Optimal Counterfactual Explanations in Tree Ensembles

Axel Parmentier [1] **Thibaut Vidal** [2 3]

Abstract

Counterfactual explanations are usually generated through heuristics that are sensitive to the search's initial conditions. The absence of guarantees of performance and robustness hinders trustworthiness. In this paper, we take a disciplined approach towards counterfactual explanations for tree ensembles. We advocate for a model-based search aiming at "optimal" explanations and propose efficient mixed-integer programming approaches. We show that isolation forests can be modeled within our framework to focus the search on plausible explanations with a low outlier score. We provide comprehensive coverage of additional constraints that model important objectives, heterogeneous data types, structural constraints on the feature space, along with resource and actionability restrictions. Our experimental analyses demonstrate that the proposed search approach requires a computational effort that is orders of magnitude smaller than previous mathematical programming algorithms. It scales up to large data sets and tree ensembles, where it provides, within seconds, systematic explanations grounded on well-defined models solved to optimality.

1. Introduction

Accountability in machine learning is quickly rising as a major concern as learning algorithms take over tasks that have a major impact on human lives. With the increasing use of profiling and automated decision-making systems, new legal provisions are being set up to protect rights to transparency. The recent interpretation of the EU General Data Protection Regulation (GDPR) by Article 29 Data Protection Working Party (2017) refers to a "right to explanations" and has triggered extensive research on algorithmic recourse and counterfactual explanations (see, e.g., Wachter et al., 2018; Karimi et al., 2020b; Verma et al., 2020).

Counterfactual explanations are contrastive arguments of the type: "To obtain this loan, you need $40,000 of annual revenue instead of the current $30,000". They correspond to small perturbations of an example that permit to modify the classification outcome, in a similar fashion as adversarial examples, though typically restricted by additional constraints ensuring actionability and plausibility (Barocas et al., 2020; Venkatasubramanian & Alfano, 2020).

Despite their conceptual simplicity, counterfactual explanations pose significant challenges related to data protection, intellectual property, ethics, along with fundamental computational tractability issues. Indeed, the scale of machine learning models has tremendously increased over a few decades. Even when restricted to the vicinity of an example, a systematic inspection of all possible explanations is intractable, and therefore most studies on counterfactual explanations rely on ad-hoc algorithms or heuristics (e.g., gradient descent in a non-convex space in Wachter et al. 2018). This poses at least three main issues:

(a) Heuristics can fail to identify the most natural and insightful explanation, and therefore do not necessarily give a trustworthy cause (Karimi et al., 2020a). This is especially true when the search involves combinatorial spaces (e.g., tree ensembles) with binary or integer features bound together by plausibility constraints.

(b) They can be sensitive to the initial conditions of the search, leading to unstable results —even for the same subject.

(c) Finally, these methods are not readily extensible to include additional constraints and domain knowledge regarding actionability and plausibility. A small change of problem formulation due to a specific application domain can very well require significant methodological adaptations.

To circumvent these issues, we advocate for a disciplined analysis of counterfactual explanations through mathematical programming lenses. We focus on tree ensembles (including random forests and gradient boosting), a popular

[1]CERMICS, École des Ponts Paristech; [2]CIRRELT & SCALE-AI Chair in Data-Driven Supply Chains, Department of Mathematics and Industrial Engineering, Polytechnique Montreal, Canada; [3]Department of Computer Science, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Brazil. Correspondence to: Thibaut Vidal <thibaut.vidal@cirrelt.ca>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

family of models with good empirical performance which is often sought as a more transparent replacement to neural networks (Rudin, 2019). We opt for a solution approach grounded on mixed integer linear and quadratic programming (MILP and MIQP) since we believe that it solves the three aforementioned issues. Firstly, the search for an optimal solution of a well-defined model permits to control the quality of the solution and ensures the stability of the results, solving key issues (a) and (b). Moreover, as seen later in this paper, the modeling capacities of MILP permit to seamlessly integrate domain information as well as many forms of plausibility and actionability constraints, therefore making meaningful progress on issue (c). Finally, the tremendous progress of MILP solution approaches over the years (estimated to a 10^{11} reduction of computational effort on the same problems – Bixby 2012) now permits to use such a modeling and search approach to its full potential.

We make the following contributions:

1. We propose the first efficient mathematical models to search for counterfactual explanations in tree ensembles with a number of binary variables that is logarithmic in the number of vertices, and therefore scales to large trees while retaining its ability to find an optimal solution. This is exponentially fewer variables than the previous models used in Cui et al. (2015) and Kanamori et al. (2020). Our approach is applicable to heterogeneous datasets with numerical, ordinal, binary, and categorical features, with possible oblique splits on numerical features, and considering single or multiple classes. In contrast with previous works, it does not require binary variables to model the numerical feature levels and has a sparse constraint matrix. Consequently, solution performance remains stable as the number of features increases.

2. We demonstrate how to integrate plausibility in our mathematical framework from an isolation forest viewpoint. Isolation forests are an effective and distribution-agnostic way to associate a plausibility score for the different regions of the feature space. This is, to our knowledge, the first time that this framework is used for counterfactual explanations, through an integration which is only possible due to our significant model-tractability improvements.

3. We discuss extensions of the model capturing important constraints regarding plausibility and actionability. We therefore provide a flexible and modular toolset that can be adapted to each specific situation.

4. Finally, we conduct an extensive and reproducible experimental campaign, which can be executed from a single self-contained Python script. Our source code is openly accessible at `https://github.com/vidalt/OCEAN` under a MIT license. We demonstrate that the approaches proposed in this work are efficient and scalable, producing *optimal* counterfactual explanations in a matter of a few seconds on data sets with over fifty features, using tree ensembles with hundreds of trees. We evaluate the impact of our plausibility constraints via isolation forests, demonstrating the flexibility of the approach and showing that these extra constraints do not significantly impact the performance of the solution process while significantly boosting the usefulness of the explanations.

