Margin Optimal Classification Trees

Federico D'Onofrio[a], Giorgio Grani[b], Marta Monaci[a], Laura Palagi[a]

[a]*Department of Computer, Control and Management Engineering Antonio Ruberti (DIAG),*
Sapienza University of Rome, Rome, Italy
[b]*Department of Statistical Sciences, Sapienza University of Rome, Rome, Italy*

Abstract

In recent years there has been growing attention to interpretable machine learning models which can give explanatory insights on their behavior. Thanks to their interpretability, decision trees have been intensively studied for classification tasks, and due to the remarkable advances in mixed-integer programming (MIP), various approaches have been proposed to formulate the problem of training an Optimal Classification Tree (OCT) as a MIP model. We present a novel mixed-integer quadratic formulation for the OCT problem, which exploits the generalization capabilities of Support Vector Machines for binary classification. Our model, denoted as Margin Optimal Classification Tree (MARGOT), encompasses the use of maximum margin multivariate hyperplanes nested in a binary tree structure. To enhance the interpretability of our approach, we analyse two alternative versions of MARGOT, which include feature selection constraints inducing local sparsity of the hyperplanes. First, MARGOT has been tested on non-linearly separable synthetic datasets in 2-dimensional feature space to provide a graphical representation of the maximum margin approach. Finally, the proposed models have been tested on benchmark datasets from the UCI repository. The MARGOT formulation turns out to be easier to solve than other OCT approaches, and the generated tree better generalizes on new observations. The two interpretable versions are effective in selecting the most relevant features and maintaining good prediction quality.

Keywords:
Machine Learning; Optimal Classification Trees; Support Vector Machines; Mixed-Integer Programming

1. Introduction

1.1. Related Work

In recent years there has been a growing interest in interpretable Machine Learning (ML) models. Decision trees are among the most popular Supervised ML tools used for classification tasks. They are famous for being easy to manage, fast in computational requirements, and the final model is easily understandable from a human perspective as opposed to other ML methods that are seen as black boxes. Given a set of points and class labels, a classification tree method builds up a binary tree structure of a maximum predefined depth. Trees are composed of branch and leaf nodes, and the branch nodes apply a sequence of dichotomic rules, called *splitting rules*, to partition the training samples into disjoint subsets. Splitting rules route samples to the left or right child node, and they can be univariate or multivariate. In the multivariate case, splits are defined by oblique hyperplanes, while in the univariate one, they are restricted to only one feature. A value for the predicted class label is assigned to each leaf node according to some simple rule, for instance the most common label rule. The key advantage of tree methods lies in their interpretability. The process behind a decision tree is transparent, and the sequential tree structure mimics human decision-making. These properties can be crucial factors in many applications, ranging from business and criminal justice to healthcare and bioinformatics. Indeed, in these domains, it is of great interest to use explainable approaches able to help humans to understand the decisions made by the model and to identify a subset of the most prominent features that influence the classification outcome. To this aim, it is preferable to build and manage shallow trees with small depth; indeed, if allowed to grow large, decision trees lose their interpretability aspect.

It is well-known that constructing a binary decision tree in an optimal way is an *NP*-complete problem [25]. For this reason, traditional approaches for finding decision trees rely on heuristics. In general, they are based on a top-down greedy strategy for growing the tree by generating splits at each node, and, once the tree is built, a bottom-up pruning procedure is applied to handle the complexity of the tree, i.e. the number of splits. Breiman et al. [12] 1984 proposed a heuristic algorithm known as CART (Classification and Regression Trees), for learning univariate decision trees. Starting from the root node, each hyperplane split is generated by minimizing a local impurity function, e.g. the Gini impurity for classification tasks.

Other univariate approaches employing different impurity functions were later proposed by Quinlan [33, 34] in his ID3 and C4.5 algorithms. These heuristic procedures lead to fast computational time but may also generate tree models with poor generalization performances. In order to overcome these drawbacks, tree ensemble methods, such as Random Forests [11], TreeBoost [21] and XGBoost [18], have been proposed. These approaches combine together decision trees using some kind of randomness; however using multiple trees leads to a lack of interpretability of the final model.

Another way to improve prediction quality is to use multivariate decision trees which employ oblique hyperplane splits. This approach involves more features per split, thus producing smaller trees but at the expense of the computational cost. Several approaches for inducing multivariate trees have been proposed (see [30], [13], [31], [38]). For instance, OC1 [30] is a greedy algorithm that searches for the best hyperplane at each node by applying a randomized perturbation strategy, while [31] presents a heuristic procedure that at each step solves a variant of the Support Vector Machine problem, where the empirical error is discretized by counting the number of misclassified samples.

Recently, several papers have been devoted to global exact optimization approaches to find an Optimal Classification Tree (OCT) using mathematical programming tools and, in particular Mixed Integer Programming (MIP) models (see the recent surveys [22, 15] and references therein). Indeed, the significant improvement in the last thirty years of both algorithms for integer optimization and computer hardware has led to an incredible increase in the computational power of mixed-integer solvers, as shown in [5]. Thus, MIP approaches became viable in the definition of ML methods, being [4] the seminal paper inaugurating a new era in the use of mixed-integer based optimization to learn OCTs. Such approaches find the decision tree in its entirety through the resolution of a single optimization model, defining each branching rule with full knowledge of all the remaining ones. In [4], Bertsimas and Dunn proposed two Mixed-Integer Linear Programming (MILP) models to build optimal trees with a given maximum depth based on univariate and multivariate splits. Along these lines, Günlük et al. [23] proposed a MIP formulation for binary classification tasks by exploiting the structure of categorical features and modeling combinatorial decisions. Further, in order to circumvent the problem of the curse of dimensionality related to the MIP approaches, Verwer and Zhang [36] presented BinOCT, a binary linear programming model, where the size is independent of the training set dimension. In [2], Aghaei et al. proposed a flow-based MILP model for binary features with a stronger linear relaxation and, by exploiting the decomposable and combinatorial structure of the model, derived a Benders' decomposition method to deal with larger instances. In addition to expressing the combinatorial nature of the decisions involved in the process, the mixed-integer framework is suitable to handle global objectives and constraints to embed desirable properties such as fairness, sparsity, cost-sensitivity, robustness, as it has been addressed in [15], [35], [1], [2] and [7].

Alongside integer optimization, continuous optimization paradigms have also been investigated in the optimal trees context. In [9], Blanquero et al. proposed a nonlinear programming model for learning an optimal "randomized" classification tree with oblique splits, where at each node a random decision is made according to a soft rule, induced by a continuous cumulative density function. Later, in [8], they addressed global and local sparsity in the randomized optimal tree model by means of regularization terms based on polyhedral norms. In their randomized framework, a sample is not assigned to a class in a deterministic way but only with a given probability.

Following a different view point, approaches using a Support Vector Machine (SVM) (see [19], [37], [32]) for each split in the tree have been investigated. First, Bennett et al. [3] provided a primal continuous formulation with a nonconvex objective function and a dual convex quadratic model to train optimal trees where each decision rule is based on a modified SVM problem. Using the kernel function, the splitting rules may be nonlinear such as sigmoidal neural networks or radial basis function networks. Nevertheless, the resulting problems are computationally hard to solve. Recently, maximum margin classifiers of the SVM type have been proposed by Blanco et al. in [7] where a Mixed-Integer Non Linear formulation for the OCT problem is introduced. The aim is to build a robust tree classifier, where the splitting rules are based on the possibility of relabelling some samples as described in [6]. In particular, the method aims to maximize the minimum margin among all the margins of the hyperplane splits in the tree, and, at the same time, it detects observations to be relabeled to obtain a more accurate classifier. The model is formulated as a Mixed Integer Second Order Cone Optimization problem.

1.2. Our contribution

Our approach falls in the basic framework of using Support Vector Machines to define optimal classification trees with multivariate splits for binary classification tasks. In particular, our model employs maximum margin hyperplanes obtained by using a linear soft SVM paradigm in a nested binary tree structure. The maximum depth of the tree is fixed, as usual in OCT approaches.

The main contribution of this paper is a novel mixed integer quadratic programming (MIQP) formulation, denoted as Margin Optimal Classification Tree (MARGOT), for learning classification trees. Our formulation differs from others in the literature as we exploit the statistical learning properties of the ℓ_2-norm soft SVM formulation. In particular, the SVM quadratic convex function is retained and the collective measure of performance of the OCT is obtained as the sum of the objective functions of each single SVM-based problem over all the branch nodes of the tree. Differently from [7], we aim to maximize the sum of the margins of the hyperplane splits rather than to maximize the minimum of these margins. Further, exploiting both the SVMs properties and the binary classification setting allow us to drastically reduce the overall number of binary variables needed in our MIQP model. Specifically, we need to introduce as many binary variables as half the number of leaf nodes, which is much less than other OCT MIP approaches. We show both on synthetic data sets in a 2-dimensional feature space and on data-sets selected from the UCI Machine Learning repository [20], that MARGOT formulation requires less computational effort than other state-of-the-art MIP models for OCT and it can be solved to certified optimality on nearly all the considered problems with a limited computational time using off-the-shelf solvers. As a consequence of the maximum margin approach, our model produces OCTs with a higher out-of-sample accuracy.

As a second contribution, we aim to enhance the interpretability of the model proposed by reducing the number of features used. Indeed, due to the intrinsic statistical properties of SVMs, such model tends to use a large number of features in the first root node of the classification tree. We propose to consider two embedded models that simultaneously train the OCT and perform feature selection. Embedded models for feature selection in SVMs have been studied in several papers (see e.g. [26, 16, 29, 27, 28]). To control the sparsity of the oblique splits, we draw inspiration from the ℓ_1-norm SVM model proposed in [27] and we use additional binary variables and a budget on the number of features used. We consider two different modellings of the budget on the number of features: hard constraints and soft penalization, respectively implemented in HFS-MARGOT and SFS-MARGOT. Numerical testing on the UCI data-sets are reported showing that the hard version seems to be easier to solve to certified optimality in a reasonable CPU time, resulting in a more sparse solution too. For all the formulations, we propose a simple greedy heuristic to obtain a first incumbent which exploits the SVM structure.

The rest of the paper is organized as follows. In Section 2, a brief introduction about Optimal Multivariate Classification Trees, proposed in [4], and Support Vector Machines is provided. In Section 3, we introduce our approach and its formulation, denoted as MARGOT. In Section 4, we present the two interpretable versions of the model with hard and soft feature selection techniques to address the sparsity of the hyperplane splits. In section 5, we provide a heuristic to generate starting feasible solutions for the analysed MIP problems. Then, in section 6, we first evaluate MARGOT on 2-dimensional synthetic datasets and we report a graphical representation of the generated trees. Finally, computational experiments on benchmark datasets from UCI repository are presented for all the proposed models.

2. Preliminaries

2.1. Multivariate Optimal Classification Trees

In this section, we introduce in more detail multivariate optimal classification trees. Given a dataset $\{(x^i, y^i) \in \mathbb{R}^n \times \{1, \dots, K\},\ i \in \mathcal{I}\}$, and a maximum depth D, an optimal classification tree is made up by $2^{(D+1)} - 1$ nodes, divided in:

- *Branch nodes*, $\mathcal{T}_B = \{0, \dots, 2^D - 2\}$: a branch node applies a splitting rule on the feature space defined by a separating hyperplane $\mathcal{H}_t(x) := \{x : h_t(x) = 0\}$, where $h_t(x) = w_t^T x + b_t$ is the hyperplane function and $w_t \in \mathbb{R}^n$ and $b_t \in \mathbb{R}$. If $h_t(x^i) \geq 0$, sample i will follow the right branch of node t, otherwise it will follow the left one;
- *Leaf nodes*, $\mathcal{T}_L = \{2^D - 1, \dots, 2^{D+1} - 2\}$: leaf nodes act as collectors and the samples which end up in the same leaf are classified with the same class label.

