

MIXED-INTEGER LINEAR OPTIMIZATION FOR SEMI-SUPERVISED OPTIMAL CLASSIFICATION TREES

JAN PABLO BURGARD, MARIA EDUARDA PINHEIRO, MARTIN SCHMIDT

ABSTRACT. Decision trees are one of the most famous methods for solving classification problems, mainly because of their good interpretability properties. Moreover, due to advances in recent years in mixed-integer optimization, several models have been proposed to formulate the problem of computing optimal classification trees. The goal is, given a set of labeled points, to split the feature space with hyperplanes and assign a class to each partition. In certain scenarios, however, labels are exclusively accessible for a subset of the given points. Additionally, this subset may be non-representative, such as in the case of self-selection in a survey. Semi-supervised decision trees tackle the setting of labeled and unlabeled data and often contribute to enhancing the reliability of the results. Furthermore, undisclosed sources may provide extra information about the size of the classes. We propose a mixed-integer linear optimization model for computing semi-supervised optimal classification trees that cover the setting of labeled and unlabeled data points as well as the overall number of points in each class for a binary classification. Our numerical results show that our approach leads to a better accuracy and a better Matthews correlation coefficient for biased samples compared to other optimal classification trees, even if only few labeled points are available.

1. INTRODUCTION

Decision trees are among the most popular approaches for supervised classification (Breiman et al. 1984; Quinlan 1986). One of the main reasons for this is that they are easy to interpret compared to other machine learning models. The core idea is to recursively partition the feature space, according to branching rules, and assign a label to each resulting part of the partition.

One way to partition the data is to use hyperplanes involving a single feature, which then leads to so-called univariate trees; see, e.g., Kotsiantis (2014) and Yildiz and Dikmen (2007). In a multivariate tree, these hyperplanes involve more than one feature and some approaches for this setting are given in Altincay (2007), Bennett and Blue (1996), and Orsenigo and Vercellis (2003). In many algorithms for univariate or multivariate trees, each separate hyperplane is generated by minimizing a local impurity function, i.e., they do not build the tree by solving just a single optimization problem.

In recent years, due to the advancement of algorithms for mixed-integer programming (MIP), many strategies for computing optimal classification trees (OCT) by globally solving an optimization problem using MIP techniques have been proposed. Some techniques are discussed in the recent surveys by Carrizosa et al. (2021) and Gambella et al. (2021). The first approaches were proposed by Bertsimas and Dunn (2017). They present two mixed-integer linear programming (MILP) models based on univariate and multivariate trees. Verwer and Zhang (2019) propose a binary

Date: January 17, 2024.

2020 Mathematics Subject Classification. 90C11, 90C90, 90-08, 68T99.

Key words and phrases. Semi-supervised learning, Optimal classification trees, Mixed-integer linear optimization.

linear formulation in which the problem size is largely independent of the size of the training data. MIP approaches that consider support vector machines (Cortes and Vapnik 1995) to split the tree also have been explored, as can be seen in Blanco et al. (2022a), Blanco et al. (2022b), and D’Onofrio et al. (2023).

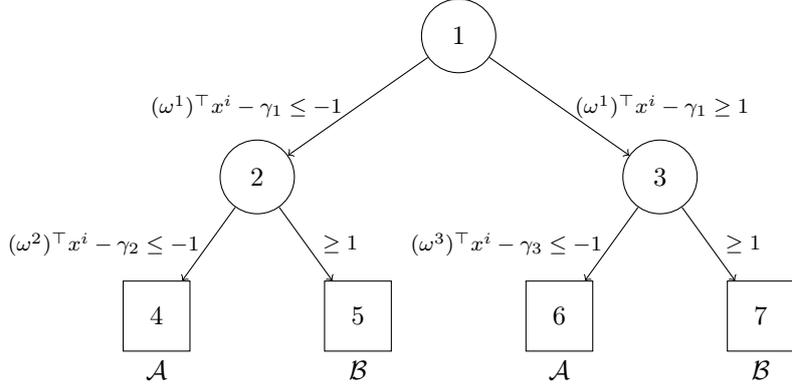
Besides the MIP models to solve an OCT, Blanquero et al. (2021) proposed a nonlinear optimization approach to compute an optimal “randomized” classification tree. In their approach, each data point is assigned to a class only with a given probability. The model uses oblique cuts with the goal of utilizing fewer predictor variables in the splits of the tree. Furthermore, Blanquero et al. (2019) aggregate local and global sparsity in the randomized tree by means of regularization with polyhedral norms.

All the strategies presented so far exclusively focus on labeled data. However, acquiring labels for every unit of interest can be expensive—in particular if classic surveys are used to obtain the labels. In this case, it would be beneficial to train the decision tree on only partly labeled data. This yields a semi-supervised learning setting (Zhu and Goldberg 2009). Algorithms for semi-supervised learning have already been proposed for SVM (Chapelle et al. 2006; Melacci and Belkin 2009), neural networks (Lee 2013; Nguyen et al. 2023; Oliver et al. 2018), and logistic regression (Amini and Gallinari 2002; Bzdok et al. 2015).

In the field of semi-supervised decision trees, Kim (2016) splits internal nodes by utilizing the structural characteristics of the data for subspace partition. Moreover, Kocev et al. (2017) consider minimizing a local impurity function and Tanha et al. (2017) propose self-training as base learners. Recently, Santhiappan and Ravindran (2021) consider a maximum mean discrepancy to estimate the class ratio at every splitting rule in a univariate decision tree. Furthermore, Zharmagambetov and Carreira-Perpinan (2022) present a graph Laplacian approach to deal with the unlabeled data. Hence, although they present different ideas, none of the approaches considers globally solving a single optimization problem.

Moreover, in many cases, external sources provide information about the total amount of elements in each class within a population. For example, in some businesses, the number of positive labels might be available, but the identification of which customer has a positive or negative label is unknown. An intuitive example is a supermarket for which the amount of cash payments is known. However, this information is not attributable to the individual customers ex-post. Another example is population surveys, where statistics agencies can provide how many people are employed. For logistic regression, the idea of aggregating this extra information is proposed by Burgard et al. (2021), who develop a cardinality-constrained multinomial logit model. In the SVM setting, Burgard et al. (2023) present a mixed-integer quadratic optimization model and iterative clustering techniques to tackle cardinality constraints for each class. Our contribution here is to propose to add this aggregated additional information to a multivariate OCT model by imposing a cardinality constraint on the predicted labels for the unlabeled data.

We develop a big- M -based MILP to solve the semi-supervised optimal classification tree (S²OCT) problem that deals with the cardinality constraint for the unlabeled data. The cardinality constraint helps to account for biased samples since the number of predictions in each class on the population is bounded by the constraint. This paper is organized as follows. In Section 2 we present preliminary concepts and our optimization problem. Afterward, the big- M -based MILP formulation is presented in Section 3. In Section 4, numerical results are reported and discussed and we conclude in Section 5.

FIGURE 1. A classification tree with depth $D = 2$

2. PRELIMINARY CONCEPTS

Let $X \in \mathbb{R}^{p \times N} = [X_l, X_u]$ be the data matrix with labeled data $X_l = [x^1, \dots, x^n]$ and unlabeled data $X_u = [x^{n+1}, \dots, x^N]$. Hence, we observe $x^i \in \mathbb{R}^p$ for all $i \in [1, N] := \{1, \dots, N\}$. We set $m := N - n$, $\mathcal{U} := [n + 1, \dots, N]$, and $c \in \{\mathcal{A}, \mathcal{B}\}^n$ as the vector of class labels for the labeled data.

In a multivariate optimal classification tree, each decision consists of a linear combination of the p components of a point x^i . Considering only the labeled data, the goal is to split the feature space into distinct regions to correctly classify each point. However, in many applications, aggregated information on the labels is available, e.g., from census data. In the following, we know the total number $\lambda \in \mathbb{N}$ of unlabeled points that belong to the class \mathcal{A} and adapt the idea of optimal classification trees such that we can use the unlabeled data as well as λ as additional information.

Given a depth $D \in \mathbb{N}$, a classification tree has $2^{D+1} - 1$ nodes, categorized in two types; the branch nodes $\mathcal{T}_B = [1, 2^D - 1]$ and the leaf nodes $\mathcal{T}_L = [2^D, 2^{D+1} - 1]$. Each branch node $b \in \mathcal{T}_B$ provides a hyperplane parameterized by (ω^b, γ_b) that splits the feature space into half-spaces. As suggested by Bertsimas and Dunn (2017), if a point x^i , $i \in [1, N]$, satisfies $(\omega^b)^\top x^i - \gamma_b \leq 0$, then x^i follows the left branch of the node b . If $(\omega^b)^\top x^i - \gamma_b > 0$ holds, the point x^i follows the right branch of the node b . For an optimization formulation, the strict inequality needs to be re-written as $(\omega^b)^\top x^i - \gamma_b \geq \varepsilon$ for a sufficiently small $\varepsilon > 0$. However, a very small value of ε might lead to numerical instabilities. To avoid this, we replace $(\omega^b)^\top x^i - \gamma_b \leq 0$ by $(\omega^b)^\top x^i - \gamma_b \leq -1$, and $(\omega^b)^\top x^i - \gamma_b \geq \varepsilon$ by $(\omega^b)^\top x^i - \gamma_b \geq 1$. Note that this change is always possible except for the rare cases that a data point actually lies on a hyperplane.