2. Background

2.1. Mixed Integer Programming

MIQPs can be cast into the following standard form:

$$\min \quad f(\mathbf{x}) \tag{1}$$
$$\text{s.t. } A\mathbf{x} \leq \mathbf{b} \tag{2}$$
$$\mathbf{x}^\intercal Q_i \mathbf{x} + \mathbf{c}_i \mathbf{x} \leq b_i \qquad i \in \{1, \dots, m\} \tag{3}$$
$$\mathbf{x} \in \mathbb{Z}^a \times \mathbb{R}^b, \tag{4}$$

where a represents the number of variables taking integer values and b is the number of continuous variables. The feasibility region of the problem is defined as the intersection of a polytope (Constraint 2) along with a set of quadratic restrictions (Constraint 3). State-of-the-art solvers (e.g., CPLEX and Gurobi) can handle separable quadratic objectives of the form $f(\mathbf{x}) = \mathbf{x}^\intercal Q \mathbf{x} + \mathbf{c}\mathbf{x}$ and therefore model a wide range of objectives (regularization terms through l_0, l_1 and squared l_2 norms, squared Euclidean and Mahalanobis distances, and variations thereof). MILP and MIQP are NP-hard in general, though astonishing progress in solution methods has permitted to handle increasingly large problems. Solver performance is, however, dependent on the quality of the problem formulation (i.e., the model). Ideally, a good model should have few binary variables, limited symmetry, and a strong continuous relaxation, i.e., a small gap between its optimal solution value and that of the same problem in which variables $\mathbf{x}$ are relaxed to the domain $\mathbb{R}^{a+b}$, as this permits to quickly prune regions of the search space during the branch-and-bound process (Wolsey, 2020).

2.2. Counterfactual Explanations in Tree Ensembles

Let $\{\mathbf{x}_k, c_k\}_{k=1}^n$ be a training set in which each $\mathbf{x}_k \in \mathbb{R}^p$ corresponds to a sample characterized by a p-dimensional feature vector and a class $c_k \in \mathcal{C}$. In the most general form, a tree ensemble $\mathcal{T}$ learns a set of trees $t \in \mathcal{T}$ returning class probabilities $F_{tc} : \mathcal{X} \to [0, 1]$. For any sample $\mathbf{x}$, the tree ensemble returns the class c that maximizes the weighted sum of the probabilities: $F_{\mathcal{T}}(\mathbf{x}) = \arg\max_c \sum w_t F_{tc}(\mathbf{x})$. Given an origin point $\hat{\mathbf{x}}$ and a desired prediction class c^*, searching for a plausible and actionable counterfactual ex-

planation consists in locating a new data point $\mathbf{x} \in \mathcal{X}$ that solves the following problem:

$$\min \quad f_{\hat{\mathbf{x}}}(\mathbf{x}) \tag{5}$$
$$\text{s.t.} \quad F_{\mathcal{T}}(\mathbf{x}) = c^* \tag{6}$$
$$\mathbf{x} \in X^{\text{P}} \cap X^{\text{A}}. \tag{7}$$

In this problem, $f_{\hat{\mathbf{x}}}$ is a separable convex cost that represents how difficult it is to move from $\hat{\mathbf{x}}$ to $\mathbf{x}$. This generic cost function includes distance metrics (e.g., squared Euclidean or Mahalanobis distance) as a special case. Moreover, it allows possible cost asymmetry (Ustun et al., 2019; Karimi et al., 2020b) and can include additional penalization terms if needed. Polytopes X^{P} and X^{A} represent the space of plausible and actionable counterfactual explanations, and will be discussed in the next paragraphs in connection with recent works.

Related studies As reviewed in Guidotti et al. (2018), Karimi et al. (2020b) and Verma et al. (2020), early studies on counterfactual explanations were conducted in majority in a model-agnostic context through enumeration and heuristic search approaches (Wachter et al., 2018). Dedicated work has been later conducted on specific models, such as tree ensembles, which presented additional challenges due to their combinatorial and non-differentiable nature. To handle this case, Lucic et al. (2019) proposed a gradient algorithm that approximates the splits of the decision trees through sigmoid functions. Tolomei et al. (2017) designed a feature tweaking algorithm that enumerates alternative paths in each tree to change the decision of the ensemble. The method is shown to provide useful counterfactual explanations on several application cases, though it does not always deliver an optimal (or even a feasible) counterfactual explanation for tree ensembles. To circumvent these issues, Karimi et al. (2020a) reformulated the search for counterfactual explanations in heterogeneous domains (with binary, numerical and categorical features) as a satisfiability problem and employed specialized solvers. The authors reported promising results on three data sets involving up to 14 features.

Cui et al. (2015) and Kanamori et al. (2020) proposed mixed-integer linear programming approaches for optimal explanations in tree ensembles. The former work considers Mahalanobis distance and introduced decision logic constraints to ensure the consistency of the split decisions through the forest (ensuring that there is exists a counterfactual example satisfying them). The later work demonstrated how to expand the formulation to consider an l_1-norm distance and plausibility constraints grounded on the Local Outlier Factor (LOF) score. Both models use a discretization of the feature space, and therefore need a large number of binary variables to express continuous features (one for each possible feature level, and one for each leaf of each decision tree). Good results were still achieved on a variety of data sets. As demonstrated in our work, better formulations relying on exponentially fewer variables can be designed, leading to reductions of CPU time by some orders of magnitude. Finally, a few other works related to mathematical programming do not necessarily consider tree ensembles explanations but provide useful additional modeling strategies. In particular, Russell (2019) modeled the choice of a feature level among multiple intervals (a special case of disjunctive constraint discussed in Jeroslow & Lowe 1984) and suggest strategies to find multiple explanations. Moreover, Ustun et al. (2019) propose to improve actionability through the definition of extra constraints representing a partial causal model.