The training phase aims at building a classification tree by finding coefficients w_t and b_t for each $t \in \mathcal{T}_B$ and by assigning class labels to the leaf nodes. According to the hierarchical tree structure, the feature space will be partitioned into disjoint regions, corresponding to each leaf node of the tree. The obtained tree is then used to classify out-of-sample data: every new sample will follow a unique path within the tree based on the splitting rules, ending up in a leaf node that will predict its class label.

In [4], Bertsimas and Dunn proposed a MILP optimization model for training multivariate OCTs, denoted as OCT-H, where in the objective function the misclassification error together with the number of features used at each split is minimized. In the optimization model, each sample is forced to end up in a single leaf, a class label for each leaf node is chosen according to the most common label rule and the classification error is computed according to the assignment of each sample to a leaf. *Routing constraints* enforce each sample to follow a unique path, while other constraints control the complexity of the tree by imposing pruning conditions and a minimum number of points accepted by each leaf.

2.2. A brief overview on Support Vector Machines

Given a binary classification instance $\{(x^i, y^i) \in \mathbb{R}^n \times \{-1, 1\},\ i \in \mathcal{I}\}$, the linear soft SVM classification problem defines a linear classifier as the function $f : \mathbb{R}^n \to \{-1, 1\}$,

$$f(x) = \operatorname{sgn}({w^*}^T x + b^*),$$

using the structural risk minimization principle (SRM) [19]. Coefficients $(w^*, b^*) \in \mathbb{R}^n \times \mathbb{R}$ identify a separating hyperplane which maximizes its margin, i.e. the distance from the closest data points. The tuple is found by solving the following convex quadratic problem:

$$\begin{aligned}
\text{(SVM)} \quad & \min_{w,b,\xi} \quad \frac{1}{2}\|w\|_2^2 + C \sum_{i \in \mathcal{I}} \xi_i && && (1)\\
& \text{s.t.} \quad y^i(w^T x^i + b) \geq 1 - \xi_i && \forall i \in \mathcal{I} && (2)\\
& \qquad\ \ \xi_i \geq 0 && \forall i \in \mathcal{I}, && (3)
\end{aligned}$$

where C is an hyperparameter that balances the two objectives: the maximization of the margin $2\|w\|_2^{-1}$ and the minimization of the misclassification cost. Variables ξ_i allow for violation of the *margin constraints* (2) and a sample i is misclassified when $\xi_i > 1$, while values $0 < \xi_i \leq 1$ correspond to correctly classified samples inside the margin. The further is a misclassified data point x^i from a feasible hyperplane, the greater will be the value of variable ξ_i (see e.g. Fig. 1). Thus, $\sum_{i \in \mathcal{I}} \xi_i$ is an upper bound on the number of samples misclassified by the hyperplane.

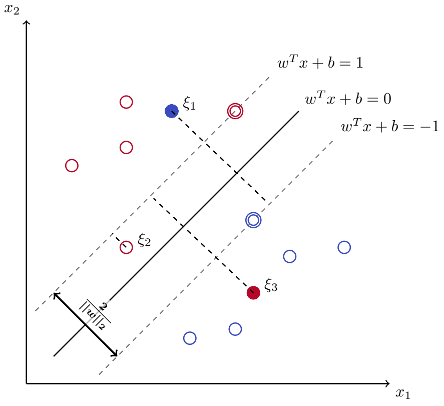


Figure 1: Example showing how errors are upper-bounded in the SVM approach by variables ξ_i: values $0 < \xi_i \leq 1$ correspond to samples that lie within the margin but are correctly classified; $\xi_i > 1$ are samples wrongly classified.

Although the objective function of the SVM problem is loosely convex, in [14] necessary and sufficient conditions are given for the support vector solution to be unique. In particular, with reference to the (1) where C is the same for all i, a necessary condition for the solution to be non-unique is that the negative and positive polarity support vectors are equal in number. Further, they proved that even when the solution is not unique, all

solutions share the same w. Thus, among the infinite separating hyperplanes, SVM selects the unique one that minimize the structural risk.
Minimizing the ℓ_2-norm of the vector w generates hyperplanes with the maximum margin, but it has little effect on the sparsity of vectors w, namely in reducing the number of components different from zero also denoted as the ℓ_0-norm, $\|w\|_0$. In this regard, it is known that using the ℓ_1-norm would be a better choice in that it is a better approximation of the ℓ_0-norm. In the literature, many papers adopted the SVM version where the ℓ_2-norm is replaced by the ℓ_1 one (see e.g. [10, 22]). In this case the problem turns to be an easier to solve linear programming problem, but generally at the expense of higher misclassification rates.

3. The Margin Optimal Classification Tree

In this section, we propose a novel MIQP model for constructing optimal classification trees which encompass the use of multivariate hyperplanes. The aim is to exploit the generalization capabilities of the soft SVM approach using maximum margin hyperplanes, thus the name Margin Optimal Classification Tree (MARGOT). For the sake of interpretability, we also analyse two alternative versions of MARGOT which reduces the number of features used at each split. These additional models are addressed in Section 4.
In order to present formally the MARGOT formulation, besides the sets of branch and leaf nodes, we use the following additional notation (see Figure 2). The set branch nodes $\mathcal{T}_B$ is partitioned in:

- $\mathcal{T}''_B$, the set of nodes in the last branching level;
- $\mathcal{T}'_B$, the set $\mathcal{T}_B \setminus \mathcal{T}''_B$.

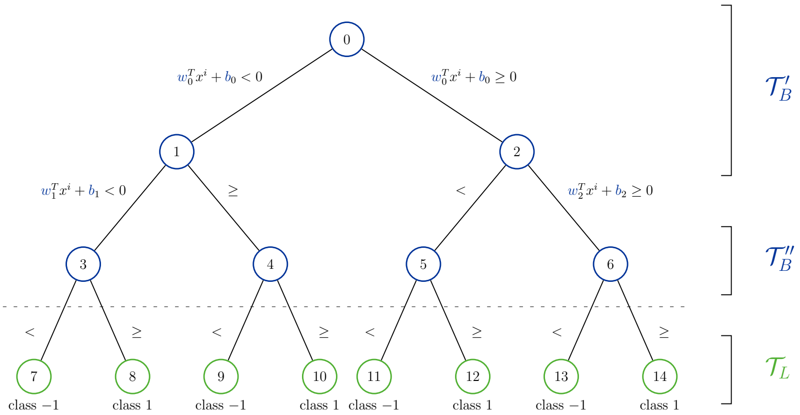


Figure 2: Representation of a tree model with depth $D = 3$ and of the sets $\mathcal{T}'_B, \mathcal{T}''_B, \mathcal{T}_L$.

We also define the following sets:

- $\mathcal{S}(t)$: the set of nodes of the subtree rooted at node $t \in \mathcal{T}_B$;
- $\mathcal{S}''(t) = \mathcal{S}(t) \cap \mathcal{T}''_B$: the subset of nodes of $\mathcal{S}(t)$ belonging to the last branching level $\mathcal{T}''_B$;
- $\mathcal{S}''_L(t)$ and $\mathcal{S}''_R(t)$: the set of nodes in $\mathcal{T}''_B$ under the left and right branch of node $t \in \mathcal{T}'_B$ such that:
    - $\mathcal{S}''(t) = \mathcal{S}''_L(t) \cup \mathcal{S}''_R(t)$
    - $\mathcal{S}''_L(t) \cap \mathcal{S}''_R(t) = \emptyset$.

In the optimization model we have to model the fact that the hyperplane at each node t needs to be trained on just a subset of samples. To this aim, let $\mathcal{I}_t \subseteq \mathcal{I}$ be the index set of samples routed to $t \in \mathcal{T}_B$. The definition of the hyperplane at each node t is obtained by means of an SVM-type problem. This means that, for each node $t \in \mathcal{T}_B$, we will have standard variables $(w_t, b_t, \xi_t) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{|\mathcal{I}_t|}$ which must satisfy the soft SVM margin constraints:

$$\begin{aligned} & y^i(w_t^T x^i + b_t) \geq 1 - \xi_{t,i} && \forall i \in \mathcal{I}_t \\ & \xi_{t,i} \geq 0 && \forall i \in \mathcal{I}_t. \end{aligned} \tag{4}$$

Samples (x^i, y^i), $i \in \mathcal{I}_t$, can be split among the right or left child node of t depending on the rules:

$$w_t^T x^i + b_t \geq 0 \;\text{ if }\; i \in \mathcal{I}_{R(t)} \;\text{ or }\; w_t^T x^i + b_t < 0 \;\text{ if }\; i \in \mathcal{I}_{L(t)}, \tag{5}$$

where sets $\mathcal{I}_{R(t)}$ and $\mathcal{I}_{L(t)}$ are the index sets of samples assigned to the right and left child nodes of t, respectively, thus $\mathcal{I}_{R(t)} \cup \mathcal{I}_{L(t)} = \mathcal{I}_t$ and $\mathcal{I}_{R(t)} \cap \mathcal{I}_{L(t)} = \emptyset$. A set of routing constraints is therefore needed, for each sample $i \in \mathcal{I}_t$, in order to impose the correct sign of the hyperplane function in x^i, $h_t(x^i) = w_t^T x^i + b_t$.
The objective function of the single SVM-type problem at node t optimizes the trade off between the maximization of the hyperplane margin and the minimization of the upper bound on the misclassification cost given by the sum of the slack variables $\xi_{t,i}$ for all $i \in \mathcal{I}_t$, weighted by a positive coefficient C_t:

$$\frac{1}{2}\|w_t\|_2^2 + C_t \sum_{i \in \mathcal{I}_t} \xi_{t,i}.$$

The aim is to train all the hyperplanes with one optimization model only, thus in the objective function we sum the previous terms over all branch nodes $t \in \mathcal{T}_B$ and all samples $i \in \mathcal{I}_t$:

$$\min \sum_{t \in \mathcal{T}_B} \Big(\frac{1}{2}\|w_t\|_2^2 + C_t \sum_{i \in \mathcal{I}_t} \xi_{t,i} \Big). \tag{6}$$

However, the route of the samples in the tree, and consequently the definition of subsets $\mathcal{I}_t$ with $t \in \mathcal{T}_B$, is not preassigned but it is defined by the optimization procedure. Hence, we need to define variables $\xi_{t,i}$ and all margin and routing constraints for every sample in $\mathcal{I}$, taking into account that constraints at node t must activate only on the subset of samples $\mathcal{I}_t$. In order to model the activation/deactivation of these constraints, we need to introduce binary variables which determine the unique path of each sample in the tree. In traditional OCT models, such variables model the assignment of each sample in $\mathcal{I}$ to either leaf nodes (resulting in $|\mathcal{T}_L| \cdot |\mathcal{I}| = 2^D |\mathcal{I}|$ variables), as in [4], or to branch nodes (resulting in $|\mathcal{T}_B| \cdot |\mathcal{I}| = (2^D - 1)|\mathcal{I}|$), as in [7]. Routing constraints are defined using these variables thus often leading to large MIP models. We aim to reduce as much as possible both the number of binary variables and the constraints used in the model to obtain a more tractable problem. Indeed, for each sample in $\mathcal{I}$ we can introduce such binary variables only for the branch nodes in $\mathcal{T}_B''$, resulting in $|\mathcal{T}_B''| \cdot |\mathcal{I}| = 2^{D-1}|\mathcal{I}|$ variables which are half the value $|\mathcal{T}_L| \cdot |\mathcal{I}|$ and less than $|\mathcal{T}_B'| \cdot |\mathcal{I}|$. This is because, following the SVM approach, we do not need to model the assignment of labels to the leaf nodes. Indeed, once hyperplanes $\mathcal{H}_t$ for $t \in \mathcal{T}_B''$ are defined, labels are then implicitly assigned to the leaves, with positive labels always assigned to right leaf nodes and negative labels to the left ones, as shown in Fig. 2. Moreover, the modelling of the leaf level is usually needed in order to evaluate the miscalssification error which is usually computed "inside" the leaves, counting, with appropriate binary variables, the number of misclassified samples assigned to each leaf. Nonetheless, in our case the misclassification cost is controlled by its upper-bound defined by the sum of slack variables which do not depend on the leaf nodes. Thus, we will model the assignment of a sample i only to a node in $\mathcal{T}_B''$ and this will be enough in order to reconstruct the unique path of the sample within the tree.
We can now define binary variables $z_{i,t}$ for all $i \in \mathcal{I}$ and $t \in \mathcal{T}_B''$, as follows:

$$z_{i,t} = \begin{cases} 1 & \text{if sample } i \text{ is assigned to node } t \in \mathcal{T}_B'' \\ 0 & \text{otherwise} \end{cases}.$$