In each leaf node, $t \in \mathcal{T}_L$, all points x^i , $i \in [1, N]$, are classified as \mathcal{A} or \mathcal{B} . In a simple example with $D = 2$, the classification tree has $\mathcal{T}_B = [1, 3]$ and $\mathcal{T}_L = [4, 7]$; see Figure 1. Regarding the classification at the leaf nodes, let $\mathcal{T}_L^{\mathcal{A}} = \{t \in \mathcal{T}_L : t \text{ is even}\}$ be the set of leaf nodes that are classified as \mathcal{A} and $\mathcal{T}_L^{\mathcal{B}} = \{t \in \mathcal{T}_L : t \text{ is odd}\}$ be the set of leaf nodes that are classified as \mathcal{B} . For the classification tree with $D = 2$, see Figure 1 again, $\mathcal{T}_L^{\mathcal{A}} = \{4, 6\}$ and $\mathcal{T}_L^{\mathcal{B}} = \{5, 7\}$ holds.

For each leaf node $t \in \mathcal{T}_L$, define $\mathcal{N}_R(t)$ as the index set of the branch nodes in which the right or “greater than” branch is traversed to reach leaf t . Moreover, we define $\mathcal{N}_L(t)$ as the index set of the branch nodes in which the left or “less than”

branch is traversed to reach leaf t . For the classification tree in Figure 1 we have

$$\begin{aligned}\mathcal{N}_L(4) &= \{1, 2\}, & \mathcal{N}_R(4) &= \emptyset, & \mathcal{N}_L(5) &= \{1\}, & \mathcal{N}_R(5) &= \{2\}, \\ \mathcal{N}_L(6) &= \{3\}, & \mathcal{N}_R(6) &= \{1\}, & \mathcal{N}_L(7) &= \emptyset, & \mathcal{N}_R(7) &= \{1, 3\}.\end{aligned}$$

Given a fixed depth D , a point x^i , $i \in [1, n]$, with $c_i \in \{\mathcal{A}, \mathcal{B}\}$, is correctly classified if all inequalities from the root to the leaf node are satisfied for some leaf node $t \in \mathcal{T}_L^{c_i}$. Hence, a point x^i is correctly classified if

$$\bigvee_{t \in \mathcal{T}_L^{c_i}} \left\{ \left\{ \bigwedge_{b \in \mathcal{N}_R(t)} [(\omega^b)^\top x^i - \gamma_b \geq 1] \right\} \wedge \left\{ \bigwedge_{b \in \mathcal{N}_L(t)} [(\omega^b)^\top x^i - \gamma_b \leq -1] \right\} \right\}. \quad (1)$$

is satisfied.

In our running example with $D = 2$, a point x^i with $c_i = \mathcal{A}$ is correctly classified if

$$\begin{aligned}& \left\{ [(\omega^1)^\top x^i - \gamma_1 \leq -1] \wedge [(\omega^2)^\top x^i - \gamma_2 \leq -1] \right\} \\ & \vee \left\{ [(\omega^1)^\top x^i - \gamma_1 \geq 1] \wedge [(\omega^3)^\top x^i - \gamma_3 \leq -1] \right\}\end{aligned}$$

holds.

An unlabeled point x^i , $i \in [n+1, N]$, is classified as \mathcal{A} if

$$\bigvee_{t \in \mathcal{T}_L^{\mathcal{A}}} \left\{ \left\{ \bigwedge_{b \in \mathcal{N}_R(t)} [(\omega^b)^\top x^i - \gamma_b \geq 1] \right\} \wedge \left\{ \bigwedge_{b \in \mathcal{N}_L(t)} [(\omega^b)^\top x^i - \gamma_b \leq -1] \right\} \right\} \quad (2)$$

is true. Thus, our goal is to find ω and γ such that Expression (1) is satisfied for all labeled data points x^i , $i \in [1, n]$, and such that the number of unlabeled points x^i , $i \in [n+1, N]$, that satisfy Expression (2) is as close as possible to λ .

For doing so, we first need to define suitable error measures that then have to be minimized. For this purpose, we define the branch and leaf error according to the decision error and leaf error proposed by Bennett and Blue (1996). The first error is related to branch nodes. For each labeled point, at each branch node, we consider the inequality that must be satisfied and then measure by how much it is violated.

Definition 1 (Branch Error). *Given a labeled point x^i , $i \in [1, n]$, in any branch node $b \in \mathcal{T}_B$, the branch errors $(y_b^R)_i$ and $(y_b^L)_i$ are defined by*

$$(y_b^R)_i := \left[-(\omega^b)^\top x^i + \gamma_b + 1 \right]^+ \quad \text{and} \quad (y_b^L)_i := \left[(\omega^b)^\top x^i - \gamma_b + 1 \right]^+$$

with $[v]^+ := \max\{0, v\}$.

The definition above can be interpreted as follows. If the point x^i satisfies $(\omega^b)^\top x^i - \gamma_b \geq 1$, and therefore follows the right branch in some node $b \in \mathcal{T}_B$, it holds $(y_b^R)_i = 0$ and $(y_b^L)_i > 0$. However, if the point follows the left branch in some node $b \in \mathcal{T}_B$, i.e., if $(\omega^b)^\top x^i - \gamma_b \leq -1$ holds, we obtain $(y_b^R)_i > 0$ and $(y_b^L)_i = 0$.

The next definition represents the error in each leaf node t . For each labeled point, we sum over the branch errors along the path from the root to the leaf node t .

Definition 2 (Leaf Error). *The leaf error of a labeled point x^i , $i \in [1, n]$, at a leaf node $t \in \mathcal{T}_L$ is defined by*

$$LE(x^i, t) := \sum_{b \in \mathcal{N}_R(t)} (y_b^R)_i + \sum_{b \in \mathcal{N}_L(t)} (y_b^L)_i. \quad (3)$$

Note that $\text{LE}(x^i, t)$ is a linear expression and for each labeled point x^i , $i \in [1, n]$, Expression (1) is satisfied if $\text{LE}(x^i, t) = 0$ for some $t \in \mathcal{T}_L^{c_i}$. Additionally, $\text{LE}(x^i, t) \geq 0$ holds for all $t \in \mathcal{T}_L$. Hence, each labeled point x^i is correctly classified if the minimum value of all leaf errors is zero, i.e., if

$$\min_{t \in c_i} \{\text{LE}(x^i, t)\} = 0$$

holds. Besides that, we want to classify λ unlabeled points as \mathcal{A} , which means λ unlabeled points must end up in some $t \in \mathcal{T}_L^{\mathcal{A}}$.

To sum up, given the data matrix $X \in \mathbb{R}^{p \times N}$ and $\lambda \in \mathbb{N}$ as well as some $s > 0$, our goal is to find optimal parameters $\omega \in \mathbb{R}^{p \times 2^D - 1}$, $\gamma \in \mathbb{R}^{2^D - 1}$ as well as $y^R, y^L \in \mathbb{R}^{2^D - 1 \times n}$ and $\xi \in \mathbb{R}$ that solve the optimization problem

$$\min_{\omega, \gamma, y^R, y^L, \xi} \sum_{i=1}^n \min_{t \in \mathcal{T}_L^{c_i}} \{\text{LE}(x^i, t)\} + C\xi \quad (\text{P1a})$$

$$\text{s.t. } (y_b^R)_i \geq -(\omega^b)^\top x^i + \gamma_b + 1, \quad b \in \mathcal{T}_B, \quad i \in [1, n], \quad (\text{P1b})$$

$$(y_b^L)_i \geq (\omega^b)^\top x^i - \gamma_b + 1, \quad b \in \mathcal{T}_B, \quad i \in [1, n], \quad (\text{P1c})$$

$$-s \leq \omega_j^b \leq s, \quad b \in \mathcal{T}_B, \quad j \in [1, p], \quad (\text{P1d})$$

$$(y_b^R)_i, (y_b^L)_i \geq 0, \quad b \in \mathcal{T}_B, \quad i \in [1, n], \quad (\text{P1e})$$

$$\lambda - \xi \leq \sum_{i=n+1}^N \sum_{t \in \mathcal{T}_L^{\mathcal{A}}} (\psi(x^i, t)) \leq \lambda + \xi, \quad (\text{P1f})$$

$$\xi \geq 0 \quad (\text{P1g})$$

where $\text{LE}(x^i, t)$ is defined in (3) and

$$\psi(x^i, t) = \begin{cases} 1, & \text{if } x^i \text{ ends in the leaf node } t, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Note that the objective function in (P1a) models a compromise between minimizing the classification error for the labeled and unlabeled data. The penalty parameter $C > 0$ aims to control the importance of the slack variable ξ . Constraints (P1b) and (P1c) enforce on which branch (left or right) the labeled point x^i should traverse in branch node b . Constraint (P1d) defines the domain of each variable ω^b . This constraint is not necessary for the correctness of the model but will serve as a big- M -type parameter that is later used for deriving certain bounds; see Section 3. Constraint (P1f) ensures that the number of unlabeled data classified as \mathcal{A} is as close to λ as possible.

The functions $\min\{\text{LE}(x^i, t) : t \in \mathcal{T}_L^{c_i}\}$ in the objective function (P1a) and $\psi(x^i, t)$ in Constraint (P1g) are discontinuous, which means that Problem (P1) cannot be solved easily by standard solvers as such. Hence, we will present a mixed-integer linear programming (MILP) formulation using binary variables to re-model the objective function (P1a) and to count the classification of unlabeled points.

3. THE MILP MODEL

We start the development of a MILP model by using classic SOS1-techniques (Beale and Tomlin 1969) and McCormick inequalities (McCormick 1976) to rephrase a min min problem as an MILP formulation.