Plausibility and actionability. Constraint (7) aims at ensuring plausibility and actionability of the counterfactual explanations. These important requirements constitute one of the cruces of recent research on counterfactual explanations (Barocas et al., 2020). Plausibility constraints (polytope X^{P}) should ensure that the explanation $\mathbf{x}$ respects the structure of the data and that it is located in a region that has a sufficiently large density of samples. To capture this notion, we will rely on the information of isolation forests (Liu et al., 2008) within our framework to restrict the search to plausible regions. In contrast, actionability constraints (polytope X^{A}) concern the trajectory between $\hat{\mathbf{x}}$ and $\mathbf{x}$. At the very least, they ensure that immutable features remain fixed and that features that are bound to evolve unilaterally are constrained to remain within a half-space. A finer-grained knowledge of correlations (or even a partial causal model as discussed in Mahajan et al. 2019, Ustun et al. 2019, Mothilal et al. 2020 and Karimi et al. 2020c) can also be integrated into the formulation through additional linear and logical constraints, as discussed in Section 3.4.

3. Methodology

Our mathematical model is presented in two main stages. First, we describe the variables and constraints that characterize the branches taken by the counterfactual example. Next, we include additional variables and constraints modeling the counterfactual example's feature values and ensuring compatibility with all the branch choices.

Our formulation relies on two main pillars. First, it uses the natural disjunctive structure of the trees to model branch choices using exponentially fewer binary variables than previous models by Cui et al. (2015) and Kanamori et al. (2020). Second, it includes continuous variables organized as order simplices (Grotzinger & Witzgall, 1984) to represent the values of numerical or ordinal features of the counterfactual example and to connect them with the branch choices. This effectively leads to a formulation requiring only $\mathcal{O}(N_v)$ non-zero terms in the constraint matrix instead of $\mathcal{O}(N_v^2)$, where N_v stands for the overall number of internal nodes in the tree ensemble. Notably, this formulation complexity

does not depend on the number of features of the data set. Our model is especially suitable for large data sets with many numerical features, and permits us to rely on isolation forests to model plausibility without sacrificing numerical tractability.

3.1. Branch Choices

For each tree $t \in \mathcal{T}$, let $\mathcal{V}_t^I$ be the set of internal vertices associated with the splits, and let $\mathcal{V}_t^L$ be the set of terminal vertices (leaves). Let $\mathcal{D}_t$ represent the possible depths values and define $\mathcal{V}_{td}^I$ as the set of internal nodes at depth d. Let $l(v)$ and $r(v)$ be the left and right children of each internal vertex $v \in \mathcal{V}_t^I$. Finally, let p_{tvc} be the class probability of $c \in \mathcal{C}$ in each leaf $v \in \mathcal{V}_t^L$. Class probabilities can be defined in $\{0, 1\}$ in the hard voting model (as in the random forest algorithm initially proposed by Breiman 2001), or in $[0, 1]$ in the general soft voting model (e.g., as in scikit-learn).

For each tree t and depth d in $\mathcal{D}_t$, we use a binary decision variable λ_{td} which will take value 1 if the counterfactual example descends towards the left branch, and 0 otherwise. This value is free if the path does not attain this depth. We also use continuous variables $y_v \in [0, 1]$ for each $v \in \mathcal{V}_t^I \cup \mathcal{V}_t^L$ to represent the flow of the counterfactual example in the decision tree. These variables are not explicitly defined as binary but will effectively take value 1 if the counterfactual example passes through vertex v, and 0 otherwise. This behavior is ensured with the following set of constraints:

$$y_{t1} = 1 \qquad t \in \mathcal{T} \tag{8}$$
$$y_{tv} = y_{tl(v)} + y_{tr(v)} \qquad t \in \mathcal{T}, v \in \mathcal{V}_t^I \tag{9}$$
$$\sum_{v \in \mathcal{V}_{td}^I} y_{tl(v)} \leq \lambda_{td} \qquad t \in \mathcal{T}, d \in \mathcal{D}_t \tag{10}$$
$$y_{tv} \in [0, 1] \qquad t \in \mathcal{T}, v \in \mathcal{V}_t^I \cup \mathcal{V}_t^L \tag{11}$$
$$\lambda_{td} \in \{0, 1\} \qquad t \in \mathcal{T}, d \in \mathcal{D}_t. \tag{12}$$

Theorem 1 Formulation (8–12) guarantees the integrality of the $\mathbf{y}$ variables.

Proof of this theorem is provided in the supplementary material. From these variables, finding the desired counterfactual class though majority vote can be expressed as:

$$z_c = \sum_{t \in \mathcal{T}} \sum_{v \in \mathcal{V}_t^L} w_t p_{tvc} y_{tv} \qquad c \in \mathcal{C} \tag{13}$$
$$z_{c^*} > z_c \qquad c \in \mathcal{C}, c \neq c^*. \tag{14}$$

3.2. Feature Consistency with the Splits

The previous variables and constraints define the counterfactual example's paths through each tree. Additional constraints are needed to ensure that there exist feature values that are consistent with all these branching decisions. To that extent, we propose efficient formulations for each main data type (numerical, binary, and categorical), which can be combined in the case of heterogeneous data sets. We will refer to I_{N}, I_{B}, and I_{C} as the index sets of each feature type.

Numerical Features. We can assume, w.l.o.g., that continuous features have been scaled into the interval $[0, 1]$. For each numerical (continuous or discrete) feature $i \in I_{\mathrm{N}}$, let k_i be the overall number of distinct split levels in the forest. Moreover, let x_i^j be the j^{th} split level for $j \in \{1, \ldots, k_i\}$, and define $x_i^0 = 0$ as well as $x_i^{k_i+1} = 1$.