Each sample has to be assigned to exactly one node $t \in \mathcal{T}_B''$, so we must impose that

$$\sum_{t \in \mathcal{T}_B''} z_{i,t} = 1 \qquad\qquad \forall i \in \mathcal{I} \tag{7}$$

$$z_{i,t} \in \{0,1\} \qquad\qquad \forall i \in \mathcal{I}, \quad t \in \mathcal{T}_B'', \tag{8}$$

where constraints (7) will be also referred to as *assignment constraints.* To model the SVM margin constraints (4), we observe that whenever a sample i belongs to $\mathcal{I}_t$, it must be assigned to a node in $\mathcal{S}''(t)$, hence we must have

$$\sum_{\ell \in \mathcal{S}''(t)} z_{i,\ell} = 1 \qquad\qquad \forall i \in \mathcal{I}_t.$$

We use this condition to activate or deactivate the SVM margin constraints when $i \in \mathcal{I}_t$ or $i \notin \mathcal{I}_t$ respectively by means of a Big-M term. Hence, we can express the SVM constraints as

$$y^i(w_t^T x^i + b_t) \geq 1 - \xi_{t,i} - M_\xi \Big(1 - \sum_{\ell \in \mathcal{S}''(t)} z_{i,\ell}\Big) \qquad \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T}_B \tag{9}$$

$$\xi_{t,i} \geq 0 \qquad \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T}_B, \tag{10}$$

where $M_\xi > 0$ is a sufficiently large value such that $M_\xi \geq 1 - y^i(w_t^T x^i + b_t)$ is satisfied for all $i \in \mathcal{I}$. When a sample $i \notin \mathcal{I}_t$, margin constraints in 9 will be always satisfied and variables $\xi_{t,i}$ at the optimum will be set to 0 because their sum is minimized in the objective function.

It remains to force each sample $i \in \mathcal{I}$ to follow a unique path from the root node to the node in $\mathcal{T}''_B$. As we have already commented, we must impose routing constraints only for the branch nodes in $\mathcal{T}'_B$. Indeed, the hyperplane at each node $t \in \mathcal{T}''_B$ is defined according to the soft SVM-type model using the ξs to measure the misclassification cost, and it does not depend on how the samples are finally routed in the leaves (namely on the predicted label). Thus, for each $t \in \mathcal{T}'_B$, we introduce the routing constraints modelling rules in (5) observing that a sample $i \in \mathcal{I}_t$ following either the left or right branch from t, must satisfy

$$\text{either} \sum_{\ell \in \mathcal{S}''_L(t)} z_{i,\ell} = 1 \qquad \text{or} \qquad \sum_{\ell \in \mathcal{S}''_R(t)} z_{i,\ell} = 1.$$

We can model the routing conditions for each $t \in \mathcal{T}'_B$ and for each $i \in \mathcal{I}$ using big-M constraints as follows:

$$w_t^T x^i + b_t \geq -M_\mathcal{H} \Big(1 - \sum_{\ell \in \mathcal{S}''_R(t)} z_{i,\ell}\Big) \qquad \forall i \in \mathcal{I}, \ \forall\, t \in \mathcal{T}'_B \tag{11}$$

$$w_t^T x^i + b_t + \varepsilon \leq M_\mathcal{H} \Big(1 - \sum_{\ell \in \mathcal{S}''_L(t)} z_{i,\ell}\Big) \qquad \forall i \in \mathcal{I}, \ \forall\, t \in \mathcal{T}'_B, \tag{12}$$

where $\varepsilon > 0$ is a sufficiently small positive value to model closed inequalities. We observe that when a sample $i \notin \mathcal{I}_t$, we have that $\sum_{\ell \in \mathcal{S}''_L(t)} z_{i,\ell} = \sum_{\ell \in \mathcal{S}''_R(t)} z_{i,\ell} = 0$ and both the constraints do not force any restriction on these samples. Thus, each separating hyperplane (w_t, b_t) is applied on a subset of samples.

The final formulation of the MARGOT model is the following:

$$\begin{array}{llll}
\text{(MARGOT)} & \displaystyle\min_{w,b,\xi,z} & \displaystyle\sum_{t \in \mathcal{T}_B} \Big(\frac{1}{2}\|w_t\|_2^2 + C_t \sum_{i \in \mathcal{I}} \xi_{t,i}\Big) & \\
& \text{s.t.} & \displaystyle y^i(w_t^T x^i + b_t) \geq 1 - \xi_{t,i} - M_\xi \Big(1 - \sum_{\ell \in \mathcal{S}''(t)} z_{i,\ell}\Big) & \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T}_B \\
& & \displaystyle w_t^T x^i + b_t \geq -M_\mathcal{H} \Big(1 - \sum_{\ell \in \mathcal{S}''_R(t)} z_{i,\ell}\Big) & \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T}'_B \\
& & \displaystyle w_t^T x^i + b_t + \varepsilon \leq M_\mathcal{H} \Big(1 - \sum_{\ell \in \mathcal{S}''_L(t)} z_{i,\ell}\Big) & \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T}'_B \\
& & \displaystyle \sum_{t \in \mathcal{T}''_B} z_{i,t} = 1 & \forall i \in \mathcal{I} \\
& & \xi_{t,i} \geq 0 & \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T}_B \\
& & z_{i,t} \in \{0, 1\} & \forall i \in \mathcal{I}, \quad \forall t \in \mathcal{T}''_B.
\end{array}$$

In Table (1), we report a summary of the number of variables and constraints as a function of the depth of the tree. For the sake of simplicity, we report in Table A.11 in the Appendix, all the notation used in the definition of the model.

	Cardinality
Variables	
Continuous variables (w, b, ξ)	$(n + 1 + \lvert I\rvert)(2^D - 1)$
Integer variables z	$\lvert I\rvert \cdot 2^{D-1}$
Constraints	
Routing constraints	$2(2^{D-1} - 1)\lvert I\rvert$
Margin constraints	$(2^D - 1)\lvert I\rvert$
Assignment constraints	$\lvert I\rvert$

Table 1: Summary of the dimensions of MARGOT.

It is worth noticing that even in the case in which the boolean variables $z_{i,t}$ are fixed to values $\hat{z}_{i,t}$ (e.g. by setting the values returned by another classification tree method such as CART), the subproblems solved at each $t \in \mathcal{T}_B$ are not pure SVM problems unless $t \in \mathcal{T}''_B$. Let us first note that sets $\mathcal{I}_t$ can be equivalently redefined as:

$$\mathcal{I}_t = \Big\{ i \in \mathcal{I} : \sum_{\ell \in \mathcal{S}''(t)} \hat{z}_{i,\ell} = 1 \Big\}. \tag{14}$$

Thus, the optimization problem (13) decomposes into the resolution of $|\mathcal{T}_B|$ problems, where the first $|\mathcal{T}'_B|$ problems, one for each $t \in \mathcal{T}'_B$, are of the type:

$$\begin{aligned}
\min_{w_t, b_t, \xi_t} \quad & \frac{1}{2}\|w_t\|_2^2 + C_t \sum_{i \in \mathcal{I}_t} \xi_{t,i} & \\
\text{s.t.} \quad & y^i(w_t^T x^i + b_t) \geq 1 - \xi_{t,i} & \forall i \in \mathcal{I}_t \\
& w_t^T x^i + b_t \geq 0 & \forall i \in \mathcal{I}_{R(t)} \\
& w_t^T x^i + b_t + \varepsilon \leq 0 & \forall i \in \mathcal{I}_{L(t)} \\
& \xi_{t,i} \geq 0 & \forall i \in \mathcal{I}_t.
\end{aligned}$$

Only the remaining $|\mathcal{T}''_B|$ problems are pure SVM problems in that routing constraints in (13) are not defined for the nodes of the last branching level.

4. MARGOT with feature selection

In Machine Learning, feature selection is the process of selecting the most relevant features of a dataset. Alongside filter and wrapped methods, the embedded ones integrate feature selection in the training process. In optimization literature, a solution is defined as sparse when the cardinality of the variables not equal to 0 is ”low”. The sparsity of an optimal solution is a requirement that is highly desirable in many application contexts. As matter of fact, the concept of embedded feature selection translates in the sparsity requirement for the solution of the optimization model used for the training process.

The MARGOT formulation does not take into account the sparsity of the hyperplane coefficients variables w_t for each node $t \in \mathcal{T}_B$. Thus, in order to improve the interpretability of our method, we propose two alternative versions of the MARGOT model where the number of features used at each branch node of the tree is either limited (”hard” approach) or penalized (”soft” approach). This, together with the tree structure of the model, yields a hierarchy scheme on the subset of features which mostly affect the classification.

In more detail, we introduce, for each node $t \in \mathcal{T}_B$ and for each feature $j \in [n]$, a new binary variable $s_{t,j} \in \{0, 1\}$ such that:

$$s_{t,j} = \begin{cases} 1 & \text{if feature } j \text{ is } \textit{selected} \text{ at node } t \ (w_{t,j} \neq 0) \\ 0 & \text{otherwise} \end{cases}.$$

Classical Big-M constraints on the variables $w_{t,j}$ must be added to model the above implication for all t, i:

$$-M_w s_{t,j} \leq w_{t,j} \leq M_w s_{t,j} \qquad \forall t \in \mathcal{T}_B, \quad \forall j = 1, \ldots, n \tag{15}$$

$$s_{t,j} \in \{0, 1\} \qquad \forall t \in \mathcal{T}_B, \quad \forall j = 1, \ldots, n, \tag{16}$$

where M_w is set to a sufficiently large value. Similarly to [27], where a MILP feature selection version of the ℓ_1-norm SVM primal problem is proposed, in the hard features selection approach, we restrict the number of features used at each node to be not greater than a budget value. We do that by introducing a hyperparameter B_t and a budget constraint for each branch node $t \in \mathcal{T}_B$:

$$\sum_{j}^{n} s_{t,j} \leq B_t. \tag{17}$$

The resulting formulation for the hard version denoted as HFS-MARGOT is the following:

$$\begin{aligned}
\text{(HFS-MARGOT)} \quad \min_{w,b,\xi,z,s} \quad & \sum_{t\in\mathcal{T}_B} \Big(\frac{1}{2}\|w_t\|_2^2 + C_t \sum_{i\in\mathcal{I}} \xi_{t,i}\Big) & \\
\text{s.t.} \quad & (7) - (12) & \\
& -M_w s_{t,j} \leq w_{t,j} \leq M_w s_{t,j} & \forall t \in \mathcal{T}_B, \quad \forall j = 1,\dots,n \\
& \sum_{j}^{n} s_{t,j} \leq B_t & \forall t \in \mathcal{T}_B \\
& s_{t,j} \in \{0,1\} & \forall t \in \mathcal{T}_B, \quad \forall j = 1,\dots,n.
\end{aligned}$$

