Informative Horseshoe Bayesian prior and Shapley values to facilitate RuleFit interpretability

Spadaccini, Giorgio

Universiteit Leiden
Statistics and Data Science
Master Thesis

Informative Horseshoe Bayesian prior and Shapley values to facilitate RuleFit interpretability

Student:
Giorgio Spadaccini

Supervisors:
Prof. Mark van de Wiel, Vrije Universiteit Amsterdam
Dr. Marjolein Fokkema, Universiteit Leiden

Defended on 13 December, 2023

Contents

Abstract

In scientific research, interpretability and high predictive performance are difficult to combine: while black-box models perform better than interpretable models, only the latter allow for transparency and inference, which are necessary tools when these models are used in decision-making or in hypothesis testing. Models such as RuleFit [1] combine the flexibility of a black-box tree ensemble with the interpretability of a sparse, LASSO linear regression. Later work substitutes Bayesian regression for the LASSO regression, thus further improving the model's prediction (Horserule, [2]). The work in this thesis was two-sided: on the one hand, we applied a different Bayesian prior (the informative Horseshoe prior) to the linear step of the RuleFit model, which can naturally take the structure of RuleFit into account. On the other hand, we used Shapley values to measure the contribution of each predictor in the RuleFit model and combined these values with the Bayesian regression to build inferential tools. The new machinery was tested on both synthetic data and the dataset from the Helius study [3].

The predictive performance of the resulting model was observed to be higher than that of the original RuleFit model, but lower than that of Horserule. Compared to Horserule, the proposed model excessively favours trees over linearity, but in doing so it more strongly enforces the choice of simpler trees. Shapley values were also compared to other importance measures mentioned in the RuleFit literature, and shown to be more accurate in reconstructing the contributions as defined in the synthetic datasets.

Keywords: RuleFit, Shapley values, Tree Ensembles, Inference, Bayesian Regression, Informative Horseshoe

Introduction

Since access to more powerful computers has become more widespread, statistical and machine learning models have been able to reach new levels of complexity, and in doing so solve more difficult tasks with higher precision. As these models typically aim at maximum predictive accuracy, they do not usually take other aspects into account. Unlike humans, for instance, most statistical models cannot incorporate factors like ethics, legality or justice in the training process. The “Correctional Offender Management Profiling for Alternative Sanctions” (COMPAS), for example, a software used in the state of New York for more than a decade, was shown to have racially biased predictions for the risk of committing a crime [4][5]. As it is essential to guarantee that statistical predictions are as free as possible from such biases, the interpretability of a model is key to performing post-hoc checks needed to ensure that a model adheres to human standards and principles.

As mentioned in [6], there are further reasons to aim for model interpretability. Often, for instance, the aim of a model is not simply that of accurately predicting a phenomenon, but also to gain insight into how the predictors and the outcome are linked. While a model often does not allow for causal attributions, it may still give clues and insight which may lead to further scientific research. Another important aspect of the interpretability of a model is that it allows us to examine its robustness against different settings or discrepancies between training and real-world data. Moreover, further insight into the model is also helpful when checking vulnerability against adversarial manipulation: in [7], Deep Neural Networks were shown to be susceptible to almost humanly imperceptible manipulations. Even tree-based ensemble learners were shown to be efficiently evaded by adversarial manipulations, in the sense that it is possible to find the smallest perturbation to apply to a datapoint in order to change its binary classification, even without having access to how the tree ensemble is defined [8]. Lastly, interpretability plays a critical role in the communication of scientific results. An ideal model would be accurate in prediction but also transparent, in terms of having a mechanism that is easily explained, and easily visualized, by producing clear pictures that give insight into the connections that the model builds between the predictors and the outcome. All the elements discussed above are ideally also inserted in an inferential framework, that can offer a scale to compare estimates against and ease the separation between noise and information. This includes tools such as hypothesis testing and credible intervals.

In this thesis, we explore RuleFit [1], an extension of a sparse linear model that enriches the predictor space with rules taken from a tree ensemble. The aim of this model is to keep easily interpretable linear effects when possible, while also enhancing prediction by adding a few simple, easily-explained rules that capture non-linearity. In doing so, one may not only obtain high predictive performance but also high interpretability. The main challenge of this machinery, however, is to enforce the use of linear terms over that of rules, since the latter are typically trained on the same data that is used for selecting the final model and therefore have an unfair advantage in terms of already fitting the training samples well. Subsequent implementations of RuleFit like HorseRule [2] address this challenge, as the LASSO regression fitted to estimate the final ensemble is replaced by a Bayesian regression with a differential Horseshoe prior that penalizes rules more strictly, according to two tunable hyperparameters.

In exploring this model and its variations, we propose an alternative prior (the Informative Horshoe prior [9]) that can naturally take into account the different nature of the predictors, and therefore needs no shrinkage parameters to be tuned in order to shrink linear terms and rules differently. Moreover, we develop tools that yield further insight into the model and quantify the importance of each predictor in the model. More specifically, we combine Shapley values and Bayesian regression to produce credible intervals for the estimated contribution of each predictor to the final prediction. An explicit formula for more direct computation of marginal Shapley values of tree ensembles is derived.

Chapter 1 contains a description of how the RuleFit model works. This includes an overview of how variable importance is measured within this context, together with the current limitations of this machinery. Chapter 2 covers the Horseshoe prior, its current use with RuleFit (HorseRule), and how it may be extended by the use of co-data, using the Informative Horseshoe prior. Chapter 3 gives an introduction to Shapley values, the TreeSHAP estimation algorithm and an explicit formula for the estimation of marginal Shapley values. It also ties in with Chapter 2, as uncertainty is included in the estimation of variable importance, and new inference-based measures are defined. The machinery described so far is applied to synthetic data in order to evaluate predictive performance, sparsity of the model and accuracy of the newly introduced importance measures. Chapter 4 shows the results of such simulations. Chapter 5 contains a small application of this model to the dataset from the Helius study [3], while Chapter 6 ends this work with a discussion of the resulting model, its advantages and its limitations. The code used to run the experiment may be found on the GitHub page linked in the Appendix, for reproducibility.

Chapter 1

An introduction to RuleFit

1.1 What is Rulefit

In 2008, Friedman and Popescu proposed their approach *RuleFit* as an extension of linear regression that can capture nonlinearity and complex interactions in an interpretable way. More specifically, it comprises of the following steps:

- A Boosted Tree Ensemble, or alternatively a Random Forest, is fitted on the dataset;
- All the rules/nodes from every tree of the ensemble are extracted;
- All these rules are coded as dummy variables and added to the predictor space;
- A LASSO linear model is fitted to the extended predictor space

The idea behind such a model is that it combines the best of both worlds: on the one hand, incorporating the rules from a tree ensemble enables us to gain some of that accuracy that is typical of a black-box method. On the other hand, the sparse use of linear terms and dummy variables allows for a more straightforward interpretation of the model.

Before the discussion is taken further, let us define some notation. Throughout this work, we consider the setting in which a dataset of n samples is used. We denote the samples by $x^{(1)}, \dots, x^{(n)}$. Each sample is a p-dimensional vector $x^{(i)} = (x_1^{(i)}, \dots, x_p^{(i)})$. A generic p-dimensional point has no specific sample index i, and is thus simply denoted by $x = (x_1, \dots, x_p)$. The generic j-th predictor is therefore x_j.

To illustrate how rules can be extracted from an ensemble of trees, we present a simplified example. A boosted tree ensemble has been fitted to five predictors, comprising only two trees, as shown in Figure 1.1. This ensemble produces six rules:

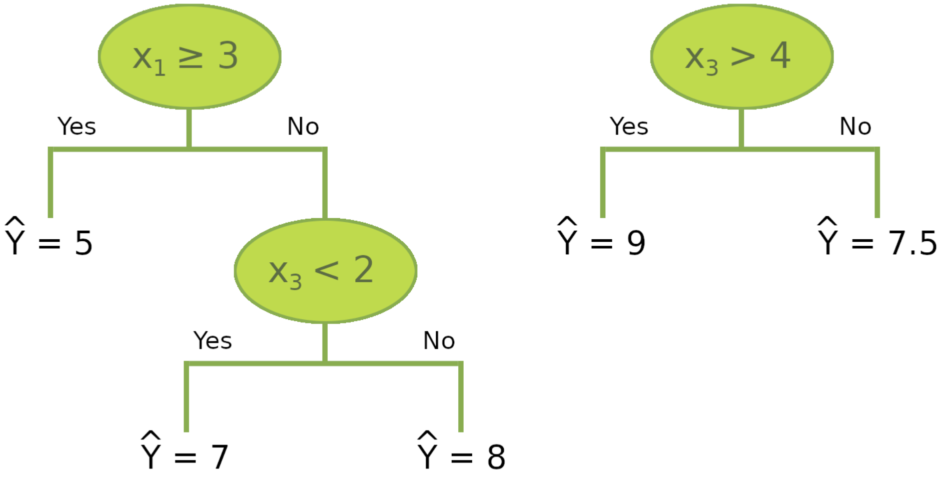

Figure 1.1: Simplified example of a boosted tree ensemble, containing only two trees of depth two and one, respectively.

$$
\begin{aligned}
&r_1(x) := I(x_1 \geqslant 3),\\
&r_2(x) := I(x_1 < 3),\\
&r_3(x) := I(x_1 < 3) \cdot I(x_3 < 2),\\
&r_4(x) := I(x_1 < 3) \cdot I(x_3 \geqslant 2),\\
&r_5(x) := I(x_3 > 4),\\
&r_6(x) := I(x_3 \leqslant 4).
\end{aligned}
$$

Table 1.1 shows how an example dataset of five samples is transformed by adding the extra components accordingly. Note how rules r_1, r_3, and r_5 would have been enough to represent the six perfectly collinear rules. However, this is not a concern since a sparse model performs selection across these collinear terms.

A possible last step in pre-treating the predictors is the winsorization of the linear terms: to avoid outliers, a continuous predictor x_j may be altered by replacing all extreme sample values with less extreme values. More specifically, x_j is substituted for:

$$
l_j(x_j) = \min(\delta_j^+, \max(\delta_j^-, x_j)) = \begin{cases} \delta_j^- & \text{if } x_j < \delta_j^- \\ x_j & \text{if } \delta_j^+ \leqslant x_j \leqslant \delta_j^+ \\ \delta_j^+ & \text{if } x_j > \delta_j^+ \end{cases}, \tag{1.1}
$$

x_1	x_2	x_3	x_4	x_5	r_1	r_2	r_3	r_4	r_5	r_6
-0.5	2	5	3.1	-1.3	0	1	0	1	1	0
3.7	1.5	-0.1	4	2.4	1	0	0	1	0	1
1.8	0.1	1	0	0.8	0	1	1	0	0	1
4.1	-0.9	2.9	2.2	2.1	0	1	0	1	0	1
3	-0.4	4.2	1.6	-0.6	1	0	0	1	1	0

Table 1.1: Example of how five exemplary datapoints in a 5-dimensional feature set would be extended to an 11-dimensional set by adding the rules defined on page 2, as taken from the tree ensemble shown in Figure 1.1.

x_1	x_2	x_3	x_4	x_5	r_1	r_2	r_3	r_4	r_5	r_6
$\mathbf{-0.27}$	**1.95**	**4.92**	3.1	$\mathbf{-1.23}$	0	1	0	1	1	0
3.7	1.5	**0.01**	**3.91**	**2.37**	1	0	0	1	0	1
1.8	0.1	1	**0.16**	0.8	0	1	1	0	0	1
4.06	$\mathbf{-0.85}$	2.9	2.2	2.1	0	1	0	1	0	1
3	-0.4	4.2	1.6	-0.6	1	0	0	1	1	0

Table 1.2: Example of how the datapoints from Table 1.1 are winsorized for $\beta = 0.025$. The numbers in bold have been winsorized.

where δ_j^- and δ_j^+ are the β-quantile and the $(1-\beta)$-quantile of $\{x_j^{(1)}, \ldots, x_j^{(n)}\}$, for a small value of β. While this step is strictly necessary, Freidman and Popescu apply this step in the interest of robustness, with a suggested default of $\beta = 0.025$. Table 1.2 shows the winsorized terms for $\beta = 0.025$. The numbers in bold are the ones that the winsorization actively changed, as they were beyond the quantiles. Note how the winsorization happens *after* the rule generation, meaning that winsorization does not change the status of a rule.

Once the predictor space is extended and winsorized, it is time to use a sparse linear regression model to select many of the rules, and possibly winsorized predictors, out of the model. This way, the rules only play a supporting role to the linear terms, whenever interactions and/or non-linearities are suggested by the data. From our example, the final model would take the form:

$$F(x) = \hat{y} = \hat{a}_0 + \sum_{k=1}^{q} \hat{a}_j r_j(x) + \sum_{j=1}^{p} \hat{b}_j l_j(x_j). \tag{1.2}$$

In this work, we refer to the terms $\hat{b}_j l_j(x_j)$ as *linear terms* and to the terms $\hat{a}_j r_j(x)$ as *rules.*

Friedman and Popescu suggest LASSO for estimating the coefficients in Equation 1.2. Since LASSO tends to favour predictors with higher variance, predictors are typically standardized before regression. Standardizing a rule r_j means dividing it by the quantity:

$$\sqrt{\overline{r}_j(1-\overline{r}_j)}, \qquad \text{where } \overline{r}_j = \frac{1}{n}\sum_{i=1}^{n} r_j(x^{(i)}). \tag{1.3}$$

The quantity $\overline{r}_j$ is the observed mean of the rule r_j on the training set and thus coincide with its observed support when the rule is $0-1$ coded as above. Dividing by the observed standard deviation, however, may be an unstable process. For this reason, Friedman and Popescu instead suggest limiting ourselves to rescaling the linear terms to make sure that they have the same variance as the average rule. More specifically, the average variance of a rule, assuming that its support $\overline{r}_j$ is uniform in $[0, 1]$, is:

$$\int_0^1 \overline{r}_j(1-\overline{r}_j)d\overline{r}_j = \Big(\frac{\overline{r}_j^2}{2} - \frac{\overline{r}_j^3}{3}\Big|_0^1 = \frac{1}{2} - \frac{1}{3} = \frac{1}{6}$$

This means that the average variance of a rule is reached when the standard deviation equals $\sqrt{1/6} \approx 0.4$. The linear terms are thus re-scaled accordingly:

$$l'_j(x_j) := 0.4\frac{l_j(x_j)}{sd(l_j(x_j))} \tag{1.4}$$

In this way, linear terms have similar variance as rules and LASSO is applied to the extended predictor space, producing the linear model in Equation 1.2.

1.2 Variable importance in a RuleFit model

As mentioned above, the goal of RuleFit is to pair accuracy levels that are typical of black-box models with higher interpretability. A common approach to obtain interpretability is variable importance, a measure that represents how much a predictor is relevant in our model's prediction of the outcome. For linear terms, variable importance may be computed as one would typically do for standardized coefficients in a linear model:

$$I_j^{(L)} := |\hat{b}_j| \cdot sd(l'_j(x_j)).$$

The analogue measure for rules is:

$$I_j^{(R)} := |\hat{a}_j| \cdot sd(r_j) = |\hat{a}_j| \cdot \sqrt{\overline{r}_j(1-\overline{r}_j)}.$$

In their initial work [1], Friedman and Popescu also discuss local measures, i.e. measures that show the contribution of a predictor to the prediction of the outcome for any given datapoint, individually. Since the contributions are additive, these are simply computed as the difference between the point-specific contribution and the average contribution, for the individual term. In other words, linear terms have a local importance measure defined as:

$$I_j^{(L)}(x) = |\hat{b}_j l_j(x_j) - \hat{b}_j \bar{l}_j| = |\hat{b}_j| \cdot |l_j(x_j) - \bar{l}_j|, \tag{1.5}$$

where $\bar{l}_j$ is the observed mean of $l_j(x_j)$ over the training data. Rules, on the other hand, have a local importance measure defined as:

$$I_j^{(R)}(x) = |\hat{a}_j r_j(x) - \hat{a}_j \overline{r}_j| = |\hat{a}_j| \cdot |r_j(x) - \overline{r}_j|. \tag{1.6}$$

The local and global measures are connected by the formula:

$$I_j^{(L)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(I_j^{(L)}(x^{(i)})\big)^2}, \qquad I_j^{(R)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\big(I_j^{(R)}(x^{(i)})\big)^2}. \tag{1.7}$$

While these importances are useful and intuitive, we also need to keep in mind that one is typically not interested in the importance of one individual rule or linear term, but rather in the importance of a predictor as a whole, which in this model would typically appear in both multiple rules and the linear term.
Friedman and Popescu propose to sum up the contributions together, while equally splitting the importance of a rule across the predictors it involves. This approach seems intuitive and compensates for rules that involve many predictors. The overall local importance of the j-th predictor for a datapoint x is therefore defined as:

$$J_j(x) := I_j^{(L)}(x) + \sum_{\substack{k \text{ s.t.} \\ x_j \in r_k}} \frac{I_k^{(R)}(x)}{m_k}, \tag{1.8}$$

where m_k denotes the number of predictors involved in the k-th rule and, with an abuse of notation, $x_j \in r_k$ is meant as "the j-th predictor is involved in the rule r_k".
A global analogue is also suggested, by replacing the local importances with the global ones:

$$J_j := I_j^{(L)} + \sum_{\substack{k \text{ s.t.} \\ x_j \in r_k}} \frac{I_k^{(R)}}{m_k}. \tag{1.9}$$

1.3 Current limitations and challenges

While the machinery described above is already quite effective at producing interpretable, accurate models, some limitations may remain. Firstly, one might be concerned with the LASSO fit: while it is true that linear terms are re-scaled to make the terms more comparable, it is important to keep in mind that this rescaling is only based on the *theoretical* average variance of the rules, which therefore does not necessarily well reflect the actual variances of the rules. Furthermore, the rules are originated from the same dataset that is also used for the linear regression, which gives an advantage to the rules in terms of fitting. This advantage should be countered for instance by reducing the shrinkage on linear terms, but the rescaling suggested by Friedman and Popescu goes in the opposite direction. Moreover, even if comparability between linear terms and rules were not a problem, it would still not be true that the rules are comparable with each other, since they have not been standardized (see for instance [10]). Friedman and Popescu do however explicitly address this, and mention that this is a deliberate choice so that rules with extreme support are penalized more than rules with a more balanced support, a choice that favours stability [1].

The variable importance measures also seem to present some limitations: for instance, the connection in Equation 1.7 between local and global importance for $I_j^{(R)}$ and $I_j^{(L)}$ does not apply to J_j: intuitively, the problem is that we cannot sum up the local importance of the different terms together, since the terms are typically are correlated. Furthermore, we have to keep in mind that these measures are all defined based on the intuition that the contributions of the rules should be equally split across the predictors involved, which does not necessarily have to be the case.