For each tree t, let $\mathcal{V}_{tij}^I$ be the set of internal nodes involving a split on feature i with level x_i^j, such that samples with feature values $x_i \leq x_i^j$ descend to the left branch, whereas others values satisfying $x_i > x_i^j$ descend to the right branch. With these definitions, the consistency of a feature i through the forest can be modeled with the help of auxiliary continuous variables μ_i^j for $j \in \{0, \ldots, k_i\}$, constrained in such a way that $\mu_i^j = 0$ implies $x_i \in [0, x_i^j]$ and $\mu_i^j = 1$ implies $x_i \in [x_i^j + \epsilon, 1]$, where ϵ is a small constant. These conditions are ensured through the following set of constraints:

$$\mu_i^{j-1} \geq \mu_i^j \qquad j \in \{1, \ldots, k_i\} \tag{15}$$
$$\mu_i^j \leq 1 - y_{tl(v)} \qquad j \in \{1, \ldots, k_i\}, t \in \mathcal{T}, v \in \mathcal{V}_{tij}^I \tag{16}$$
$$\mu_i^{j-1} \geq y_{tr(v)} \qquad j \in \{1, \ldots, k_i\}, t \in \mathcal{T}, v \in \mathcal{V}_{tij}^I \tag{17}$$
$$\mu_i^j \geq \epsilon y_{tr(v)} \qquad j \in \{1, \ldots, k_i\}, t \in \mathcal{T}, v \in \mathcal{V}_{tij}^I \tag{18}$$
$$\mu_i^j \in [0, 1] \qquad j \in \{0, \ldots, k_i\}, \tag{19}$$

and the feature level x_i can be derived from these auxiliary variables (if needed) as:

$$x_i = \sum_{j=0}^{k_i} (x_i^{j+1} - x_i^j) \mu_i^j. \tag{20}$$

Binary Features. We assume, w.l.o.g., that all splits on binary features send values 0 to the left branch and values 1 to the right branch. Let $\mathcal{V}_{ti}^I$ be the set of all vertices splitting on a binary feature $i \in I_{\mathrm{B}}$. The consistency of this feature value can be ensured as follows:

$$x_i \leq 1 - y_{tl(v)} \qquad t \in \mathcal{T}, v \in \mathcal{V}_{ti}^I \tag{21}$$
$$x_i \geq y_{tr(v)} \qquad t \in \mathcal{T}, v \in \mathcal{V}_{ti}^I \tag{22}$$
$$x_i \in \{0, 1\}. \tag{23}$$

Categorical Features. Let k_i be the number of possible categories for feature $i \in I_{\mathrm{C}}$. Let ν_i^j be a variable that will take value 1 if x_i belongs to category $j \in \{1, \ldots, k_i\}$ and 0 otherwise. Decision trees usually handle categorical variables through one-vs-all splits, sending the samples of a given category j to the right branch and the rest of the samples to the left. Let $v \in \mathcal{V}_{tij}^I$ be the set of vertices

splitting category j of feature i. The consistency of this feature through the forest is modeled as:

$$\nu_i^j \leq 1 - y_{tl(v)} \qquad j \in C_i, t \in \mathcal{T}, v \in \mathcal{V}^I_{tij} \tag{24}$$
$$\nu_i^j \geq y_{tr(v)} \qquad j \in C_i, t \in \mathcal{T}, v \in \mathcal{V}^I_{tij} \tag{25}$$
$$\nu_i^j \in \{0,1\} \qquad j \in C_i \tag{26}$$
$$\sum_{j \in C_i} \nu_i^j = 1. \tag{27}$$

Theorem 2 Formulation (8–27)
(i) yields feature values that are consistent with all splits in the forest;
(ii) involves only $\mathcal{O}(N_v)$ non-zero terms in the constraint matrix overall;
(iii) achieves an equal or tighter linear relaxation than the decision logic constraints used in Cui et al. (2015).

Using a reduced number of integer variables is usually beneficial for computational performance. As discussed in the supplementary material, as a consequence of the integrality of the $\boldsymbol{\lambda}$ and $\mathbf{y}$ variables, the domain of the $\mathbf{x}$, $\boldsymbol{\nu}$ variables in Formulation (8–27) can be relaxed to the continuous interval $[0,1]$ while retaining integrality of the linear-relaxation solutions. We also show in the supplementary material how to efficiently handle ordinal features (or, generally, any ordered feature in which the open intervals between successive levels bear no meaning), multivariate splits on numerical features (Brodley & Utgoff, 1995), and combinatorial splits involving several categories for categorical features.

3.3. Objective Function

In a similar fashion as Gower's distance (Gower, 1971), we use a general objective $f(\mathbf{x}, \boldsymbol{\mu}, \boldsymbol{\nu}) = f^{\text{N}}(\boldsymbol{\mu}) + f^{\text{B}}(\mathbf{x}) + f^{\text{C}}(\boldsymbol{\nu})$ which contains different terms to model the objective contributions of the numerical, binary, and categorical features. Moreover, as actionability depends on each feature and direction of action, we will use asymmetric extensions of common distances metrics with feature-dependent weights.

The model presented in the previous sections gives a direct access to the values of the **binary** and **categorical** features through $\mathbf{x}$ and $\boldsymbol{\nu}$. We can therefore generally associate any discrete cost value for each of these choices:

$$f^{\text{B}}(\mathbf{x}) = \sum_{i \in I_{\text{B}}} (c_i^{\text{TRUE}} x_i + c_i^{\text{FALSE}} (1 - x_i)) \tag{28}$$
$$f^{\text{C}}(\boldsymbol{\nu}) = \sum_{i \in I_{\text{C}}} \sum_{j \in C_i} c_i^j \nu_i^j. \tag{29}$$

The cost coefficients corresponding to the origin state $\hat{\mathbf{x}}$ typically have zero cost, though negative values could be used if needed to model possible feature states that appear more desirable.

Numerical features can be directly accessed through the $\mathbf{x}$ variables defined in Equation (20), or indirectly through the $\boldsymbol{\mu}$ variables. Most classical objectives used in counterfactual explanations can be directly expressed from $\mathbf{x}$, but modeling via $\boldsymbol{\mu}$ can in some cases lead to a better linear relaxation (e.g., for l_0). To that end, we add the origin level $\hat{x}_i$ (with index denoted as $\hat{j}_i$) to the list of hyperplane levels defining the $\boldsymbol{\mu}_i$ variables. The l_0, l_1 and l_2 objectives with asymmetric and feature-dependent weights $\{c_i^-, c_i^+\}$ can then be expressed as:

$$l_0 : \begin{cases} f_0^{\text{N}}(\boldsymbol{\mu}) = \sum_{i \in I_{\text{N}}} (c_i^- z_i^- + c_i^+ z_i^+) & \\ z_i^- \geq 1 - \mu_i^{j-1}, z_i^+ \geq \mu_i^j & i \in I_{\text{N}}, j = \hat{j}_i \\ z_i^- \in \{0,1\}, z_i^+ \in \{0,1\} & i \in I_{\text{N}} \end{cases} \tag{30}$$