Lemma 1. Consider a set of continuous functions $f_k : \mathbb{R}^p \rightarrow \mathbb{R}$, $k \in [1, d]$, for some $d \in \mathbb{N}$, and let $\Omega \subseteq \mathbb{R}^p$ be given. Suppose further that there exist values $u_k > 0$ such that

$$0 \leq f_k(x) \leq u_k$$

holds for all $x \in \Omega$ and $k \in [1, d]$. Then, $x^* \in \mathbb{R}^p$ is a solution to the problem

$$\min_x \min_{k \in [1, d]} f_k(x) \quad (5a)$$

$$\text{s.t. } x \in \Omega \quad (5b)$$

if and only if there exist $\alpha^*, \beta^* \in \mathbb{R}^d$ such that (x^*, α^*, β^*) is a solution to the problem

$$\min_{x, \alpha, \beta} \sum_{k=1}^d \beta_k \quad (6a)$$

$$\text{s.t. } x \in \Omega, \quad (6b)$$

$$\sum_{k=1}^d \alpha_k = 1, \quad (6c)$$

$$\alpha_k \in \{0, 1\}, \quad k \in [1, d], \quad (6d)$$

$$\beta_k \leq u_k \alpha_k, \quad k \in [1, d], \quad (6e)$$

$$\beta_k \leq f_k(x), \quad k \in [1, d], \quad x \in \Omega, \quad (6f)$$

$$\beta_k \geq f_k(x) - u_k(1 - \alpha_k), \quad k \in [1, d], \quad x \in \Omega, \quad (6g)$$

$$\beta_k \geq 0, \quad k \in [1, d]. \quad (6h)$$

Proof. By introducing the binary variables α_k , we obtain that x^* is a solution to Problem (5) if and only if (x^*, α^*) is a solution of the problem

$$\min_{x, \alpha} \sum_{k=1}^d \alpha_k f_k(x) \\ \text{s.t. } (6b)-(6d).$$

Besides that, for any (x^*, α^*) solution of the problem above, if $\alpha_k^* = 0$ holds for some $k \in [1, d]$, Constraints (6e) and (6h) enforce that $\beta_k^* = 0 = \alpha_k^* f_k(x^*)$. On the other hand, if $\alpha_k^* = 1$ holds, by Constraints (6f) and (6g), we obtain $\beta_k^* = \alpha_k^* f_k(x^*)$.

Hence, x^* is a solution to Problem (5) if and only if (x^*, α^*, β^*) exists such that

$$\beta_k^* = \alpha_k^* f_k(x^*), \quad k \in [1, d],$$

is a solution to Problem (6). \square

To apply the previous lemma to Problem (P1) it is necessary that $LE(x^i, t)$ has lower and upper bounds, which are given in following proposition. Here and in what follows, $\|\cdot\|$ denotes the Euclidean norm.

Proposition 1. Given $[x^1, \dots, x^N]$ with $x^i \in \mathbb{R}^p$ for all $i \in [1, N]$ and $s > 0$, every optimal solution $(\omega, \gamma, y^R, y^L, \xi)$ to Problem (P1) satisfies

$$|(\omega^b)^\top x^i - \gamma_b| \leq \eta s \sqrt{p}, \quad i \in [1, N], \quad b \in \mathcal{T}_B,$$

and

$$(y_b^R)_i \leq \eta s \sqrt{p} + 1, \quad (y_b^L)_i \leq \eta s \sqrt{p} + 1, \quad i \in [1, n], \quad b \in \mathcal{T}_B,$$

where $\eta := \max_{i, k \in [1, N]} \|x^i - x^k\|$. Moreover,

$$0 \leq LE(x^i, t) \leq B(s), \quad t \in i \in [1, n], \quad t \in \mathcal{T}_L,$$

holds with

$$B(s) := D(\eta s \sqrt{p} + 1). \quad (7)$$

Proof. Let \mathcal{H} be the set of hyperplanes that separates $[x^1, \dots, x^N]$ into two sets, and let the hyperplane $(\omega^b, \gamma_b) \in \mathcal{H}$ be the hyperplane with the minimal maximum distance to any point in X . Then, according to Blanco et al. (2022a), the maximum distance from a point $x^i, i \in [1, N]$, to (ω^b, γ_b) is η , i.e.,

$$\frac{|(\omega^b)^\top x^i - \gamma_b|}{\|\omega^b\|} \leq \eta.$$

Moreover, Constraint (P1d) enforces that $\|\omega^b\| \leq s\sqrt{p}$. Hence,

$$|(\omega^b)^\top x^i - \gamma_b| \leq \eta s\sqrt{p}, \quad i \in [1, N], \quad b \in \mathcal{T}_B,$$

holds and

$$-(\omega^b)^\top x^i + \gamma_b + 1 \leq \eta s\sqrt{p} + 1 \quad \text{as well as} \quad (\omega^b)^\top x^i - \gamma_b + 1 \leq \eta s\sqrt{p} + 1$$

are satisfied for all $b \in \mathcal{T}_B$ and $i \in [1, n]$.

Note further that Problem (P1) is a minimization problem and $\text{LE}(x^i, t)$ is a sum of the D non-negatives terms y_b^R and y_b^L . Thus, Constraints (P1b) and (P1c) imply that

$$(y_b^R)_i \leq \eta s\sqrt{p} + 1, \quad (y_b^L)_i \leq \eta s\sqrt{p} + 1, \quad i \in [1, n], \quad b \in \mathcal{T}_B,$$

and

$$0 \leq \text{LE}(x^i, t) \leq B(s), \quad t \in \mathcal{T}_L,$$

holds. \square

Note that Constraint (P1d) is decisive for obtaining the upper bound $B(s)$ in the previous lemma. Finally, to overcome the discontinuity of the function $\psi(\cdot)$ defined in (4), we also add binary variables and use SOS techniques again to turn on or off the enforcement of a constraint. More formal, by introducing the matrices $\beta \in \mathbb{R}^{n \times 2^{D-1}}$, $\alpha \in \{0, 1\}^{n \times 2^{D-1}}$, $z \in \{0, 1\}^{m \times 2^{D-1}}$, and $\delta \in \{0, 1\}^{m \times 2^{D-1}}$ of binary variables, we can reformulate the optimization problem (P1) as follows:

$$\min_{\omega, \gamma, y^R, y^L, \xi, \alpha, \beta, z, \delta} \sum_{i=1}^n \sum_{t \in \mathcal{T}_L^{c_i}} \beta_t^i + C\xi \quad (\text{P2a})$$

$$\text{s.t.} \quad (\text{P1b})\text{--}(\text{P1e}),$$

$$\sum_{t \in \mathcal{T}_L^{c_i}} \alpha_t^i = 1, \quad i \in [1, n], \quad (\text{P2b})$$

$$\alpha_t^i \in \{0, 1\}, \quad i \in [1, n], \quad t \in \mathcal{T}_L^{c_i}, \quad (\text{P2c})$$

$$\beta_t^i \leq \text{LE}(x^i, t), \quad i \in [1, n], \quad t \in \mathcal{T}_L^{c_i}, \quad (\text{P2d})$$

$$\beta_t^i \geq \text{LE}(x^i, t) - B(s)(1 - \alpha_t^i), \quad i \in [1, n], \quad t \in \mathcal{T}_L^{c_i}, \quad (\text{P2e})$$

$$0 \leq \beta_t^i \leq B(s)\alpha_t^i, \quad i \in [1, n], \quad t \in \mathcal{T}_L, \quad (\text{P2f})$$

$$(\omega^b)^\top x^i - \gamma_b \leq z_i^b M - 1, \quad b \in \mathcal{T}_B, \quad i \in \mathcal{U}, \quad (\text{P2g})$$

$$(\omega^b)^\top x^i - \gamma_b \geq -(1 - z_i^b)M + 1, \quad b \in \mathcal{T}_B, \quad i \in \mathcal{U}, \quad (\text{P2h})$$

$$z_i^b \in \{0, 1\}, \quad b \in \mathcal{T}_B, \quad i \in \mathcal{U}, \quad (\text{P2i})$$

$$\delta_i^t \leq z_i^b, \quad b \in \mathcal{N}_R(t), \quad t \in \mathcal{T}_L^A, \quad i \in \mathcal{U}, \quad (\text{P2j})$$

$$\delta_i^t \leq -z_i^b + 1, \quad b \in \mathcal{N}_L(t), \quad t \in \mathcal{T}_L^A, \quad i \in \mathcal{U}, \quad (\text{P2k})$$

$$\delta_i^t \geq \sum_{b \in \mathcal{N}_R(t)} z_i^b + \sum_{b \in \mathcal{N}_L(t)} (-z_i^b + 1) - (D - 1), \quad t \in \mathcal{T}_L^A, \quad i \in \mathcal{U}, \quad (\text{P2l})$$

$$\delta_i^t \in \{0, 1\}, \quad t \in \mathcal{T}_L^A, \quad i \in \mathcal{U}, \quad (\text{P2m})$$

$$\lambda - \xi \leq \sum_{i=1+n}^N \sum_{t \in \mathcal{T}_L^A} \delta_i^t \leq \lambda + \xi, \quad (\text{P2n})$$

$$\xi \geq 0, \quad (\text{P2o})$$

where M needs to be chosen sufficiently large. The constraints in (P2c) enforce that the minimum value of $\text{LE}(x^i, t)$ is selected for each $x^i \in [1, n]$ and Constraints (P2d)–(P2f) ensure

$$\beta_t^i = \alpha_t^i \text{LE}(x^i, t), \quad i \in [1, n], \quad t \in \mathcal{T}_L.$$

As z_i^b is binary, Constraints (P2g) and (P2h) lead to

$$\begin{aligned} (\omega^b)^\top x^i - \gamma_b \geq 1 &\implies z_i^b = 1, \quad b \in \mathcal{T}_B, \quad i \in \mathcal{U}, \\ (\omega^b)^\top x^i - \gamma_b \leq -1 &\implies z_i^b = 0, \quad b \in \mathcal{T}_B, \quad i \in \mathcal{U}. \end{aligned}$$

Furthermore, as δ_i^t is binary as well, for all $i \in \mathcal{U}$ and $t \in \mathcal{T}_L^A$, Constraints (P2j)–(P2l) lead to

$$\delta_i^t = \begin{cases} 1, & \text{if } z_i^b = 1 \text{ for } b \in \mathcal{N}_R(t) \text{ and } z_i^b = 0 \text{ for } b \in \mathcal{N}_L(t), \\ 0, & \text{otherwise,} \end{cases}$$

i.e.,

$$\delta_i^t = \begin{cases} 1, & \text{if } x^i \text{ ends up in the leaf node } t, \\ 0, & \text{otherwise.} \end{cases}$$

Constraint (P2n) ensures that the number of unlabeled data classified as \mathcal{A} is as close to λ as possible. Reformulation (P2) is an MILP. We refer to this problem as S²OCT. As usual in mixed-integer optimization, the big- M -value is crucial as is the choice of s in Constraint (P1d). However, precisely based on s we have an exact value for M , as discussed in the following theorem.