Lastly, neither the coefficients nor the importance measures can benefit from uncertainty quantification: the bias induced by the frequentist LASSO regression makes the variance of the estimation unreliable in inferential statements [11], and LASSO's Bayesian counterpart, i.e. the Laplace prior, is also inadequate for uncertainty quantification [12].

Chapter 2

Bayesian linear regression: changing the sparse model

2.1 Why go Bayesian?

As we have mentioned above, LASSO is an efficient way of performing sparse linear regression, but it also has limitations: more complex structures and relationships between the data may require coefficients to be shrunken in different ways. This is a context where Bayesian approaches thrive: within this realm, we are allowed to specify complex priors that may very accurately reflect our prior knowledge about the data or, like in this case, the desired structure of the coefficients (think, for instance, of differentiation between linear terms and rules, but also between more general and more specific rules). An advantage of using priors is that these differentiations can be formulated hierarchically in Bayesian models: to better represent heterogeneity in the predictors, we may decide to not only define a prior distribution for the coefficients of our linear model but also to establish prior distributions for the hyperparameters of said priors, as a way to indicate that some coefficients should be shrunken more than others. This represents a drastic departure from LASSO regression, which does not penalize each predictor individually and therefore tends to overshrink large effects.

Inducing specific prior knowledge is also helpful in contexts where the sample size is small, especially when we are performing regression on a large predictor space (which may very easily be our case, as we expand it by adding many binary rules). In these situations, maximum likelihood solutions (and LASSO estimates, as pointed out in [13]) tend to be infeasible or less stable. Lastly, using Bayesian regression yields not only a point estimate for the coefficients but rather the full distribution of the coefficient estimates and, therefore, of any quantity that depends on said coefficients. From this distribution, we may compute credible intervals and perform inference, tools that all go towards the direction of more transparent, interpretable machine learning.

2.2 The Horseshoe prior

The Horseshoe prior was first implemented by Carvalho, Polson and Scott in 2009. As described in [14], the Horseshoe prior is particularly good at avoiding overshrinkage of larger coefficients and being stable in its estimate.
The way this prior is defined, for a coefficient vector β, goes as follows:

$$\begin{aligned} y|X,\beta,\sigma^2 &\sim \mathcal{N}(X\beta,\sigma^2 I_n), \\ \beta_j|\lambda_j,\tau,\sigma^2 &\sim \mathcal{N}(0,\lambda_j^2\tau^2\sigma^2), \\ \lambda_j &\sim \mathcal{C}^+(0,1), \\ \tau &\sim \mathcal{C}^+(0,1), \\ \sigma^2 &\sim \sigma^{-2}d\sigma^2. \end{aligned}$$

In other words, the shrinkage of a coefficient β_j has two components: a global shrinkage, represented by the parameter τ, and a local shrinkage, represented by the parameter λ_j, which is individual to each coefficient. The idea behind this prior is that some coefficients need much more shrinkage than others, therefore λ_j determines how much the global shrinkage τ needs to be dampened or amplified.

In [2], Nalenz and Villani implement such horseshoe prior in the context of RuleFit, while also customizing the prior in a way that more aggressively shrinks coefficients to zero when they involved rules that were not as valuable for interpretability and accuracy. More specifically, they favour rules that have:

- **less extreme support:** rules with low variance (i.e. that cover almost every datapoint or that, conversely, only cover very few) might be more unstable and induce overfitting. Furthermore, the fewer rules are used, the more interpretable is the model, suggesting therefore that the model should focus only on rules that really make a difference in many training samples.
- **fewer predictors:** if many predictors are involved in the definition of a rule, then the rule becomes less interpretable, and should therefore not be included unless strictly necessary.

In order to do so, the horseshoe prior was redefined by changing the prior distribution of the local shrinkage parameters λ_j, which takes the form:

$$\lambda_j \sim \mathcal{C}^+(0,A_j), \tag{2.1}$$

where A_j may be defined as a quantity that is low for very unfavourable predictors and high for very favourable ones. Note that A_j is the median value of a random variable with distribution $\mathcal{C}^+(0,A_j)$, meaning that a higher value of A_j likely induces a higher λ_j, which in turn makes the prior $\beta_j \sim \mathcal{N}(0,\lambda_j^2\tau^2\sigma^2)$ less informative and reduce the shrinkage. The choice that Nalenz and Villani make in [2] is to give linear terms a value of $A_j = 1$, while for rules the formula is:

$$A_j = \frac{\big(2\cdot\min(1-\overline{r}_j,\overline{r}_j)\big)^{\mu}}{(m_j)^{\eta}}, \tag{2.2}$$

where m_j denotes the number of predictors involved in the definition of the j-th rule. The parameters $\mu, \eta > 0$ may be set manually according to how heavily one wants to shrink rules with extreme support or many predictors. These two hyperparameters may also be determined by cross-validation, however the authors report high accuracy with the default of $\mu = 1, \eta = 2$. The definition of A_j is designed so that $A_j \leqslant 1$ always holds, for any values of μ and η, with $A_j = 1$ being satisfied only for rules of depth one that have the most evenly distributed support possible (which is $\overline{r}_j = 1 - \overline{r}_j = 0.5$). In particular, linear terms are being penalized as much as these types of ideal rules, and less than any other rule.

The method of using this type of structured horseshoe regularization on the extended predictor space obtained from the Rulefit machinery is part of what the authors Nalenz and Villani named *Horserule*. Aside from the use of this horseshoe prior with structured penalization, another innovation of the Horserule method is that it not only collects rules from a boosted tree ensemble but also from a Random Forest. According to Nalenz and Villani, the Random Forest brings additional randomness and diversity in the rules, therefore increasing predictive performance. We, however, do not focus on this aspect of the Horserule model, and instead keep our discussion simply on the use of the structured horseshoe regularization.

There is one last fundamental difference between Horserule and Rulefit: while the latter rescales the linear terms as shown in Equation 1.4 in order to make them comparable to the rules, the former approach simply centers and standardizes everything, both rules and linear terms. Concerns about the instability of this type of approach have already been raised above, in the context of LASSO regression, but in Horserule the structured penalization also helps reducing instability. To discuss this further, let us notice that the HorseShoe prior may be conveniently rewritten to incorporate the re-scaling of predictors. Let us study the conditional density of β, for any given $\lambda = (\lambda_1, \ldots, \lambda_p)$, in the case where the predictors $x_1, \ldots, x_p$ are re-scaled by some coefficients $a_1, \ldots, a_p$. We write it up to a multiplicative constant, and before it is integrated over all values of λ:

$$\begin{aligned} f_\beta^{(a)}(b|\lambda) &\propto e^{-\frac{1}{2\sigma^2}||y - \sum_{j=1}^p (a_j x_j) b_j||^2} \cdot e^{-\frac{1}{2\tau^2\sigma^2}\left(\sum_{j=1}^p \frac{b_j^2}{\lambda_j^2}\right)} \\ &= e^{-\frac{1}{2\sigma^2}||y - \sum_{j=1}^p x_j (a_j b_j)||^2} \cdot e^{-\frac{1}{2\tau^2\sigma^2}\left(\sum_{j=1}^p \frac{(a_j b_j)^2}{(a_j \lambda_j)^2}\right)} \\ &\propto f_\beta\big((a_j b_j)_{j=1,\ldots,p} | (a_j \lambda_j)_{j=1,\ldots,p}\big). \end{aligned}$$

In other words, multiplying a predictor x_j by a scaling parameter a_j before fitting a Horseshoe model is equivalent to directly fitting the Horseshoe model with an individual shrinkage parameter that is scaled by the same parameter. The coefficient estimate also automatically accommodates the new scale of the predictor, if we do so, as b_j is replaced by $a_j b_j$. Within the context of using Horserule, this means that standardizing the predictors before regression is equivalent to replacing $\lambda_j \sim \mathcal{C}^+(0, A_j)$ with:

$$\lambda'_j := \frac{\lambda_j}{sd(x_j)} \sim \mathcal{C}^+(0, \frac{A_j}{sd(x_j)}).$$

Now we apply this equivalence to our Horserule setting: assuming that $\mu \geqslant 0.5$, we notice that, for a rule r_j, standardization is equivalent to a local shrinkage parameter $\lambda'_j \sim \mathcal{C}^+(0, A'_j)$, where:

$$\begin{aligned} A'_j &= \frac{A_j}{sd(r_j)} \\ &= \frac{A_j}{\sqrt{\overline{r}_j(1-\overline{r}_j)}} \\ &= \frac{(2\min(\overline{r}_j, 1-\overline{r}_j))^{\mu}}{m_j^{\eta}\sqrt{\overline{r}_j(1-\overline{r}_j)}} \\ &= \frac{(2\min(\overline{r}_j, 1-\overline{r}_j))^{\mu-0.5)}}{m_j^{\eta}} \cdot \sqrt{\frac{2\min(\overline{r}_j, 1-\overline{r}_j)}{\overline{r}_j(1-\overline{r}_j)}} \\ &= \frac{(2\min(\overline{r}_j, 1-\overline{r}_j))^{\mu-0.5)}}{m_j^{\eta}} \cdot \sqrt{\frac{2}{\max(\overline{r}_j, 1-\overline{r}_j)}} \end{aligned} \tag{2.3}$$

In other words, up to a multiplicative factor $\sqrt{\frac{2}{\max(\overline{r}_j, 1-\overline{r}_j)}}$, standardization has the effect of only reducing the parameter μ by 0.5. This multiplicative factor, however, is quite stable and has little span across different supports: Figure 2.1 compares the function $\sqrt{\frac{2}{\max(\overline{r}_j, 1-\overline{r}_j)}}$ with the function $\frac{1}{\sqrt{\overline{r}_j(1-\overline{r}_j)}}$. Considering that the latter represents how much shrinkage is dampened when performing standardization before LASSO, we see how standardizing rules beforehand does not produce the significant instability that the classical Rulefit had to address.

2.3 Informative HorseShoe: using co-data

Complementary data, or just co-data, as described for instance by [15] and [16], is a term typically referring to prior information that one has about the predictors. This information may be used to structure the prior in a Bayesian model with more insight and to take advantage of historical/expert knowledge that would otherwise be lost. Examples of co-data could for instance be -omics annotations, but also p-values or estimates from a previous experiment. In our context, our prior knowledge includes (but is not necessarily limited to) the different nature of the predictors (linear terms vs. rules, rules of different depth, rules with different support).

From now on, let us denote the co-data as a collection $\{Z^{(d)}\}_{d=1,\ldots,D}$ of matrices $Z^{(d)} \in \mathbb{R}^{p \times h_d}$ containing categorical and/or continuous information about the p predictors. Many steps have already been taken in the direction of incorporating co-data in one's analysis, both for a single source of co-data (see [17] [18] for grouped LASSO, for categorical co-data, and [19] [20] [21], for Bayesian use of co-data) and multiple sources of co-data (see [22] and [9]). Since we have multiple types of information about the predictors and since further, experiment-specific sources could also be available in a general setting, we focus on multiple sources of co-data.

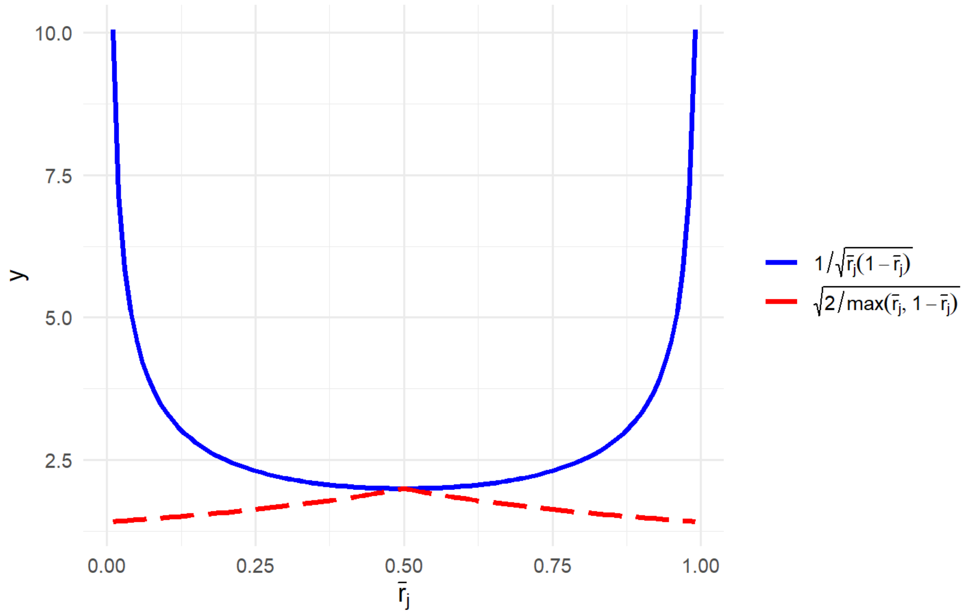

Figure 2.1: Plot of the functions $y = 1/\sqrt{\overline{r}_j(1-\overline{r}_j)}$ (blue, solid line) and $y = \sqrt{\frac{2}{\max(\overline{r}_j, 1-\overline{r}_j)}}$ (red, dotted line). Given the observed frequency $\overline{r}_j$ of a rule r_j. The red line, which represents the change in shrinkage induced by the standardization under structured penalization of rules, is much more stable than the blue line, representing the change in shrinkage induced by the standardization when no penalization of rules is present.

Both [22] and [9] proceed with a prior similar to the one written above, but then connect the parameter λ_j with a linear combination obtained from the co-data matrices. In [9], for instance, the informative horseshoe prior is defined as:

$$y|X, \beta, \sigma^2 \sim \mathcal{N}(X\beta, \sigma^2 I_n), \tag{2.4}$$

$$\beta_j|\lambda_j, \tau, \sigma^2 \sim \mathcal{N}(0, \lambda_j^2\tau^2\sigma^2), \tag{2.5}$$

$$\lambda_j|Z, \gamma \sim \mathcal{C}^+(\sum_{d=1}^{D}(Z_j^{(d)})^t\gamma_d, 1), \tag{2.6}$$

$$\gamma_d|\kappa_d^2 \sim \mathcal{N}(0, \kappa_d^2 I_{s_d}), \tag{2.7}$$

$$\kappa_d^2 \sim \mathcal{IG}(1, 1), \tag{2.8}$$

$$\tau \sim \mathcal{C}^+(0, 1), \tag{2.9}$$

$$\sigma^2 \sim \sigma^{-2} d\sigma^2. \tag{2.10}$$

In this approach, unlike HorseRule, the differentiation in the distribution of λ_j is made at the level of the location of the Cauchy distribution, rather than its scale. As mentioned in [9], this is more intuitive in contexts where covariates are not only categorized into strong and insignificant but also allow for more nuanced, weaker contributions. With this in mind, the informative Horseshoe regression developed by Busatto and van de Wiel in [9] seems to be a potential improvement to Horserule, since multiple types of information about our setting may be specified, and since this opens up our model to further, external sources of information.

As we replace the Horseshoe prior with this co-data-based counterpart, we may also decide whether we want to keep the structured penalization or not: thanks to our discussion about shrinkage and scaling from our previous chapter, we can consider the structured penalization of the rules as an ordinary horseshoe fit, where the linear terms are being standardized and the rules are being rescaled by a factor A'_j as defined in Equation 2.3. With this mindset, the horseshoe fit may be replaced with the informative horseshoe. Alternatively, one may consider the natural adaptiveness of the informative horseshoe prior and write undesirable properties like extreme support, high depth and rule-like nature as co-data sources. In this manner, specifying the structured penalization can be avoided altogether. Since there is no structured penalization, standardization of rules may be unstable and should therefore be avoided in this case. As mentioned above, this naturally favours rules with more balanced support. Not standardizing rules also naturally favours linear terms, a phenomenon that Friedman and Popescu have countered by shrinking the linear terms by a factor of 0.4. When using the informative Horseshoe prior, however, the co-data naturally creates a distinction in shrinkage between rules and linear terms.

Chapter 3

Interpreting individual contributions: Shapley values

3.1 What are Shapley values

As discussed above, a key feature of Rulefit is that high predictive performance is paired with better interpretability. Yet, the above-discussed limitations of the current variable importance measures motivate looking further into other ways to interpret a predictor's effect in our model. A significant step in the direction of interpretable machine learning was the introduction of Shapley values for fitted regression models.

Shapley values were first introduced by game theorist Lloyd Shapley [23] as a way to measure the individual contribution of multiple players in a game. They were only later proposed as a way to make black-box models more transparent (see for instance [24] and [25]). In particular, they defined the so-called SHAP, SHapley Additive exPlanations, to be the Shapley values for the specific game of using the fitted model to predict the outcome. The different players of the game are, in this context, the individual predictors, and they are each assigned their contribution to the prediction. Because SHAP values are a specific case of Shapley values, the two names are often used interchangeably in the Machine Learning literature. That being said, "SHAP" tends to refer to more specific, practical algorithms to estimate Shapley values, while "Shapley" is used in a more theoretical context. Variations of the original Shapley values tend to retain the original name, as is the case for Marginal Shapley values or Baseline Shapley values. Following this common practice, we mainly use the expression "Shapley values" and resort to using "SHAP" when referring to well-established practices, libraries and algorithms.

To illustrate the reasoning behind the definition of Shapley values, let us consider a fitted model $F(x)$. For any datapoint x^* and any predictor index j, Shapley values are a local measure that tries to estimate how much the j-th predictor contributes to the specific prediction $F(x^*)$. An intuitive way of computing such contributions would be fitting two models: a model $F(x_1, \dots, x_p)$ which uses the j-th predictor and a second model $F_j(x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_p)$ which is fit without x_j. Then a possible measure could be:

$$I_j(x^*) := F(x_1^*, \dots, x_p^*) - F_j(x_1^*, \dots, x_{j-1}^*, x_{j+1}^*, \dots, x_p^*).$$

This measure, however, requires fitting a new model for every predictor whose importance we need to measure. More importantly, it does not properly consider the collinearity between two predictors: let us consider an example where a linear model is fitted on data with the following structure:

$$Y = X_1 + X_2, \qquad\qquad X_1 = 2X_2.$$

In this context, $F(x_1, x_2) := x_1 + x_2$,$F_1(x_2) = \frac{3}{2}x_2$ and $F_2(x_1) = 3x_1$ share the same image, meaning that the importance $I_j(x^*)$ as defined above would always be null. Another option is to replace F_j with the marginalization of the model $F(x)$ over the predictor x_j, as a way to "remove" the extra information given by the j-th predictor. This method does not require re-fitting the model, and results in the measure:

$$I'_j(x^*) := F(x_1^*, \dots, x_p^*) - \mathbb{E}[F(x_1, \dots, x_p)|x_j = x_j^* \forall i \neq j]. \tag{3.1}$$

Applying this measure to the linear model built above, however, still produces null contributions:

$$\begin{aligned} I'_1(x^*) &= F(x_1^*, x_2^*) - \mathbb{E}[F(x_1, x_2)|x_2 = x_2^*] \\ &= x_1^* + x_2^* - \mathbb{E}[x_1 + x_2|x_2 = x_2^*] \\ &= x_1^* + x_2^* - (x_1^* + x_2^*) \\ &= 0. \end{aligned}$$

This problem goes beyond collinearity and is generally related to the fact that this measure still does not pick up cooperation between predictors very well, even when predictors are independent. Let us consider the following example, where x_1 and x_2 are independent predictors:

$$F(x_1, x_2) = I(x_1 > 0 \vee x_2 < 0).$$

The contribution of the first predictor to the outcome prediction for the datapoint $x^* = (1, -1)$, as defined in Equation 3.1, would be:

$$\begin{aligned} I'(x^*) &= F(x_1^*, x_2^*) - \mathbb{E}[F(x_1, x_2)|x_1 = x_1^*] \\ &= I(1 > 0 \vee -1 < 0) - \mathbb{E}[I(1 > 0 \vee x_2 < 0)] \\ &= 1 - \mathbb{E}[1] = 0. \end{aligned}$$

The same would happen for the second predictor, implying a model where no predictor is important. The problem here is that both predictors add no further relevant information if the other predictor is already there. Their power relies on adding information when no predictor is being used yet. In other words, to make this more general, different predictors may work together towards prediction and therefore the effect of a predictor x_j may depend on which predictors x_j is being added to.