$$l_1 : \begin{cases} f_1^{\text{N}}(\boldsymbol{\mu}) = \sum_{j=0}^{k_i} (\phi_i^{j+1} - \phi_i^j) \mu_i^j & \\ \text{with parameter } \phi_i^j = c_i^- \max(\hat{x}_i - x_i^j, 0) & \\ \qquad\qquad + c_i^+ \max(x_i^j - \hat{x}_i, 0) & \end{cases} \tag{31}$$

$$l_2 : \begin{cases} f_2^{\text{N}}(\mathbf{x}) = \sum_{i \in I_{\text{N}}} \left(c_i^- (z_i^-)^2 + c_i^+ (z_i^+)^2\right) & \\ z_i^+ + z_i^- = x_i - \hat{x}_i & i \in I_{\text{N}} \\ z_i^+ \in \{0,1\},\ z_i^- \in [0,1] & i \in I_{\text{N}} \end{cases} \tag{32}$$

Finally, observe that Equation (31) can be extended to model any objective expressed as a piecewise-linear convex function by defining different $\boldsymbol{\phi}$ parameters and introducing extra $\boldsymbol{\mu}$ variables for any additional breakpoint needed.

Overall, our model gives an extensible framework for efficiently modeling most existing data types, decision-tree structures, and objectives. As it mathematically represents the space of all feasible counterfactual explanations, solving it to optimality using state-of-the-art MILP solvers for the objective of choice permits to locate optimal explanations.

3.4. Domain Knowledge and Actionability

As seen in the previous sections, Formulation (8–27) accounts for different data types (e.g., numerical, binary and categorical) without transformation. For categorical variables in particular, this effectively ensures that counterfactual explanations respect the structure of the data (select exactly one category). Our model's flexibility also permits us to integrate, as needed, additional domain knowledge and actionability requirements through linear and logical constraints. Table 1 summarizes several of these constraints based on recent proposals from the literature. In the next section, we will detail our proposal to exploit isolation forests

Table 1. Domain knowledge and actionability constraints

Domain Knowledge	**Constraints**
Fixed features	$x_i = \hat{x}_i, \mu_i = \hat{\mu}_i, \nu_i = \hat{\nu}_i$
Monotonic features	$x_i \geq \hat{x}_i, \mu_i \geq \hat{\mu}_i, \nu_i \geq \hat{\nu}_i$
Known **linear relations** between features (i.e., joint actionability – Venkatasubramanian & Alfano 2020)	$A(x_i - \hat{x}_i) \leq \mathbf{b}$
Known **logical implications** between features,	
Example for binary features $(x_1 = \text{TRUE}) \Rightarrow (x_2 = \text{TRUE})$	$x_2 \geq x_1$
Example for categorical features $x_1 \in \{\text{CAT1}, \text{CAT2}\} \Rightarrow x_2 \in \{\text{CAT3}, \text{CAT4}\}$	$\nu_2^3 + \nu_2^4 \geq \nu_1^1 + \nu_1^2$
Resource constraints (e.g., time) as modeled by additional functions $g_i(\mathbf{x}, \boldsymbol{\nu}, \boldsymbol{\mu})$	$g_i(\mathbf{x}, \boldsymbol{\nu}, \boldsymbol{\mu}) \leq b_i$

to ensure counterfactual explanations plausibility within our mathematical framework.

3.5. Isolation Forests for Plausibility

We propose to rely on isolation forests (Liu et al., 2008) for a fine-grained representation of explanation plausibility. Isolation forests are trained to return an outlier score for any sample, inversely proportional to its average path depth within a set of randomized trees grown to full extent on random sample subsets. Therefore, constraining this average depth controls the outlier score (and consequently the plausibility) of the counterfactual explanation.

To include this constraint, we train the isolation forest $\mathcal{T}_\text{I}$ on the training samples from the target class. Then, we mathematically express it through Formulation (8–12) and collect the $\boldsymbol{\mu}$ levels from the union of the two forests in Equations (15–19). Lastly, we constrain the average depth in $\mathcal{T}_\text{I}$ as follows:

$$\sum_{t \in \mathcal{T}_\text{I}} \sum_{v \in \mathcal{V}_t^L} d_{tv} y_{tv} \geq \delta |\mathcal{T}_\text{I}|, \tag{33}$$

where d_{tv} represents the depth of a vertex v in tree t, and δ is a fixed threshold defining the average depth under which samples are declared as outliers. In our experiments, we will set this threshold to capture 10% of the training data as an outlier, and therefore we seek a counterfactual explanation typical of the 90% most common cases of the target class.

4. Computational Experiments

We conduct an extensive experimental campaign to fulfill two main goals:

- Evaluating the performance of our approach in terms of CPU time and solution quality. We measure the time needed to find optimal solutions on different data sets and evaluate the impact of the number of trees in the ensemble and their depth. We also compare the quality of our counterfactual explanations relatively to previous algorithms on a common objective (l_1 distance).
- Assessing the impact of the *plausibility* constraints obtained from isolation forests on the tractability of the model and the quality of the counterfactuals.

Our algorithm, referred to as OCEAN (Optimal Counterfactual Explanations) in the remainder of this paper, has been developed in Python 3.8 and can be readily executed from a single script that builds the most suitable mathematical model for the data set at hand. The complete data and source code needed to reproduce our experiments is provided at `https://github.com/vidalt/OCEAN`. The supplementary material of this paper also includes additional detailed results.

We use scikit-learn v0.23.0 for training random forests and Gurobi 9.1 (via gurobipy) for solving the mathematical models. All experiments have been run on four threads of an Intel Core i9-9880H 2.30GHz CPU with 64GB of available RAM, running Ubuntu 20.04.1 LTS.

We now discuss the preparation of the data and describe each experiment. We limit the scope of our experiments to the search of a single —optimal— explanation for each subject. If needed, diverse explanations could be generated by iteratively applying our framework, collecting its solution, and excluding it in subsequent iterations via an additional linear constraint or penalty term.