In the soft approach, we remove the budget constraints and control their violations by adding a penalization term in the objective function weighted by an appropriate hyperparameter α:

$$\sum_{t\in\mathcal{T}_B} \max\Big\{0, \sum_{j}^{n} s_{t,j} - B_t\Big\}.$$

The resulting version allows for more than B_t features to be selected at each splitting node t penalizing the number of the features that exceed the budget. The max functions can be linearized with the introduction of a new continuous variable u_t, for all $t \in \mathcal{T}_B$, thus obtaining the following MIQP formulation denoted as SFS-MARGOT:

$$\begin{aligned}
\text{(SFS-MARGOT)} \quad \min_{w,b,\xi,z,s} \quad & \sum_{t\in\mathcal{T}_B} \Big(\frac{1}{2}\|w_t\|_2^2 + C_t \sum_{i\in\mathcal{I}} \xi_{t,i} + \alpha u_t\Big) & \\
\text{s.t.} \quad & (7) - (12) & \\
& (15) - (16) & \\
& u_t \geq \sum_{j}^{n} s_{t,j} - B_t & \forall t \in \mathcal{T}_B \\
& u_t \geq 0 & \forall t \in \mathcal{T}_B.
\end{aligned}$$

Both models induce *local* sparsity on each single vector w_t and this local approach may be preferable rather than addressing *global* sparsity on the full vector w, as in [4]. Indeed, global sparsity of the vector w has little effect on the "spreadness" of the features among the splitting rules in the tree, often leading to trees with fewer and less interpretable splits, usually in the higher levels. In this sense, inducing local sparsity can generate models which better exploit the hierarchical structure of the tree. A solution which is more "spread" in terms of features, can thus result in a more interpretable machine learning model because it yields a hierarchy scheme among the features which mostly affect the classification. Of course, other constraints facing additional requirements on the selected features can be added to these formulations. Indeed, sparsity of the w_t variables may not be the only interesting property when addressing the interpretability of the decision.

5. Heuristic for a starting feasible solution

We develop a simple greedy heuristic algorithm to find a feasible solution to be used as a good-quality warm start for the optimization procedure. As well known, the value of the warm start solution is an upper bound

Variable	Meaning	Model
$w \in \mathbb{R}^{\|\mathcal{T}_B\| \times n}$	split coefficients	MARGOT
$b \in \mathbb{R}^{\|\mathcal{T}_B\|}$	split biases	MARGOT
$\xi \in \mathbb{R}^{\|\mathcal{T}_B\| \times \|\mathcal{I}\|}$	slack variables	MARGOT
$z \in \{0,1\}^{\|\mathcal{I}\| \times \|\mathcal{T}_B''\|}$	samples assignment to nodes in $\mathcal{T}_B''$	MARGOT
$s \in \{0,1\}^{\|\mathcal{T}_B\| \times n}$	feature selection	HFS/SFS-MARGOT
$u \in \mathbb{R}^{\|\mathcal{T}_B\|}$	soft FS penalization parameter	SFS-MARGOT

Table 2: Overview of all the variables used in the proposed formulations and their meaning.

on the optimal one and it can be used to prune nodes of the branch and bound tree of the MIP solver, yielding eventually to shorter computational times. Thus developing a good warm start solution is usually addressed in MIP formulation and implemented in off-the-shelf softwares at the root node of the branching tree. In [4], several warm start procedures are proposed, from the simplest one, which consists in using the solution provided by CART, to more tailored ones.

The general heuristic scheme, denoted as Local SVM Heuristic, exploits the special structure of the MIQP models addressed and can be applied to obtain feasible solutions for MARGOT, HFS-MARGOT, and SFS-MARGOT models. The Local SVM Heuristic is based on a greedy recursive top-down strategy. Starting from the root node, the maximum margin hyperplane is computed using an SVM model embedding, when needed, hard or soft constraints for features selection. More in details, for each node $t \in \mathcal{T}_B$, given a predefined index set $\mathcal{I}_t \subseteq \mathcal{I}$, the heuristic solves the following problem:

$$
\begin{array}{llll}
(\text{WS-SVM}_t) & \displaystyle\min_{w,b,\xi,s,u} & \displaystyle\frac{1}{2}\|w\|_2^2 + C_t \sum_{i \in \mathcal{I}_t} \xi_i + \alpha u & \\
& \text{s.t.} & y^i(w^T x^i + b) \geq 1 - \xi_i & \forall i \in \mathcal{I}_t \\
& & -M_w s_j \leq w_{t,j} \leq M_w s_j & \forall j = 1, \dots, n \\
& & u \geq \displaystyle\sum_{j}^{n} s_j - B_t & \\
& & u \geq 0 & \\
& & \xi_i \geq 0 & \forall i \in \mathcal{I} \\
& & s_j \in \{0,1\} & \forall j = 1, \dots, n.
\end{array}
$$

where hyperparmeters B_t, α and variable u may be fixed to specific values to get warm start solutions for the three different models, MARGOT, HFS-MARGOT, and SFS-MARGOT. In particular, when $B_t = n$ we do not impose restrictions on the number of features. When $B_t < n$ and u is set to zero we obtain an ℓ_2-norm SVM model with a hard constraint on the number of features, similarly to the approach in [27]. Finally, when $\alpha > 0, B_t < n$ and u is variable, we impose a soft constraint on the number of features. Given the optimal tuple $(\widehat{w}_t, \widehat{b}_t, \widehat{\xi}_t, \widehat{s}_t) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{|\mathcal{I}|} \times \{0,1\}^n$ obtained at node t, the samples are partitioned to the left or right child node in the subsequent level in the tree according to the routing rules defined by the hyperplane $\mathcal{H}_t = \{x \in \mathbb{R}^n : \widehat{w}_t^T x + \widehat{b}_t = 0\}$. Thus, each node t works on a different subset of samples $\mathcal{I}_t \subseteq \mathcal{I}$ and $\mathcal{I}_{L(t)}$ and $\mathcal{I}_{R(t)}$ are the index sets of samples assigned to the left or right child node of t. At the end of the procedure, for each $t \in \mathcal{T}_B$, the solutions $(\widehat{w}_t, \widehat{b}_t, \widehat{\xi}_t) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^{|\mathcal{I}|}$, together with solutions $\widehat{s}_t \in \{0,1\}^n$, when needed, constitute a feasible solution for the original problem. In the very last step, variables $z_{i,t}$ are set according to the the definitions of sets $\mathcal{I}_t$, $t \in \mathcal{T}_B''$. The general scheme which encompasses the three different strategies is reported in Algorithm 1.

The heuristic procedure requires the solution of $2^D - 1$ MIQP problems with a decreasing number of constraints (depending on the size of $\mathcal{I}_t$) that can be easily handled by off-the-shelf MIP solvers. We show in the computational experiments that the use of our heuristics improves the quality of the warm start solution with respect to the chosen optimization solver.

Algorithm 1: Local SVM Heuristic

```
Data: $\{(x^i, y^i) \in \mathbb{R}^n \times \{-1, 1\}, i \in \mathcal{I}\}$;

Parameters: $\{C_t > 0, t \in \mathcal{T}_B\}$, $\widehat{\alpha} > 0$, $M_w > 0$, $D$, $\varepsilon > 0$, $\{\widehat{B}_t > 0, t \in \mathcal{T}_B\}$;
Input: Model $\in$ {MARGOT, HFS-MARGOT, SFS-MARGOT};

Initialize: $\mathcal{I}_0 = \mathcal{I}$, $\mathcal{I}_t = \emptyset \; \forall t \in \mathcal{T}_B \setminus \{0\}$, $\widehat{z}_{i,t} = 0, \forall t \in \mathcal{T}_B'', \forall \; i \in \mathcal{I}$, $\widehat{\xi}_{t,i} = 0, \forall t \in \mathcal{T}_B, \forall \; i \in \mathcal{I}$;
for level $k = 0, \dots, D-1$ do
    for node $t = 2^k - 1, \dots, 2^{k+1} - 2$ do

        if model = MARGOT then
            set $B_t = n$
        end
        if model = HFS-MARGOT then
            set $B_t = \hat{B}_t$ and $u = 0$
        end
        if model = SFS-MARGOT then
            set $B_t = \widehat{B}_t$ and $\alpha = \widehat{\alpha}$
        end

        Find $(\widehat{w}_t, \widehat{b}_t, \widehat{\xi}_t, \widehat{s}_t)$ optimal solution of WS-SVM$_t$

        Set $\mathcal{I}_{L(t)} = \{i \in \mathcal{I}_t : \widehat{w}_t^T x^i + \widehat{b}_t + \varepsilon < 0\}$ and $\mathcal{I}_{R(t)} = \{i \in \mathcal{I}_t : \widehat{w}_t^T x^i + \widehat{b}_t \geq 0\}$
    end
end
for $t \in \mathcal{T}_B''$ do
    if $i \in \mathcal{I}_t$ then
        Set $\widehat{z}_{i,t} = 1$
    end
end

Output: A feasible solution $(\widehat{w}, \widehat{b}, \widehat{\xi}, \widehat{s}, \widehat{z})$ for all the input models.
```

6. Computational Results

In this section we present different computational results where models MARGOT, HFS-MARGOT and SFS-MARGOT are compared to other two benchmark OCT models:

- OCT-H, as described in [4];
- MM-SVM-OCT, as proposed in [7] where no relabeling is allowed.

All mathematical programming models were coded in Python and solved using Gurobi 9.5.1 on a server Intel(R) Xeon(R) Gold 6252N CPU processor at 2.30 GHz and 96 GB of RAM. We used two groups of datasets:

- 3 non-linearly separable synthetic datasets in a 2-dimensional feature space, in order to give a graphical representation of the maximum margin approach (presented in section 6.1);
- 10 datasets from UCI Machine Learning Repository [20], to test effectiveness of the formulations as regards both the out-of-sample error and the optimization performance (presented in section 6.2).

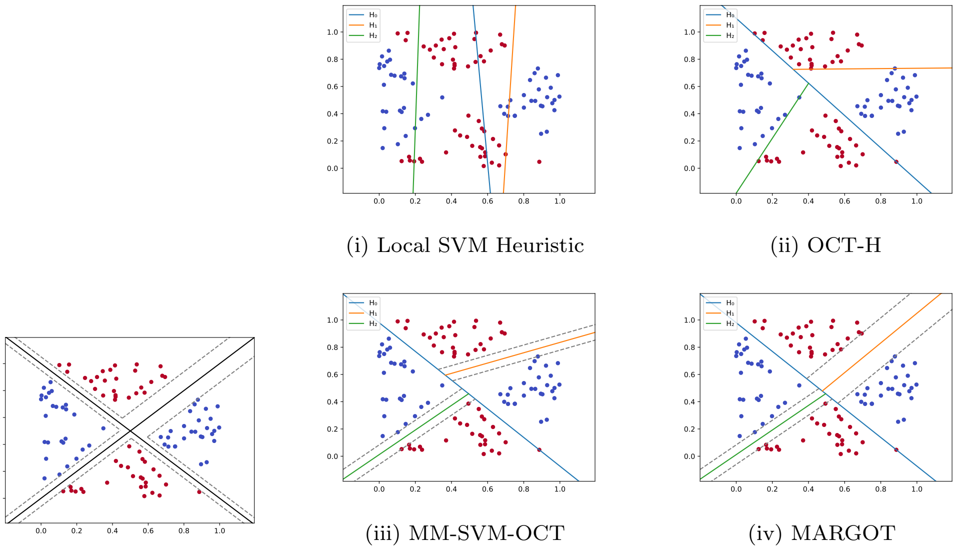
For the results on the UCI datasets, we performed a 4-fold cross validation for the parameters C which is detailed in the specific sections below. In section 6.3, we eventually present a brief analysis on the Local SVM algorithm presented in section 5, in order to motivate the use of the warm start solution in input to the solver.