Therein lies the novelty of Shapley values, where contrasts between marginalizing and not marginalizing x_j are not only computed once, while fixing all other predictors. Instead, Shapley values compute this contrast for any combination of all remaining predictors being used or marginalized. More specifically, the Shapley value of the j-th predictor for a datapoint x^* is computed as:

$$\phi_j(x^*) = \sum_{S \subseteq \{1,\dots,p\}\setminus\{j\}} \frac{1}{p\binom{p-1}{|S|}} \Big(\mathbb{E}[F(x)|x_S = x_S^*, x_j = x_j^*] - \mathbb{E}[F(x)|x_S = x_S^*] \Big), \quad (3.2)$$

where the notation $x_S = x_S^*$ means $x_k = x_k^* \; \forall k \in S$. The weights $\frac{1}{p\binom{p-1}{|S|}}$ are denoted by $w_p(S)$, or simply $w(S)$ when p is clear from the context. In this formula, these weights serve the purpose of taking into account the fact that there are fewer subsets S where $|S|$ is very small or very large. A set S here represents which predictors are not being marginalized for a specific contrast, and therefore are playing a role in determining $F(x^*)$. This is why the sets S are typically referred to as *sets of active players*.

There are multiple benefits to Shapley values as defined in Equation 3.2.
First, they allow for an intuitive interpretation: they may be described as the average change in prediction that is induced in the model when an unknown number of the predictors is expanded by adding the j-th predictor.
Second, Shapley values account for cooperation between predictors: going back to our example with $F(x_1, x_2) = I(x_1 > 0 \vee x_2 < 0)$ and $x^* = (1, -1)$, the Shapley value for the first predictor is:

$$\begin{aligned}
\phi_1(x^*) &= \frac{1}{2}\big(\mathbb{E}[F(x)|x_1 = x_1^*, x_2 = x_2^*] - \mathbb{E}[F(x)|x_2 = x_2^*]\big) \\
&\qquad\qquad + \frac{1}{2}\big(\mathbb{E}[F(x)|x_1 = x_1^*] - \mathbb{E}[F(x)]\big) \\
&= \frac{1}{2}\big(F(x^*) - \mathbb{E}[F(x)|x_2 = x_2^*]\big) + \frac{1}{2}\big(\mathbb{E}[F(x)|x_1 = x_1^*] - \mathbb{E}[F(x)]\big) \\
&= \frac{1}{2}\big(\mathbb{E}[F(x)|x_1 = x_1^*] - \mathbb{E}[F(x)]\big).
\end{aligned}$$

We indeed see how the formula for Shapley values has produced an extra non-null term that gives non-null importance to x_1, and analogously to x_2.

Lastly, Shapley values have the following four properties:

- **Efficiency:** Summing up all values produces the overall deviation from the average prediction: $\sum_{j=1}^{p} \phi_j(x^*) = F(x^*) - \mathbb{E}[F(x)]$;
- **Symmetry:** Two predictors that equally change the prediction when added to any same coalition of active players S have the same Shapley value;
- **Dummy:** If a predictor does not alter the prediction when added to any coalition of active players, then its Shapley value is zero. This is a very helpful property, as it gives us a natural null against which to contrast significant predictors, in an inferential framework;

- **Additivity:** Computing the Shapley values for the model $F + G$ is equivalent to adding up the Shapley values from the model F and the model G, computed separately. Thus, for RuleFit, the Shapley values for the full model may be computed by summing up the Shapley values of the individual linear terms and the rules.

Shapley values are the only type of measure in game theory to have these four properties [23], which makes them a particularly desirable measure.

3.2 Limitations, challenges and alternatives for Shapley values

Despite the many benefits described above, there are two main issues that complicate the use of Shapley values:

- **Conditional expectations:** Expanding on the previous point, one needs to not only make assumptions about the distribution of the predictors but also be able to estimate the conditional means in Equation 3.2, which may be very challenging, especially when the model F is sophisticated or these expectations are being conditioned on continuous predictors.
- **Computational cost:** In most cases, the summation in Equation 3.2 does not simplify, and has to be explicitly computed over all possible subsets S. The number of subsets, however, is 2^{p-1} and thus grows exponentially with the number of predictors, making these values hard to compute in most applications.

These two challenges have motivated researchers to find approximations of Shapley values that are computationally more affordable. The computational cost was reduced by approximations such as sampled Shapley Values [24][25], which estimates Shapley values from only sampling *some* active player subsets S. Starting from m points $t^{(1)}, \dots, t^{(m)}$ selected uniformly at random and m orderings $\pi_1, \dots, \pi_m$ selected uniformly at random from the group $\mathcal{S}_p$ of all permutations of the p predictors, the sampled Shapley values for a point x^* for the j-th feature are computed as:

$$\widehat{\phi}_j(x^*) := \frac{1}{m}\sum_{i=1}^{m}(f(\tau(x^*, t^{(i)}, \pi_i)) - f(\tau'(x^*, t^{(i)}, \pi_i))),$$

with $\tau(x^*, t^{(i)}, \pi_i))$ and $\tau'(x^*, t^{(i)}, \pi_i))$ are p-dimensional vectors whose k-th entry is defined as:

$$(\tau(x^*, t^{(i)}, \pi_i))_k = \begin{cases} x_k^* & \text{if } k = j \text{ or } k \text{ is before or } j \text{ w.r.t. } \pi_i \\ y_k & \text{otherwise} \end{cases},$$

$$(\tau'(x^*, t^{(i)}, \pi_i))_k = \begin{cases} x_k^* & \text{if } k \text{ is strictly before } j \text{ w.r.t. } \pi_i \\ y_k & \text{otherwise} \end{cases}.$$

Implicitly, the set of active players S is the set of all predictors that come before the j-th feature, in the randomly extracted feature order π_i. This way of defining S from a permutation allows for the sampling of S to fairly distribute between different sizes of active players sets, so that no weighting $w(S)$ is needed.

KernelSHAP [26][27] also shortens computation times by taking advantage of the Sampling of active player subsets, but combines it with a clever rewriting of Shapley values as matrix product and the use of statistical modelling to approximate the joint distribution of the features. In doing so, the conditional expectations are quite accurately accounting for dependencies between predictors, but the computational cost is still considerable [27].

Another popular alternative is Marginal Shapley values, also called Random Baseline Shapley values [28], which simply do not take conditional expectations, but rather the expectations $\mathbb{E}[F(x^*_S, x_{S'})]$, where $F(x^*_S, x_{S'})$ denotes the function $F(x)$ where the entries in the set S have been fixed to the value x^*_S, while the others are left as variables. This substitution in expectations is de facto the same as assuming independence between predictors when computing the expectations. This approach may allow for simplifications in the definition of Shapley values from Equation 3.2, as discussed in the next sections.

It is important to point out that these simplifications come at the cost of less accurate estimates, especially when predictors are highly correlated, since marginal Shapley values ignore correlation when computing the expectations. However, the fact that the marginal Shapley values may differ from their exact counterpart does not necessarily need to be a source of concern: in the context of a linear model, for instance, marginal Shapley values coincide with the interpretation that we already give to the coefficients, i.e. $\phi_j(x^*) = \beta_j(x^*_j - \overline{x}_j)$ [27]. Exact Shapley values, on the other hand, account for the correlation between predictors and therefore do not coincide with our former notion of the contribution of a linear term. In particular, this means that if the purpose is to have a measure that generalizes the way we already interpret coefficients for the linear terms in a linear model, then we should use marginal Shapley values, rather than the exact ones. The use of marginal Shapley values instead of exact ones is also important for consistency: for the linear terms, only the marginal Shapley values are readily available, and their exact counterparts would need methods such as KernelSHAP to be computed [27]. Adding marginal Shapley values for the linear terms to non-marginal Shapley values for the rules would thus hinder the interpretability of our importance measure.

3.3 TreeSHAP

While Shapley values can be difficult to compute for a generic model, the problem is simplified in the context of RuleFit: by additivity of Shapley values, we can compute said importance measure for the linear model and for the tree ensemble separately. Since the marginal Shapley values for the linear terms are readily computed from the regression coefficients, the problem is reduced to estimating the marginal Shapley values for the tree ensemble. Thanks to their simpler structure, it is easier to estimate Shapley values for trees. An example of this is given by [29], who produce an explicit formula for computing marginal Shapley values. This formula, however, is only applicable to oblivious (also called symmetric) trees. In Section 3.4, we take a different approach and propose a simpler formula that applies to any kind of tree and, by the additivity of Shapley values, tree ensembles.

The most widely used estimation of Shapley values for trees and tree ensembles, however, is TreeSHAP. Firstly introduced by Lundberg, Erion and Lee [30], TreeSHAP is an approximation of Shapley values for trees that tries to both account for correlations between features and, simultaneously, speed up computations. Its main strength is that it does not directly have to sample through all subsets S of active players, but rather runs through an individual tree in a top-down manner, and for every splitting node, it acknowledges the contribution that the associated predictor would bring if it were added to a coalition. The weighting of the contributions according to the size S is stored separately. Another strength is that the expectations partially account for the conditioning on $x_S = x_S^*$: the quantities $\mathbb{E}[F(x)|x_S = x_S^*]$ are approximated by averaging the predictions $F(t)$ from all samples t that satisfy the same rules as x^* for the predictors that are active players. More specifically, as explained in [31], the algorithm is equivalent to estimating a single conditional mean $\mathbb{E}[F(x)|x_S = x_S^*]$ of a single tree $F(x)$ by agglomerating the nodes in a bottom-up manner. A parent of two terminal nodes collapses into a newly generated terminal node whose prediction $F(x)$ is determined as follows:

- if the splitting predictor is not an active player (and therefore needs to be marginalized), the new $F(x)$ is the weighted average of the two predictions of the child nodes. The average is weighted according to how many points belong in each node.
- if the splitting predictor is an active player (and therefore its rule needs to be enforced by discarding the points that do not follow it), the new $F(x)$ is the prediction of the child node whose defining rule is satisfied by x^*.

This collapsing process is repeated until only a single terminal point is left. Its prediction value estimates the conditional mean. Figure 3.1 depicts how an exemplary tree undergoes the process, for the active player subset $S = \{1, 4\}$ and the point $x^* = (3, 2, 1, 0)$. In this example, the estimate for $\mathbb{E}[F(x)|x_S = x_S^*]$ is 7.7. Instead of repeating the process in Figure 3.1 for every possible subset S, TreeSHAP is built to scan through every tree in a top-down matter, and keep track of these conditional means as the scanning proceeds. These conditional means are being approximated by another similar quantity which is sometimes referred to as *TreeSHAP game value* [29].

The algorithm drastically reduces computation times, while also partially accounting for the conditioning in the expectations that appear in the definition of Shapley values as in Equation 3.2. This algorithm, however, has been shown to have flaws such as:

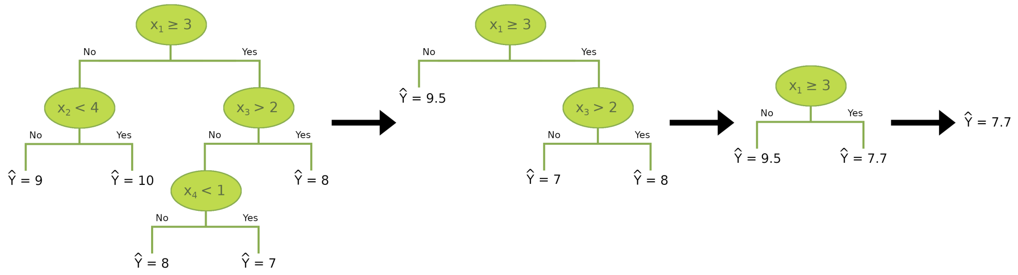

Figure 3.1: Example of how TreeSHAP approximates the conditional mean $\mathbb{E}[F(x)|x_S = x_S^*]$ for $x^* = (3, 2, 1, 0)$ and $S = \{1, 4\}$. In the first step, two nodes are collapsed: the two left-most leaves result from a split on the inactive player x_2; they are collapsed into their parent node whose prediction becomes the weighted mean of the two child nodes. The middle two leaves result from a split on the active player x_4; they are collapsed into their parent node whose prediction becomes the prediction of the child node x^* was sent to. In the second step, the two right-most leaves result from a split on the inactive player x_3; they are collapsed into their parent node whose prediction becomes the weighted mean of the two child nodes. The resulting value is the estimate of $\mathbb{E}[F(x)|x_S = x_S^*]$. Here, it is estimated as 7.7.

- **False influence of predictors:** Even when a predictor does not actually contribute to the outcome, TreeSHAP may produce (typically small) non-zero estimates of its Shapley value [32].

- **Structure dependence:** TreeSHAP is not implementation-invariant [29]. In other words, trees that split the predictor space in exactly the same way and with exactly the same predictions may produce different Shapley value estimates, if the splitting predictors are re-arranged in order (e.g. the tree splits by predictor 1 before predictor 3, instead of the other way around). More specifically, TreeSHAP tends to attribute more importance to the predictors that appear earlier in the trees. Combining this with the fact that trees are greedy and therefore strongly correlated predictors tend to be split first, this results in an underestimation of small contributions and an overestimation of larger ones [33].

- **Loss of interpretability and bias:** Since this algorithm is using TreeSHAP game values instead of the conditional means, we are technically estimating a quantity that is conceptually different from Shapley values, and therefore bound to be a somewhat biased approximation [29]. In the context where non-marginal Shapley values are really needed, which is when features are strongly correlated, TreeSHAP may still be inaccurate in estimating the conditional means and the bias may therefore be quite substantial [27].

These considerations add to the aforementioned hesitations about the use of non-marginal estimations of Shapley values.

It is important to point out that Lundberg, Erion and Lee also implemented a version of TreeSHAP that computes marginal Shapley values. This version is not structure-dependent anymore and alleviates the problem of loss of interpretability. For marginal Shapley values, however, we show in the next section that a more explicit formula can be derived.

3.4 A formula for marginal Shapley values

In this section, we compute an explicit formula for marginal Shapley values. In particular, we focus on the Shapley values for an individual rule. This represents no loss of generality, as a tree ensemble is a (weighted) sum of trees and a tree may be seen as a linear combination of individual, 0-1-coded rules: consider for instance the two trees in Figure 1.1 on page 2. They can explicitly be written as:

$$T_1(x) = 5 \cdot I(x_1 \geqslant 3) + 7 \cdot I(x_1 < 3, x_3 < 2) + 8 \cdot I(x_1 < 3, x_3 \geqslant 2),$$

$$T_2(x) = 9 \cdot I(x_3 > 4) + 7.5 \cdot I(x_1 \leqslant 4).$$

Since Shapley values are additive, this means that we can focus on the contributions of the individual rules, which is especially convenient in our context, where RuleFit only retains the most relevant rules in the model.

Before we prove the formula, let us first note that marginal Shapley values allow us to focus only on the predictors that really are involved in the definition of a rule:

Lemma 1. *Consider a function $F : \mathbb{R}^p \to \mathbb{R}$, and take $q < p$ such that $F(x_1, \ldots, x_p)$ only depends on q of the p total predictors, say $x_{j_1}, \ldots, x_{j_q}$. Then the Shapley values for F may be computed by only focussing on $x_{j_1}, \ldots, x_{j_q}$: if $j \in \{j_1, \ldots, j_q\}$, then:*

$$\phi_j(x^*) = \sum_{S \subseteq \{j_1, \ldots, j_q\} \setminus \{j\}} \frac{1}{q\binom{q-1}{|S|}} \Big(\mathbb{E}[F(x_1, \ldots, x_p) | x_j = x_j^*, x_S = x_S^*] - \mathbb{E}[F(x_1, \ldots, x_p) | x_S = x_S^*] \Big).$$

If $j \notin \{j_1, \ldots, j_q\}$, then $\phi_j(x^*) = 0$.

Proof. See Subsection 6 of the Appendix. □

This Lemma becomes convenient as it bridges the gap between intuition and practice: if a rule only focuses on some predictors, then it should only alter the importance of said predictors, and the contribution should not be different according to how many correlated noisy predictors are included in the model.

Before we prove our result, we present a variation of Vandermonde's identity, also required in order to simplify the Equation 3.2 defining Shapley values:

Lemma 2. *For any $a, b, c \in \mathbb{N}$ such that $c \leqslant b$, the following equality holds:*

$$\sum_{l=0}^{c} \binom{a+l}{l} \binom{b-l}{c-l} = \binom{a+b+1}{c}.$$

Proof. See Subsection 6 of the Appendix for a combinatorial argument. □

With this last Lemma in our toolbox, we can now prove our main result. Let us prove it in the case where a rule involves all p predictors. Thanks to Lemma 1, we already know that, even if that is not the case, we may focus on the involved predictors only, when computing the Shapley values of a rule.

Theorem 3. *Assume to have a dataset $\mathcal{T}$ of size n. Consider a 0-1 coded rule of the form $R(x_1, \ldots, x_p) = \prod_{j=1}^{p} R_j(x_j)$, with $R_j : \mathbb{R} \to \{0, 1\}$. Then an unbiased estimator of the marginal Shapley value of $F(x) = \beta R(x)$ for the j-th predictor and the datapoint x^* is:*

$$\widehat{\phi}_j(x^*) = \beta \cdot \left(\frac{1}{n(p - q_j(x^*))} \sum_{\substack{t \in \mathcal{T} \ s.t. \\ \Omega(t) \supseteq \Omega(x^*)'}} \frac{R_j(x_j^*) - R_j(t_j)}{\binom{2p - q_j(x^*) - q_j(t) - 1}{p - q_j(x^*)}} \right),$$

where $\Omega(t)$ and $\Omega(x^*)$ are sets of predictor indices defined as:

$$\Omega(t) = \{k \in \{1, \ldots, p\} | R_k(t_k) = 1\}, \qquad \Omega(x^*) = \{k \in \{1, \ldots, p\} | R_k(x_k^*) = 1\},$$

and $q_j(t)$ and $q_j(x^*)$ are set sizes defined as:

$$q_j(t) = |\Omega(t) \backslash \{j\}|, \qquad q_j(x^*) = |\Omega(x^*) \backslash \{j\}|.$$

Note that $\Omega(x^*)'$ denotes the complementary subset of $\Omega(x^*)$ with respect to $\{1, \ldots, p\}$.