4.1. Data Preparation

We conduct our experiments on eight data sets representative of diverse applications such as loan approval, socio-economical studies, pretrial bail, news performance prediction, and malware detection. Table 2 reports their number of samples (n), number of features (total = p, numerical = p_N, binary = p_B, and categorical = p_C), and source of origin.

All these data sets include heterogeneous feature types. For data sets AD, CC and CP, we used the same preprocessing

Table 2. Characteristics of the data sets

Data set	n	p	p_N	p_B	p_C	Src.
AD: Adult	45222	11	5	2	4	UCI
CC: Credit Card Default	29623	14	11	3	0	UCI
CP: COMPAS	5278	5	2	3	0	ProPublica
GC: German Credit	1000	9	5	1	3	UCI
ON: Online News	39644	47	43	2	2	UCI
PH: Data Phishing	11055	30	8	22	0	UCI
SP: Spambase	4601	57	57	0	0	UCI
ST: Students Performance	395	30	13	13	4	UCI

as indicated in Karimi et al. (2020a). GC preprocessing was done as suggested in German credit preprocessing (2017). Finally, for ON we set "data channel" and "weekday" as categorical. The three remaining data sets are used in their original form. Each data set has been randomly split into 80% training and 20% test set.

To standardize the analyses between different data sets, we opted to set actionability constraints on two columns wherever applicable: "age" is constrained to be non-decreasing, and "sex" always stays fixed.

4.2. Performance and Scalability

In a first analysis, we evaluate the CPU time of OCEAN on each data sets for the asymmetric and weighted l_0, l_1, and l_2 objectives, without the plausibility restrictions. To simulate differences of actionability among features, the marginal weights c_i^- and c_i^+ of each feature in the objective have been independently drawn in the uniform distribution $U(0.5, 2)$. For each data set, we generated a single random forest with 100 trees limited at depth 5. We selected 20 different negative samples from the test set to serve as origin points for the counterfactual explanations. Figure 1 reports the CPU time needed to find an optimal counterfactual explanation in each case. Each boxplot represents 20 CPU time measurements, one for each counterfactual.

As visible in this experiment, OCEAN can locate *optimal* counterfactual explanations in a matter of seconds, even for data sets including over fifty numerical features with a large number of levels. Finding counterfactual explanations with variants of the l_1 or l_2 norms also appears slightly faster than with l_0. For small data sets with a few dozen features, CPU times of the order of a second are typically achieved, making our framework applicable even in time-constrained environments, e.g., for interactive tasks. Finally, as counterfactual search has a decomposable geometrical structure, CPU time could be even further reduced by additional parallel computing if the need arises.

Next, we compare the performance of OCEAN with previous approaches: the heuristic feature tweaking (FT) algorithm of Tolomei et al. (2017), the exact model-agnostic

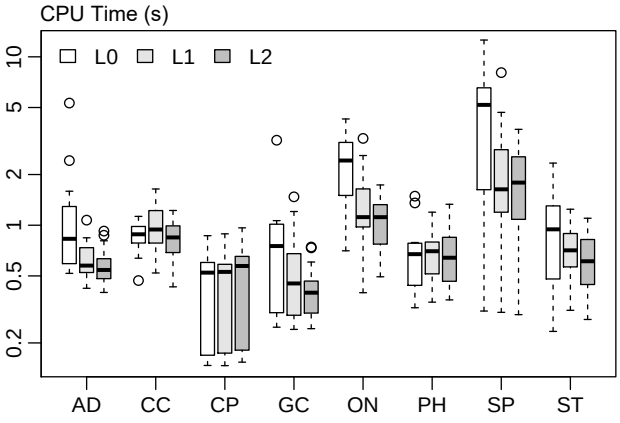

Figure 1. CPU time to find an optimal counterfactual explanations, considering different data sets and objectives

counterfactual explanations (MACE) algorithm from Karimi et al. (2020a) and the optimal action extraction (OAE) approach proposed in Cui et al. (2015) and extended in Kanamori et al. (2020). We used the implementation of FT and MACE (with precision 10^{-3}) provided at `https://github.com/amirhk/mace`, and we provide a re-implementation of OAE within the same code base as OCEAN. For this comparative analysis, we use the l_1 objective with homogeneous weights, as this is a common objective handled by all the considered methods. For a fair comparison, all the random forests and origin points for the counterfactual explanations have been saved in a serialized format, and identically loaded for each method.

We start from a baseline size of 100 trees with a maximum depth of 5 for the random forest and then extend our analysis to a varying number of trees in $\{10, 20, 50, 100, 200, 500\}$ and depth limit in $\{3, 4, 5, 6, 7, 8\}$. Figures 2 and 3 report, for each method, the mean CPU time with its 95% confidence interval as a function of these parameters for data sets AD and CC. The same figures are provided in the supplementary material for the other data sets.

As seen in these experiments, OCEAN locates optimal counterfactual explanations in a CPU time orders of magnitude smaller than MACE and OEA. Even in the most complex configurations (e.g., ST with 500 trees), OCEAN completed the optimization within two minutes, whereas MACE and OEA ran for more than five hours without terminating.

Next, we evaluate the quality of the counterfactual explanations produced by these methods. We measure, for each method and dataset, the average CPU time per explanation and the ratio $R = D/D_{\text{OPT}}$ between the sum of the l_1 distances of the 20 counterfactual explanations produced by the method (D) and that of optimal explanations (D_{OPT}). Table 3 provides these values for the baseline setting (100 trees in the random forest limited at depth 5), and similar

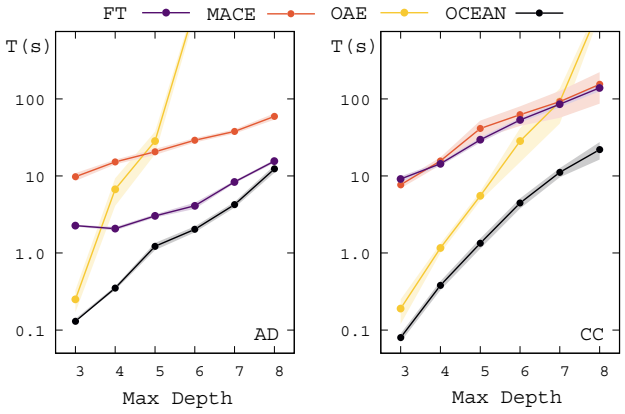

Figure 2. Comparative analysis of CPU time as a function of the maximum depth of the trees. Number of trees fixed to 100.

