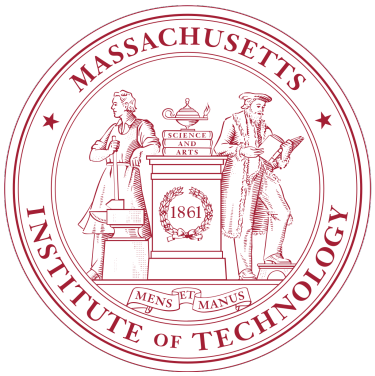
ORFA - An Optimal RuleFit Algorithm

Machine Learning Under a Modern Optimization Lens



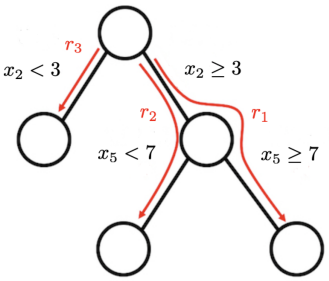
Ryan Lucas & Paul Roeseler

Abstract

We showcase a modification of the RuleFit algorithm (Friedman and Popescu (2008)) using optimal regression trees (Bertsimas, Dunn, and Paschalidis (2017)) as the rule generators. The corresponding algorithm, which we call an Optimal RuleFit Algorithm (ORFA), is capable of mining for interaction effects in an efficient number of splits. We show that this simple modification outperforms the original RuleFit algorithm on various simulated datasets, as well as 84 real-world regression datasets provided by PMLB. We go on to show that the current RuleFit setup involves training a tree and a linear regression in two separate steps. Moving beyond this disaggregated training pipeline, we propose a mixed-integer optimization formulation to solve the problem all at once. This algorithm, which we call an Integrated Optimal RuleFit Algorithm (IORFA), is a provably optimal variant of RuleFit. We show that the algorithm can be solved to optimality for problems of moderate size. Moreover, it has a comparable number of decision variables to ORT, and hence can easily be sped up using techniques such as local search or non-linear methods going forward.

1 Introduction

Linear regression models cannot mine for interaction effects in high-dimensional datasets. Accounting for interaction effects thus requires manual specification, often based on intuition or prior knowledge. Decision trees are capable of uncovering such interaction effects but struggle to account for ordinary linear effects. The RuleFit algorithm (Friedman and Popescu (2008)) uses the paths of a decision tree to generate interaction features, which become input for a linear model. The resulting RuleFit algorithm thus captures both linear and interaction effects. See below for a demonstrative example. Each rule r_i, corresponding to a unique path from root to leaf node, is encoded as a feature in a linear model, and a parameter $\hat{\delta}_i$ is estimated.



$$\begin{aligned}\hat{\boldsymbol{Y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} &+ \hat{\delta}_1(1\{x_2 \geq 3\} \cdot 1\{x_5 \geq 3\}) \\ &+ \hat{\delta}_2(1\{x_5 < 7\}) \\ &+ \hat{\delta}_3(1\{x_2 < 3\})\end{aligned}$$

Despite the clear motivation of RuleFit, the algorithm faces a major drawback: decision trees require many splits to achieve strong performance, and hence produce overly sparse features for a linear model. This is because the number of observations satisfying each rule shrinks as the number of splits increases. For arbitrarily long rules, RuleFit is potentially fitting a parameter to an indicator feature, which applies to only a small fraction of observations. Moreover, rules become increasingly difficult to interpret as they increase in size. Optimal Regression Trees (Bertsimas et al. (2017)) circumvent these problems by achieving superior performance at lower depths. In light of these advances, we propose ORFA as a revivification of the RuleFit algorithm, with ORTs as the rule generators. We hypothesize that this method should outperform RuleFit, simply due to the fact that ORT yields shorter and more efficient splits than CART.

2 Method

In ORFA, we use the rules generated from an ORT as input to a linear regression model. Notice that ORTs generate rules of the form:

$$r_m(\boldsymbol{x}) = \prod_{j \in T_m} 1\left\{\boldsymbol{a}_j^\top \boldsymbol{x}_j \lessgtr b_j\right\}$$

where m is the m-th rule generated by tree T, $\boldsymbol{a}_j$ is the linear splitting criterion and b_j is the threshold at which we perform the split. For a particular observation i, these rules are fed to a linear regression model, where we combine the tree-generated rules with the original features:

$$f(\boldsymbol{x}^{(i)}) = \beta_0 + \sum_{i=1}^{p} \beta_p x_p^{(i)} + \sum_{m=1}^{M} \delta_m r_m(\boldsymbol{x}^{(i)})$$