Theorem 2. Consider $\eta := \max_{i,k \in [1, N]} \|x^i - x^k\|$. Any feasible point for Problem (P2) satisfies (P2g) and (P2h) for $M = \eta s \sqrt{p} + 1$.

Proof. Note that if $z_i^b = 1$ for some $i \in \mathcal{U}$ and $b \in \mathcal{T}_B$, Constraints (P2g) and (P2h) imply

$$M - 1 \geq (\omega^b)^\top x^i - \gamma_b \geq 1.$$

Moreover, since $(\omega^b)^\top x^i - \gamma_b \geq 0$, because of Proposition 1, we get

$$(\omega^b)^\top x^i - \gamma_b \leq \eta s \sqrt{p}.$$

This means that $M = \eta s \sqrt{p} + 1$ does not cut off any feasible solution.

On the other hand, if $z_i^b = 0$ holds for some $i \in \mathcal{U}$ and $b \in \mathcal{T}_B$, due to Constraint (P2g) and (P2h), we obtain

$$1 - M \leq (\omega^b)^\top x^i - \gamma_b \leq -1,$$

and, similarly, $M = \eta s \sqrt{p} + 1$ does not cut off any feasible solution as well. \square

4. NUMERICAL RESULTS

In this section, we present and discuss our computational results that exemplify the advantages of considering the known total amount of each class. We analyze this on different test sets from the literature. The test sets are discussed in Section 4.1, while the computational setup is described in Section 4.2. The evaluation criteria are depicted in Section 4.3. Finally, the numerical results are discussed in Section 4.4.

4.1. Tests Sets. For the computational analysis of the proposed approach, we consider the subset of instances presented by Olson et al. (2017) that are applicable to classification problems and that have at most three classes. Repeated instances are eliminated, and all instances are reduced to complete cases only. If an instance contains three classes, we convert them into two classes, such that the class with label 1 represents the class \mathcal{A} and the other two classes represent the class \mathcal{B} . This results in a final test set of 97 instances, as listed in Table 1. To avoid numerical instabilities, all data sets are scaled as follows. For each coordinate $j \in [1, p]$, we compute

$$l_j = \min_{i \in [1, N]} \{x_j^i\}, \quad u_j = \max_{i \in [1, N]} \{x_j^i\}, \quad m_j = 0.5(l_j + u_j)$$

and shift each coordinate j of all data points x^i via $\bar{x}_j^i = x_j^i - m_j$. Furthermore, if a coordinate j of the re-scaled points is still large, i.e., if $\bar{l}_j = l_j - m_j < -10^2$ or $\bar{u}_j = u_j - m_j > 10^2$ holds, it is re-scaled via

$$\tilde{x}_j^i = (\bar{v} - \underline{v}) \frac{\bar{x}_j^i - \bar{l}_j}{\bar{u}_j - \bar{l}_j} + \bar{v}$$

with $\bar{v} = 10^2$ and $\underline{v} = -10^2$. The corresponding 10 instances that we re-scale are marked with an asterisk in Table 1.

Table 1: Overview over the entire test set with number of points (N) and dimension (p)

ID	Instance	N	p
1	prnm_synth	250	2
2*	analcatdata_asbestos	73	3
3*	lupus	87	3
4	analcatdata_boxing1	120	3
5	analcatdata_boxing2	132	3
6	haberman	289	3
7	analcatdata_happiness	60	3
8*	analcatdata_aids	50	4
9	analcatdata_lawsuit	263	4
10	iris	147	4
11	hayes_roth	93	4
12	balance_scale	625	4
13	parity5	32	5
14*	bupa	341	5
15	irish	470	5
16	phoneme	5349	5
17	tae	110	5
18	new_thyroid	215	5
19*	analcatdata_bankruptcy	50	6
20*	analcatdata_creditscore	100	6
21	mux6	64	6
22	monk3	357	6
23	monk1	432	6
24	monk2	432	6
25	appendicitis	106	7
26	prnm_crabs	200	7
27*	penguins	333	7
28	postoperative_patient_data	78	8

29*	biomed	209	8
30*	pima	768	8
31*	cars	392	8
32	analcatdata_japansolvent	52	9
33	glass2	162	9
34	breast_cancer	272	9
35	saheart	462	9
36	threeOf9	512	9
37	profb	672	9
38	breast_w	463	9
39	tic_tac_toe	958	9
40	xd6	512	9
41	cmc	1425	9
42	analcatdata_cyyoung9302	92	10
43	analcatdata_cyyoung8092	97	10
44	breast	691	10
45	flare	315	10
46	parity5+5	1024	10
47	magic	18905	10
48	analcatdata_fraud	42	11
49	heart_statlog	270	13
50	heart_h	293	13
51	hungarian	293	13
52*	cleve	302	13
53*	heart_c	302	13
54	wine_recognition	178	13
55*	australian	690	14
56*	adult	48790	14
57*	schizo	340	14
58*	buggyCrx	690	15
59	labor	57	16
60	house_votes_84	342	16
61	hepatitis	155	19
62*	credit_g	1000	20
63	gametes_e_0.1H	1599	20
64	gametes_e_0.4H	1600	20
65	gametes_e_0.2H	1600	20
66	gametes_h_50	1592	20
67	gametes_h_75	1599	20
68*	churn	5000	20
69*	ring	7400	20
70	twonorm	7400	20
71	waveform_21	5000	21
72	ann_thyroid	7129	21
73	spect	228	22
74	horse_colic	357	22
75	agaricus_lepiota	8124	22
76*	hypothyroid	3086	25
77*	dis	3711	29
78*	allhypo	3709	29
79*	allbp	3711	29
80*	breast_cancer_wisconsin	569	30

81	backache	180	32
82	ionosphere	351	34
83	chess	3196	36
84	waveform_40	5000	40
85	connect_4	67557	42
86	spectf	267	44
87*	tokyo1	959	44
88	molecular_biology_promoters	106	57
89*	spambase	4210	57
90	sonar	208	60
91	splice	2903	60
92	coil2000	8380	85
93*	Hill_Valley_without_noise	1212	100
94*	clean1	476	168
95*	clean2	6598	168
96	dna	3002	180
97	gametes_e_1000atts	1600	1000

In our computational study, we aim to emphasize the significance of cardinality constraints, particularly in the context of non-representative biased samples. Biased samples are frequently observed in non-probability surveys, which are surveys where the inclusion process is not monitored and, hence, the inclusion probabilities are unknown as well. Therefore, correction methods like inverse inclusion probability weighting are not applicable. For an understanding of inverse inclusion probability weighting, see Skinner and D’arrigo (2011) and references therein.

To simulate this scenario, we create 5 biased samples with 10 % of the data being labeled for each instance. In contrast to a simple random sample, where each point has an equal probability of being chosen as labeled data, in the biased sample the labeled data are chosen with probability 85 % for being on class \mathcal{A} . Then, for each instance, with a time limit of 7200 s each, we apply the methods listed in Section 4.2.

Additionally, in Appendix B, we provide the results under simple random sampling, which produces unbiased samples. We see that the results from the proposed methods are similar to OCT-H (Bertsimas and Dunn 2017) in this setting. Despite the extra computational costs, we do not observe any drawbacks by using S²OCT in more simple survey designs.

4.2. Computational Setup. For each one of the 485 instances described in Section 4.1, the following approaches are compared:

- (a) OCT-H as described in Bertsimas and Dunn (2017), where only labeled data are considered.
- (b) S²OCT as given in Problem (P2) with $B(s)$ as in Expression (7) and M as given in Theorem 2.

Our comparison has been implemented in Julia 1.8.5 and we use Gurobi 9.5.2 and JuMP (Dunning et al. 2017) to solve OCT-H as well as Problem (P2). All computations were executed on the high-performance cluster “Elwetritsch”, which is part of the “Alliance of High-Performance Computing Rheinland-Pfalz” (AHRP). We used a single Intel XEON SP 6126 core with 2.6 GHz and 64 GB RAM.

To give the same importance to labeled and unlabeled data, we set the parameter $C = 1$ in S²OCT. Furthermore, we set the complexity parameter $\alpha = 0$ in OCT-H. Also, as required by OCT-H, all points x^i belong to $[0, 1]^d$. For this to hold, we

re-scaled the data as discussed in Section 4.1, with the difference that $\tilde{l}_j < 0$ and $\tilde{u}_j > 1$.