6.1. Results on 2-dimensional synthetic datasets

As regards the 2-dimensional datasets, we used two artificially constructed problems, `4-partitions` and `6-partitions`, and the more complex synthetic `fourclass` data set ([24]) as reported in LIBSVM Library

[17]. Our aim is to offer a glimpse of the differences in the hyperplanes generated by the different optimal tree models. We also reported the solution returned by the Local SVM Heuristic (Algorithm 1) to show how far the greedy solution is from the optimal ones. For all three synthetic datasets, there exist a set of hyperplanes that can reach perfect classification on the training data. In particular, `4-partitions` and `6-partitions` were constructed by defining hyperplanes with margins and plotting 108 and 96 random points, respectively, in regions outside the margin (see Figures 3a, 4a). Although reaching zero classification error on the training data is not desirable in ML models, in these cases, we want to highlight the power of using hyperplanes with margins to derive more robust classifiers. We did not account for the out-of-sample performances; thus, the whole dataset is used to train the optimal tree. Of course, in the 2-dimensional case, we do not consider HFS-MARGOT and SFS-MARGOT. The results are commented below, and a cumulative view is reported in Table 3. In all the experiments, the time limit of the solver has been set to 4 hours.

In Figures 3b, 4b, 5 we report the hyperplanes generated by the Local SVM Heuristic, OCT-H, MM-SVM-OCT, and MARGOT in figures (i), (ii), (iii), and (iv) respectively. Different colors correspond to different branch nodes of the tree, as reported in the legend. To highlight the margin of the hyperplanes of the last splitting branch nodes, in the case of MM-SVM-OCT and MARGOT, we also plotted the two supporting hyperplanes at a distance $2||w||^{-1}$.



(i) Local SVM Heuristic

(ii) OCT-H

(iii) MM-SVM-OCT

(iv) MARGOT

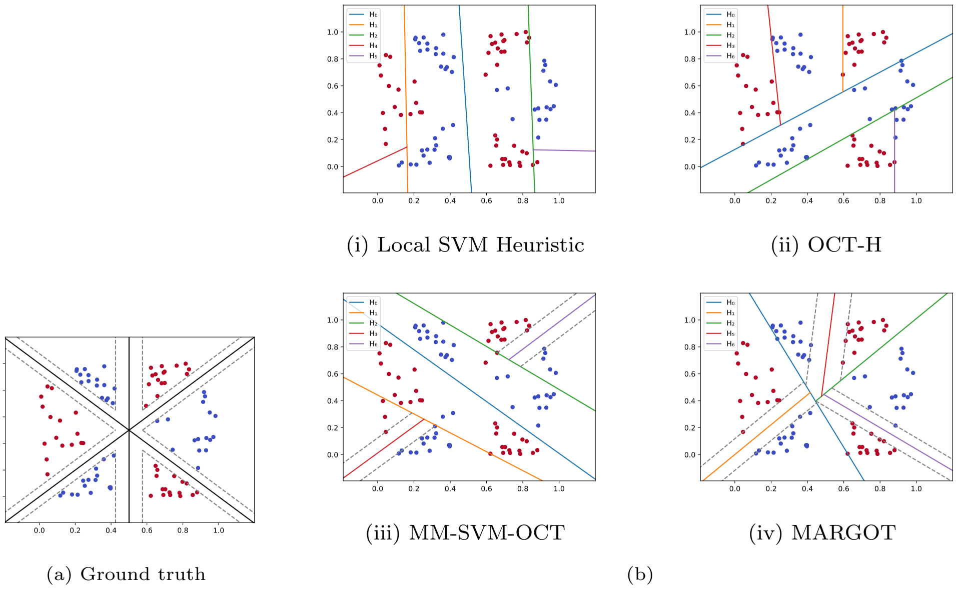
(a) Ground truth

(b)

Figure 3: Results on the `4-partitions` synthetic dataset.

For the `4-partitions` dataset we consider trees with depth $D = 2$. Fig. 3b (i) represents the tree obtained by the Local SVM Heuristic. Hyperplane $\mathcal{H}_0$ at the root node colored in blue is obtained on the whole dataset, while the hyperplanes at its child nodes, $\mathcal{H}_1$ in green and $\mathcal{H}_2$ in orange, are obtained considering the partition of the points given by $\mathcal{H}_0$ as the splitting rule on the whole dataset. The heuristic returns a classification tree that does not classify all data points correctly, obtaining an accuracy of 86.1%. Concerning the OCT approaches in Fig. 3b (ii), (iii), and (iv), the solver reaches the certified optimal solution on all three models, thus obtaining 0% MIP gap in different computational times. All three OCT models reach an accuracy of 100%. We note that OCT-H creates hyperplanes that do not consider the margin. Indeed, the objective of this approach is to minimize the misclassification cost and the number of features used across the whole tree. Thus, as it happens for the orange hyperplane $\mathcal{H}_1$, OCT-H tends to select axis-aligned hyperplanes to split the points. As regards the tree produced by the MM-SVM-OCT model, only the minimum margin among all the hyperplanes is maximized. Consequently, only the green hyperplane $\mathcal{H}_2$ lies in the center between the partitions of points, while the others do not. Instead, the MARGOT tree is the one that most resembles the ground truth in Fig. 3a and both the orange $\mathcal{H}_1$ and green $\mathcal{H}_2$ hyperplanes have a wider margin.

Fig.4, shows the more complex synthetic dataset `6-partitions` where we set $D = 3$, the minimum depth to correctly classify all samples is. MARGOT and OCT-H reach perfect classification on the whole dataset, and both Local SVM Heuristic and MM-SVM-OCT return good accuracies above 90%. Moreover, the solutions



(i) Local SVM Heuristic

(ii) OCT-H

(iii) MM-SVM-OCT

(iv) MARGOT

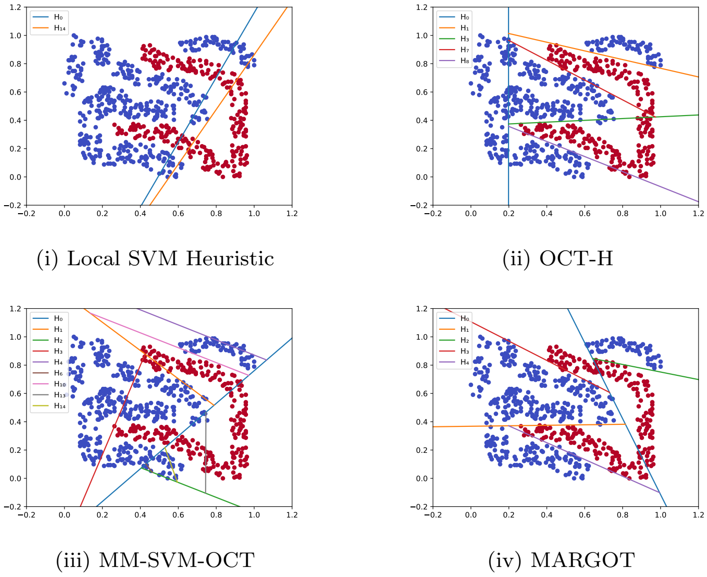
(a) Ground truth

(b)

Figure 4: Results on the `6-partitions` synthetic dataset.

produced by MARGOT and MM-SVM-OCT models are optimal, while OCT-H optimization reaches the time limit with a MIP gap of 50%. Also here, among all the trees generated, MARGOT seems to yield the more reliable classifier being the one closer to the ground truth.

Finally, we evaluated the four methods on the `fourclass` dataset. In this case, being the problem the most complex of the three, none of the models has been solved to certified optimality, and OCT-H, MM-SVM-OCT and MARGOT optimization procedures reach a MIP gap of 100%, 74.5% and 67.2% respectively. MARGOT and OCT-H approaches were able to correctly classify almost all the samples, outperforming MM-SVM-OCT which reaches an accuracy of 88.4%. In this highly non-linearly separable case, it is possible to observe how the greedy fashion of the Local SVM Heuristic may generate models not able to capture the underlying truth of the data.



(i) Local SVM Heuristic

(ii) OCT-H

(iii) MM-SVM-OCT

(iv) MARGOT

Figure 5: Results on the `fourclass` synthetic dataset.

It is worth noticing that both the approaches on the `6-partitions` and `fourclass` datasets tend to minimize the complexity of the tree, i.e. the number of hyperplane splits created. In OCT-H model, this is a consequence of both the penalization of the selection of features in the objective function and the presence of specific constraints and variables which model the pruning of the tree. Similarly, MM-SVM-OCT controls the complexity of the tree with a penalization term in the objective function by introducing binary variables and related constraints. On the contrary, in MARGOT model the complexity is implicitly minimized in the objective function. Indeed, creating an hyperplane split in a node t leads to new coefficients $w_t \neq 0$ and variables $\xi_t \neq 0$ that appear in the objective function.

			Local SVM			OCT-H			MM-SVM-OCT			MARGOT		
Dataset	$\lvert\mathcal{I}\rvert$	D	Time	Gap	ACC	Time	Gap	ACC	Time	Gap	ACC	Time	Gap	ACC
`4-partitions`	108	2	0.1	-	86.1	53.5	0	100	5.2	0	100	0.2	0	100
`6-partitions`	96	3	0.1	-	91.7	14400	50	100	181.2	0	96.9	5410.7	0	100
`fourclass`	689	4	1.0	-	77.9	14400	100.0	97.8	14400	74.5	88.4	14400	67.2	99.6

Table 3: Time (s), Gap (%) and ACC (%) performances on the synthetic datasets (time limit = 14400s = 4h).

6.2. Results on UCI datasets

We compare the different OCT models computing two predictive measures: the accuracy (ACC) and the area under the curve (AUC). The ACC is the percentage of correctly classified samples and the AUC is the mean of the percentage of correctly classified samples with positive label and the percentage of correctly classified samples with negative label.

Hence,

$$ACC = \frac{TP + TN}{TP + TN + FP + FN},$$

$$AUC = \frac{\frac{TP}{TP+FN} + \frac{TN}{TN+FP}}{2},$$

where TP are true positives, TN are true negatives, FP false positives and FN false negatives. We first partitioned each dataset in training (80%) and test (20%) sets and then performed a 4-fold-cross validation (4-FCV) on the training set in order to find the best hyperparameters. In the first set of results where feature selection is not taken into account, the selected hyperparameters are the ones which gave the best average validation accuracy in the cross validation process. As it will be explained later in this section, a more specific tuning of the hyperparameters was implemented for the second set of results where the sparsity of the hyperplanes is addressed. Once the best hyperparameters are selected, we compute the predictive measures on the training and test set. The results on the training dataset, as well as all the hyperparameters used in this paper, can be found in the appendix.

Dataset	$\lvert\mathcal{I}\rvert$	n	Class (%)
Breast Cancer Diagnostic	569	30	63/36
Breast Cancer Wisconsin	449	9	47/53
Climate Model Crashes	540	18	9/91
Ionosphere	350	33	36/64
Parkinsons	195	22	25/75
Heart Disease Cleveland	297	13	54/46
Sonar	208	60	53/47
SPECTF Heart	267	44	21/79
Tic-Tac-Toe	958	18	35/65
Wholesale	440	7	81/19

Table 4: Information about the datasets considered.