From this, we add a regularisation parameter ρ and extract coefficient estimates via least squares:

$$\begin{aligned}\left(\{\delta\}_1^M, \{\beta\}_0^p\right) &= \operatorname{argmin}_{\{\hat{\delta}\}_1^M, \{\hat{\beta}\}_0^p} \textstyle\sum_{i=1}^{n} (y^{(i)} - f(\boldsymbol{x}^{(i)}))^2 \\ &+ \rho \cdot \left(\textstyle\sum_{j=1}^{p} |\beta_j| + \sum_{m=1}^{M} |\delta_m|\right)\end{aligned}$$

The coefficients $\hat{\boldsymbol{\beta}}$ correspond to the standard features of the linear model, while the $\hat{\boldsymbol{\delta}}$ coefficients relate to the rules generated by the ORT.

3 Simulations

We evaluate linear regression, decision trees, RuleFit, and ORFA across multiple simulated datasets. Our simulations differ from one another in the inclusion of linear terms and/or interaction effects. Here we expect linear regression to be optimal when we assume a linear data generating process (DGP). Similarly, we expect decision trees such as CART or ORT to perform best when interaction effects are present. Lastly, we suspect that RuleFit and ORFA will perform best when both linear terms and interaction effects are present, as we will argue is often the case in practice. Our simulation study can be summarised as follows:

- DGP-1: Linear effects.
- DGP-2: Interaction effects.
- DGP-3: Linear and interaction effects.

We start by simulating normally distributed exogenous variables $\boldsymbol{X} \in \mathbb{R}^{n \times p}, n = 2000, p = 8$ with $\boldsymbol{X} \sim N(0, I)$, which we use to generate multiple targets that follow either DGP-1, DGP-2 or DGP-3.

3.1 DGP-1

The first target variable, $\boldsymbol{y}_1$, is generated using purely linear effects:

$$\boldsymbol{y}_1 = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

$$\boldsymbol{\beta} \in \mathbb{R}^p = \begin{bmatrix} 0.15 & -0.2 & 0 & \cdots & 0 \end{bmatrix}^T \text{ and } \boldsymbol{\varepsilon} \sim N(0, 0.1I).$$

Here $\beta_1 = 0.15$, $\beta_2 = -0.2$ control the true linear effects, while $\beta_3, ..., \beta_8$ being 0 forces these features to have no effect. Nonetheless, our model will be fit on these features in order to simulate a realistic scenario where we have noisy, uninformative features.

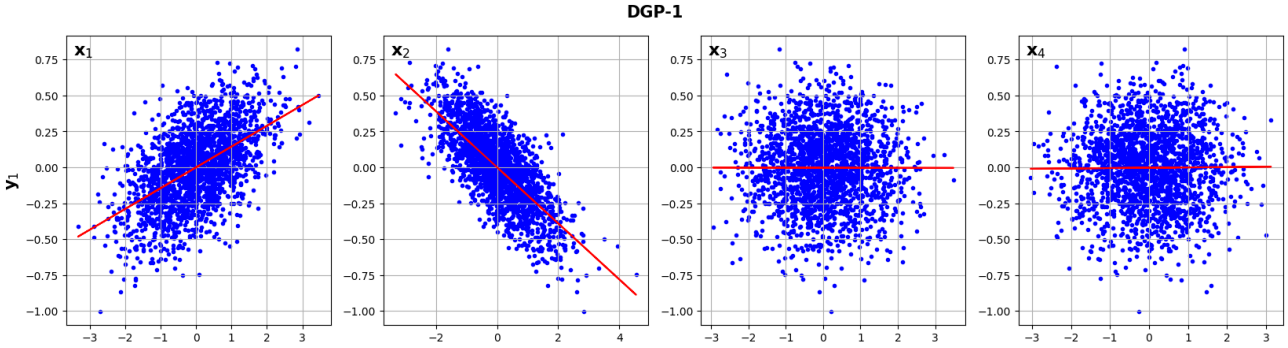


Figure 1: DGP-1 generated using $\boldsymbol{y}_1 = \boldsymbol{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}$. Here the two left plots are legitimate linear effects ($\beta_1 = 0.15, \beta_2 = -0.2$) while the right plots have no linear relationships ($\beta_3, \beta_4 = 0$).