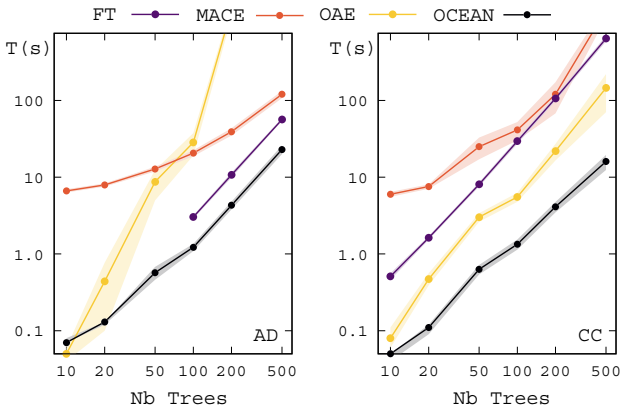

Figure 3. Comparative analysis of CPU time as a function of the number of trees in the ensemble. Maximum depth fixed to 5.

Table 3. Time and solution quality comparison

Data	FT		MACE		OAE		OCEAN	
	T(s)	R	T(s)	R	T(s)	R	T(s)	R
AD	3.03	15.9	20.60	1.1	28.37	1.0	1.22	1.0
CC	29.44	10.2	41.25	1.2	5.52	1.0	1.34	1.0
CP	22.68	4.5	15.82	1.0	0.38	1.0	0.52	1.0
GC	16.26	4.8	19.03	1.0	5.08	1.0	1.16	1.0
ON	10.05	31.7	>900	—	>900	—	2.97	1.0
PH	10.95	1.4	>900	—	0.94	1.0	0.52	1.0
SP	NA	—	>900	—	>900	—	2.73	1.0
ST	NA	—	>900	—	69.64	1.0	1.10	1.0

results are provided in the supplementary material for the other settings. A value of "NA" means that no feasible counterfactual explanation has been found, whereas ">900" means that the CPU time limit of 900 seconds was exceeded.

As observed in this experiment, it is notable that the CPU time of OCEAN is shorter or comparable to that of the FT heuristic. Yet, despite its similar speed, FT does not guarantee optimality and effectively reached, on average, distance values that are 1.4 to 31.7 times greater than the optima. It also regularly failed to find feasible counterfactual explanations (i.e., attaining the desired class) for SP and ST. These observations confirm the fact that heuristic techniques for counterfactual search can provide explanations that are considerably more complex than needed, and sensitive to the subject or subject group. In contrast, optimal counterfactual explanations are deterministic and fully specified through their mathematical definition —independently of the search approach designed to find them— providing us greater control and accountability. Arguably, "optimal" counterfactual explanations should be gradually established as a requirement for transparency and trustworthiness.

Finally, the computational efficiency of OCEAN's can be explained, in part, by the sparsity of its mathematical model. For data set AD with baseline parameters, OEA's formulation included 163,605 non-zero terms, whereas OCEAN included only 18,330. This difference becomes even more marked as the maximum depth increases. With a maximum tree depth of 8, the number of non-zero terms rises to 3,223,586 for OEA compared to 127,755 for OCEAN. This permits integrating additional linear constraints, logical constraints, and a wide range of actionability definitions (Table 1). The next section will show how to profit from this performance gain to model fine-grained plausibility restrictions through isolation forests.

4.3. Isolation Forests for Plausibility

We finally evaluate how isolation forests can ensure better plausibility in OCEAN's counterfactual explanations. Figure 4 first provides an illustrative example of our method on a small case. As observed in this example, random forests tend to make arbitrary class choices in low-density regions. Without any plausibility constraint, the counterfactual algorithm exploits the proximity to one such region to find a nearby counterfactual explanation. However, this explanation is not plausible as it represents an outlier among the samples of the desired class.

Introducing additional isolation forest constraints in our mathematical model as suggested in Section 3.5 permits us to circumvent this issue, as it successfully restricts the search for counterfactual examples to the core of the distribution of the desired class. Our restriction, therefore, builds on the same logic as the convex density constraints based on Gaussian mixtures proposed in (Artelt & Hammer, 2020), though it has the general advantage to be distribution-agnostic and applicable to most data types.

To evaluate the impact of these plausibility constraints in our model, we conduct a final experiment which consists of measuring the prevalence of plausible explanations (P), the cost of the explanations (C), and the computational effort (T)

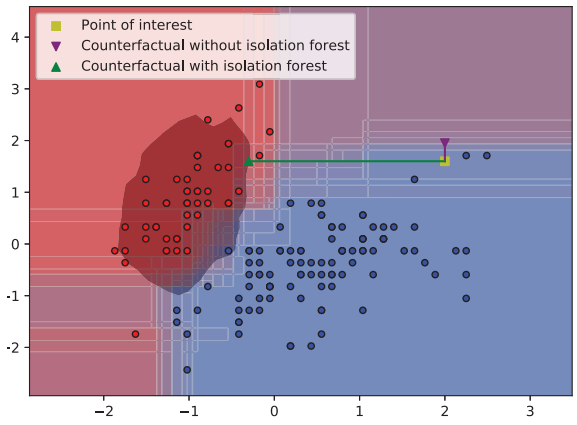

Figure 4. Isolation forests in counterfactual explanations for the Iris data set: x-axis = "sepal length", y-axis = "sepal width"

of OCEAN with and without isolation-forest constraints. The results of this analysis are reported in Table 4, with additional details in the supplementary material.