Theorem 2 establishes a relationship between s and M . To keep M at least 500 and s sufficiently large, from preliminary numerical tests we set, in S²OCT,

$$s = \begin{cases} \max\{10, 499/(\eta\sqrt{d})\}, & \text{if } N \in [1, 650), \\ \max\{20, 499/(\eta\sqrt{d})\}, & N \in [650, 1500). \\ \max\{40, 499/(\eta\sqrt{d})\}, & \text{otherwise.} \end{cases}$$

By default, MIP solvers such as Gurobi aim to achieve a balance between exploring new feasible solutions and verifying the optimality of the current solution. In preliminary numerical tests, we observed that the solver required significant time to find feasible solutions. Therefore, we selected Gurobi's parameter MIPFocus = 1, i.e., the solver focuses more on finding feasible solutions. Moreover, D in OCT-H and in S²OCT are fixed as

$$D = \begin{cases} 2, & \text{if } N \in [1, 1000), \\ 3, & \text{otherwise.} \end{cases}$$

4.3. Evaluation Criteria. The first evaluation criterion is the run time of OCT-H and S²OCT. To compare run times, we use empirical cumulative distribution functions (ECDFs). Specifically, for S being a set of solvers (or approaches as above) and for \bar{P} being a set of problems, we denote by $t_{\bar{p},s} \geq 0$ the run time of the approach $s \in S$ applied to the problem $\bar{p} \in \bar{P}$ in seconds. If $t_{\bar{p},s} > 7200$, we consider problem \bar{p} as not being solved by approach s . With these notations, the performance profile of approach s is the graph of the function $\gamma_s : [0, \infty) \rightarrow [0, 1]$ given by

$$\gamma_s(\sigma) = \frac{1}{|\bar{P}|} |\{\bar{p} \in \bar{P} : t_{\bar{p},s} \leq \sigma\}|.$$

Furthermore, since the true labels of all points are known in the simulation, we categorize them into four distinct categories: true positive (TP) or true negative (TN) if the point is classified correctly in classes \mathcal{A} or \mathcal{B} , respectively, as well as false positive (FP) if the point is misclassified in the class \mathcal{A} and as false negative (FN) if the point is misclassified in the class \mathcal{B} . Using this information, we calculate two classification metrics. The first one is accuracy (AC). It measures the proportion of correctly classified points and is given by

$$\text{AC} := \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \in [0, 1]. \quad (8)$$

Observe that for AC the greater the value, the better the classification. The second metric is Matthews correlation coefficient (MCC). It measures the correlation coefficient between the observed and predicted classifications and is computed by

$$\text{MCC} := \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}} \in [-1, 1]. \quad (9)$$

As for accuracy, the higher the MCC, the better the classification. The main question is the following: For a specific instance, does S²OCT or OCT-H have a higher accuracy and MCC? Hence, we compute the instance-wise difference of the accuracy and MCC according to

$$\overline{\text{AC}} := \text{AC}_{\text{S}^2\text{OCT}} - \text{AC}_{\text{OCT-H}} \quad \overline{\text{MCC}} := \text{MCC}_{\text{S}^2\text{OCT}} - \text{MCC}_{\text{OCT-H}}, \quad (10)$$

where $\text{AC}_{\text{OCT-H}}$ and $\text{AC}_{\text{S}^2\text{OCT}}$ are computed as in (8), and $\text{MCC}_{\text{OCT-H}}$ and $\text{MCC}_{\text{S}^2\text{OCT}}$ as in (9). To keep the numerical results section concise, we report on precision and recall in Appendix A.

4.4. Numerical Results.

TABLE 2. Different quantile values for the number of continuous and binary variables

	Continuous		Binary	
	OCT-H	S ² OCT	OCT-H	S ² OCT
min	31	43	42	151
25 %	61	195	123	846
50 %	97	366	226	1843
75 %	319	3028	1451	16 469
max	14 039	121 910	54 373	695 835

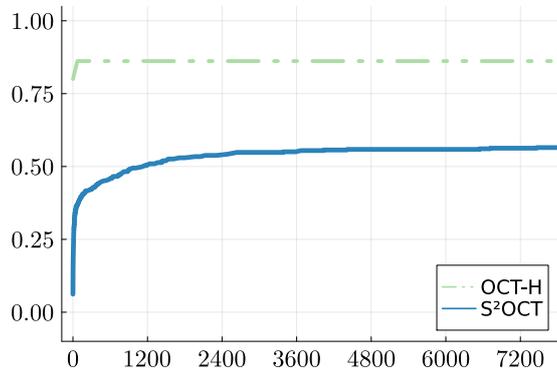


FIGURE 2. ECDFs for run times (in seconds).

4.4.1. *Run Time.* The number of continuous and binary variables is an important property for comparing different approaches. For this purpose, we computed the number of these variables for all instances presented in Section 4.1. Table 2 provides a quantile analysis of these quantities.

Observe that S²OCT has more variables than OCT-H. Therefore, it can be expected that OCT-H solves more instances than S²OCT within the time limit. However, note that the two approaches are designed for different purposes. Our approach considers labeled and unlabeled points together with the respective cardinality constraint, while OCT-H only deals with labeled points. Figure 2 shows ECDFs of run times of OCT-H and S²OCT. OCT-H solved 86 % of the instances within the time limit, while S²OCT does so for 58 %. As expected, OCT-H has significantly shorter run times.

4.4.2. *Accuracy and MCC.* Note that for both metrics, \overline{AC} and \overline{MCC} , a value greater than zero indicates that S²OCT had a better result than OCT-H and lower than zero indicates that OCT-H had a better result than S²OCT. Moreover, the box in the boxplot depicts the range of the medium 50 % of the values; 25 % of the values are below and 25 % are above the box. As can be seen in Figure 3, the \overline{AC} values are greater than zero in 75 % of the results (rows 1 and 2). Therefore, our proposed method takes advantage of the additional information on the total number of cases for the classes and has a better accuracy than OCT-H. When comparing all instances (column 1), the negative outliers indicate worse accuracy for S²OCT than OCT-H in some cases. This happens because in some instances our method does not terminate within the time limit while OCT-H does. Since for those instances that

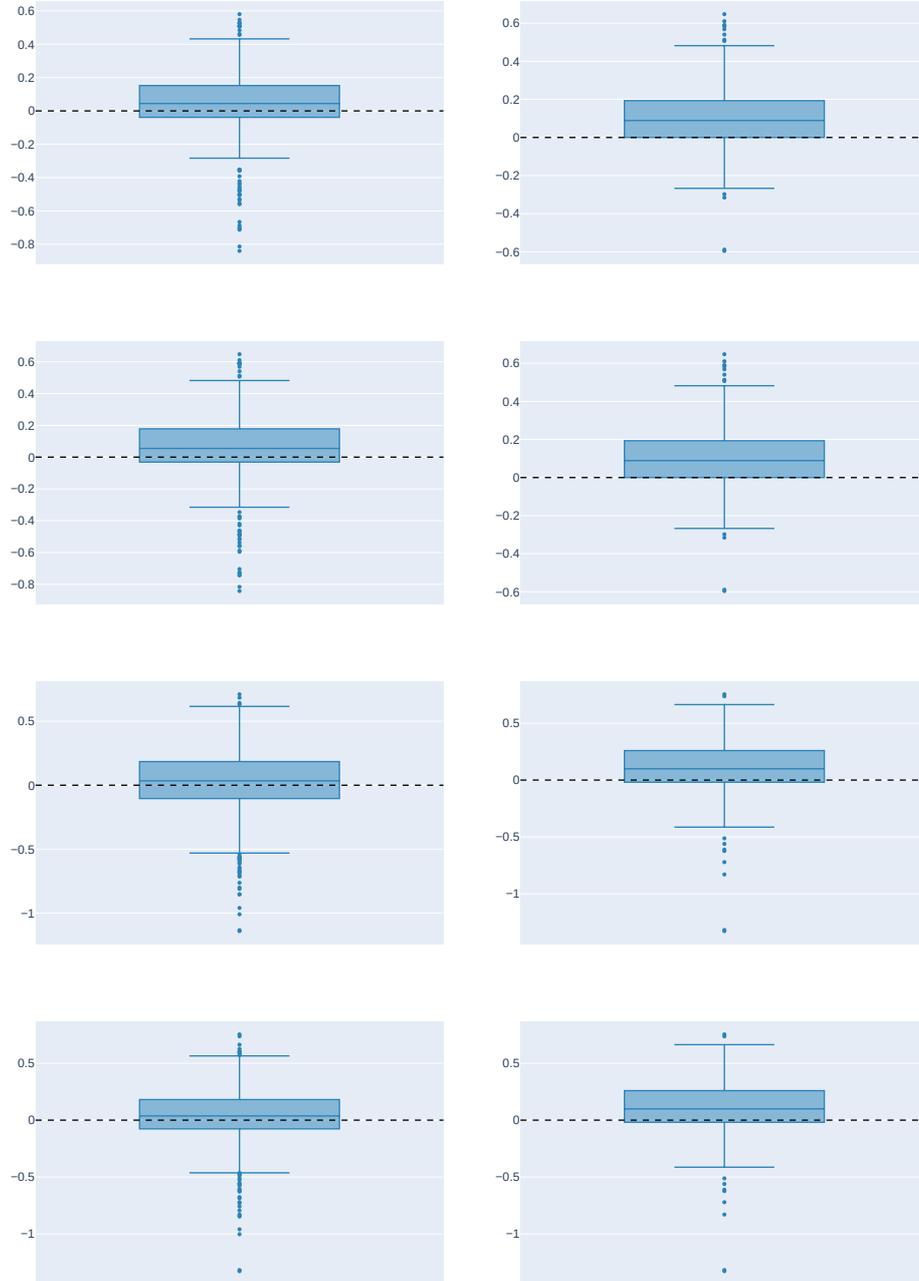


FIGURE 3. First row: Comparison of accuracy \overline{AC} as described in (10) for the entire data set. Second row: Comparison of \overline{AC} for unlabeled data. Third row: Comparison of precision \overline{MCC} as described in (10) for the entire data set. Last row: Comparison of \overline{MCC} for unlabeled data. Left: Comparison for all instances. Right: Comparison only for those instances for which both approaches terminate within the time limit.

terminate within the time limit (column 2), we have few outliers in accuracy (rows 1 and 2), we expect that the number of instances with lower precision will decrease if we would increase the time limit. Figure 3 also shows that the $\overline{\text{MCC}}$ values are greater than zero in most cases (rows 3 and 4), especially when comparing only the instances that terminate in the time limit (column 2). This means that our method has a better MCC than OCT-H. The consequences of the results so far are that using the unlabeled points as well as the cardinality constraint allows to correctly classify the points with higher accuracy and better MCC than with the optimal decision tree approach OCT-H. Moreover, further numerical tests revealed that if the percentage of labeled points is decreased, OCT-H tends to decrease in accuracy and MCC, while the deterioration for S²OCT is much less pronounced. This is especially relevant as in typical social surveys the sample proportion is seldomly over 1% of the population.