As can be in Figure 2, when only linear effects are present, linear regression performs much better than tree-based methods CART and ORT. RuleFit and ORFA perform as well as the linear model and similarly to the true model (which matches the DGP exactly, except without noise).

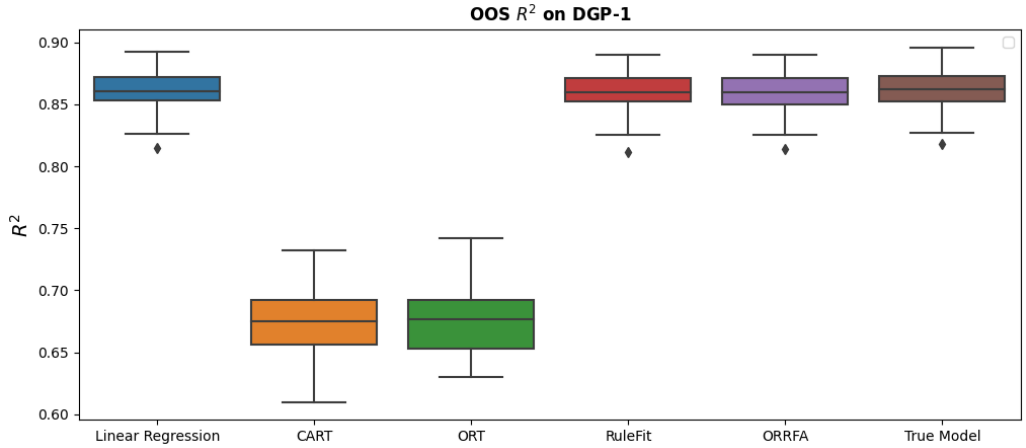


Figure 2: DGP-1: Linear regression and rule-based methods outperform tree methods with only linear effects.

3.2 DGP-2

The second target $\boldsymbol{y}_2$ is generated using purely interaction effects, which mimic a tree structure (recursive binary splitting). Here we simulate two interaction effects, one between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and another between $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$. This DGP is shown below mathematically. In Figure 3, we plot these two interaction effects, as well as a non-existent interaction effect. The points are colored such that green points represent $\boldsymbol{y}_2 = 1$.

$$\boldsymbol{y}_2 = 1\{\boldsymbol{x}_1 \leq -0.1\} \times 1\{\boldsymbol{x}_2 \geq -0.5\} + 1\{\boldsymbol{x}_3 \leq 0.4\} \times 1\{\boldsymbol{x}_4 \geq -0.3\} + \boldsymbol{\varepsilon}$$
$$\boldsymbol{\varepsilon} \sim N(0, 0.1I)$$

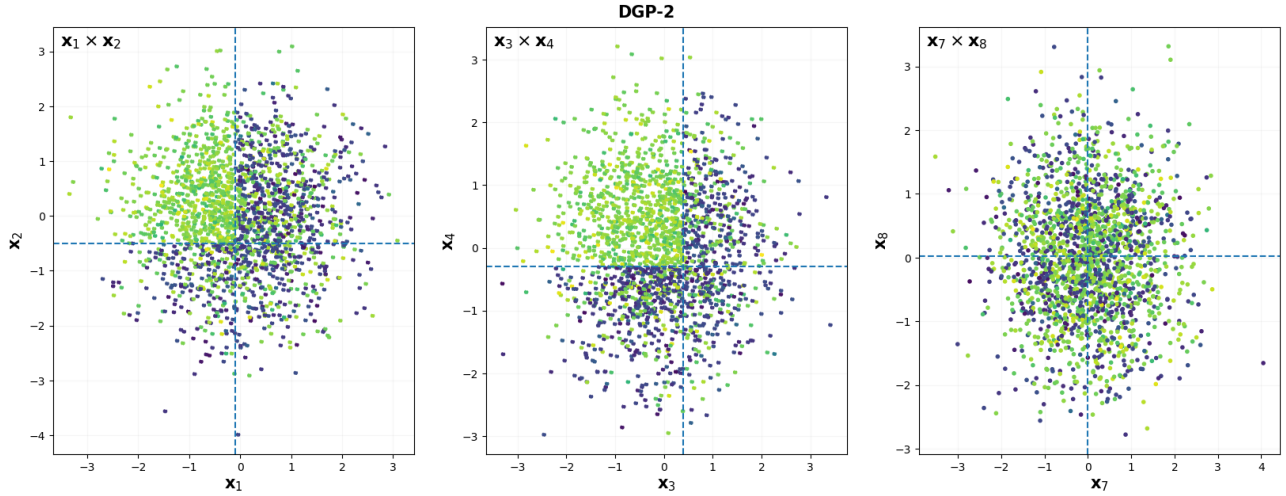


Figure 3: DGP-2 generated with $1\{\boldsymbol{x}_1 \leq -0.1\} \times 1\{\boldsymbol{x}_2 \geq -0.5\} + 1\{\boldsymbol{x}_3 \leq 0.4\} \times 1\{\boldsymbol{x}_4 \geq -0.3\}$. Here we plot two legitimate interaction effects (left, middle) and one non-existent interaction effect (right).

