Tan C (2020) Explaining machine learning classifiers through diverse counterfactual explanations. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp 607–617
Murthy SK, Kasif S, Salzberg S (1994) A system for induction of oblique decision trees. J Artif Intell Res 2:1–32
Narodytska N, Ignatiev A, Pereira F, Marques-Silva J (2018) Learning Optimal Decision Trees with SAT. In: Proceedings of the Twenty-Seventh international joint conference on artificial intelligence (IJCAI-18), pp 1362–1368
Nijssen S, Fromont E (2010) Optimal constraint-based decision tree induction from itemset lattices. Data Min Knowl Disc 21(1):9–51
Norouzi M, Collins M, Johnson MA, Fleet DJ, Kohli P (2015) Efficient non-greedy optimization of decision trees. Adv Neural Inf Process Syst 2015:1729–1737
Orsenigo C, Vercellis C (2003) Multivariate classification trees based on minimum features discrete support vector machines. IMA J Manag Math 14(3):221–234
Óskarsdóttir M, Ahmed W, Antonio K, Baesens B, Dendievel R, Donas T, Reynkens T (2020) Social network analytics for supervised fraud detection in insurance. arXiv:2009.08313
Palagi L (2019) Global optimization issues in deep network regression: an overview. J Global Optim 73(2):239–277
Pangilinan JM, Janssens GK (2011) Pareto-optimality of oblique decision trees from evolutionary algorithms. J Global Optim 51(2):301–311
Pardalos PM, Boginski VL, Vazacopoulos A (eds) (2007) Data mining in biomedicine. Springer optimization and its applications, Springer
Pfetsch ME, Pokutta S (2020) IPBoost—non-convex boosting via integer programming. arxiv:2002.04679
Piccialli V, Sciandrone M (2018) Nonlinear optimization and support vector machines. 4OR 16(2):111–149
Pospisil T, Lee AB (2019) (f)RFCDE: Random forests for conditional density estimation and functional data. arXiv:1906.07177
Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann, San Mateo
Rahman R, Dhruba SR, Ghosh S, Pal R (2019) Functional random forest with applications in dose-response predictions. Sci Rep 9(1):1–14
Ramon Y, Martens D, Provost F, Evgeniou T (2020) A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C. Adv Data Anal Classif 2020:5

Ribeiro MT, Singh S, Guestrin C (2016) Why Should I Trust You?: explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144
Ridgeway G (2013) The pitfalls of prediction. Natl Inst Justice J 271:34–40
Romei A, Ruggieri S (2014) A multidisciplinary survey on discrimination analysis. Knowl Eng Rev 29(5):582–638
Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Mach Intell 1(5):206–215
Rudin C, Ertekin Ş (2018) Learning customized and optimized lists of rules with mathematical programming. Math Program Comput 10(4):659–702
Ruggieri S (2019) Complete search for feature selection in decision trees. J Mach Learn Res 20(104):1–34
Saha A, Basu S, Datta A (2020) Random forests for dependent data. arXiv:2007.15421
Savickỳ P, Klaschka J, Antoch J (2000) Optimal classification trees. In: COMPSTAT, pp 427–432, Springer
Scornet E (2016) On the asymptotics of random forests. J Multivariate Anal 146:72–83
Scornet E, Biau G, Vert J-P (2015) Consistency of random forests. Ann Stat 43(4):1716–1741
Sherali HD, Hobeika AG, Jeenanunta C (2009) An optimal constrained pruning strategy for decision trees. INFORMS J Comput 21(1):49–61
Sokol K, Flach PA (2019) Counterfactual explanations of machine learning predictions: opportunities and challenges for AI safety. In: SafeAI @ AAAI
Souillard-Mandar W, Davis R, Rudin C, Au R, Libon DJ, Swenson R, Price CC, Lamar M, Penney DL (2016) Learning classification models of cognitive conditions from subtle behaviors in the digital clock drawing test. Mach Learn 102(3):393–441
Street WN (2005) Oblique multicategory decision trees using nonlinear programming. INFORMS J Comput 17(1):25–31
Strobl C, Boulesteix A-L, Kneib T, Augustin T, Zeileis A (2008) Conditional variable importance for random forests. BMC Bioinf 9(1):307