Proof. See Subsection 6 of the Appendix. □

The theorem and its proof do not depend on the choice of dataset $\mathcal{T}$ used to estimate the marginal means. In particular, this implies that $\mathcal{T}$ does not need to coincide with the training set, but rather may be chosen to be a subset of the observations that more accurately reflects the distributional properties of the whole population.

The formula in this theorem only selects the datapoints t for which the specific condition $\Omega(x^*)' \subseteq \Omega(t)$ holds. This condition is symmetric:

$$\Omega(t)' \subseteq \Omega(x^*) \iff \Omega(x^*)' \subseteq \Omega(t),$$

which means that t contributes to the Shapley value of x^* if and only if x^* contributes to the Shapley value of t. In such a case, the contributions are precisely identical but swap signs. This contribution only depends on the sample size n, the number of predictors involved in the rule and the quantities $q_j(t), q_j(x^*)$. The function $q_j(x)$ counts how many sub-rules R_k are activated by the datapoint x, if we exclude the j-th predictor. Let us clarify these definitions with an example. Consider the following rule:

$$R(x_1, x_2, x_3) = R_1(x_1) \cdot R_2(x_2) \cdot R_3(x_3),$$

$$R_1(x_1) = I(x_1 > 0), \quad R_2(x_2) = I(x_2 < 0.4), \quad R_3(x_3) = I(x_3 = 1).$$

We define an exemplary dataset $\mathcal{T}$ consisting of the following 4 points:

$$t^{(1)} = (0.5, 0, 1), \;\; t^{(2)} = (-0.5, 0.2, 1), \;\; t^{(3)} = (1, 0.6, 0), \;\; t^{(4)} = (-0.3, 0.1, 0),$$

and choose $x^* = (0.2, 0, 0)$, for which:

$$R_1(x_1^*) = 1, \qquad R_2(x_2^*) = 1, \qquad R_3(x_3^*) = 0.$$

This means that:

$$\Omega(x^*) = \{1, 2\}, \qquad \Omega(x^*)' = \{3\},$$

$$q_1(x^*) = |\{2\}| = 1, \qquad q_2(x^*) = |\{1\}| = 1, \qquad q_3(x^*) = |\{1, 2\}| = 2.$$

Table 3.1 shows the objects Ω, Ω' and q as described in Theorem 3, for all the datapoints in $\mathcal{T}$, together with which ones contribute to the marginal Shapley values of x^*. Note that, for instance, $t^{(3)}$ and $t^{(4)}$ do not contribute to the estimation of the marginal Shapley values of x^*, since $R_3(x_3^*)$, $R_3(t_3^{(3)})$ and $R_3(t_3^{(4)})$ are all null.

$t^{(i)}$	$R_1(t_1^{(i)})$	$R_2(t_2^{(i)})$	$R_3(t_3^{(i)})$	$\Omega(t^{(i)})$	$\Omega(t^{(i)})'$	$\Omega(t^{(i)}) \supseteq \Omega(x^*)'$	$q_1(t^{(i)})$	$q_2(t^{(i)})$	$q_3(t^{(i)})$
$t^{(1)}$	1	1	1	$\{1,2,3\}$	$\varnothing$	Yes	2	2	2
$t^{(2)}$	0	1	1	$\{2,3\}$	$\{1\}$	Yes	2	1	1
$t^{(3)}$	1	0	0	$\{1\}$	$\{2,3\}$	No	0	1	1
$t^{(4)}$	0	1	0	$\{2\}$	$\{3,1\}$	No	1	0	1

Table 3.1: Example of Ω, Ω' and q_j for the datapoints defined on page 21, as defined in Theorem 3. The column "$\Omega(t^{(i)}) \supseteq \Omega(x^*)'$" denotes whether the sample $t^{(i)}$ satisfies the condition $\Omega(t^{(i)}) \supseteq \Omega(x^*)'$ and therefore contributes in the formula to compute $\phi_j(x^*)$, for any j.

The intuition behind the condition $\Omega(x^*)' \subseteq \Omega(t)$ was already given in the proof: it is satisfied by a pair of datapoints if, excluding the predictor whose Shapley value is being computed, every predictor x_k has $R_k(x_k) = 1$ for at least one of the two datapoints in the pair. It is important to point out, however, that satisfying this condition is only equivalent to two samples contributing to each other's Shapley values, it is no guarantee that the contribution is nonzero. For that, looking at Theorem 3, we need to also have $R_p(x_p^*) \neq R_p(t_p)$. In the next section, we develop a possible implementation for an algorithm that computes the formula from Theorem 3.

3.5 Vectorial re-writing of the condition $\Omega(t) \supseteq \Omega(x^*)'$

Now that we have an explicit formula to compute marginal Shapley values, let us describe an algorithm that computes the contribution to the Shapley values of each rule. In particular, the main challenge of computing them is identifying which datapoints give a contribution; these contributions then need to be summed up. Knowing that the following equality between logical statements:

$$A \vee B = \neg\big((\neg A) \wedge (\neg B)\big),$$

we can re-write the condition $\Omega(x^*)' \subseteq \Omega(t)$ for contributing to the estimated Shapley value as:

$$\begin{aligned}
\Omega(x^*)' \subseteq \Omega(t) &\iff (R_k(x_k^*) = 1) \vee (R_k(t_k) = 1) \;\; \forall k \in \{1, \dots, p\} \\
&\iff p = \sum_{k=1}^{p} \Big((R_k(x_k^*) = 1) \vee (R_k(t_k) = 1) \Big) \\
&\iff p = \sum_{k=1}^{p} \Big(1 - \big((1 - R_k(x_k^*)) \cdot (1 - R_k(t_k)) \big) \Big) \\
&\iff p = p - \sum_{k=1}^{p} \big((1 - R_k(x_k^*)) \cdot (1 - R_k(t_k)) \big) \\
&\iff \sum_{k=1}^{p} \big((1 - R_k(x_k^*)) \cdot (1 - R_k(t_k)) \big) = 0.
\end{aligned}$$

This new formulation is particularly convenient, as it allows a faster algorithm implementation. Define $\mathcal{V}$ as:

$$\mathcal{V} : \mathbb{R}^p \to \{0, 1\}^p, \qquad \mathcal{V}(x) := (R_1(x_1), \dots, R_p(x_p)).$$

Then, using a dataset $\mathcal{T}$, we may estimate marginal Shapley values for a datapoint x^* as follows:

1. Compute $\langle \mathcal{V}(x^*), \mathcal{V}(t) \rangle$ for all samples $t \in \mathcal{T}$, then filter out all datapoints in $\mathcal{T}$ for which $\langle \mathcal{V}(x^*), \mathcal{V}(t) \rangle = 0$ does not hold.

2. For each of the datapoints t mentioned in step 1, compute the contribution as $\frac{R_j(x_j^*) - R_j(t_j)}{\binom{2p - q_j(x^*) - q_j(t) - 1}{p - q_j(x^*)}}$. The Shapley value estimate needs to be updated by this value.

The computations in step 1 may be written in matrix notation, which makes the computation of Shapley values for multiple datapoints easier to optimize.

3.6 Advantages and limitations of our approach

The main issue with the non-marginal TreeSHAP algorithm that we raised was the incompatibility between the marginal Shapley values for linear terms and the exact Shapley values which TreeSHAP tries to estimate. The use of marginal Shapley values estimates, which is what we are proposing, solves this issue, together with the issue of loss of interpretability and structure dependence. On the other hand, this comes at the cost of estimating a quantity that still does not coincide with the exact Shapley values, since the means are not conditional.

The already existing TreeSHAP machinery is a very fast algorithm: the non-marginal approach has computational complexity $O(TLD^2)$, where T is the number of trees, L is the maximum number of leaves appearing in a tree and D is the maximum depth of a tree. Our approach, on the other hand, is slower. More specifically, our implementation has to repeat the steps described on page 23 for every individual rule. The cost of estimating the marginal Shapley values for a datapoint x^* from a background dataset $\mathcal{T}$ is, for a single rule:

- $O(|\mathcal{T}| \cdot D)$ operations in order to compute the scalar products $\langle \mathcal{V}(x^*), \mathcal{V}(t) \rangle$ in step 1;
- $O(|\mathcal{T}| \cdot D)$ binomial coefficients and subtraction, in the most time-consuming scenario where every sample is compatible with x^* in step 1, and every predictor has to be updated. Let us interpret the computation of a single binomial coefficient as a single operation.

Overall, this means that we use $O(|\mathcal{T}| \cdot D)$ operations for every rule, and there are $O(TL)$ rules from the whole tree ensemble, meaning that our algorithm takes $O(TLD|\mathcal{T}|)$ operations to run. Since the background dataset $\mathcal{T}$ most likely needs to have multiple datapoints per leaf in any tree, we can say that $|\mathcal{T}| > L \approx 2^D$, which makes our algorithm slower. If we compare our approach to the marginal version of TreeSHAP, however, the computational complexity is the same [31]. This, however, is a pessimistic estimate in which no sparsity is induced: if a small number of rules is selected, then the computational gain may be higher for our algorithm than for TreeSHAP. However, as both algorithms are quite fast and applicable to the same settings, this difference might not be relevant.

Our version, on the other hand, has other advantages:

- **Diagnostics and transparency:** The marginal Shapley values may now be seen by the individual contribution of each rule: if a particular rule is overshrinking or overinflating the estimates, this can be easily ascertained.
- **Convergence and sampling:** Re-writing the Shapley values as means of individual quantities over all points in the sample allows us to use convergence-like arguments and, for instance, gradually add new samples to the background dataset until some convergence criterion is satisfied. Similarly, this formula may also be used to compute marginal Shapley values under theoretical joint distributions of the predictors and take advantage of approaches like importance sampling to reach better estimates more rapidly.
- **Compatibility with (Bayesian) RuleFit:** In the context of using Bayesian regression on RuleFit, not using our formula would mean re-converting the rule ensemble into a tree ensemble, and then applying TreeSHAP. More importantly, in order to produce the posterior distribution of the Shapley values estimates, this procedure would have to be repeated for each individual MCMC draw that was produced for Bayesian regression. Our method, on the other hand, can store the weights for every rule, meaning that the algorithm only has to be run once.

3.7 Showing Shapley values

As discussed in [6], a typical way of enhancing the interpretability of a model is through visualizations. As Shapley values became more and more commonly used, some types of visualizations of the Shapley contributions have become golden standards. Some typical examples of how variable importance is displayed are:

- **Force plots:** For the prediction of a given datapoint x^*, the individual Shapley value of every predictor may be seen as a contribution to the estimated outcome $F(x^*)$. In doing so, these individual contributions are seen as "forces" that push the estimate either below or above the average, and either work together or against each other. These forces are represented by bars.
- **Heatmaps:** A heatmap with all samples on the x-axis and predictors on the y-axis, where the colour represents the intensity (and sign) of the Shapley value associated with the pair of sample x^* and predictor x_j.
- **Scatter plots:** For a given j-th predictor, a scatterplot is produced where the x axis shows x_j^* and the y axis shows $\phi_j(x^*)$.
- **Summary plots:** A sina plot with the predictors on the y axis and the Shapley value on the x axis. Colour is used to represent whether the feature has higher or lower values.
- **Feature importance values:** For any j-th predictor, the average $|\phi_j(x^*)|$ across all datapoints x^* is computed as a measure of deviation that is induced by a predictor. These measures are plotted in a histogram.

Examples of such plots are shown in Chapter 4. In the next section, we focus on how these visualizations may be enhanced through means of credible intervals and uncertainty measures.

3.8 Credible intervals for Shapley values

On page 7, we discussed how substituting a Bayesian regression for a LASSO regression grants us a posterior distribution of the coefficients. On the other hand, Theorem 3 provides an explicit formulation of Marginal Shapley values that only depends on the coefficients. Combining these two elements within the RuleFit framework allows us to deduce the posterior distribution of the Shapley values for each datapoint and each predictor. Since Shapley values have a natural null that we can contrast influence against, credible intervals for Shapley values become a powerful tool for measuring the significance of predictors in the prediction of the outcome $F(x^*)$.

The benefits of using credible intervals trickle down to the Shapley visualizations discussed above: feature importance values can now be presented with credible intervals, both locally and globally. Furthermore, the uncertainty described by the credible intervals may actively be incorporated into the definition of importance itself. For instance, let us denote the credible interval for $\phi_j(x^*)$ as $[\phi_j^{\text{low}}(x^*), \phi_j^{\text{up}}(x^*)]$. Then one may define a local importance measure given by:

$$\psi_j(x^*) = \begin{cases} \phi_j^{\text{low}}(x^*) & \text{if } \phi_j^{\text{up}}(x^*) \geqslant \phi_j^{\text{low}}(x^*) \geqslant 0 \\ \phi_j^{\text{up}}(x^*) & \text{if } 0 \geqslant \phi_j^{\text{up}}(x^*) \geqslant \phi_j^{\text{low}}(x^*) \\ 0 & \text{if } \phi_j^{\text{up}}(x^*) \geqslant 0 \geqslant \phi_j^{\text{low}}(x^*) \end{cases} .$$

In other words, $\psi_j(x^*)$ is the (signed) distance of the credible interval from the null reference point 0, i.e. the minimum effect that can be estimated within the chosen confidence level. This distance is therefore null if 0 lies in the interval.

This significance-sensitive measure may be used instead of Shapley values in all the plots within the SHAP machinery, to specifically highlight the predictors that significantly contribute to the model. An example may be the following alternative to Feature importance values:

$$\text{CI-Dist}_j := \frac{1}{n}\sum_{i=1}^{n} |\psi_j(x^{(i)})|, \tag{3.3}$$

where $x^{(1)}, \dots, x^{(n)}$ is a given representative dataset. Another useful source of information is how many samples $x^{(i)}$ have $\psi_j(x^{(i)}) \neq 0$, i.e. the proportion of datapoints with a value of $\phi_j(x^{(i)})$ that significantly differs from zero. The two measures, combined, may give an overview of variable importance within a more inferential framework.

Chapter 4

Empirical evaluation

4.1 Methods

RuleFit-based models

The models that were compared were RuleFit as described by Friedman and Popescu, Horserule as described by Nalenz and Nillani and informative Horseshoe RuleFit, as described in Chapter 2. All three models included winsorization of linear terms by Equation 1.1, with $\beta = 0.025$.

Rule generation

In order to fit the three models, the set of rules to be added to the predictors had to be determined. Since the packages *pre* and *horserule* generate rules in different ways and rule generation is not part of our focus for this work, we decided to fit both models and combine their rules. The rules in *pre* and in *horserule* were built under standard settings, which thus produced an esnemble of rules produced by *partykit*'s *ctree* function [34] (from *pre*), *xgboost* [35] (from *horserule*) and *randomForest* [36] (also from *horserule*). The aim is to ensure that such a diverse set of rules produces more general comparisons of the different linear regressions that do not depend on how the rules are generated. This initial ensemble of rules was thus kept constant for all three models.

RuleFit

RuleFit was implemented manually by fitting a LASSO regression with the *cv.glmnet* function from the *glmnet* package [37]. This was applied on the set of predictors and rules described just above. RuleFit required the tuning of the ℓ_1 penalization parameter λ, which is cross-validated by the *cv.glmnet* function. The 1 standard error rule was applied, and the optimal λ was chosen to be as strong as possible within 1 standard deviation of the MSE from the optimal performance.

Horserule

Horserule was implemented manually by fitting the Horseshoe model from the *horseshoe* package [38], on the predictors rescaled by the factor A'_j as in Equation 2.3. This was shown above to be equivalent to the HorseRule structured penalization. HorseRule required the tuning of μ and η. We tried out five choices in total: the suggested default of $(\mu, \eta) = (1, 2)$ and then four variations in which μ and η were in turn doubled or halved: $(\mu, \eta) \in \{(0.5, 2), (2, 2), (1, 1), (1, 4)\}$.

Informative Horseshoe RuleFit

The informative Horseshoe RuleFit was implemented by fitting a Bayesian linear regression with informative Horseshoe prior through means of the package *infHS*, available on [39]. The regression was performed both with and without structured penalization of rules. Structured penalization (combined with standardization) was applied by rescaling the predictors by the factor A'_j as in Equation 2.3. In this case, the model was fit for the same hyperparamweter choices as Horserule, i.e. $(\mu, \eta) \in \{(1, 2), (0.5, 2), (2, 2), (1, 1), (1, 4)\}$. When no structured penalization was applied, rules were left unstandardized.

For both the penalized and unpenalized case, multiple sources of co-data were tried:

- A Linear vs. Rule co-data source, indicating whether the predictor is a rule or a linear term;
- A depth co-data source, distinguishing between rules of different depth (linear terms and depth-one rules are on the same level: they are equally favourable in terms of simplicity, and the first source already distinguishes between them);
- A co-data source for support, set to 1 for linear terms and to $\min(\bar{r}_j, 1 - \bar{r}_j)$ for rules.

The sources were added sequentially, in the order listed above, for a total of three different models, ranging from one to three sources of co-data.

Experiment

The machinery described in the previous chapters was tested on two synthetic datasets:

- **Friedman dataset 1 (D1):** First introduced in [40], with:

$$x_1, \ldots, x_5 \sim \mathcal{U}(0, 1),$$

$$\epsilon \sim \mathcal{N}(0, \sigma^2),$$

$$y = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \epsilon.$$

By default $\sigma^2 = 1$. However, this produces an irreducible error of less than 5% of the total variance. We therefore chose $\sigma^2 = 5$, which produces about 17% of irreducible error, for a more realistic setting.

- **Extra dataset (D2):** Proposed by us, to stimulate conflicting effects between rules and linear terms, but also to test interactions where splitting the importance evenly as RuleFit does might be unfair:

$$x_1, \ldots, x_{10} \sim \mathcal{U}(0,1),$$

$$\epsilon \sim \mathcal{N}(0,2),$$

$$\begin{aligned} y =& 2x_1(1 - \sin(3\pi x_1)) \\ &+ 4x_2 \\ &+ (2 \cdot I(x_3 < 0.5) \\ &+ 4 \cdot (x_3 \geqslant 0.5))x_3 \\ &+ 4I(x_4 > 0.5)x_4 \\ &+ 4I(|x_5 - 0.5| > 0.25)x_5 \\ &+ 4I(x_7 > 0.75)x_6 \\ &+ \epsilon. \end{aligned}$$

Both datasets were extended to have $p = 100$ predictors by adding noisy predictors from a uniform distribution in $[0, 1]$. For this experiment, the models were fit on $n = 1\,000$ training samples. All predictors were generated independently.