5. CONCLUSION

In many classification problems, acquiring labels for the entire population of interest can be expensive. Fortunately, external sources oftentimes can provide aggregated information on how many points are in each class. For this context, we proposed an MILP model for semi-supervised multivariate OCTs that considers the setting of labeled and unlabeled data points as well as additional aggregated information for the unlabeled data for a binary classification.

Under the condition of simple random sampling, our proposed approach has a slightly better accuracy and a better MCC than the conventional optimal classification tree. In many applications, however, the available data is coming from non-probability samples, where the data collection mechanism is largely unknown. Assuming simple random sampling in this setting is at least optimistic. Consequently, there is the risk of obtaining biased samples. Our numerical results show that our model has better accuracy, MCC, and precision than the existing approach from the literature, even with a small number of labeled points and biased samples. As expected, the drawback of introducing the cardinality constraint is that we get larger computational costs.

For future work, we will adapt our approach to a multiclass OCT. Furthermore, more research is needed to further reduce the computational burden.

ACKNOWLEDGEMENTS

The authors thank the DFG for their support within RTG 2126 “Algorithmic Optimization”.

REFERENCES

- Altincay, H. (2007). “Decision trees using model ensemble-based nodes.” In: *Pattern Recognition* 40, pp. 3540–3551. DOI: [10.1016/j.patcog.2007.03.023](https://doi.org/10.1016/j.patcog.2007.03.023).
- Amini, M.-R. and P. Gallinari (2002). “Semi-Supervised Logistic Regression.” In: *Proceedings of the 15th European Conference on Artificial Intelligence. ECAI’02*. Lyon, France: IOS Press, pp. 390–394.
- Beale, E. and J. Tomlin (1969). “Special facilities in a general mathematical programming system for nonconvex problems using ordered sets of variables.” In: *Operational Research* 69, pp. 447–454.
- Bennett, K. P. and J. A. Blue (1996). “Optimal Decision Trees.” In: *Rensselaer Polytechnic Institute Math Report* 214.
- Bertsimas, D. and J. Dunn (2017). “Optimal classification trees.” In: *Machine Learning* 106.7, pp. 1039–1082. DOI: [10.1007/s10994-017-5633-9](https://doi.org/10.1007/s10994-017-5633-9).

- Blanco, V., A. Japón, and J. Puerto (2022a). “Robust optimal classification trees under noisy labels.” In: *Advances in Data Analysis and Classification* 16.1, pp. 155–179. DOI: [10.1007/s11634-021-00467-2](https://doi.org/10.1007/s11634-021-00467-2).
- Blanco, V., A. Japón, and J. Puerto (2022b). “A mathematical programming approach to SVM-based classification with label noise.” In: *Computers & Industrial Engineering* 172, p. 108611. DOI: [10.1016/j.cie.2022.108611](https://doi.org/10.1016/j.cie.2022.108611).
- Blanquero, R., E. Carrizosa, C. Molero-Río, and D. Romero Morales (2019). “Sparsity in optimal randomized classification trees.” In: *European Journal of Operational Research* 284.1, pp. 255–272. DOI: [10.1016/j.ejor.2019.12.002](https://doi.org/10.1016/j.ejor.2019.12.002).
- (2021). “Optimal randomized classification trees.” In: *Computers & Operations Research* 132, p. 105281. DOI: [10.1016/j.cor.2021.105281](https://doi.org/10.1016/j.cor.2021.105281).
- Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks.
- Burgard, J. P., J. Krause, and S. Schmaus (2021). “Estimation of regional transition probabilities for spatial dynamic microsimulations from survey data lacking in regional detail.” In: *Computational Statistics & Data Analysis* 154, p. 107048. DOI: [10.1016/j.csda.2020.107048](https://doi.org/10.1016/j.csda.2020.107048).
- Burgard, J. P., M. E. Pinheiro, and M. Schmidt (2023). *Mixed-Integer Quadratic Optimization and Iterative Clustering Techniques for Semi-Supervised Support Vector Machines*. arXiv: [2303.12532v2](https://arxiv.org/abs/2303.12532v2) [[math.OC](https://arxiv.org/abs/2303.12532v2)].
- Bzdok, D., M. Eickenberg, O. Grisel, B. Thirion, and G. Varoquaux (2015). “Semi-Supervised Factored Logistic Regression for High-Dimensional Neuroimaging Data.” In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc.
- Carrizosa, E., C. Molero-Río, and D. R. Morales (2021). “Mathematical optimization in classification and regression trees.” In: *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research* 29.1, pp. 5–33. DOI: [10.1007/s11750-021-00594-1](https://doi.org/10.1007/s11750-021-00594-1).
- Chapelle, O., M. Chi, and A. Zien (2006). “A Continuation Method for Semi-Supervised SVMs.” In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML ’06. New York, NY, USA: Association for Computing Machinery, pp. 185–192. DOI: [10.1145/1143844.1143868](https://doi.org/10.1145/1143844.1143868).
- Cortes, C. and V. Vapnik (1995). “Support Vector Networks.” In: *Machine Learning* 20, pp. 273–297. DOI: [10.1007/BF00994018](https://doi.org/10.1007/BF00994018).
- D’Onofrio, F., G. Grani, M. Monaci, and L. Palagi (2023). *Margin Optimal Classification Trees*. arXiv: [2210.10567](https://arxiv.org/abs/2210.10567) [[math.OC](https://arxiv.org/abs/2210.10567)].
- Dunning, I., J. Huchette, and M. Lubin (2017). “JuMP: A Modeling Language for Mathematical Optimization.” In: *SIAM Review* 59.2, pp. 295–320. DOI: [10.1137/15M1020575](https://doi.org/10.1137/15M1020575).
- Gambella, C., B. Ghaddar, and J. Naoum-Sawaya (2021). “Optimization problems for machine learning: A survey.” In: *European Journal of Operational Research* 290.3, pp. 807–828. DOI: [10.1016/j.ejor.2020.08.045](https://doi.org/10.1016/j.ejor.2020.08.045).
- Kim, K. (2016). “A hybrid classification algorithm by subspace partitioning through semi-supervised decision tree.” In: *Pattern Recognition* 60, pp. 157–163. DOI: [10.1016/j.patcog.2016.04.016](https://doi.org/10.1016/j.patcog.2016.04.016).
- Kocev, M. C. D., J. Levatić, and S. Džeroski (2017). “Semi-supervised classification trees.” In: *Journal of Intelligent Information Systems* 49, pp. 461–486. DOI: [10.1007/s10844-017-0457-4](https://doi.org/10.1007/s10844-017-0457-4).
- Kotsiantis, S. (2014). “A hybrid decision tree classifier.” In: *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology* 26, pp. 327–336. DOI: [10.3233/IFS-120741](https://doi.org/10.3233/IFS-120741).