Su X, Wang M, Fan J (2004) Maximum likelihood regression trees. J Comput Graph Stat 13(3):586–598
Therneau T, Atkinson B, Ripley B (2015) rpart: recursive partitioning and regression trees, 2015. R package version 4.1-10
Truong A (2009) Fast growing and interpretable oblique trees via logistic regression models. In: Ph.D. thesis, University of Oxford, UK
Tuncel KS, Baydogan MG (2018) Autoregressive forests for multivariate time series modeling. Pattern Recogn 73:202–215
Turney PD (1995) Cost-sensitive classification: empirical evaluation of a hybrid genetic decision tree induction algorithm. J Artif Intell Res 2:369–409
Ustun B, Rudin C (2016) Supersparse linear integer models for optimized medical scoring systems. Mach Learn 102(3):349–391
Van Vlasselaer V, Eliassi-Rad T, Akoglu L, Snoeck M, Baesens B (2017) GOTCHA! Network-based fraud detection for social security fraud. Manage Sci 63(9):3090–3110
Verhaeghe H, Nijssen S, Pesant G, Quimper C-G, Schaus P (2019) Learning optimal decision trees using constraint programming. In: The 25th International Conference on Principles and Practice of Constraint Programming (CP2019)
Verma S, Dickerson J, Hines K (2020) Counterfactual explanations for machine learning: a review. arXiv:2010.10596
Verwer S, Zhang Y (2017) Learning decision trees with flexible constraints and objectives using integer optimization. In Salvagnin D, Lombardi M (eds) Integration of AI and OR techniques in constraint programming: 14th International Conference, CPAIOR 2017, Padua, Italy. Proceedings, pp 94–103
Verwer S, Zhang Y, Ye QC (2017) Auction optimization using regression trees and linear models as integer programs. Artif Intell 244:368–395
Verwer S, Zhang Y, Ye QC (2019) Learning optimal classification trees using a binary linear program formulation. Proc AAAI Conf Artif Intel 33:1625–1632
Vidal T, Pacheco T, Schiffer M (2020) Born-again tree ensembles. arXiv:2003.11132
Visani G, Bagli E, Chesani F, Poluzzi A, Capuzzo D (2020) Statistical stability indices for LIME: obtaining reliable explanations for machine learning models. arXiv:2001.11757
Wachter S, Mittelstadt B, Russell C (2017) Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harvard J Law Technol 31:841–887

Wager S, Athey S (2018) Estimation and inference of heterogeneous treatment effects using random forests. J Am Stat Assoc 113(523):1228–1242
Weston J, Elisseeff A, Schölkopf B, Tipping M (2003) Use of the zero norm with linear models and kernel methods. J Mach Learn Res 3:1439–1461
Wickramarachchi DC, Robertson BL, Reale M, Price CJ, Brown J (2016) HHCART: an oblique decision tree. Comput Stat Data Anal 96:12–23
Wickramasuriya SL, Athanasopoulos G, Hyndman RJ (2019) Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. J Am Stat Assoc 114(526):804–819
Yang L, Liu S, Tsoka S, Papageorgiou LG (2017) A regression tree approach using mathematical programming. Expert Syst Appl 78:347–357
Yang Y, Garcia Morillo I, Hospedales TM (2018) Deep neural decision trees. arXiv:1806.06988
Yu J, Ignatiev A, Stuckey PJ, Le Bodic P (2020) Computing Optimal Decision Sets with SAT. arXiv:2007.15140
Zafar MB, Valera I, Gomez Rodriguez M, Gummadi KP (2017) Fairness constraints: mechanisms for fair classification. In: Artificial Intelligence and Statistics, pp 962–970, PMLR
Zantedeschi V, Kusner MJ, Niculae V (2020) Learning binary trees via sparse relaxation. arXiv:2010.04627
Zeng J, Ustun B, Rudin C (2017) Interpretable classification models for recidivism prediction. J R Stat Soc Ser A 180(3):689–722
Zhang Y, Song K, Sun Y, Tan S, Udell M (2019) Why should you trust my explanation? Understanding Uncertainty in LIME Explanations. arXiv:1904.12991
Zhu H, Murali P, Phan DT, Nguyen LM, Kalagnanam JR (2020) A scalable MIP-based method for learning optimal multivariate decision trees. Adv Neural Inf Process Syst 2020:33