As can be seen in Figure 4, the tree-based methods and the rule-based methods achieve similar performance on DGP-2. Thus when interaction effects are present and linear effects are not, RuleFit and ORFA do not seem to suffer in performance, relative to models that only incorporate interaction effects. Linear regression, on the other hand, performs very poorly in the absence of linear effects.

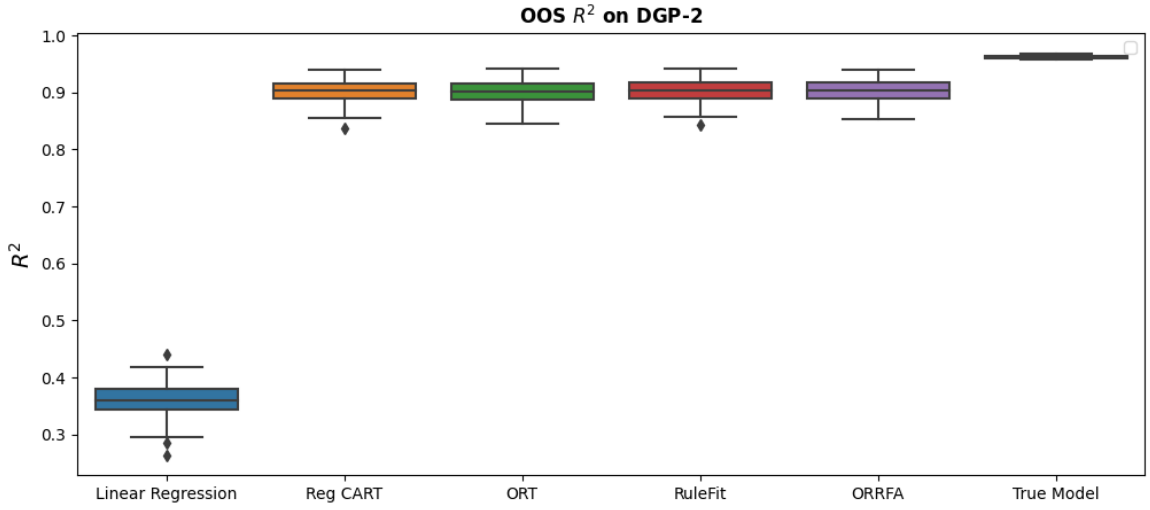


Figure 4: DGP-2 performance: When only interaction effects are present, tree-based methods and rule-based methods greatly outperform linear regression.

3.3 DGP-3

We finally consider a DGP in which we have both linear effects and interaction effects. Here the target is positively related to features $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, while we also have interaction terms involving $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$. We weight more or less towards linear effects and interaction effects using a parameter λ. In essence, we employ multiple data-generating processes, some of which have a higher presence of linear features, and others where interaction features weigh more strongly. Mathematically, the DGP is specified as follows:

$$\boldsymbol{y}_4 = \lambda \times \underbrace{(1\{\boldsymbol{x}_3 \leq 0.3\} \times 1\{\boldsymbol{x}_4 \geq -0.5\})}_{\text{interaction effects}} + (1-\lambda) \times \underbrace{(0.5\boldsymbol{x}_1 + 0.1\boldsymbol{x}_2)}_{\text{linear effects}} + \boldsymbol{\varepsilon}$$

$$\boldsymbol{\varepsilon} \sim N(0, 0.1I)$$

where $\lambda \in [0,1]$. λ closer to zero equates to more linearity and fewer interaction effects. Values closer to one indicate the opposite.