We injected warm start solutions, produced by the Local SVM Heuristic, for MARGOT, HFS-MARGOT and SFS-MARGOT models. Following the warm start procedure presented in [4], OCT-H was also given its starting solution. A time limit of 30 seconds was given for every warm start procedure and an overall time limit of 10 minutes was set for training the models. The maximum depth of the trees generated was set to $D = 2$ and the ranges of the hyperparameters used in the 4-FCV for the different models are the following:

- For OCT-H the minimum number of samples per leaf was set to 5% of the total number of training samples and the grid used for the hyperparameter α is $\{0\} \cup \{2^i : i \in \{-8, \dots, 2\}\}$.
- For MM-SVM-OCT we used the same grid as the one specified in the related paper; $c_1 \in \{10^i : i \in \{-5, \dots, 5\}\}$ and the complexity hyperparameter $c_3 \in \{10^i : i \in \{-2, \dots, 2\}\}$.
- For MARGOT, we consider all possible combination resulting from $C_t \in \{10^i : i \in \{-5, \dots, 5\}\}$ for all $t \in Tb$, and $C_1 = C_2$, imposing the same C_t values for all nodes t belonging to the same branching level.
- For HFS-MARGOT, we used the same grid for the C_t values as in MARGOT and, concerning the budget parameters $\{B_t, \text{ for all } t \in \mathcal{T}_B\}$, we varied all possible combinations resulting from values of $B_0 \in \{1, 2, 3\}$ and values of $B_1 = B_2 \in \{2, 3, 4\}$, considering only combinations where $B_0 \leq B_1 = B_2$, with the value 4 regarded as the maximum number of features a node can admit in order to be interpretable.
- For SFS-MARGOT, the grid used for the hyperparameter α is $\{2^i : i \in \{0, \dots, 10\}\}$ and we varied C_t values as in MARGOT but in a smaller grid $\{10^i : i \in \{-4, -2, 0, 2, 4\}\}$. We set all budget values $B_t = 1$ for all $t \in \mathcal{T}_B$, allowing the model to have full flexibility on where to use more features than the budget value.

Regarding the ε parameter used in our formulation, for similar reasons as the one stated in [4], we set $\varepsilon = 0.001$ as a compromise between choosing small values that lead to numerical issues and large values that can affect the feasible region excluding possible solutions. Moreover, the big-M parameters on our model are fixed to a sufficiently large amount ($M_\xi = M_w = 50, M_\mathcal{H} = 100$), while the ones in MM-SVM-OCT are fixed as indicated in the relative paper.

6.2.1. First set of results

The first set of results are shown in Table 5, where we compare the predictive performances of MARGOT against OCT-H and MM-SVM-OCT. We can see how MARGOT takes full advantage of the generalization capabilities deriving from the maximum margin approach resulting in much higher ACC and AUC scores on the test set.

	OCT-H		MM-SVM-OCT		MARGOT	
Dataset	ACC	AUC	ACC	AUC	ACC	AUC
Breast Cancer Diagnostic	94.7	94.3	93.9	92.7	**97.4**	**96.9**
Breast Cancer Wisconsin	94.4	94.3	**96.7**	**96.6**	**96.7**	**96.6**
Climate Model Crashes	93.5	86.4	**97.2**	**88.4**	**97.2**	**88.4**
Ionosphere	84.3	79.8	85.7	80.0	**87.1**	**82.0**
Parkinsons	82.1	**84.7**	**87.2**	81.6	84.6	83.1
Heart Disease Cleveland	80.0	79.7	81.7	81.3	**83.3**	**83.0**
Sonar	66.7	65.9	**73.8**	**73.0**	71.4	70.7
SPECTF Heart	74.1	53.3	**79.6**	50.0	**79.6**	**56.8**
Tic-Tac-Toe	96.4	95.5	96.9	96.2	**97.9**	**97.0**
Wholesale	84.1	79.8	83.0	74.2	**87.5**	**85.1**

Table 5: Results on the predictive performances of the OCT methods evaluated: test ACC (%) and test AUC (%).

Computational times and MIP gaps can be found in Table 6. It is clear how MARGOT optimization problem is much easier to solve than OCT-H and MM-SVM-OCT. Indeed, 9 times out of 10, MARGOT reaches the optimal solution with a mean computational time and MIP gap of 121.2 seconds and 0.3% respectively. MM-SVM-OCT reaches the optimal solution 3 times out of 10, with a mean running time and gap of 470.7 seconds and 11.5% respectively, and OCT-H reaches the optimum 1 times out of 10, with a mean time and gap of 568.3 seconds and 81.8% respectively.

	OCT-H		MM-SVM-OCT		MARGOT	
Dataset	Time	Gap	Time	Gap	Time	Gap
Breast Cancer Diagnostic	600.4	72.8	600.2	27.5	**7.4**	**0.0**
Breast Cancer Wisconsin	600.3	83.7	213.8	**0.0**	**3.0**	**0.0**
Climate Model Crashes	600.2	90.6	600.2	0.6	**10.7**	**0.0**
Ionosphere	62.2	**0.0**	600.0	12.3	**15.1**	**0.0**
Parkinsons	600.3	90.8	600.1	12.5	**207.4**	**0.0**
Heart Disease Cleveland	600.1	91.9	600.0	28.8	**318.1**	**0.0**
Sonar	600.1	96.1	262.7	**0.0**	**1.3**	**0.0**
SPECTF Heart	600.1	96.9	**30.3**	**0.0**	600.2	3.3
Tic-Tac-Toe	600.4	100.0	600.1	0.9	**2.7**	**0.0**
Wholesale	600.4	95.6	600.0	32.4	**46.4**	**0.0**

Table 6: Results on the optimization performances of the OCT methods evaluated: computational times (s) and MIP Gaps (%).

6.2.2. Second set of results: feature selection

In the following set of results, we compare HFS-MARGOT and SFS-MARGOT with OCT-H. Both MARGOT and MM-SVM-OCT do not appear in this set of results because these models do not address sparsity of the hyperplanes. For this analysis, a different hyperparameter selection was carried out. This was done in order to consider the fact that we are not just comparing the predictive performances of the methods, but we are also evaluating the feature selection aspect. Still the 4-FCV was conducted, but this time we did not select the hyperparameters which gave the highest mean validation accuracy. Indeed, highest mean validation accuracy values yield to models which select a high number of features, thus contrasting the aim to create more interpretable trees. At the same time, it is not useful to just consider interpretability in that the hyperparameters which gave the best results in terms of feature selection may result in less performing classifiers.

Thus, we proceeded as follows. For each dataset, after performing the standard 4-FCV, we highlighted the combinations of hyperparameters which resulted in a mean validation accuracy in the range $[0.975\gamma, \gamma]$, where γ is the best mean validation accuracy value. This way, we selected the combinations of hyperparameters corresponding to "good" classifiers. Among these combinations, we selected the ones corresponding to the less number of feature used and among these the one with the best validation accuracy. We denote by OCT-H*, HFS-MARGOT* and SFS-MARGOT* the tree models generated with this feature selection driven hyperperameter tuning. To the best of our knowledge, this is the first time that a tailored cross validation was carried out in order to fairly compare different ML models, produced by optimization based methods which take into account feature selection aspects.

	OCT-H*		HFS-MARGOT*		SFS-MARGOT*	
Dataset	ACC	AUC	ACC	AUC	ACC	AUC
Breast Cancer Diagnostic	94.7	94.3	**95.6**	**95.5**	94.7	94.3
Breast Cancer Wisconsin	**95.6**	95.3	**95.6**	**95.4**	94.4	94.5
Climate Model Crashes	**98.1**	**93.9**	94.4	76.8	96.3	82.8
Ionosphere	**88.6**	**85.8**	87.1	83.8	85.7	81.8
Parkinsons	**87.2**	81.6	**87.2**	**84.8**	**87.2**	**84.8**
Heart Disease Cleveland	80.0	79.5	83.3	82.6	**86.7**	**86.2**
Sonar	71.4	71.1	71.4	70.9	**73.8**	**73.6**
SPECTF Heart	75.9	51.1	75.9	57.8	**83.3**	**65.9**
Tic-Tac-Toe	95.8	94.7	76.6	74.0	**97.9**	**97.0**
Wholesale	87.5	86.1	86.4	85.2	**88.6**	**86.9**

Table 7: Results on the predictive performances of the OCT models resulting from the feature selection driven tuning of the hyperparameters: test ACC (%) and test AUC (%).

	OCT-H*		HFS-MARGOT*		SFS-MARGOT*	
Dataset	Time	Gap	Time	Gap	Time	Gap
Breast Cancer Diagnostic	600.5	72.8	**313.9**	**0.0**	314.7	**0.0**
Breast Cancer Wisconsin	600.2	70.4	309.1	**0.0**	**65.1**	**0.0**
Climate Model Crashes	600.2	81.4	**145.3**	**0.0**	600.1	14.2
Ionosphere	600.2	89.1	511.5	**0.0**	**26.2**	**0.0**
Parkinsons	600.1	82.4	600.3	**28.4**	600.0	62.0
Heart Disease Cleveland	600.2	86.9	**106.5**	**0.0**	600.0	28.2
Sonar	600.1	91.4	600.9	98.3	600.1	**42.2**
SPECTF Heart	600.1	94.5	600.1	**93.8**	600.1	99.9
Tic-Tac-Toe	600.3	91.9	600.6	98.1	600.1	**1.0**
Wholesale	600.2	73.5	**36.4**	**0.0**	600.0	80.1

Table 8: Results on the optimization performances of the OCT models resulting from the features selection driven tuning of the hyperparameters: computational times (s) and MIP Gap values (%).

In tables 7 and 8, we report both predictive and optimization performances of the compared models. Table 9 shows the difference in the features selected among the analysed OCT models. We denote by F the set of distinct features used overall in the tree, and we denote by F_t the set of features selected at node t. The first two columns of Table 9, refer to OCT-H and MARGOT as they were implemented in Table 5, while the subsequent ones are related to models for which the specific feature selection hyperparameter tuning was adopted. As expected, MARGOT model tends to use all the features available because it has no feature selection constraint or term in the objective function. Also OCT-H, for which a standard 4-FCV is carried out too, often produces tree models which are not interpretable even though the number of features selected is penalized in its formulation. OCT-H*, HFS-MARGOT* and SFS-MARGOT* generate models selecting lower number of features, maintaining good prediction performances, as we can see in tables 9 and 7. In particular, we can notice how SFS-MARGOT* presents better ACC and AUC values and this is probably due to the fact that the SFS-MARGOT model benefits both from the maximum margin approach and the soft feature selection. Indeed, we can see from table 9 how SFS-MARGOT*, where violation of the budget constraints is allowed, has more freedom in the selection of feature compared to the more restrictive approach of HFS-MARGOT*, which at each node cannot exceed a predifined number of selected features. One thing to notice is that on these results, both OCT-H and OCT-H* tend to use the selected features just in the first branch node of the tree. Similar thing is done by the SFS-MARGOT* model, while, on the contrary, the features selected by HFS-MARGOT* tend to be more spread throughout the branch nodes. Using hard budget constraint limiting the number of feature selected at each branch node, has indeed two consequences: on the one hand it spreads more evenly the features selected among the tree structure, on the other this restriction, which is made a priori by setting parameters B_t, might result insufficient in order achieve the best performances. This is the case of the Tic-Tac-Toe dataset where the HFS-MARGOT* model clearly did not perform well, and this is most likely because the values we adopted for budget parameters B_t were too limiting. Of course, we could have used higher budget values but this, apart from leading to less interpretable tree structures and time consuming hyperparameters tuning, does not attain the scope of our computational results.