- Lee, D.-H. (2013). “Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks.” In: *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*.
- McCormick, G. P. (1976). “Computability of Global Solutions to Factorable Non-convex Programs: Part I – Convex Underestimating Problems.” In: *Mathematical Programming* 10.1, pp. 147–175. DOI: [10.1007/BF01580665](https://doi.org/10.1007/BF01580665).
- Melacci, S. and M. Belkin (2009). “Laplacian Support Vector Machines Trained in the Primal.” In: *Journal of Machine Learning Research* 12. DOI: [10.48550/ARXIV.0909.5422](https://doi.org/10.48550/ARXIV.0909.5422).
- Nguyen, T. N. N., B. Veeravalli, and X. Fong (2023). “A Semi-Supervised Learning Method for Spiking Neural Networks Based on Pseudo-Labeling.” In: *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. DOI: [10.1109/IJCNN54540.2023.10191317](https://doi.org/10.1109/IJCNN54540.2023.10191317).
- Oliver, A., A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow (2018). “Realistic Evaluation of Deep Semi-Supervised Learning Algorithms.” In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc. DOI: [10.48550/arXiv.1804.09170](https://doi.org/10.48550/arXiv.1804.09170).
- Olson, R. S., W. La Cava, P. Orzechowski, R. J. Urbanowicz, and J. H. Moore (2017). “PMLB: a large benchmark suite for machine learning evaluation and comparison.” In: *BioData Mining* 10.36, pp. 1–13. DOI: [10.1186/s13040-017-0154-4](https://doi.org/10.1186/s13040-017-0154-4).
- Orsenigo, C. and C. Vercellis (2003). “Multivariate classification trees based on minimum features discrete support vector machines.” In: *IMA Journal of Management Mathematics* 14.3, pp. 221–234. DOI: [10.1093/imaman/14.3.221](https://doi.org/10.1093/imaman/14.3.221).
- Quinlan, J. R. (1986). “Induction of Decision Trees.” In: *Machine Learning* 1, pp. 81–106. DOI: doi.org/10.1007/BF00116251.
- Santhiappan, S. and B. Ravindran (2021). “A Semi-Supervised Approach to Growing Classification Trees.” In: *Proceedings of the 3rd ACM India Joint International Conference on Data Science & Management of Data (8th ACM IKDD CODS & 26th COMAD)*. CODS-COMAD '21. Bangalore, India: Association for Computing Machinery, pp. 29–37. DOI: [10.1145/3430984.3431009](https://doi.org/10.1145/3430984.3431009).
- Skinner, C. J. and D’arrigo (2011). “Inverse probability weighting for clustered nonresponse.” In: *Biometrika* 98.4, pp. 953–966. DOI: [10.1093/biomet/asr058](https://doi.org/10.1093/biomet/asr058).
- Tanha, J., M. van Someren, and H. Afsarmanesh (2017). “Semi-supervised self-training for decision tree classifiers.” In: *International Journal of Machine Learning and Cybernetics* 8, pp. 355–370. DOI: [10.1007/s13042-015-0328-7](https://doi.org/10.1007/s13042-015-0328-7).
- Verwer, S. and Y. Zhang (2019). “Learning Optimal Classification Trees Using a Binary Linear Program Formulation.” In: vol. 33. AAAI Press, pp. 1625–1632. DOI: [10.1609/aaai.v33i01.33011624](https://doi.org/10.1609/aaai.v33i01.33011624).
- Yildiz, O. and O. Dikmen (2007). “Parallel Univariate Decision Trees.” In: *Pattern Recognition Letters* 28, pp. 825–832. DOI: [10.1016/j.patrec.2006.11.009](https://doi.org/10.1016/j.patrec.2006.11.009).
- Zharmagambetov, A. and M. A. Carreira-Perpinan (2022). “Semi-Supervised Learning with Decision Trees: Graph Laplacian Tree Alternating Optimization.” In: *Advances in Neural Information Processing Systems*. Vol. 35, pp. 2392–2405.
- Zhu, X. and A. B. Goldberg (2009). *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers. DOI: [10.2200/S00196ED1V01Y200906AIM006](https://doi.org/10.2200/S00196ED1V01Y200906AIM006).

APPENDIX A. FURTHER NUMERICAL RESULTS

Besides the measures of accuracy and MCC, we compare two further measures that, depending on the application, can be highly relevant. The first metric is precision (PR). It measures the proportion of correctly classified points among all

positively classified points and is thus defined as

$$\text{PR} := \frac{\text{TP}}{\text{TP} + \text{FP}} \in [0, 1]. \quad (11)$$

This quantity is important in some application such as for fraud detection systems, where identifying legitimate transactions as fraudulent is better than identify fraudulent transactions as legitimate. Moreover, precision can be higher when there are more positives in the dataset.

Second, we consider recall (RE), which quantifies the proportion of positive instances that are correctly classified as positive. It is formally given by

$$\text{RE} := \frac{\text{TP}}{\text{TP} + \text{FN}} \in [0, 1]. \quad (12)$$

This quantity is important in some applications such as in cancer diagnosis, where evaluating recall is relevant as it is more significant to identify potential cancer cases than to do not. Different from precision, recall can be higher when there are more negatives in the dataset.

As accuracy and MCC, the main question is how much better precision and recall of S²OCT are compared to the one of the OCT-H. Hence, we compute the difference of the precision and recall according to

$$\overline{\text{PR}} := \text{PR}_{\text{S}^2\text{OCT}} - \text{PR}_{\text{OCT-H}} \quad \overline{\text{RE}} := \text{RE}_{\text{S}^2\text{OCT}} - \text{RE}_{\text{OCT-H}}, \quad (13)$$

where $\text{PR}_{\text{OCT-H}}$ and $\text{PR}_{\text{S}^2\text{OCT}}$ are computed as in (11) for the OCT-H and S²OCT, respectively. In the same way, $\text{RE}_{\text{OCT-H}}$ and $\text{RE}_{\text{S}^2\text{OCT}}$ are computed as in (12) for the OCT-H and S²OCT. Note that as in Section 4, for both $\overline{\text{PR}}$ and $\overline{\text{RE}}$, a value greater than zero indicates that S²OCT has a better result than OCT-H and lower than zero indicates that S²OCT has a worse result than OCT-H. As can be seen in Figure 4, the $\overline{\text{PR}}$ values are greater than zero in more than 75% of the results (rows 1 and 2). This means that S²OCT classifies the points with higher precision than OCT-H. Hence, OCT-H has more false-positive results. The negative outliers most likely are due to the same reason as those for the respective $\overline{\text{AC}}$ and $\overline{\text{MCC}}$ values.

On the other hand, Figure 4 also show that $\overline{\text{RE}}$ is, in general, lower than 0. This means that OCT-H has better recall than our method. The results of precision and recall can be justified by the fact that the biased sample is more likely to have labeled data in class \mathcal{A} and having no information about the unlabeled data, the OCT-H ends up classifying points on the positive side.

APPENDIX B. NUMERICAL RESULTS FOR SIMPLE RANDOM SAMPLES

Our computational study in Section 4 focuses on the analysis of non-representative biased samples. The typical baseline scenario for evaluating the performance of estimators is to apply them on simple random samples. Therefore, to complement our numerical results, we also present the results under simple random sampling. In a simple random sampling, each unit in the data set has the same probability $\pi_i = n/N$ to be included in the sample of labeled data of size n . The instances are the same as described in Section 4.1. The computational setup follows the description in Section 4.2. As before, the used evaluation criteria are $\overline{\text{AC}}$ and $\overline{\text{MCC}}$ as in (10) and $\overline{\text{PR}}$ and $\overline{\text{RE}}$ as in (13).

Figure 5 shows that for all the instances (column 1), $\overline{\text{AC}}$ (rows 1 and 2) and $\overline{\text{MCC}}$ (rows 3 and 4) have value greater than 0 and lower than 0 in 50% of the cases. This means both approaches have similar accuracy and MCC. However, when comparing only those instances that terminate within the time limit (column 2), it can be seen that S²OCT has slightly better accuracy and MCC, but not as much as for biased samples; see Section 4.4.2. This is expected because the sample is not

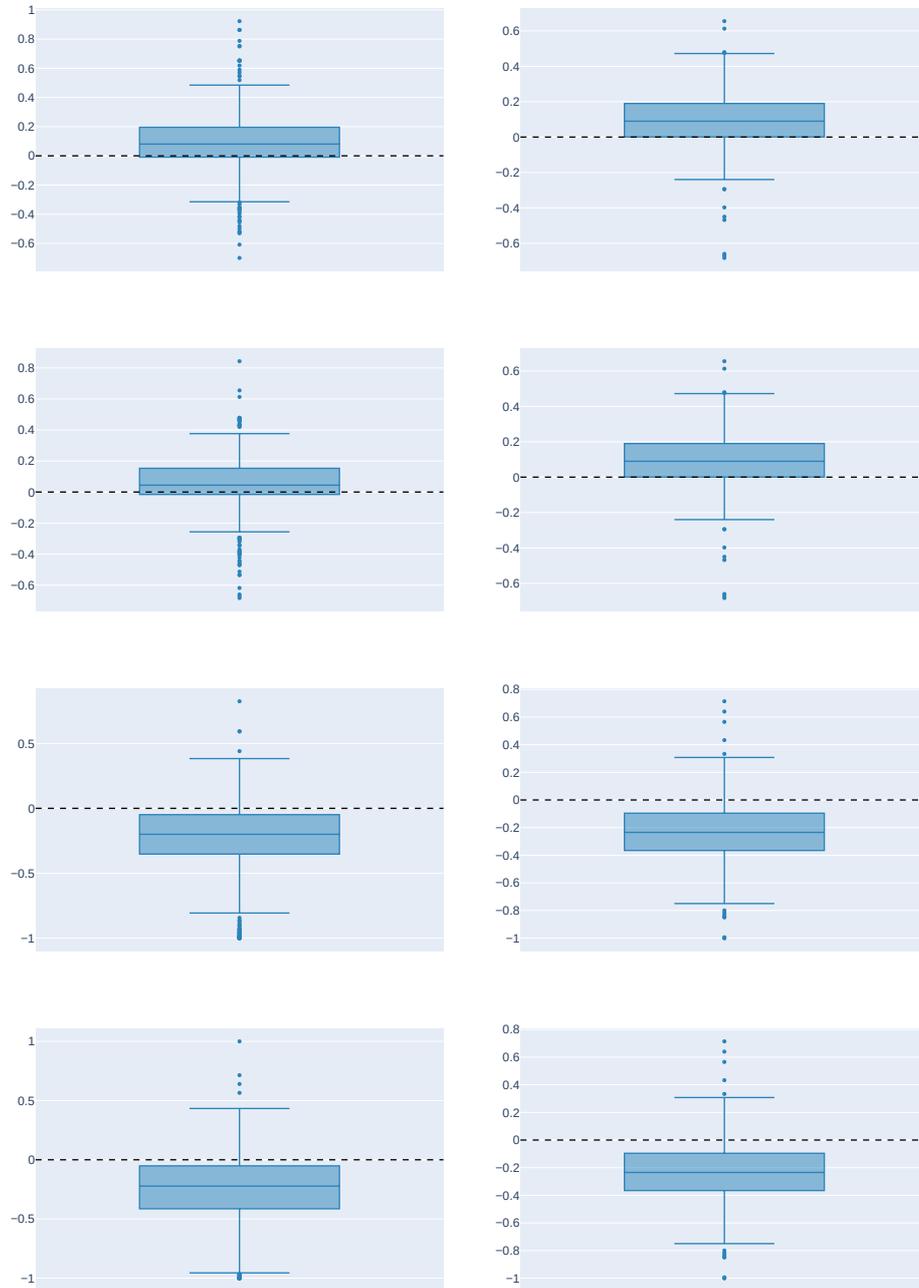


FIGURE 4. First row: Comparison of precision \overline{PR} as described in (13) for the entire data set. Second row: Comparison of \overline{PR} for unlabeled data. Third row: Comparison of recall \overline{RE} as described in (13) for the entire data set. Last row: Comparison of \overline{RE} for unlabeled data. Left: Comparison for all instances. Right: Comparison only for those instances for which both approaches terminate within the time limit.