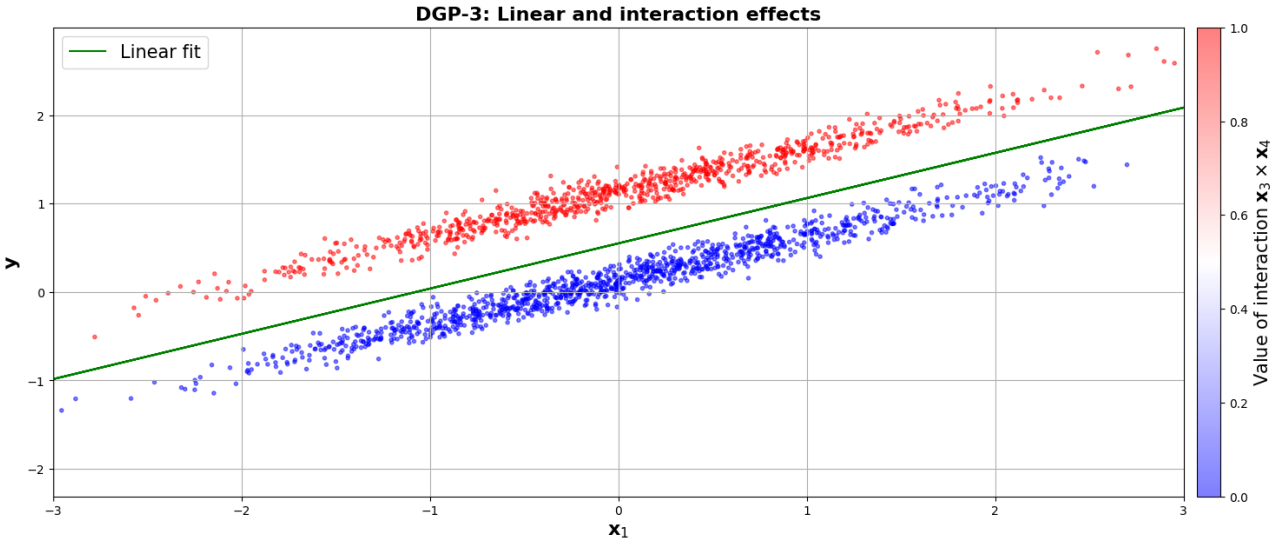


Figure 5: DGP-3 example with $\lambda = 0.5$, indicating equal parts linearity and interaction effects

As is shown in Figure 6, linear regression performs well when only linear effects are present ($\lambda = 0$) and likewise tree methods are superior when λ is closer to 1. Across all values of λ, ORFA is dominant over linear regression and ORT in terms of performance, indicating that it can capture varying degrees of linear and interaction effects.

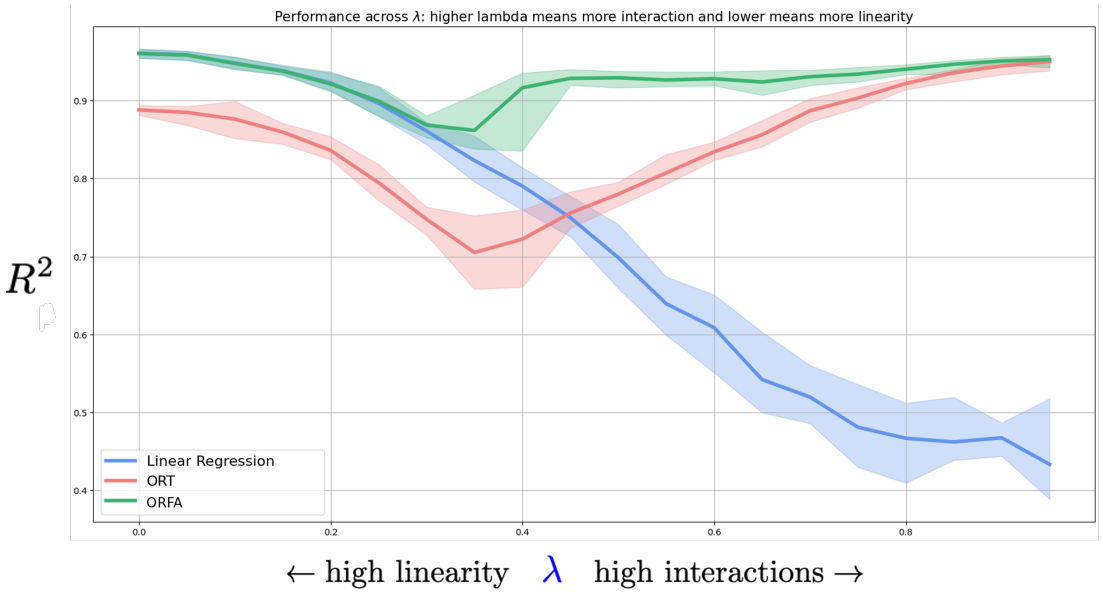


Figure 6: Performance by λ , weighting more or less towards linearity and interaction effects.