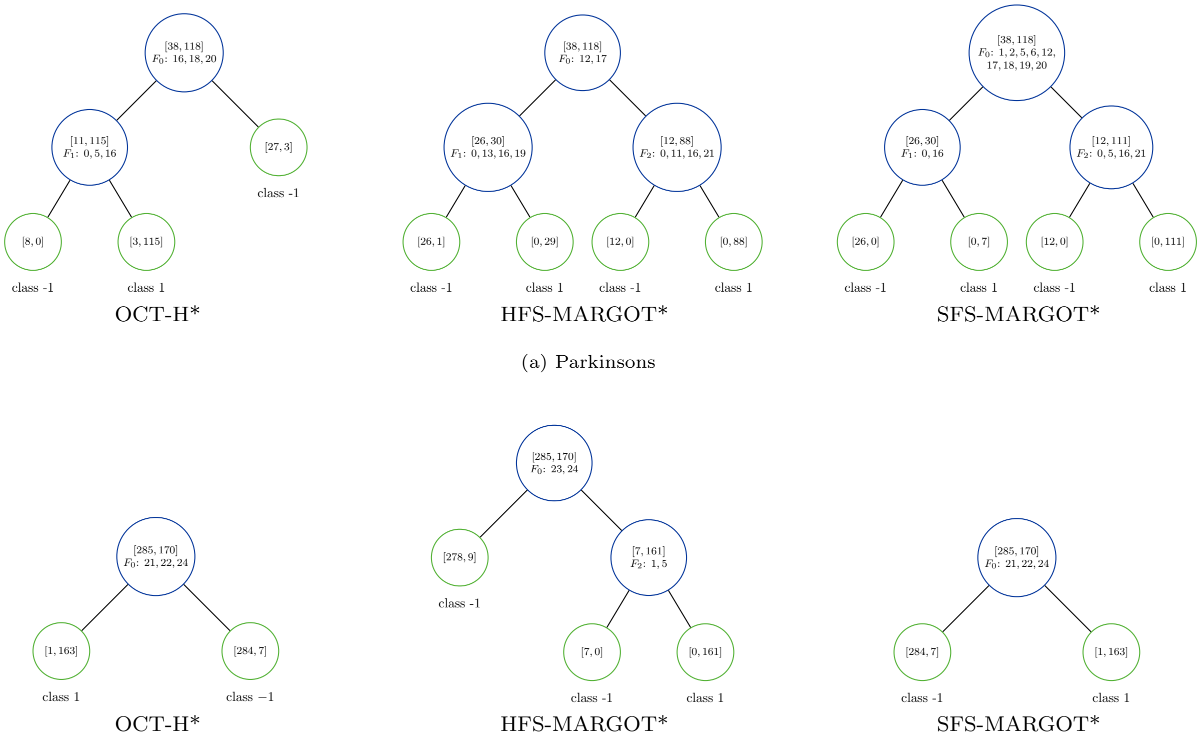
In Figures 6, we give more insight on how the features selected by the different OCT models divide the training samples. We focused on two datasets, the Parkinsons and the Breast Cancer Wisconsin ones, in that we found them explicative of the behaviour of all the trees outputed by OCT-H*, HFS-MARGOT* and SFS-MARGOT* models.

6.3. Warm Start

During the branch and bound process implemented by Gurobi, after finding a first incumbent solution, the solver applies heuristics to improve the quality of the incumbent solution before further exploring the branch and bound tree. In general, a warm start input solution is accepted as the first incumbent if its value is better than the initial solution found by the solver. In Table 10, we analyse the quality of the warm start input solution produced by the Local SVM Heuristic. To this aim, we introduce the following definitions: f_0 refers to the value of the first incumbent solution and f_1 is the value of the best incumbent solution after all heuristics at

		OCT-H		MARGOT		OCT-H*		HFS-MARGOT*		SFS-MARGOT*	
Dataset	n	$\|F\|$	$\|F_0\|, \|F_1\|, \|F_2\|$	$\|F\|$	$\|F_0\|, \|F_1\|, \|F_2\|$	$\|F\|$	$\|F_0\|, \|F_1\|, \|F_2\|$	$\|F\|$	$\|F_0\|, \|F_1\|, \|F_2\|$	$\|F\|$	$\|F_0\|, \|F_1\|, \|F_2\|$
Breast Cancer Diagnostic	30	3	3, 0, 0	30	30, 30, 0	3	3, 0, 0	4	2, 0, 2	3	3, 0, 0
Breast Cancer Wisconsin	9	4	4, 0, 0	9	9, 8, 0	2	2, 0, 0	4	2, 2, 0	4	3, 1, 0
Climate Model Crashes	18	11	11, 0, 0	18	18, 0, 18	7	7, 0, 0	6	2, 4, 0	4	4, 0, 0
Ionosphere	33	33	32, 28, 22	33	33, 32, 32	3	3, 0, 0	4	2, 0, 2	2	1, 0, 1
Parkinsons	22	13	7, 0, 8	22	22, 22, 22	5	3, 3, 0	8	2, 4, 4	12	9, 2, 4
Heart Disease Cleveland	13	4	4, 0, 0	13	13, 0, 6	3	3, 0, 0	3	1, 2, 2	6	6, 0, 0
Sonar	60	23	23, 0, 0	49	49, 0, 0	15	15, 0, 0	5	1, 2, 2	8	7, 0, 1
SPECTF Heart	44	25	10, 4, 17	44	44, 0, 42	5	5, 0, 0	6	2, 1, 3	10	9, 1, 0
Tic-Tac-Toe	18	18	18, 1, 18	18	18, 0, 0	18	18, 0, 0	5	2, 2, 1	18	18, 0, 0
Wholesale	7	7	6, 0, 5	7	7, 0, 6	2	2, 0, 0	5	1, 2, 2	3	3, 0, 0

Table 9: Comparison on the number of features selected by OCT-H and MARGOT in Table 5 and models in Table 7; F is is the set of features used in the tree and F_t are the features used in the node $t = 0, 1, 2$.



(a) Parkinsons

(b) Breast Cancer Wisconsin

Figure 6: Trees generated by models in Tables 7 on the Parkinsons and Breast Cancer Wisconsin datasets; for each branch node, we report the number of positive and negative training samples in the squared brackets and below the features selected at each node. For each leaf node, the number of positive and negative samples is indicated, together with the assigned class label.

root node of the branch and bound tree have been applied. We compute these values in two different cases. In the first one, no input solution is given to the solver, while in the second, the solver was given the warm start computed with the Local SVM Heuristic. From the results in the tables, we can see how generally the value of the Local SVM solution is better than the one of the first incumbent solution found by Gurobi. Only with the SPECTF Heart dataset, our warm start for HFS-MARGOT and SFS-MARGOT models did not produce a better first solution. Similarly, in the most of the cases, f_1 values are better when the Local SVM solution is injected. Moreover we can notice how, almost everytime the Local SVM input solution was injected, values f_0 and f_1 are equal. Finally, we note that finding all the warm start solutions with the Local SVM Heuristic did not take more than 1 second.

MARGOT	f_0		f_1	
Dataset	No warm start	Local SVM	No warm start	Local SVM
Breast Cancer Diagnostic	412.69	**71.33**	73.31	**71.33**
Breast Cancer Wisconsin	245.32	**70.85**	71.27	**70.85**
Climate Model Crashes	2592000000398.93	**94.14**	113.17	**94.14**
Ionosphere	1680000000000.00	**107.08**	110.87	**107.08**
Parkinsons	306298.02	**196369.04**	213242.43	**196369.04**
Heart Desease Cleveland	29.56	**18.48**	**18.48**	**18.48**
Sonar	11.02	**11.01**	**0.17**	0.35
SPECTF Heart	127800000000.00	**17.47**	**17.47**	**17.47**
Tic-Tac-Toe	4596000000000.00	**84.00**	337.00	**84.00**
Wholesale	349170.61	**104787.61**	**104787.61**	**104787.61**

HFS-MARGOT	f_0		f_1	
Dataset	No warm start	Local SVM	No warm start	Local SVM
Breast Cancer Diagnostic	569527.09	**78042.45**	569527.09	**78042.45**
Breast Cancer Wisconsin	99327.00	**44848.75**	99327.00	**44848.75**
Climate Model Crashes	1480.00	**94.14**	1480.00	**94.14**
Ionosphere	2785244.38	**1491734.49**	2785244.38	**1491734.49**
Parkinsons	5694086.97	**2914676.52**	5694086.97	**2914676.52**
Heart Desease Cleveland	107570.28	**98193.89**	107570.28	**98193.89**
Sonar	1434.07	**990.21**	1434.07	**990.21**
SPECTF Heart	**0.89**	**0.89**	**0.89**	**0.89**
Tic-Tac-Toe	535.30	**469.26**	535.26	**469.26**
Wholesale	7532.43	**7256.24**	7532.43	**7256.24**

SFS-MARGOT	f_0		f_1	
Dataset	No warm start	Local SVM	No warm start	Local SVM
Breast Cancer Diagnostic	125004.07	**7382.54**	24619.24	**7382.54**
Breast Cancer Wisconsin	629.32	**130.04**	230.79	**130.04**
Climate Model Crashes	61588.840	**14800.00**	**14800.00**	**14800.00**
Ionosphere	1120000005651.30	**249.77**	400.00	**249.77**
Parkinsons	376661.06	**194921.43**	344525.19	**194921.43**
Heart Desease Cleveland	414.07	**215.00**	299.28	**215.00**
Sonar	902.33	**170.07**	311.757	**170.07**
SPECTF Heart	**85200000033077.50**	**85200000033077.50**	**8800.01**	**8800.01**
Tic-Tac-Toe	121.81	**95.40**	**53.51**	**53.51**
Wholesale	40023.52	**12177.22**	12438.92	**12177.22**

Table 10: Warm start analysis for MARGOT, HFS-MARGOT and SFS-MARGOT

7. Conclusions and future research

In this paper, we propose a novel MIQP model, MARGOT, to train multivariate optimal classification trees which employ maximum margin hyperplanes by following the soft SVM paradigm. The proposed model presents fewer binary variables and constraints than other OCT approaches by exploiting the SVM approach and the binary classification setting, resulting in a much more compact formulation. The computational experience shows that MARGOT results in a much easier model to solve compared to state-of-the-art OCT models, and, thanks to the statistical properties inherited by the SVM approach, it reaches better predictive performances. In the case sparsity of the hyperplane splits is a desirable requirement, HFS-MARGOT, and SFS-MARGOT, represent two valid interpretable alternatives, which model feature selection with hard budget constraints and soft penalization, respectively. Indeed, rather than inducing global sparsity as in [4], optimizing the local one generates trees that better exploit the structure of the OCT classifier. Both the feature selection versions are comparable to the other OCT approaches in terms of prediction quality but are easier to solve. On the one hand, HFS-MARGOT, thanks to its hard budget constraints, results in a more interpretable model where the selected features are evenly spread among tree branch nodes without losing too much prediction quality. On the other, SFS-MARGOT presents better out-of-sample performances thanks to its soft feature selection approach, at the expense of longer computational times.
Plenty of future directions of this work are of interest. Firstly, the method can be extended to deal with the multi-class case. In addition, being SVMs widely used for regression tasks, a similar version of MARGOT to learn optimal regression trees can be further addressed. Lastly, the development of a tailored optimization algorithm for the resolution of the proposed models can be investigated to improve computational times on real-world instances.

References

[1] Aghaei, S., Azizi, M. J., and Vayanos, P. (2019). Learning optimal and fair decision trees for non-discriminative decision-making. *CoRR*, abs/1903.10598.

[2] Aghaei, S., Gómez, A., and Vayanos, P. (2021). Strong optimal classification trees. *CoRR*, abs/2103.15965.

[3] Bennett, K. P. and Blue, J. A. (1998). A support vector machine approach to decision trees. *1998 IEEE International Joint Conference on Neural Networks Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36227)*, 3:2396–2401 vol.3.

[4] Bertsimas, D. and Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106(7):1039–1082.