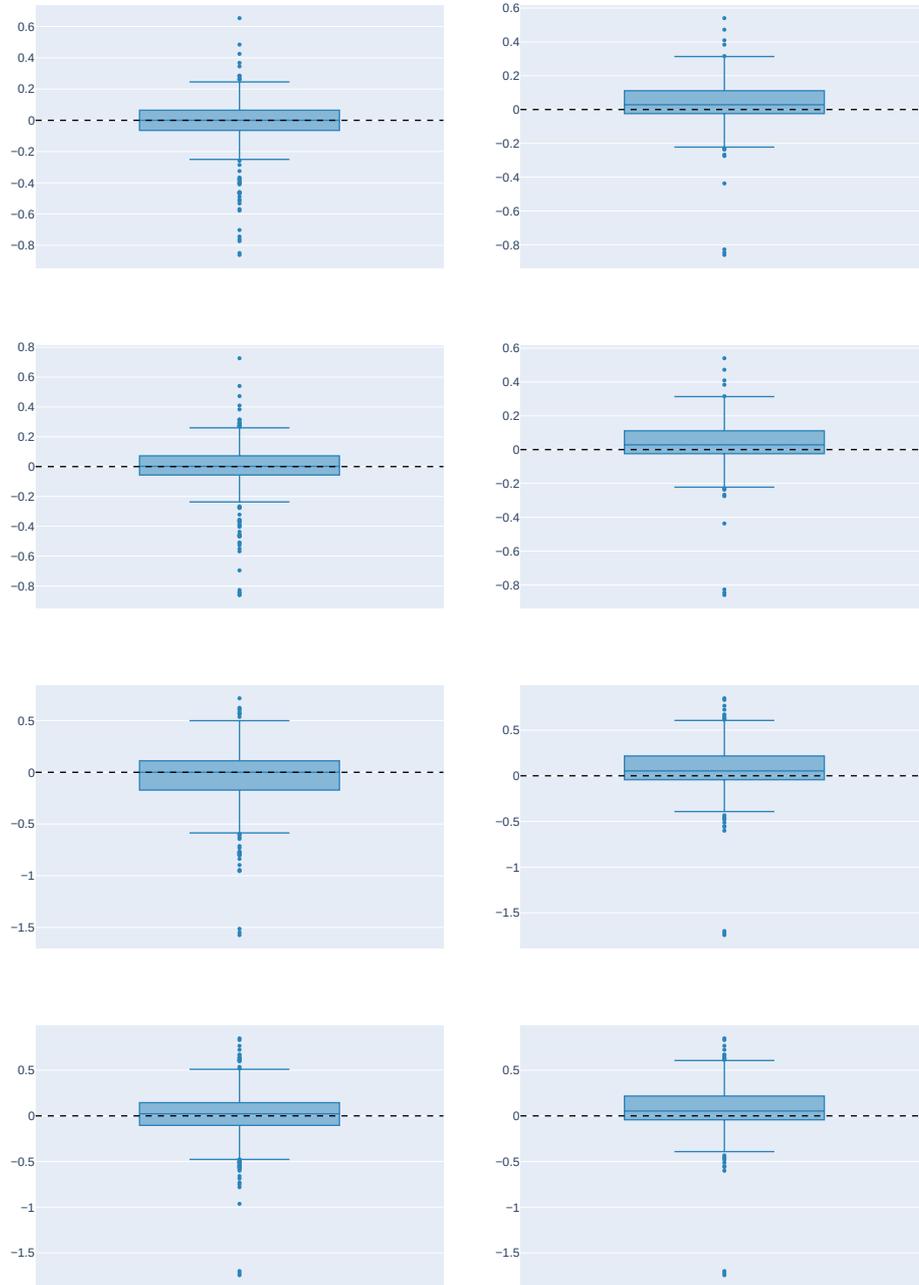


FIGURE 5. First row: Comparison of accuracy \overline{AC} as described in (10) for the entire data set. Second row: Comparison of \overline{AC} for unlabeled data. Third row: Comparison of precision \overline{MCC} as described in (10) for the entire data set. Last row: Comparison of \overline{MCC} for unlabeled data. Left: Comparison for all instances. Right: Comparison only for those instances for which both approaches terminate within the time limit.

biased. Consequently, the cardinality constraint, which aims to balance the class distribution, does not introduce additional meaningful information to the problem. As can be seen in Figure 6, precision and recall are similar for both approaches. Therefore, for the simple random samples, our approach has almost the same results as OCT-H, with slight improvements in accuracy and MCC.

(J. P. Burgard) TRIER UNIVERSITY, DEPARTMENT OF ECONOMIC AND SOCIAL STATISTICS,
UNIVERSITÄTSRING 15, 54296 TRIER, GERMANY
Email address: burgardj@uni-trier.de

(M. E. Pinheiro, M. Schmidt) TRIER UNIVERSITY, DEPARTMENT OF MATHEMATICS, UNIVER-
SITÄTSRING 15, 54296 TRIER, GERMANY
Email address: pinheiro@uni-trier.de
Email address: martin.schmidt@uni-trier.de

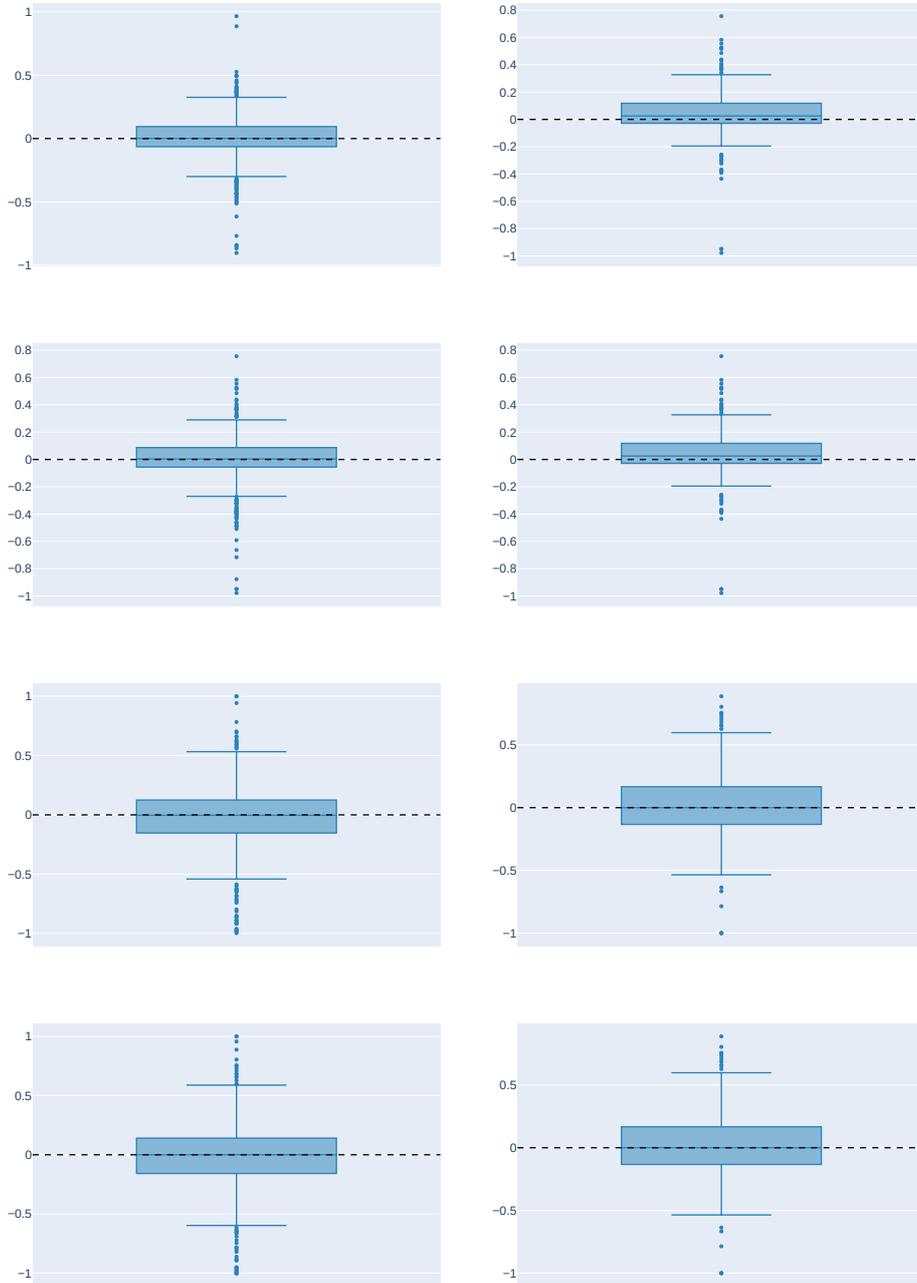


FIGURE 6. First row: Comparison of precision \overline{PR} as described in (13) for the entire data set. Second row: Comparison of \overline{PR} for unlabeled data. Third row: Comparison of recall \overline{RE} as described in (13) for the entire data set. Last row: Comparison of \overline{RE} for unlabeled data. Left: Comparison for all instances. Right: Comparison only for those instances for which both approaches terminate within the time limit.