The methods mentioned above were compared to two tree ensembles:

Random Forest, as implemented in the *randomForest* function from the package *randomForest*[36]. Default values for the number of trees and the number of bagged predictors per split were chosen and the trend of the Out-of-Bag error over the increasing number of trees (Figure 4.1) suggests that the choice was appropriate.

XGBoost tree ensemble, as implemented in the *xgboost* function from the *xgboost* package [35]. The extreme gradient boosting was also chosen with its default parameters. A maximum depth of three was set, to facilitate interpretability.

The second part of our experiment consisted of a comparison of local and global importance measures. For this goal, we focussed on an informative Horseshoe RuleFit with a single co-data source, as further sources did not produce relevant changes. Moreover, as the purpose was a comparison between measures in contexts where multiple rules interact together and compensate eachother, this informative HorseShoe fit entailed no structured penalization of rules. For this comparison, the local importance measure suggested by Friedman and Popescu was compared to Shapley values. Friedman and Popescu's measure was computed twice: once in its original form, where absolute values were taken for every coefficient as in Equations 1.5, 1.6 and 1.8, and once in a sign-sensitive manner that did not take the absolute value into account, but still split the effect of the rule equally across predictors, in the spirit of Friedman and Popescu's definition. These measures were compared to the original contributions in the generating functions that defined the synthetic data, on page 28.

Shapley values and Friedman and Popescu's measures were also compared globally. In this case, however, there is no ideal measure to compare these importance metrics to. In the context of the effects of individual predictors, one might wish the following properties to hold:

- For a contribution $f_j(x_j)$ of the predictor x_j that is *centered* (i.e. $\mathbb{E}[f(x_j)] = 0$), the importance of the contribution should be proportional to the domain it covers: if we halve its support, for instance, then the importance should also be halved.
- For a *centered* contribution $f_j(x_j)$ of the predictor x_j, the importance of the contribution should be proportional to the magnitude of f_j: the contribution $g_j := 2 \cdot f_j$, for instance, should have twice as much importance as the contribution f_j.

These two properties combined require the importance to be proportional to the area under the centered contribution function $f_j(x_j)$. One may therefore define a target importance measure as $\mathbb{E}_{x_j}[|f_j(x_j)|]$, for contributions of an individual predictor. The measure we are proposing as benchmark for our experiment thus consists of the expected contribution of x_j, in absolute value. With this in mind, one might generalize this expected contribution to two-way interactions by marginalizing the expectation over the other predictor. If x_j forms a two-way-interacting contribution with $x_{j'}$ denoted by $f_{j,j'}(x_j, x_{j'})$, then this would mean writing this benchmark as $\mathbb{E}_{x_j}[\mathbb{E}_{x_{j'}}[|f_{j,j'}(x_j, x_{j'})|]]$. However, in marginalizing $|f_{j,j'}(x_j, x_{j'})|$, we do not have a centered function anymore. In other words, a more sensible extension would be:

$$\mathbb{E}_{x_j}[|\mathbb{E}_{x_{j'}}[f_{j,j'}(x_j, x_{j'})] - \mathbb{E}_{x_j}[\mathbb{E}_{x_{j'}}[f_{j,j'}(x_j, x_{j'})]]|],$$

which simplifies to $\mathbb{E}_{x_j}[|\mathbb{E}_{x_{j'}}[f_{j,j'}(x_j, x_{j'})]|]$ if we again consider $f_{j,j'}(x_j, x_{j'})$ to be centered. Note that, due to the presence of the absolute value, the importance of two predictors contributing in a two-way contribution is not the same. This measure was used as a benchmark against which to compare the candidate measures of global importance. While it is true that this generalization to two-way contributions is completely arbitrary, it did not impact the observed results, as these were also observed in the case of univariate contributions.

Inferential measures were also incorporated in the experiment, as te minimum effect $\psi_j(x^*)$ substituted $\phi_j(x^*)$ both for the computation of global imortance measures and for some local importance visualizations.

4.2 Results

Predictive accuracy

The comparison was firstly restricted to HorseRule and unpenalized informative Horseshoe. Table 4.1 shows the different mean squared errors on a test set of 10 000 new samples, for the fitted models listed above. In particular, we observe that our approach outperforms RuleFit, but not HorseRule. Moreover, the use of further sources of co-data does not seem to alter the resulting model: a distinction between rules and linear terms

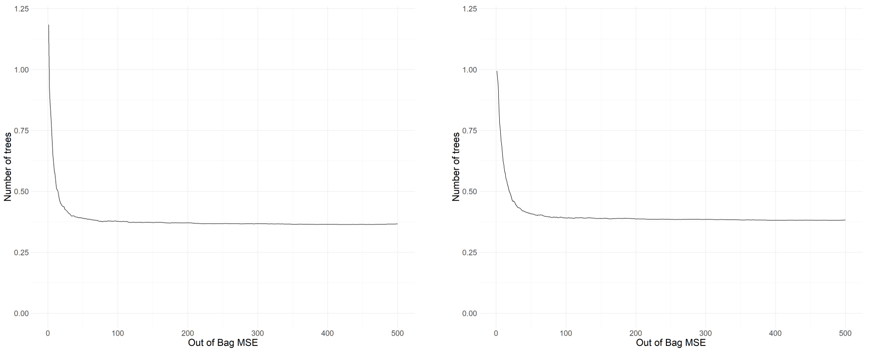

Figure 4.1: Out-of-bag Mean Squared Error of the Random Forest fits on the D1 dataset (left) and D2 dataset (right). The MSE is shown against the number of trees in the Random Forest.

is sufficient.

Table 4.2 shows the test Mean Squared Error when the same training and test sets are used for an informative Horseshoe RuleFit model with structrued penalization. The 5 combinations of parameters μ, η tried above were tested together with the use of one, two and three sources of co-data. Upon correct specification of the hyperparameters μ, η, the introduction of structured penalization improves the performance of the model on the dataset D2, but worsens it for the dataset D1. Compared to Horserule, our model also shows more variation under different shrinkage hyperparameters.

Coefficient distribution

For RuleFit, HorseRule and Informative Horshoe RuleFit, it is also possible to explore the distribution of the coefficients: the stacked bar charts in Figure 4.2 and 4.3 show the distribution of the 10% highest absolute values of the coefficients, on a square-root scale, for HorseRule and unpenalized Informative Horshoe RuleFit fitted on the D1 dataset. RuleFit, on the other hand, performs explicit variable selection, and it selects 103 terms out of 2651 for dataset D1 and 110 terms out of 3173 for dataset D2. Figures 4.4 show the distribution of its few non-zero coefficients for the datasets D1 respectively. We once again observe no significant difference between the case of one, two and three co-data sources, which in particular means that adding the second and third sources does not penalize more complex rules any further. As we can see, the Informative HorseShoe RuleFit relies less on linear terms, compared to HorseRule, while RuleFit does not select any linear terms at all.

When a stronger penalization of (complex) rules is enforced by augmenting the Informative HorseShoe RuleFit with structured penalization, the coefficient distributions change, as shown in Figure 4.5 for dataset D1. The figure only shows the distribution of the coefficients for the use of a single source co-data, as we once again point out no significant difference when multiple co-data sources are introduced. From these figures,

Model	Test MSE (D1)	Test MSE (D2)
Random Forest	0.358	0.397
XGBoost	0.270	0.269
RuleFit	0.242	0.253
Horserule, $\mu = 1, \eta = 2$	0.225	0.224
Horserule, $\mu = 0.5, \eta = 2$	0.225	0.224
Horserule, $\mu = 2, \eta = 2$	0.230	0.225
Horserule, $\mu = 1, \eta = 1$	0.227	0.228
Horserule, $\mu = 1, \eta = 4$	0.232	0.220
Inf. Horseshoe RuleFit, 1 source	0.234	0.244
Inf. Horseshoe RuleFit, 2 sources	0.233	0.244
Inf. Horseshoe RuleFit, 3 sources	0.232	0.244

Table 4.1: Mean Squared Error in prediction for a test set of $n = 10\,000$ samples, generated from the Friedman D1 generating function and the extra D2 generating function (see page 28). The outcome is scaled to have variance equal to 1, and the irreducible error amounts to about 0.173 for D1 and 0.177 for D2.

μ	η	Friedman Dataset (D1)			Extra Dataset (D2)		
		1 source	2 sources	3 sources	1 source	2 sources	3 sources
1	2	0.239	0.240	0.238	0.235	0.234	0.234
0.5	2	0.238	0.236	0.236	0.230	0.230	0.229
2	2	0.253	0.252	0.252	0.253	0.253	0.253
1	1	0.239	0.239	0.237	0.235	0.234	0.234
1	4	0.257	0.257	0.257	0.244	0.244	0.244

Table 4.2: Mean Squared Error in prediction applied to test sets of $n = 10\,000$ samples, generated from as discussed on page 28. The fitted model is an Informative Horshoe Rulefit, with a further structured penalization of complex rules, given by parameters μ, η.

we note that the informative HorseShoe prior gives more room for penalization of rule complexity, compared to its HorseRule counterpart, as the differentiation in shrinkage of rules of different complexity becomes stronger. Linear terms, however, are not relied picked up more.

Figures 6.1, 6.2, 6.3 and 6.4 in the Appendix show the same same plots for the dataset D2.

Local and global importance measures

Figure 4.6 shows the scatterplots for said local importance measures for the defining predictors and three noise variables, on dataset D2. The measures are compared against the true contributions (in purple), as defined in the data-generating function. Since predictors x_1 and x_2 from dataset D1 and x_6 and x_7 from dataset D2 have a joint effect, local importances are summed up and compared to the joint effect (all the importance measures used are additive) and shown separately in Figure 4.7. While all importance measures are adequately good at giving little to no importance to noise variables, these figures show Shapley values to be consistently more accurate estimations of the original contributions defined by the data generating functions on page 28, since the original measure by Friedman and Popescu flattens and overinflates the contributions while the sign-sensitive variant underestimates them.

Shapley values and the importance measure defined by Friedman and Popescu were also compared globally. Figure 4.8 compares the SHAP Feature Importance values ("Shapley", in blue) and the global importance measure introduced by Friedman and Popescu ("Friedman-Popescu", in yellow) to the benchmark measure discussed in the Methods section ("Target", in purple), for dataset D2. This figure also contains the measure defined in Equation 3.3 ("CI distance", in red) and the frequency of datapoints with a significantly non-null Shapley value ("Percentage significant", in green), which can take into account the uncertainty in the estimation. The SHAP Feature Importance values and the proposed CI distance measure prove to match the shape of the target measure quite accurately, while the original measure suggested by Friedman and Popescu overinflates feature importance when linear terms are combined with rules. This can be seen for instance from the extra dataset D2, where the contribution of x_1 is a nonlinear perturbation of x_2 which amounts to about the same effect in practice but is defined to encourage the concurrence of rules and the linear term. The Friedman-Popescu importance score for x_1 is nonetheless considerably higher than its counterpart for x_2.

Figures 6.5, 6.6 and 6.7 in the Appendix show the same trends for dataset D1, both locally and globally.

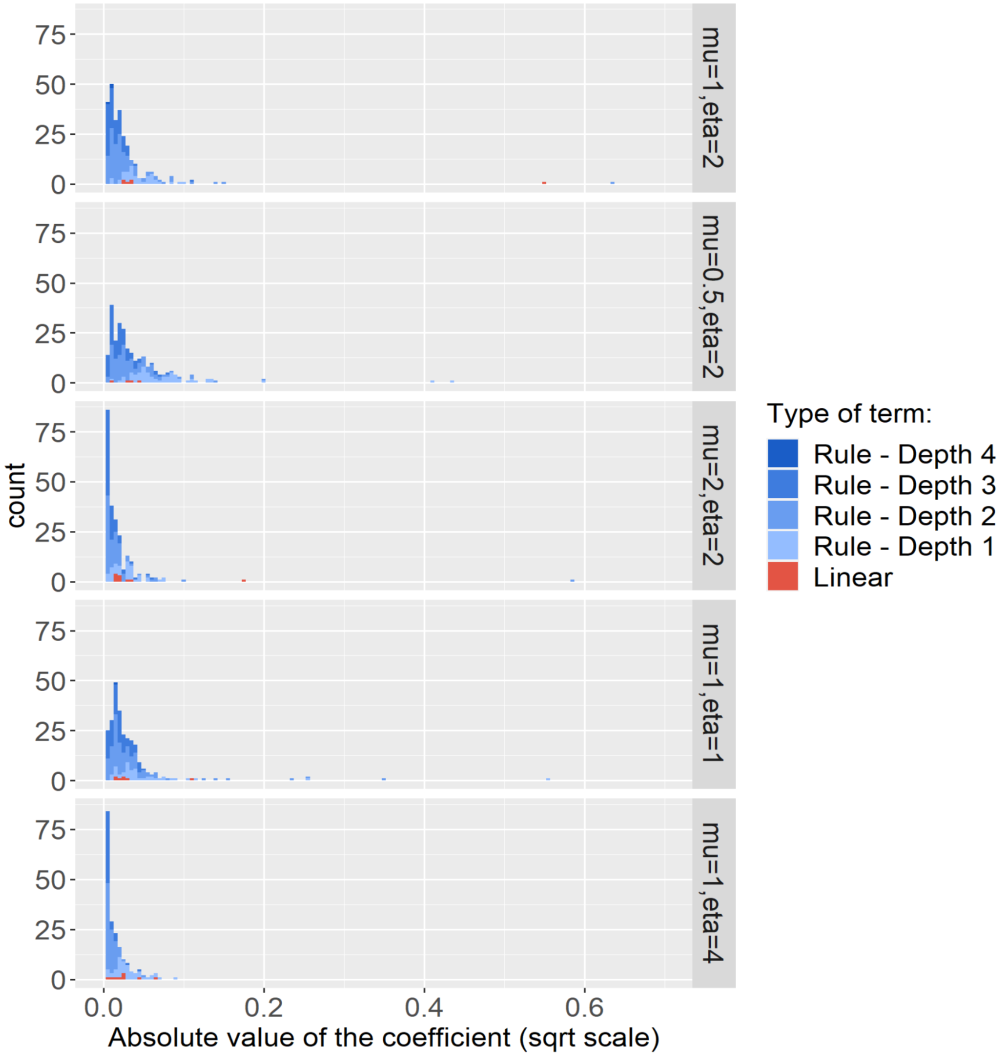

Figure 4.2: Stacked bars for the distribution of the absolute value of the coefficients, for a HorseRule model fit on the Friedman D1 dataset. Structured regularization is implemented with different combinations of parameters μ, η. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale.

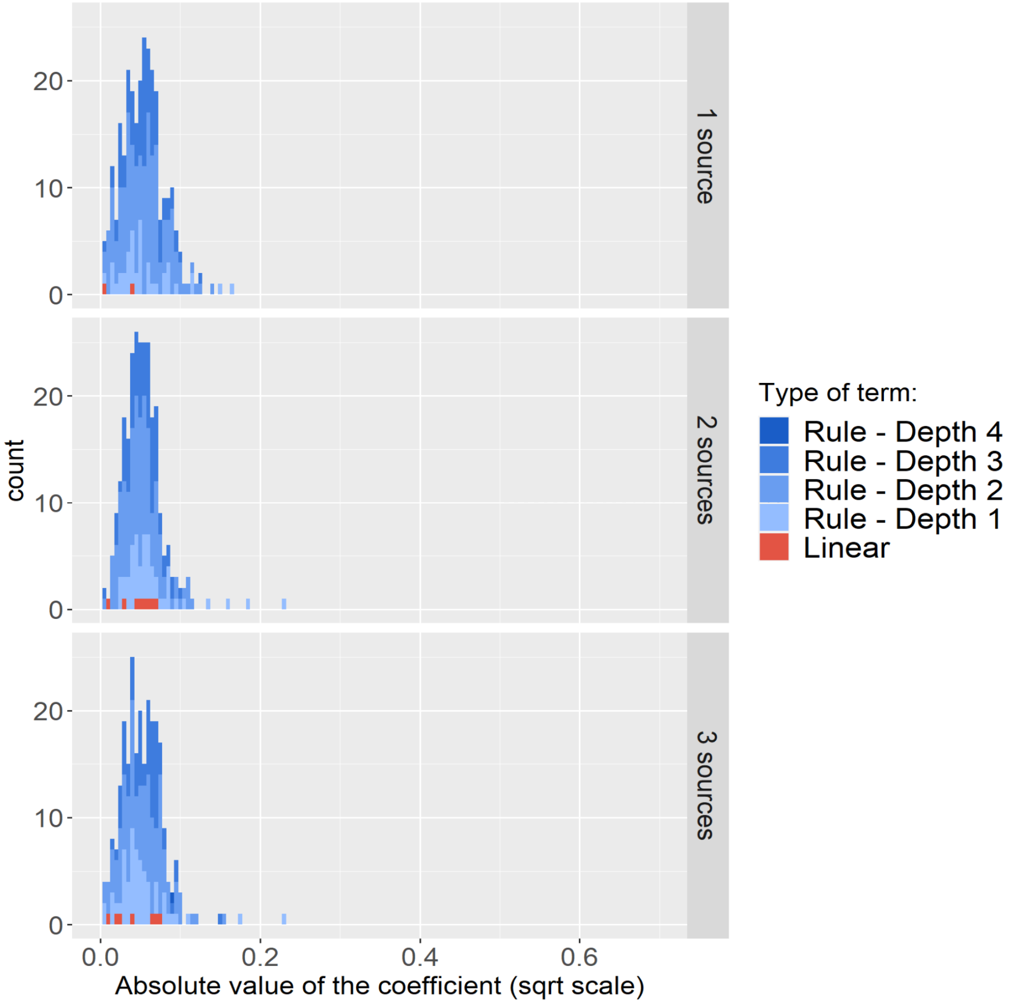

Figure 4.3: Stacked bars for the distribution of the absolute value of the coefficients, for an Informative Horseshoe RuleFit model without standardization or structured penalization, fit on the Friedman D1 dataset. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale.