[5] Bixby, R. E. (2012). A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, Extra Volume: Optimization Stories(2012):107–121.

[6] Blanco, V., Japón, A., and Puerto, J. (2020). A mathematical programming approach to binary supervised classification with label noise. *CoRR*, abs/2004.10170.

[7] Blanco, V., Japón, A., and Puerto, J. (2022). Robust optimal classification trees under noisy labels. *Advances in Data Analysis and Classification*, 16(1):155–179.

[8] Blanquero, R., Carrizosa, E., Molero-Río, C., and Romero Morales, D. (2020). Sparsity in optimal randomized classification trees. *European Journal of Operational Research*, 284(1):255–272.

[9] Blanquero, R., Carrizosa, E., Molero-Río, C., and Romero Morales, D. (2021). Optimal randomized classification trees. *Computers & Operations Research*, 132:105281.

[10] Bradley, P. and Mangasarian, O. (2000). Massive data discrimination via linear support vector machines. *Optimization methods and software*, 13(1):1–10.

[11] Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

[12] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. (1984). *Classification and Regression Trees.* Chapman and Hall/CRC.

[13] Brodley, C. E. and Utgoff, P. E. (1995). Multivariate decision trees. *Machine Learning*, 19:45–77.

[14] Burges, C. J. and Crisp, D. (1999). Uniqueness of the SVM solution. *Advances in neural information processing systems*, 12.

[15] Carrizosa, E., del Rio, C., and Romero Morales, D. (2021). Mathematical optimization in classification and regression trees. *TOP*, 29(1):5–33. Published online: 17. Marts 2021.

[16] Carrizosa, E., Martín-Barragán, B., and Romero Morales, D. (2011). Detecting relevant variables and interactions in supervised classification. *European Journal of Operational Research*, 213(1):260–269.

[17] Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[18] Chen, T. and Guestrin, C. (2016). XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

[19] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

[20] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[21] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232.

[22] Gambella, C., Ghaddar, B., and Naoum-Sawaya, J. (2021). Optimization problems for machine learning: A survey. *European Journal of Operational Research*, 290(3):807–828.

[23] Günlük, O., Kalagnanam, J., Li, M., Menickelly, M., and Scheinberg, K. (2021). Optimal decision trees for categorical data via integer programming. *Journal of Global Optimization*, 81:233–260.

[24] Ho, T. K. and Kleinberg, E. M. (1996). Building projectable classifiers of arbitrary complexity. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 2, pages 880–885. IEEE.

[25] Hyafil, L. and Rivest, R. L. (1976). Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett.*, 5:15–17.

[26] Jiménez-Cordero, A., Morales, J. M., and Pineda, S. (2021). A novel embedded min-max approach for feature selection in nonlinear support vector machine classification. *European Journal of Operational Research*, 293(1):24–35.

[27] Labbé, M., Martínez-Merino, L. I., and Rodríguez-Chía, A. M. (2019). Mixed integer linear programming for feature selection in support vector machine. *Discrete Applied Mathematics*, 261:276–304. GO X Meeting, Rigi Kaltbad (CH), July 10–14, 2016.

[28] Lee, I. G., Yoon, S. W., and Won, D. (2022). A mixed integer linear programming support vector machine for cost-effective group feature selection: Branch-cut-and-price approach. *European Journal of Operational Research*, 299(3):1055–1068.

[29] Maldonado, S., Perez, J., Weber, R., and Labbé, M. (2014). Feature selection for support vector machines via mixed integer linear programming. *Information Sciences*, 279:163–175.

[30] Murthy, S. K., Kasif, S., and Salzberg, S. L. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32.

[31] Orsenigo, C. and Vercellis, C. (2003). Multivariate classification trees based on minimum features discrete support vector machines. *IMA Journal of Management Mathematics*, 14(3):221–234.

[32] Piccialli, V. and Sciandrone, M. (2018). Nonlinear optimization and support vector machines. *4OR*, 16:111–149.

[33] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

[34] Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[35] Verwer, S. and Zhang, Y. (2017). Learning decision trees with flexible constraints and objectives using integer optimization. In *CPAIOR*.

[36] Verwer, S. and Zhang, Y. (2019). Learning optimal classification trees using a binary linear program formulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):1625–1632.

[37] Wang, L. (2005). Support vector machines: Theory and applications. *Studies in fuzziness and soft computing, v 177*, 302.

[38] Wickramarachchi, D. C., Robertson, B. L., Reale, M., Price, C. J., and Brown, J. (2016). Hhcart: An oblique decision tree. *Computational Statistics & Data Analysis*, 96:12–23.

Appendix A. Additional tables

Here we present additional tables. Table A.11 presents a summary on all the notation of sets, parameters and hyperparameters adopted in the paper. ACC and AUC performances on training samples are reported in tables A.12 and A.13). Finally, for the sake of replicability, we present all the hyperparameters that were chosen to carry out our computational experiments in tables A.14 and A.15.

Notation	Description
Sets	
$\mathcal{T}_B$	Branch nodes
$\mathcal{T}_L$	Leaf nodes
$\mathcal{T}'_B$	Branch nodes excluded the ones in the last branching level
$\mathcal{T}''_B$	Branch nodes of the last branching level
$\mathcal{S}(t)$	Branch nodes of the subtree rooted at node $t \in \mathcal{T}_B$
$\mathcal{S}''(t)$	Nodes of $\mathcal{S}(t)$ in the last branching level $\mathcal{T}''_B$
$\mathcal{S}''_L(t)$	Nodes in $\mathcal{T}''_B$ under the left branch of $t \in \mathcal{T}'_B$
$\mathcal{S}''_R(t)$	Nodes in $\mathcal{T}''_B$ under the right branch of $t \in \mathcal{T}'_B$
$\mathcal{I}$	Index set of data samples
$\mathcal{I}_t$	Index set of data samples assigned to node $t \in \mathcal{T}_B$
$\mathcal{I}_{L(t)}$	Index set of data samples assigned to the left child node of $t \in \mathcal{T}_B$
$\mathcal{I}_{R(t)}$	Index set of data samples assigned to the right child node of $t \in \mathcal{T}_B$
Parameters	
n	Number of features
ε	Parameter to model the strict inequality in routing constraints
$\{M_w, M_\xi, M_\mathcal{H}\}$	Set of Big-M parameters used MARGOT formulations
Hyperparameters	
D	Maximum depth of the tree
C_t	Penalty parameter on the misclassification error at node $t \in \mathcal{T}_B$
B_t	Budget value on the number of features at node $t \in \mathcal{T}_B$
α	Penalty parameter for the soft feature selection

Table A.11: Notation: sets, parameters and hyperparameters.

	OCT-H		MM-SVM-OCT		MARGOT	
Dataset	ACC	AUC	ACC	AUC	ACC	AUC
Breast Cancer Diagnostic	98.2	97.8	98.7	98.2	**99.3**	**99.1**
Breast Cancer Wisconsin	96.9	96.8	96.7	96.6	**97.5**	**97.4**
Climate Model Crashes	98.6	91.9	**99.3**	**95.9**	**99.3**	**95.9**
Ionosphere	**100.0**	**100.0**	98.9	98.7	96.4	95.2
Parkinsons	99.4	98.7	96.8	93.4	**100.0**	**100.0**
Heart Disease Cleveland	85.7	85.6	88.6	88.5	**89.0**	**88.9**
Sonar	98.8	98.8	**100.0**	**100.0**	97.0	96.8
SPECTF Heart	**98.6**	**96.6**	79.3	50.0	96.7	92.0
Tic-Tac-Toe	**99.5**	**99.2**	99.3	99.1	98.4	97.7
Wholesale	96.0	**95.5**	83.5	75.0	**96.3**	95.2

Table A.12: Results on the predictive performances of the OCT models evaluated: train ACC (%) and train AUC (%).

	OCT-H*		HFS-MARGOT*		SFS-MARGOT*	
Dataset	ACC	AUC	ACC	AUC	ACC	AUC
Breast Cancer Diagnostic	**98.2**	**97.8**	98.0	97.4	**98.2**	**97.8**
Breast Cancer Wisconsin	95.0	94.9	**96.4**	96.2	**96.4**	**96.4**
Climate Model Crashes	**98.1**	**91.6**	95.8	76.9	96.8	84.8
Ionosphere	91.4	88.4	**92.1**	**89.9**	90.0	86.4
Parkinsons	96.2	94.8	99.4	99.6	**100.0**	**100.0**
Heart Disease Cleveland	85.7	85.4	82.7	82.1	**86.5**	**86.1**
Sonar	**97.0**	**96.8**	78.9	78.2	91.0	90.3
SPECTF Heart	88.3	73.3	85.0	74.6	**91.1**	**80.9**
Tic-Tac-Toe	**99.1**	**98.7**	75.5	72.1	98.6	97.9
Wholesale	91.5	89.1	93.8	**94.7**	**95.2**	94.6

Table A.13: Results on the predictive performances of the OCT models resulting from the feature selection driven tuning of the hyperparameters: train ACC (%) and train AUC (%).

	OCT-H	MM-SVM-OCT		MARGOT	
Dataset	α	c_1	c_3	C_0	$C_1 = C_2$
Breast Cancer Diagnostic	2^{-5}	10^{4}	10^{0}	10^{0}	10^{0}
Breast Cancer Wisconsin	2^{-7}	10^{4}	10^{-1}	10^{0}	10^{0}
Climate Model Crashes	2^{-6}	10^{2}	10^{-2}	10^{0}	10^{0}
Ionosphere	0	10^{1}	10^{0}	10^{0}	10^{0}
Parkinsons	2^{-8}	10^{3}	10^{1}	10^{0}	10^{4}
Heart Disease Cleveland	2^{-5}	10^{1}	10^{-2}	10^{-1}	10^{-1}
Sonar	2^{-7}	10^{0}	10^{0}	10^{-3}	10^{-1}
SPECTF Heart	2^{-6}	10^{-5}	10^{-2}	10^{-1}	10^{-1}
Tic-Tac-Toe	0	10^{4}	10^{1}	10^{0}	10^{0}
Wholesale	2^{-7}	10^{3}	10^{-1}	10^{3}	10^{3}

Table A.14: Hyperparameters selected for results in Table 5 and Table 6.

	OCT-H*		HFS-MARGOT*				SFS-MARGOT*	
Dataset	α	C_0	$C_1 = C_2$	B_0	$B_1 = B_2$	C_0	$C_1 = C_2$	α
Breast Cancer Diagnostic	2^{-5}	10^{3}	10^{3}	2	2	10^{2}	10^{2}	2^{10}
Breast Cancer Wisconsin	2^{-4}	10^{2}	10^{3}	2	2	10^{0}	10^{0}	2^{4}
Climate Model Crashes	2^{-4}	10^{1}	10^{1}	2	4	10^{2}	10^{2}	2^{10}
Ionosphere	2^{-5}	10^{4}	10^{4}	2	2	10^{0}	10^{0}	2^{8}
Parkinsons	2^{-5}	10^{4}	10^{5}	2	4	10^{2}	10^{4}	2^{10}
Heart Disease Cleveland	2^{-4}	10^{1}	10^{3}	1	2	10^{0}	10^{0}	2^{2}
Sonar	2^{-7}	10^{-4}	10^{1}	1	2	10^{0}	10^{0}	2^{2}
SPECTF Heart	2^{-5}	10^{-4}	10^{-2}	2	3	10^{-4}	10^{2}	2^{8}
Tic-Tac-Toe	2^{-8}	10^{-2}	10^{0}	2	3	10^{-1}	10^{-1}	2^{0}
Wholesale	2^{-6}	10^{-2}	10^{2}	1	2	10^{2}	10^{2}	2^{8}

Table A.15: Hyperparameters selected for results in Table 7 and Table A.13.