Empirical Experiments

Complementing our experiments on synthetic datasets, we also test the performance of ORFA on real-world datasets. For this purpose, we use the Penn Machine Learning Benchmark (PMLB) database collection assembled by Olson and La Cava (2017). This database contains over 200 datasets for regression problems. All datasets have been pre-processed so that they can be directly used. Still, we exclude 120 datasets to reduce the runtime of our benchmark experiments. We excluded datasets that have more than 10000 observations, 50 features, or are based on simulated data.

On the remaining 84 real-world datasets, we evaluate ORFA against various benchmarks. The results are compared to those of RuleFit, linear regression, CART, and ORT. For all tree-based methods, we select the same set of hyperparameters to ensure comparable results. In order to produce stable results, we apply all methods 5 times to each dataset, using each time a different training and test split. An overview of the results of these experiments is given in Table 1.

		Lin. Regression	CART	ORT	RuleFit	ORFA
Dataset	**CPU**	0.721	0.951	0.956	**0.976**	**0.978**
	Automobile	0.759	0.847	**0.879**	0.806	**0.874**
	Rabe	0.984	0.882	0.882	**0.986**	**0.987**
	Puma	0.375	0.567	**0.601**	0.571	**0.608**
	PW	0.710	0.780	0.762	**0.820**	**0.822**
	Wind	**0.754**	0.663	0.667	**0.754**	0.753
	Sleep Apnea	0.193	**0.845**	0.852	0.836	**0.844**
	Bodyfat	**0.974**	0.944	0.946	**0.974**	0.973
	CPU Small	0.707	0.936	0.947	**0.963**	**0.969**
	FRI	0.265	0.580	**0.684**	0.614	**0.749**
	Chatfield	0.851	0.704	0.679	0.781	**0.750**
	Geyser	**0.800**	**0.775**	0.755	0.779	0.762
⋮	⋮	⋮	⋮	⋮	⋮	
	Average	0.424	0.625	0.633	**0.707**	**0.724**

Table 1: Out-of-sample R^2 across 84 real-world regression datasets provided by PMLB. The best performer on each dataset is highlighted in **blue**, while **purple** denotes the second best.

In general, we observe that ORFA achieves a better average out-of-sample R^2 than RuleFit across the 84 datasets. Beyond that we were able to show that the rules generated by ORFA are smaller than those of RuleFit, resulting in an overall more interpretable method. Compared to linear regression, CART, and ORT, the out-of-sample R^2 of ORFA is 0.3, 0.1, and 0.09 higher, respectively. The overall poor performance of linear regression in this experiment is not necessarily surprising considering the simulation results from section 3. Recall that while trees provide (somewhat) stable results in data environments that have either linear effects or interaction effects, linear models struggle when interaction effects predominate (Figure 6).

The overall findings of this benchmark study clearly show that methods accounting for both linear and interaction effects in the data are superior to methods focusing on only one of the two. The magnitude of these differences is quite surprising. Especially when considering that the rule-based methods are still inherently interpretable. These findings, consequently, do not only show that our extension of the original RuleFit algorithm leads to improved performance but also motivate the question of why rule-based methods are not more widely used. Real-world data nearly always include both linear and interaction effects. OFRA provides an easy-to-apply, interpretable, and automated approach to account for these effects while reaching higher performance than all other tested methods.

4 Beyond a Heuristic: Integrated Optimal RuleFit Algorithm

Notice that RuleFit and ORFA adopt a disaggregated training pipeline. Both algorithms start by fitting a tree and thereafter feed the resultant rules to a linear regression model. However, this invariably leads to a sub-optimal solution, since the rules are not purpose-fit for use in linear regression. Consider even that the tree splits may be highly collinear, and hence in the disaggregated case, the tree may look for splits that have already been captured by a linear relationship with the original features. In light of this drawback, we propose an Integrated Optimal RuleFit (IORFA) algorithm, which fits the tree and the linear regression all in one step. We do this by considering an alternative MIO formulation to ORTs, using a regression loss as the objective function. The IORFA optimization problem is as follows:

$$
\begin{aligned}
&\min_{\boldsymbol{\beta},\boldsymbol{\delta},\boldsymbol{z}} \sum_i \left(y_i - \boldsymbol{x}_i^T \boldsymbol{\beta} - \sum_{t \in L} \delta_t z_{i,t} \right)^2 \\
&\text{subject .to.} \\
&N_t = \sum_{i=1}^{n} z_{it}, \quad \forall t \in T_L \\
&\boldsymbol{a}_m^\top \boldsymbol{x}_i \geq b_t - (1 - z_{it}), \quad i = 1, \ldots, n, \quad \forall t \in T_B, \quad \forall m \in A_R(t), \\
&\boldsymbol{a}_m^\top (\boldsymbol{x}_i + \boldsymbol{\epsilon}) \leq b_t + (1 + \epsilon_{\max})(1 - z_{it}), \; i = 1, \ldots, n, \forall t \in T_B, \forall m \in A_L(t) \\
&\sum_{t \in T_L} z_{it} = 1, \quad i = 1, \ldots, n, \\
&z_{it} \leq l_t, \quad \forall t \in T_L, \\
&\sum_{i=1}^{n} z_{it} \geq N_{\min} l_t, \quad \forall t \in T_L, \\
&\sum_{j=1} a_{jt} = d_t, \quad \forall t \in T_B, \\
&0 \leq b_t \leq d_t, \quad \forall t \in T_B, \\
&d_t \leq d_{p(t)}, \quad \forall t \in T_B \backslash \{1\} \\
&z_{it}, l_t \in \{0, 1\}, \quad i = 1, \ldots, n, \quad \forall t \in T_L, \\
&a_{jt}, d_t \in \{0, 1\}, \quad j = 1, \ldots, p, \quad \forall t \in T_B
\end{aligned}
$$

The main modification compared to the formulation of ORT is in the objective function. Here we optimize over linear regression coefficients $\boldsymbol{\beta}$, as well as the tree binning variables $\boldsymbol{z}$, and the rule coefficients $\boldsymbol{\delta}$. One can think of δ_t as fitting a linear coefficient to each of the leaf nodes $t \in L$ of the tree. This is equivalent to fitting a linear coefficient to each group identified by sequential splits of the tree. One may ask whether optimizing over these different types of decision variables introduces additional complexity. However, notice that we only introduce $p + M$ decision variables to the problem, which for most problems are a fraction of the size of $\boldsymbol{z}$, ORTs primary decision variables. A contrast between RuleFit and ORFA versus IORFA is provided in Figure 7.

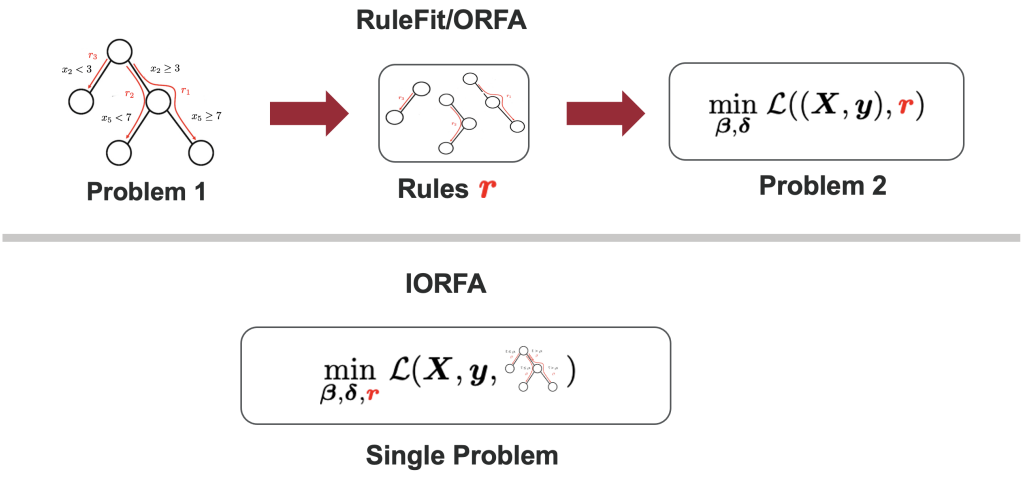


Figure 7: Example of the RuleFit/ORFA training pipeline versus the IORFA training pipeline.

The first comparisons on simulated and real-world datasets between this integrated approach and the previously presented methods are very promising. Not only can the MIO formulation be solved in a reasonable time for small to medium-sized data sets, but the performance in various simulations is also exceeding that of the other presented methods. The results for the same simulations conducted in DGP-3 with a random λ and 100 iterations are shown in Figure 8. We expect that the time required for solving the MIO formulation can be significantly reduced by using warm starts, a local search variant, or cutting plane methods.