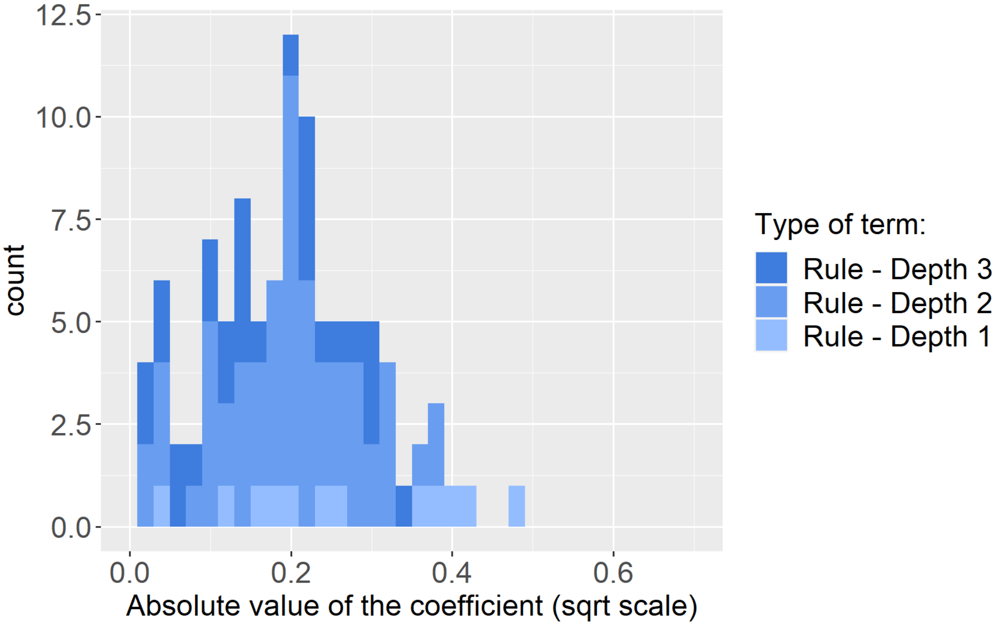

Figure 4.4: Stacked bars for the distribution of the absolute value of the non-zero coefficients, for a RuleFit model fit on the Friedman D1 dataset. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale. In this case, no rules are present, as they are all selected out of the model.

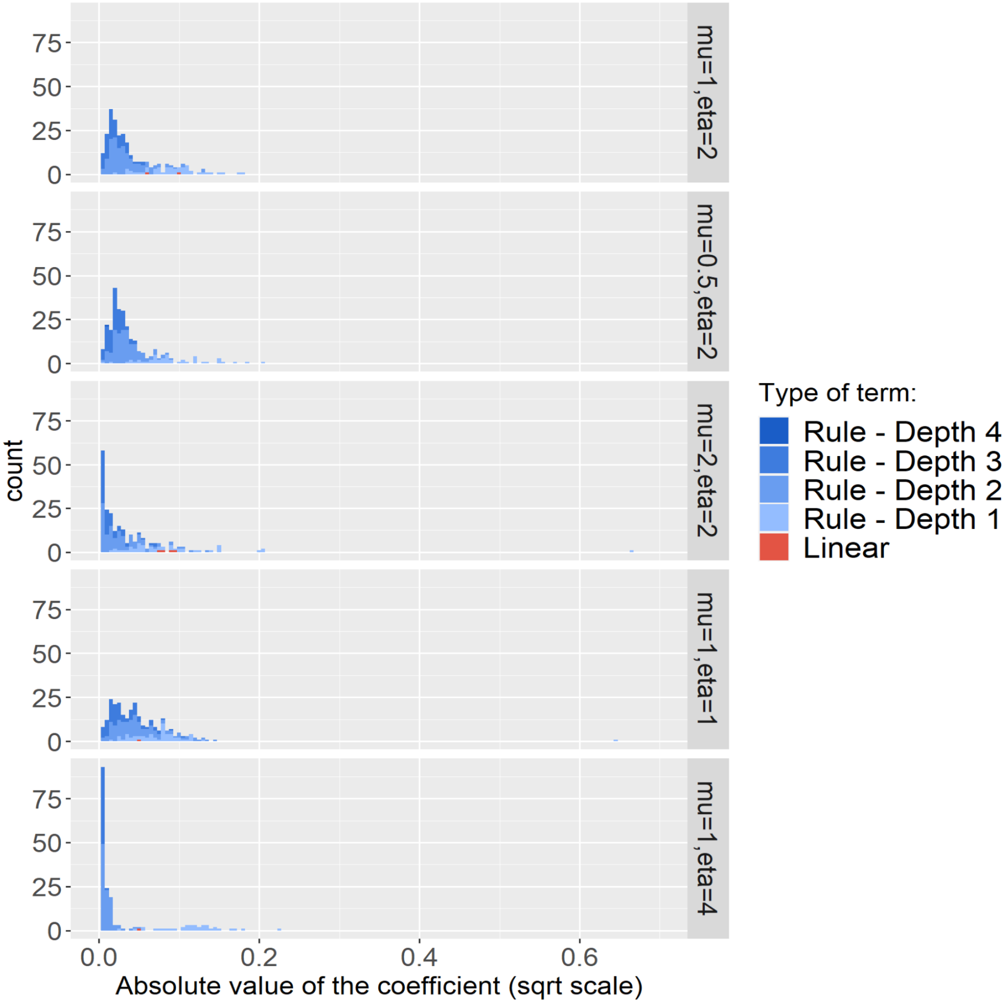

Figure 4.5: Stacked bars for the distribution of the absolute value of the coefficients, for an Informative Horseshoe RuleFit model with one source of co-data, fit on the Friedman D1 dataset. Structured regularization is also implemented, and different parameters μ, η are tested. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale. The coefficient distribution for the use of two and three co-data sources are similar.

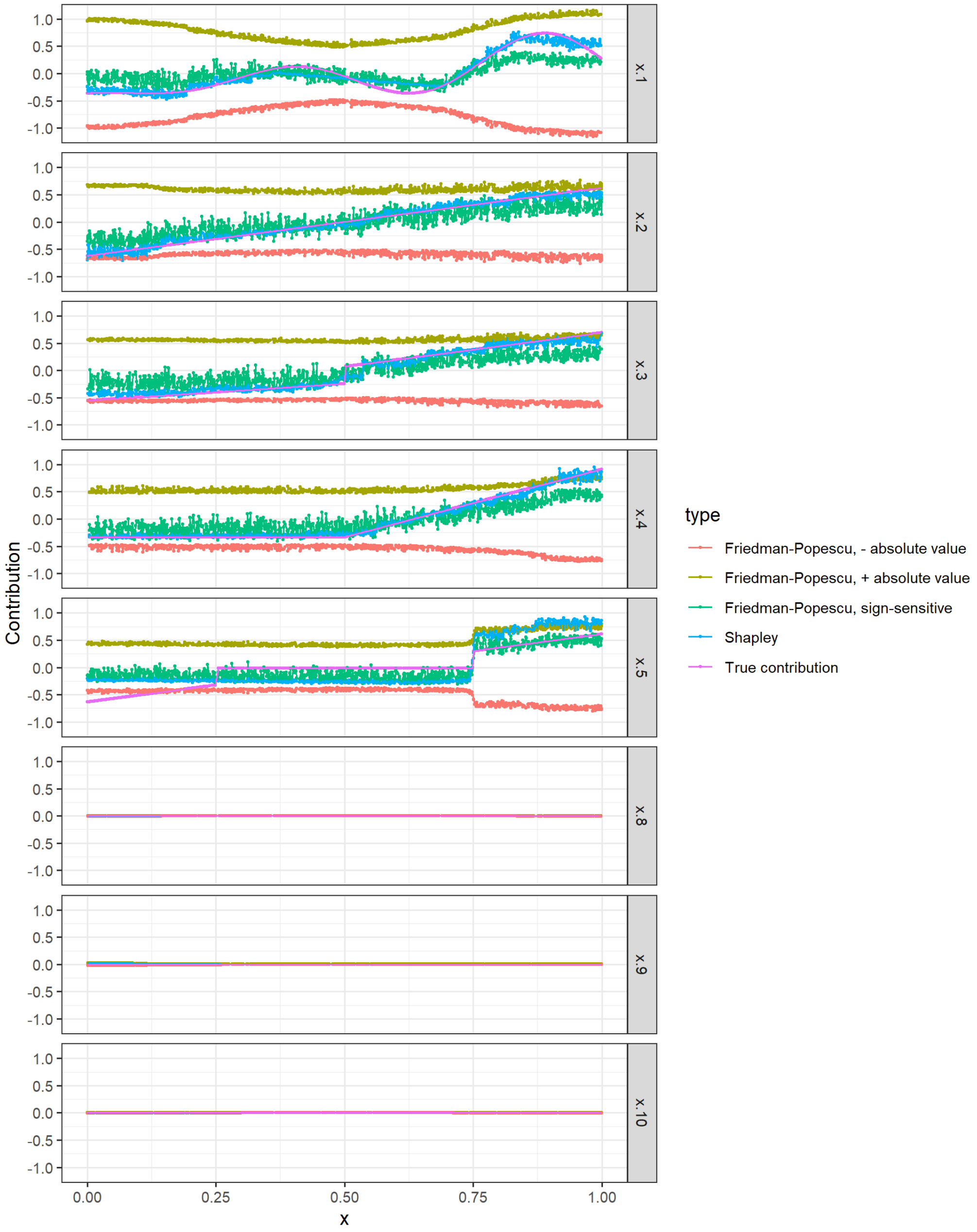

Figure 4.6: Estimated local importance of the first ten predictors for all points of the D2 training set, compared to the true underlying contribution, as defined by the generation function of the Extra dataset D2 (in purple). Predictors x_6 and x_7, which share a joint contribution, are excluded from the plot. Figure 4.7 is added to explicitly measure the estimated joint effect of these two variables.

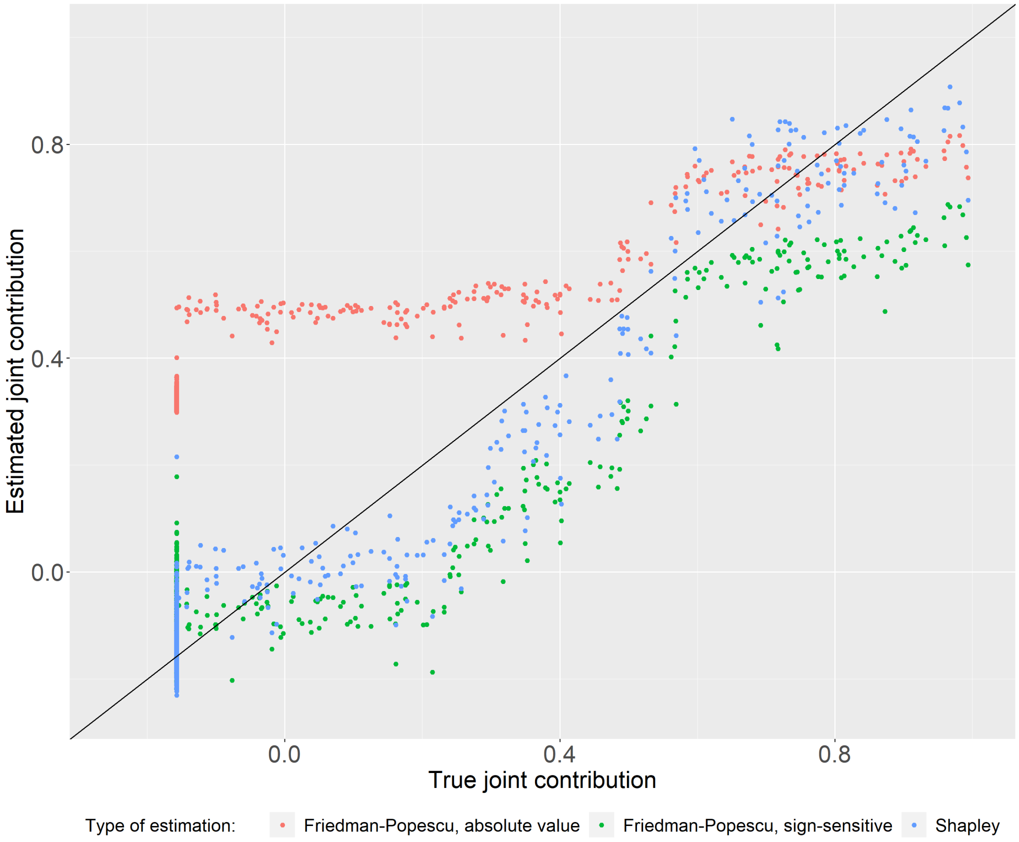

Figure 4.7: Estimated joint local importance of predictors x_6 and x_7 for all points of the D2 training set, compared to the true underlying contribution, as defined by the generation function of the Extra dataset D2. The black line represents the ideal positioning of points where the estimated and true contributions coincide.

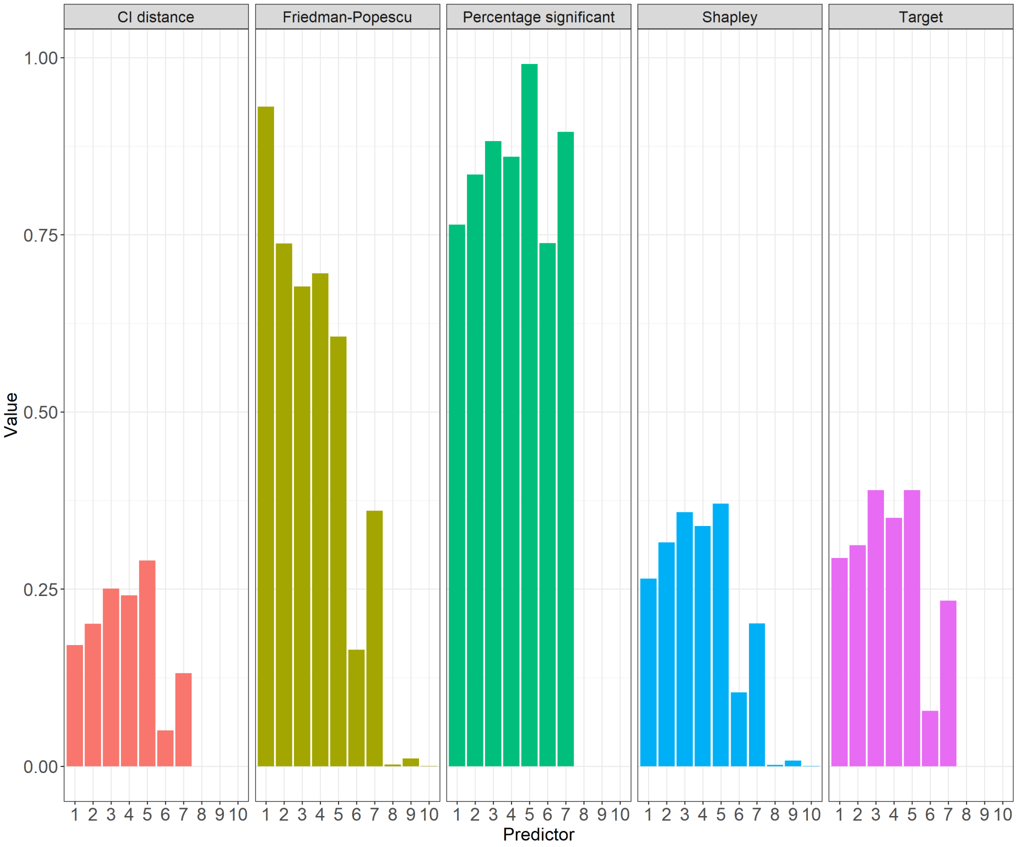

Figure 4.8: Different global importance measures compared, for the first 10 predictors, for an informative Horseshoe Rulefit model fit on the D1 dataset. In green, the percentage of points that have a 95% credible interval not containing 0. In purple, a possible target measure, as defined on page 30, based on the generating function.

To conclude our experiment, we extended the incorporation of uncertainty to the local importance measures, as discussed in Chapter 3. Figure 6.8 shows how a Shapley scatter plot may be enhanced with confidence bars. The heatmaps in Figure 6.9 and the summary plot in Figure 6.10 are examples of how the SHAP plots discussed in Chapter 3 may be updated by substituting ψ_j for ϕ_j. As we can see, these plots maintain the same trends depicted by the marginal Shapley values, but become more legible and produce higher contrasts between significant and non-significant contributions.

Chapter 5

Applying the machinery to the Helius study

To showcase how RuleFit can be applied in practice, we fit this model to an altered version of the data from the Helius study [3], where Cholesterol level was picked as the target outcome. The variables in this dataset are a mix of continuous (cholesterol level, packs of cigarettes smoked per year, age, systolic blood pressure on the logarithmic scale, BMI), dichotomous (sex, smoking yes/no, coffee consumption yes/no) and categorical predictors (ethnicity). The continuous variables were centered and standardized. The dichotomous variables were contrast-coded with values -1 and 1, which centers and standardizes them as if they had balanced supports. The categorical predictor for ethnicity was coded with four dummy variables. Each of them was centered and standardized assuming equal distribution among the five categories, using therefore a reference mean of 0.2 and reference standard deviation of '$sqrt5/2$. This dataset was then augmented with five extra variables, synthetically generated from a standard normal distribution. Four of such variables were kept as noise, while the fifth artificial predictor was used to synthetically induce a quadratic effect on cholesterol. After being standardized, cholesterol was replaced as follows:

$$chol \leftarrow chol + \frac{1}{8} artificial^2.$$

The newly obtained outcome variable has a variance of approximately 1.033. In total, this dataset contained $p = 12$ predictors of the outcome variable cholesterol. An Ordinary Least Squares (OLS) linear model and an informative Horseshoe RuleFit with structured rule penalization for $\mu = 1, \eta = 2$ were fit on the same training subset of size $n = 2\,000$, while the remaining 19 570 observations were used as test samples to assess predictive accuracy. Table 5 shows the test Mean Squared Error of both models. As we can see, the informative HorseShoe RuleFit is better at prediction. Looking at the Shapley value scatterplots in Figure 5.1, we can connect the higher accuracy of the informative Horseshoe with its ability to reconstruct the nonlinear contribution of age and the synthetic predictor.

The global importance measure that we introduced in Chapter 3 is shown in Figure

Model	Test MSE	Test R^2
Linear Model	0.950	0.090
Informative Horseshoe RuleFit	0.920	0.119

Table 5.1: Test Mean Squared Error and R squared score observed on 19570 of the 21570 observations from the Helius dataset. Performance is compared between an Ordinary Least Squares linear model and an Informative Horseshoe RuleFit. The latter performs sensibly better. The outcome variable has a variance of approximately 1.033.

5.2 and suggests that age is by far the most influential predictor, followed by a moderate contribution of the ethnicity and small contributions of the BMI and the quadratic, artificial effect.