Table 4. Impact of the plausibility constraints in OCEAN

Data set	**OCEAN-noIF**			**OCEAN-IF**		
	P	C	T(s)	P	C	T(s)
AD	55%	0.21	0.75	100%	0.40	1.82
CC	0%	0.03	0.99	100%	0.56	1.54
CP	25%	0.12	0.44	100%	0.57	0.85
GC	25%	0.09	0.60	100%	1.13	1.71
ON	100%	0.01	1.34	100%	0.01	1.64
PH	35%	0.78	0.36	100%	2.40	1.81
SP	100%	0.02	2.71	100%	0.02	4.43
ST	40%	1.18	0.62	100%	1.63	1.36

As seen in this experiment, plausibility comes with extra costs that depend on the data set. Nevertheless, one cannot refer to a trade-off between cost and plausibility since misleading targets do not necessarily help to fulfill any given goal and may even trigger additional losses. Finding plausible and actionable explanations should therefore be the over-arching goal in counterfactual search. Our experiments also demonstrate that unconstrained solutions are rarely plausible and that dedicated constraints (e.g., through isolation forests) should be set up. Finally, the computational effort of OCEAN has roughly doubled due to the addition of the isolation forests in the model. This appears to be a reasonable increase given the current efficiency of OCEAN and the high practical importance of fine-grained plausibility constraints.

5. Discussion

We have shown that it is possible to generate optimal plausible counterfactual explanations at scale for tree ensembles through careful integration of isolation forests and mathematical programming. By doing so, we have circumvented a major trustworthiness and accountability issue faced by heuristic approaches, and provided mathematical guarantees for tree ensembles interpretation. Our contribution also includes a modeling toolkit that can be used as a building block for a disciplined evaluation of counterfactual search models, plausibility constraints, and actionability paradigms in various applications.

The research perspectives are numerous. From a methodology standpoint, one can always challenge tractability limits and attempt to apply the OCEAN methodology to increasingly larger data sets and tree ensembles. To that end, we suggest to investigate new formulations and valid inequalities, as well as model compression and geometrical decomposition strategies (see, e.g., Vidal & Schiffer, 2020) which have the potential to speed up the solution process. We also recommend pursuing a disciplined evaluation of compression and explanation of white-box models through mathematical programming lenses, as performance guarantees are critical for a fair access to algorithmic recourse.

Acknowledgements

This research has been partially supported by CAPES, CNPq [grant number 308528/2018-2] and FAPERJ [grant number E-26/202.790/2019] in Brazil.

References

Artelt, A. and Hammer, B. Convex density constraints for computing plausible counterfactual explanations. In *International Conference on Artificial Neural Networks*, pp. 353–365. Springer Cham, 2020.

Article 29 Data Protection Working Party. Guidelines on Automated individual decision-making and profiling for the purposes of Regulation 2016/679, 2017.

Barocas, S., Selbst, A., and Raghavan, M. The hidden assumptions behind counterfactual explanations and principal reasons. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 80–89. ACM, 2020.

Bixby, R. A brief history of linear and mixed-integer programming computation. *Documenta Mathematica*, pp. 107–121, 2012.

Breiman, L. Random forests. *Machine Learning*, 45(1): 5–32, 2001.

Brodley, C. and Utgoff, P. Multivariate decision trees. *Machine Learning*, 19(1):45–77, 1995.

Cui, Z., Chen, W., He, W., and Chen, Y. Optimal action extraction for random forests and boosted trees. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 179–188, 2015.

German credit preprocessing, 2017. URL https://www.kaggle.com/uciml/german-credit.

Gower, J. A general coefficient of similarity and some of its properties. *Biometrics*, 27(4):857–871, 1971.

Grotzinger, S. and Witzgall, C. Projections onto order simplexes. *Applied Mathematics and Optimization*, 270(12): 247–270, 1984.

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. A survey of methods for explaining black box models. *ACM Computing Surveys*, 51(5):93:1–93:42, 2018.

Jeroslow, R. and Lowe, J. Modelling with integer variables. *Mathematical Programming Study*, 22:167–184, 1984.

Kanamori, K., Takagi, T., Kobayashi, K., and Arimura, H. DACE: Distribution-aware counterfactual explanation by mixed-integer linear optimization. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 2855–2862, 2020.

Karimi, A.-H., Barthe, G., Balle, B., and Valera, I. Model-agnostic counterfactual explanations for consequential decisions. In Chiappa, S. and Calandra, R. (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 895–905. PMLR, 2020a.

Karimi, A.-H., Barthe, G., Schölkopf, B., and Valera, I. A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. *arXiv:2010.04050*, 2020b.

Karimi, A. H., Schölkopf, B., and Valera, I. Algorithmic recourse: From counterfactual explanations to interventions. *arXiv:2002.06278*, 2020c.

Liu, F., Ting, K., and Zhou, Z.-H. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

Lucic, A., Oosterhuis, H., Haned, H., and de Rijke, M. FOCUS: Flexible optimizable counterfactual explanations for tree ensembles. *arXiv:1911.12199*, 2019.

Mahajan, D., Tan, C., and Sharma, A. Preserving causal constraints in counterfactual explanations for machine learning classifiers. *arXiv:1912.03277*, 2019.

Mothilal, R., Sharma, A., and Tan, C. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, FAT*'20, pp. 607–617, New York, NY, USA, 2020. Association for Computing Machinery.

Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

Russell, C. Efficient search for diverse coherent explanations. *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 20–28, 2019.

Tolomei, G., Silvestri, F., Haines, A., and Lalmas, M. Interpretable predictions of tree-based ensembles via actionable feature tweaking. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 465–474, New York, NY, 2017.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 10–19, 2019.

Venkatasubramanian, S. and Alfano, M. The philosophical basis of algorithmic recourse. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pp. 284–293, 2020.

Verma, S., Dickerson, J., and Hines, K. Counterfactual explanations for machine learning: A review. *arXiv:2010.10596*, 2020.

Vidal, T. and Schiffer, M. Born-again tree ensembles. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 9743–9753, Virtual, 2020. PMLR.

Wachter, S., Mittelstadt, B., and Russell, C. Counterfactual explanations without opening the black box: Automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31:841, 2018.

Wolsey, L. A. *Integer Programming*. John Wiley & Sons, Hoboken, NJ, 2020. ISBN 9781119606475.