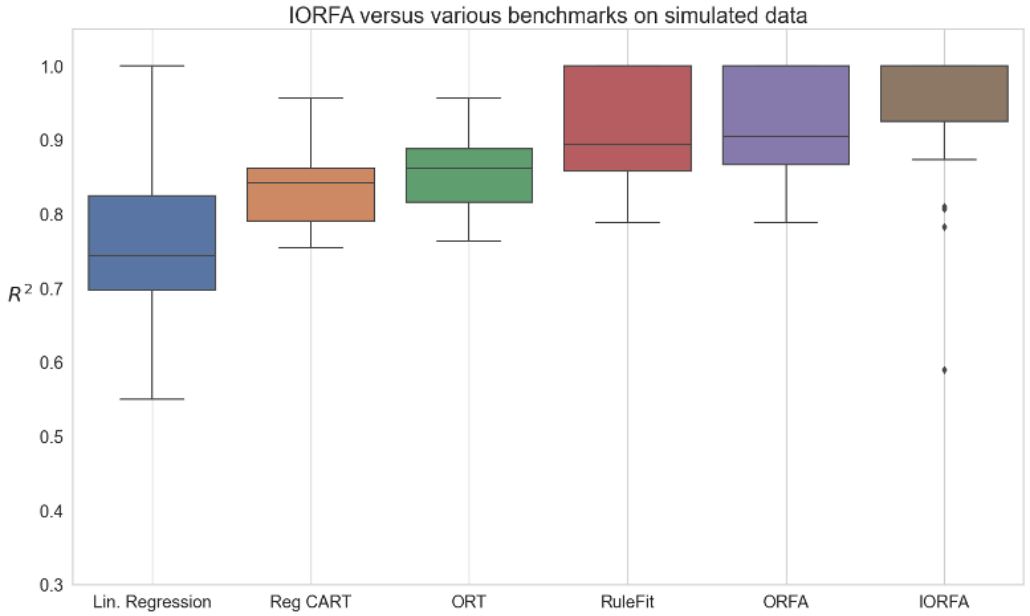


Figure 8: Initial results of IORFA on DGP-3 with random λ. Experiments are repeated 100 times.

Conclusion

The purpose of our project was to determine whether the recently emerging optimal tree algorithms can be used to extend the original RuleFit method. We found that incorporating ORT-generated rules instead of CART-based ones not only leads to performance improvement but also to smaller rules that can be interpreted more easily. ORFA is, consequently, both more interpretable and performant than RuleFit. Compared to algorithms such as CART, ORT, and linear regression, ORFA produces substantially better results.

In light of these promising findings, we developed a first version of an integrated algorithm, which solves the linear optimization problem and the rule generation simultaneously and optimally. The first results for this algorithm seem promising. However, both the performance needs to be evaluated in more detail and the solving process needs to be sped up. Only then can this method be applied to most real-world datasets in a reasonable amount of time.

Beyond these remarks, there are some limitations regarding the here presented experiments and results. First, the simulated datasets are generated on the basis of simplified relationships that may not be reflective of the complex interactions in real-world data. Moreover, not all non-linear effects are interaction based. ORFA may consequently not capture all types of non-linear relationships. Additionally, while we tested the performance of ORFA on a great number of real-world datasets, with such a large number of experiments, we're unable to process every dataset individually. Thus we rely on the appropriateness of feature engineering applied by Olson and La Cava (2017).

Contributions

We contributed equally to all aspects of the project. We both handled Simulations, Results, and the integrated algorithm together. We can't state which parts were performed by whom individually. We are roommates so this happened very naturally ☺.

References

Bertsimas, D., Dunn, J., & Paschalidis, A. (2017). Regression and classification using optimal decision trees. In *2017 ieee mit undergraduate research technology conference (urtc)* (p. 1-4). DOI: 10.1109/URTC.2017.8284195

Friedman, J. H., & Popescu, B. E. (2008, sep). Predictive learning via rule ensembles. *The Annals of Applied Statistics*, *2*(3). Retrieved from https://doi.org/10.1214%2F07-aoas148 DOI: 10.1214/07-aoas148

Olson, R. S., & La Cava, W. (2017, Dec 11). Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, *10*(1), 36. Retrieved from https://doi.org/10.1186/s13040-017-0154-4 DOI: 10.1186/s13040-017-0154-4