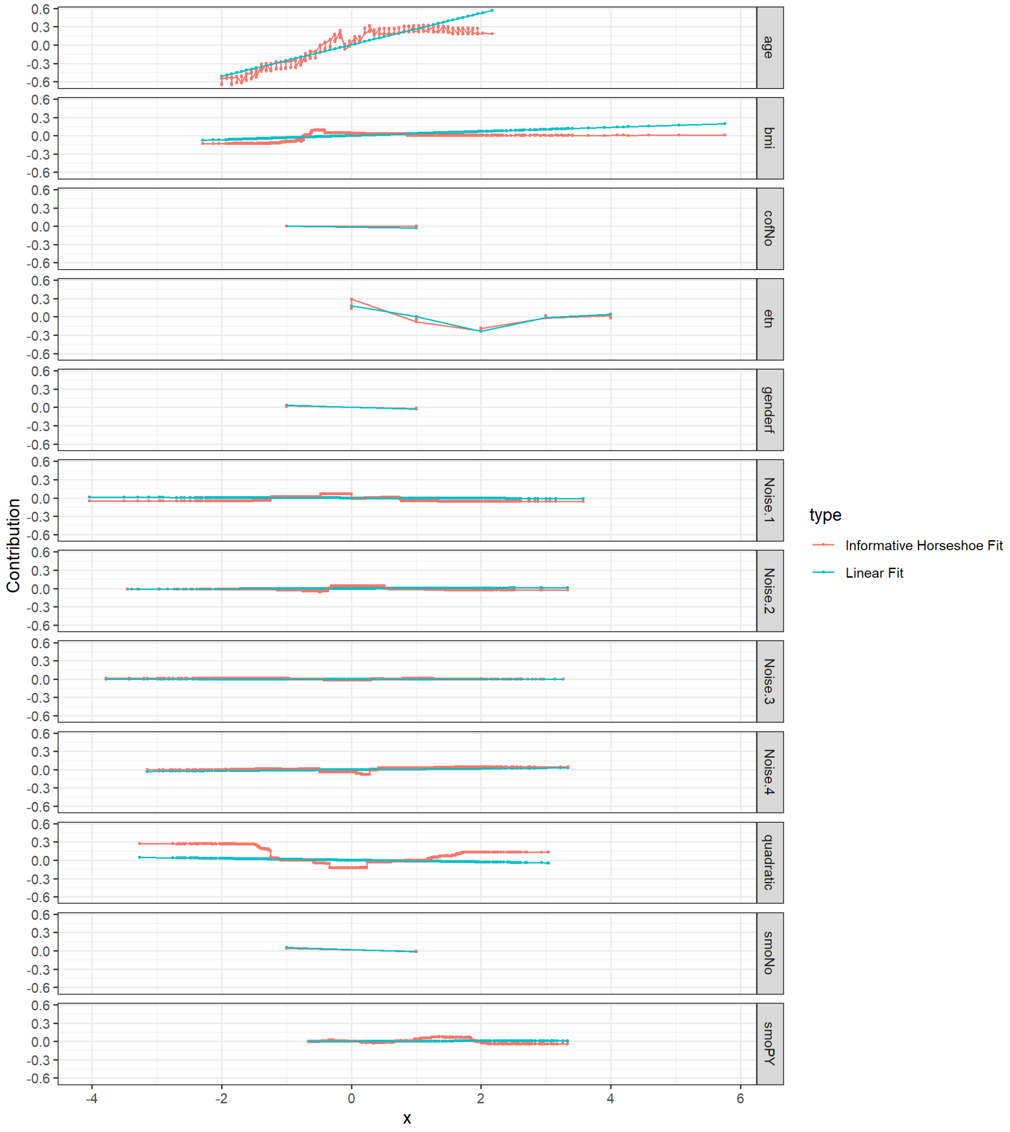

Figure 5.1: Estimated local importance for all predictors and all training datapoints of the Helius dataset, under an Informative Horseshoe RuleFit model.

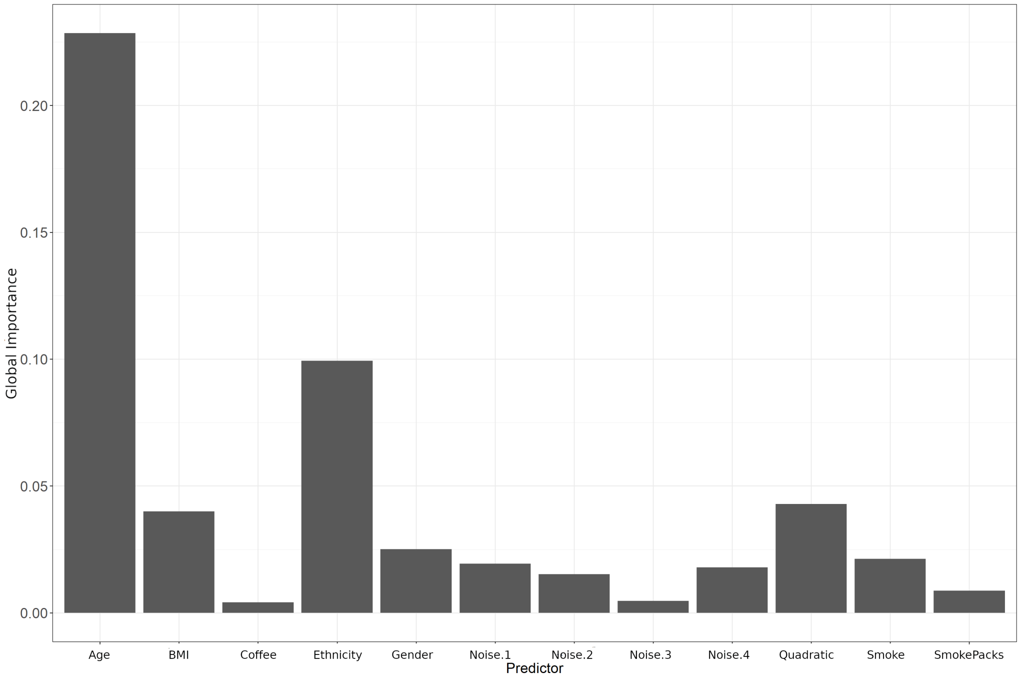

Figure 5.2: Estimated global importance for all predictors of the Helius dataset, under an Informative Horseshoe RuleFit model.

Chapter 6

Discussion

We presented a new approach that combines RuleFit, Horserule, co-data and Shapley values. In doing so, we bridged rule standardization and structured shrinkage. Our substitution of the horseshoe prior by the informative horseshoe renders a model whose adaptive shrinkage can naturally distinguish between linear and rule-based terms. This differentiation, however, currently favours rules over linear terms. This may be explained by the fact that the rules have been generated on the same dataset as the final fit, and are therefore prioritized by the adaptivity of the informative Horseshoe. On the other hand, when combined with the structured penalization of rules, the use of co-data enhances the structurization of shrinkage, leaving room for a more aggressive shrinkage of complex rules. While the predictive performance of our approach is lower than that of HorseRule, the model we introduce opens to the possibility of incorporating further sources of expert knowledge on the predictors.

Our experiments also arguably gave us more insight into the contribution of each feature in prediction: Shapley values were introduced in the context of RuleFit as a new importance measure that has a more rigorous origin and a more easily communicable interpretation, compared to their pre-existing counterpart suggested by Friedman and Popescu. In our simulations, this new measure proved to be not only more accurate on a local level but also to generalize better globally and to reflect the desirable properties of scalability with respect to the magnitude and support. Combining Shapley values with Bayesian regression also produces credible intervals, which enables us to both perform inference and consider a parallel measure that takes into account the uncertainty in the estimation. This measure also has an interpretable definition as minimum effect estimated under a certain confidence level.

In implementing this machinery, we derived an explicit formula to estimate marginal Shapley values for tree ensembles, and an algorithm to calculate them at a computational speed that is comparable to the TreeeSHAP algorithm, while also being more suitable to our context, where the tree ensemble is decomposed as a rule ensemble and where a Bayesian regression requires the Shapley values to be re-computed multiple times. Furthermore, this formula gives us more transparency with respect to the individual contributions of each rule. As such, this allows to open up the black box of Shapley values, which now can be traced back to the contributions of individual model components (rules and or/linear terms). It also grants us a chance to decide convergence criteria and use importance sampling when marginal Shapley values are computed under a predetermined joint distribution of the predictors.

On the other hand, our approach comes with limitations: first, the use of the informative Horseshoe prior does not induce exact sparsity like the RuleFit based on LASSO regression. This may be solved by performing posterior selection [9] [41]. Second, even when compared to another Bayesian regression such as HorseRule, the informative horseshoe still shrinks rules less, at the expense of linear terms, without any gain in predictive performance.

In terms of feature importance scores, our uncertainty estimation is only measured in terms of the posterior distribution of the coefficients, and the instability in the generation of the rules is not taken into account. Without any quantification of this instability, it is hard to tell whether the uncertainty is correctly estimated, especially since our experience suggests that the different ways of generating rules have an impact on the predictive performance of the model.

The aforementioned limitations would suggest further research into how the rules are being generated: for instance, a co-data matrix may be added to take the source of the rule into account, so as to encourage a higher shrinkage of rules that are generated with a less accurate algorithm. A broader, more diverse way to construct rules might drastically reduce the uncertainty in rule generation. Furthermore, to encourage the use of linear terms over rules, one could leave the linear terms unpenalized, to correct for the unfair advantage of rules, or fit the the tree ensembles on the residuals from a linear model, rather than on the full outcome y. The latter approach, however, might induce more instability in the generation of rules, and might therefore require a more rigorous analysis of how uncertainty propagates from the rule generation to the final model. Another possible direction for further research is the development of ways to use co-data: more specifically, the information about predictors needs to be transferred to the rule terms. For instance, such co-data may be defined as a weighted average of the co-data of the predictors involved in each rule. The weights may be chosen as the support of each sub-rule or, even better, the marginal Shapley values themselves, as they already measure the contribution of each predictor in a rule. Hierarchical approaches, where the first splitting predictor determines the co-data value for the rule, might also be justified by the greedy implementation of most tree-generating algorithms.

Shapley-based importance measures may also be explored further: the manner in which the importance is naturally decomposed into contributions of individual rules has not yet been linked to interactions between predictors. Our formula can straightforwardly be adjusted to also compute marginal Shapley interaction values, to give more insight into moderation effects between predictors.

Aknowledgments and Code

The code for both experiments is available on the GitHub page `https://github.com/GiorgioSpadaccini/informative-horseshoe-rulefit`.
The dataset from the Helius study is not available to the public, and is therefore not included in the repository. The synthetic datasets and their generating functions discussed on Chapter 4 are available and can be used to reproduce the code.

Appendix

Figures

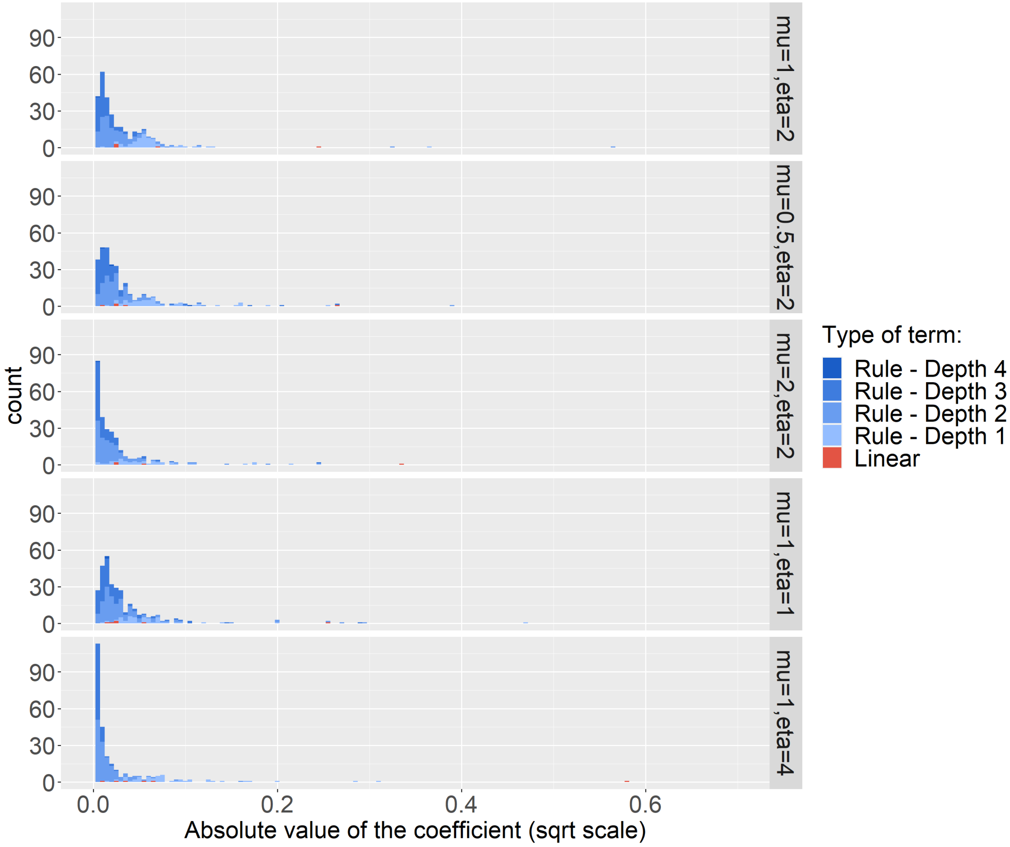

Figure 6.1: Stacked bars for the distribution of the absolute value of the coefficients, for a HorseRule model fit on the Extra D2 dataset. Structured regularization is implemented with different combinations of parameters μ, η. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale.

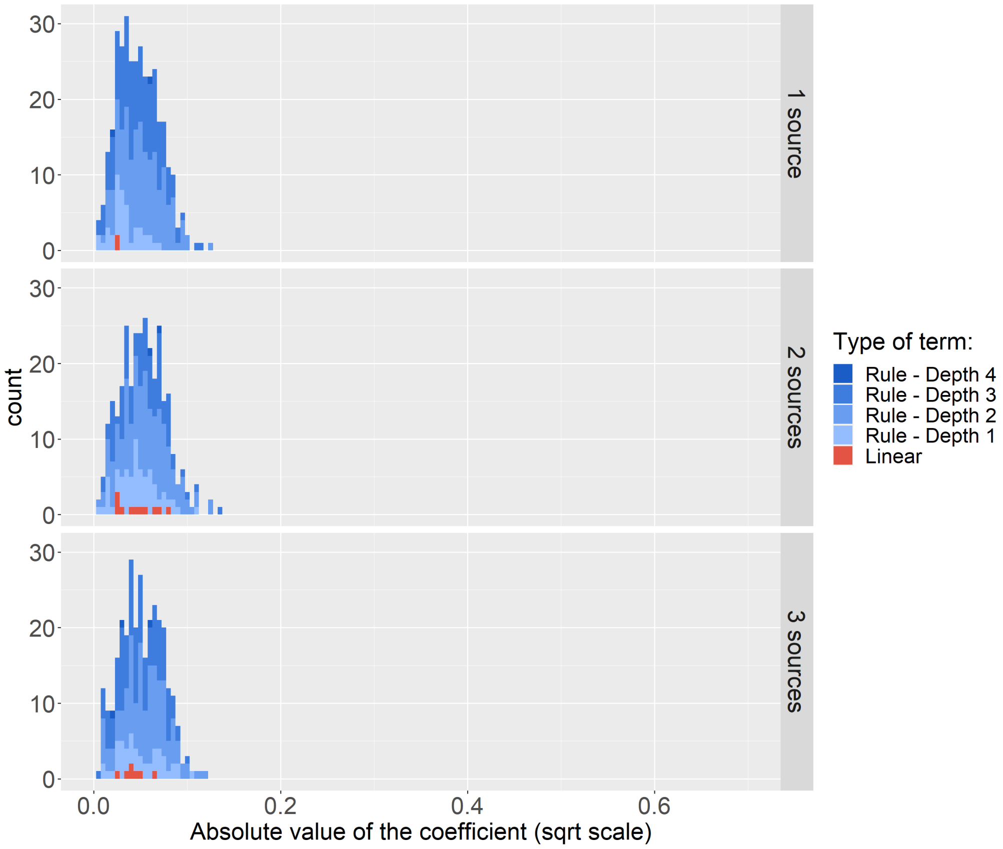

Figure 6.2: Stacked bars for the distribution of the absolute value of the coefficients, for an Informative Horseshoe RuleFit model without standardization or structured penalization, fit on the Extra D2 dataset. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale.

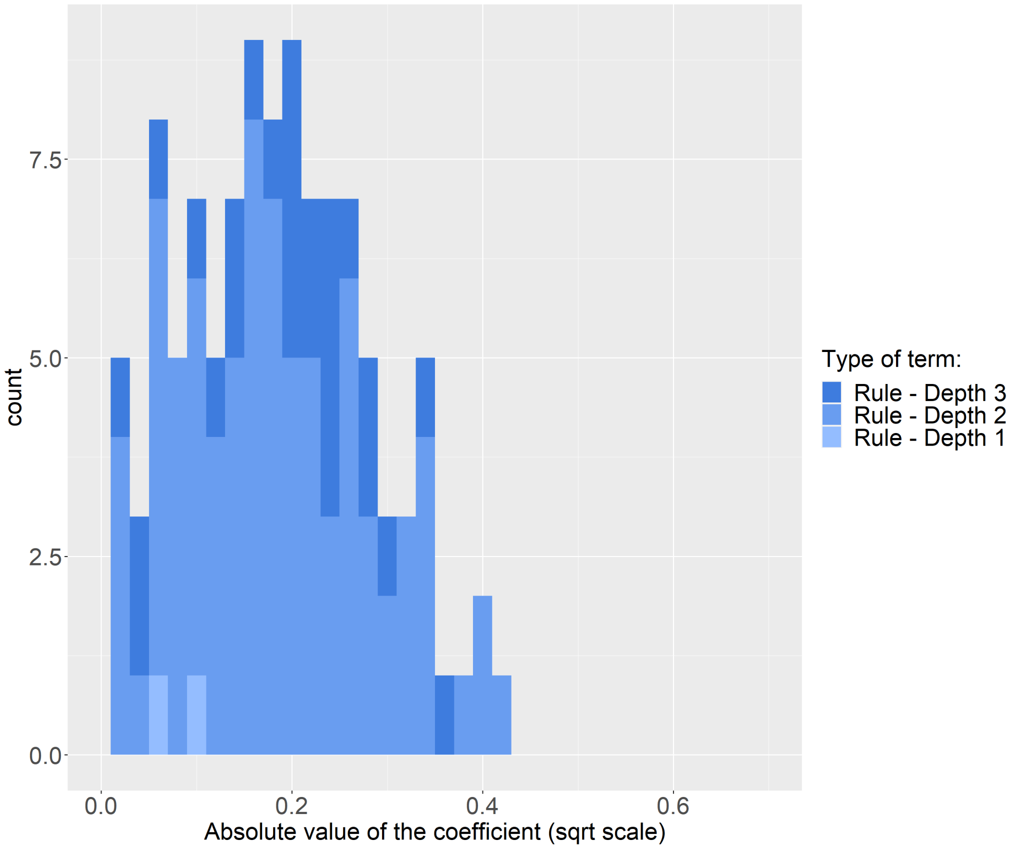

Figure 6.3: Stacked bars for the distribution of the absolute value of the non-zero coefficients, for a RuleFit model fit on the Friedman D1 dataset. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale. In this case, no rules are present, as they are all selected out of the model.

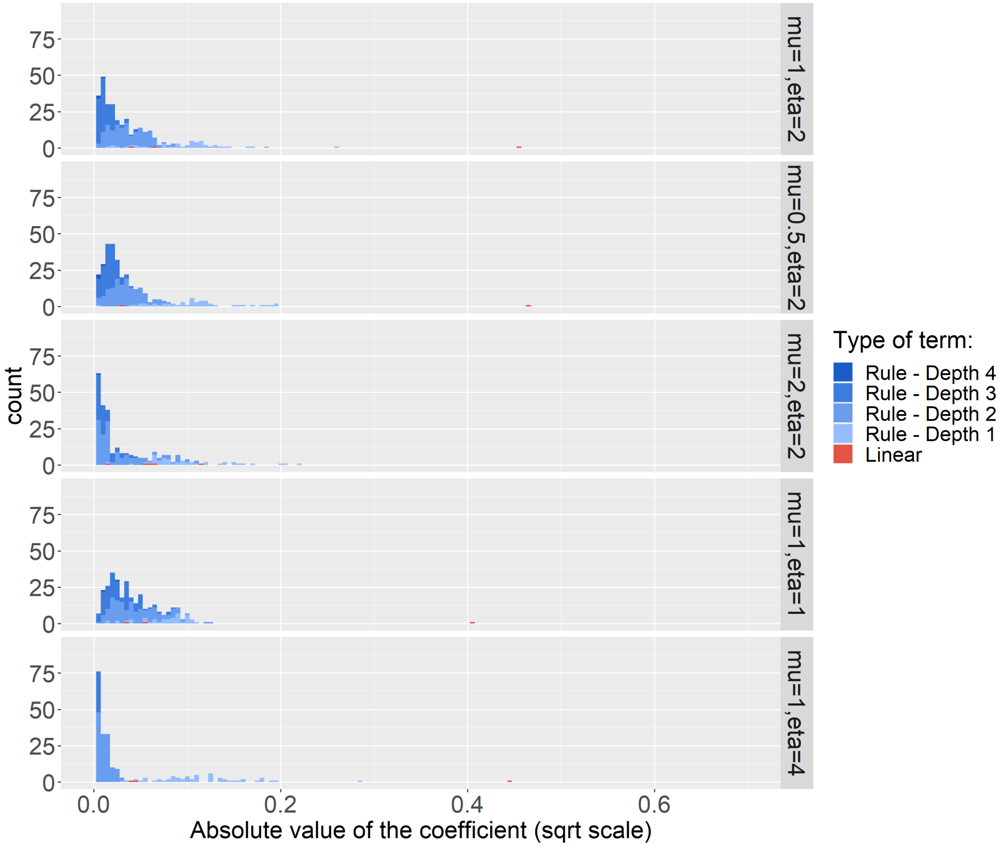

Figure 6.4: Stacked bars for the distribution of the absolute value of the coefficients, for an Informative Horseshoe RuleFit model with one source of co-data, fit on the Extra D2 dataset. Structured regularization is also implemented, and different parameters μ, η are tested. The stacked bars are split by colour: blue for rules and red for linear terms. Darker shades of blue represent deeper rules. The coefficients are on a square-root scale. The coefficient distribution for the use of two and three co-data sources are similar.

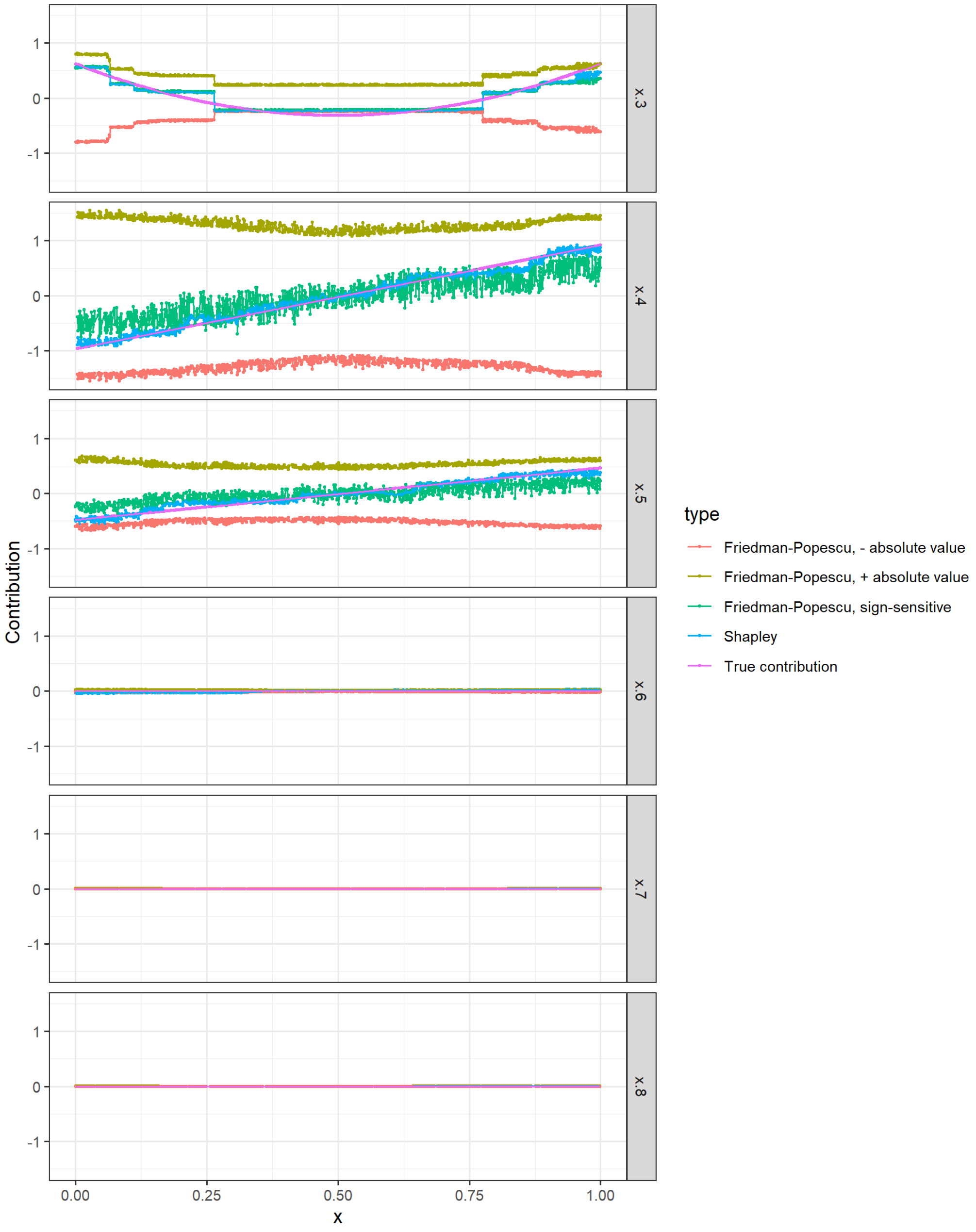

Figure 6.5: Estimated local importance of the first eight predictors for all points of the D1 training set, compared to the true underlying contribution, per Friedman 1 generation function (in purple), excludingthe first two variables. For predictors x_1 and x_2, which share a joint contribution, Figure 6.6 is added to explicitly measure the estimated joint effect.

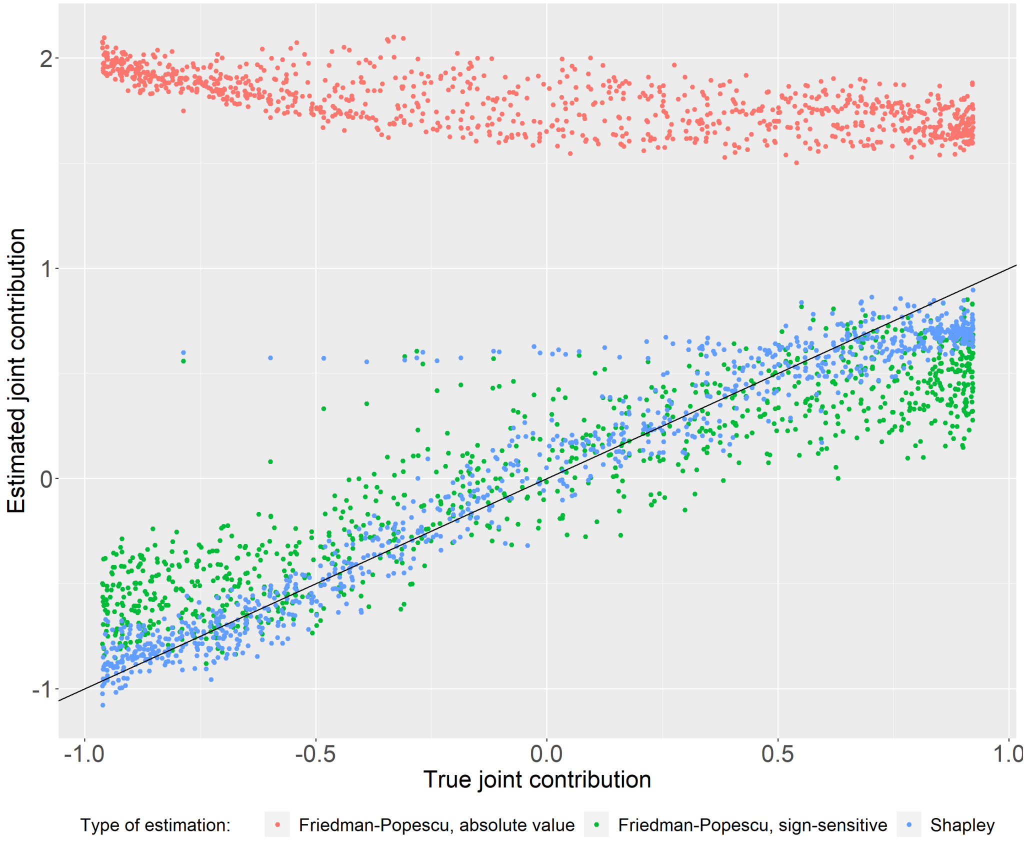

Figure 6.6: Estimated joint local importance of predictors x_1 and x_2 for all points of the D1 training set, compared to the true underlying contribution, per Friedman 1 generation function. The black line represents the ideal positioning of points where the estimated and true contributions coincide.

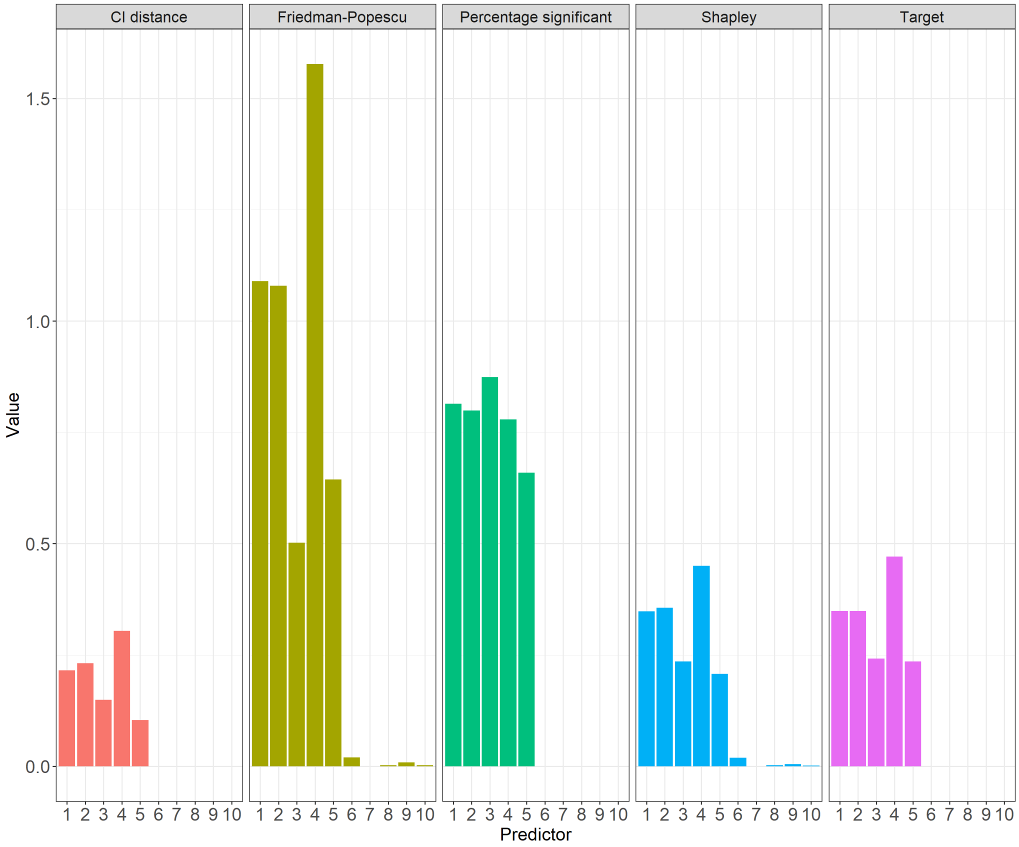

Figure 6.7: Different global importance measures compared, for the first 10 predictors, for an informative Horseshoe Rulefit model fit on the D1 dataset. In green, the percentage of points that have a 95% credible interval not containing 0. In purple, a possible target measure, as defined on page 30, based on the generating function.

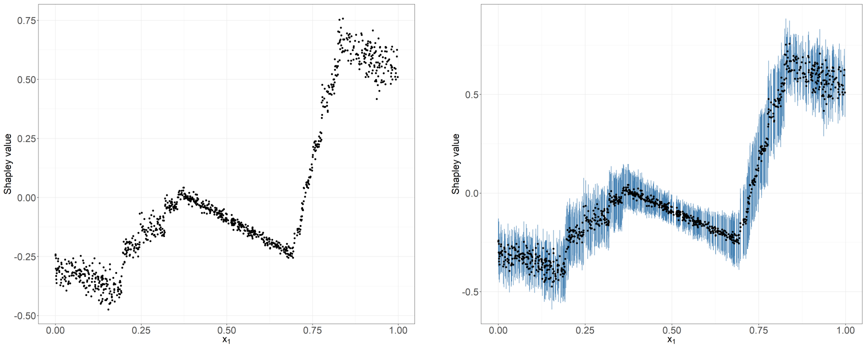

Figure 6.8: Scatterplot for local importance measures of predictor x_1 from Dataset D2, before (left) and after (right) being enhanced with credible intervals.

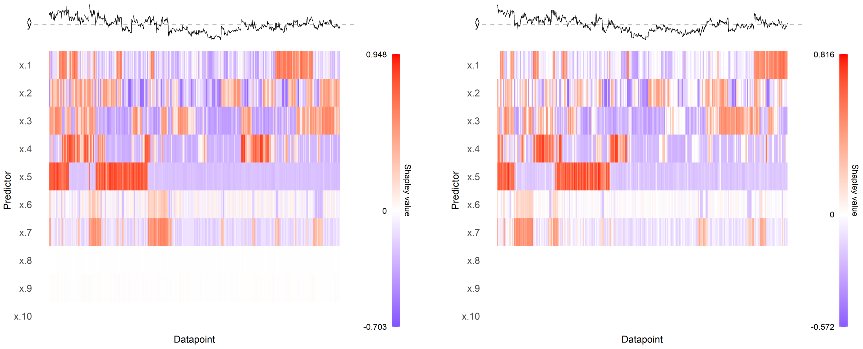

Figure 6.9: SHAP heatmap for Dataset D2, with marginal Shapley values ϕ_j (left) and signed credible interval distance ψ_j (right).

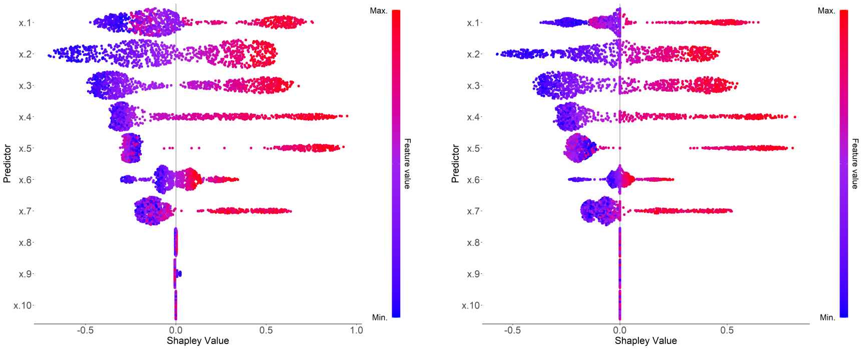

Figure 6.10: SHAP summary plot for Dataset D2, with marginal Shapley values ϕ_j (left) and signed credible interval distance ψ_j (right).

Proofs

Proof of Lemma 1

Lemma. *Consider a function $F : \mathbb{R}^p \to \mathbb{R}$, and take $q < p$ such that $F(x_1, \ldots, x_p)$ only depends on q of the p total predictors, say $x_{j_1}, \ldots, x_{j_q}$. Then the Shapley values for F may be computed by only focussing on $x_{j_1}, \ldots, x_{j_q}$: if $j \in \{j_1, \ldots, j_q\}$, then:*

$$\phi_j(x^*) = \sum_{S \subseteq \{j_1,\ldots,j_q\}\setminus\{j\}} \frac{1}{q\binom{q-1}{|S|}} \Big(\mathbb{E}[F(x_1, \ldots, x_p)|x_j = x_j^*, x_S = x_S^*] \\ - \mathbb{E}[F(x_1, \ldots, x_p)|x_S = x_S^*] \Big).$$

If $j \notin \{j_1, \ldots, j_q\}$, then $\phi_j(x^*) = 0$.

Proof. Without loss of generality, we may assume that $j_1 = 1, \ldots, j_q = q$. Furthermore, it suffices to prove this Lemma for $q = p - 1$. If we then iterate the argument multiple times and remove all non-contributing predictors one by one, the argument is generally proven.
Let us also use the notation:

$$\Delta\mathbb{E}_S := \mathbb{E}[F(x_1, \ldots, x_p)|x_j = x_j^*, x_S = x_S^*] - \mathbb{E}[F(x_1, \ldots, x_p)|x_S = x_S^*].$$

Note that $\phi_p(x^*) = 0$ is trivial, since the independence of F from x_p means that $\Delta\mathbb{E}_S = 0 \forall S$.
For the remaining cases, let us assume that we need to compute the Shapley value for the first predictor, for simplicity.
Now we can write:

$$\begin{aligned}\phi_1(x^*) &= \sum_{S \subseteq \{2,\ldots,p\}} \frac{1}{p\binom{p-1}{|S|}} \Delta\mathbb{E}_S \\ &= \sum_{\substack{S \subseteq \{2,\ldots,p\} \\ S \not\ni p}} \frac{1}{p\binom{p-1}{|S|}} \Delta\mathbb{E}_S + \sum_{\substack{S \subseteq \{2,\ldots,p\} \\ S \ni p}} \frac{1}{p\binom{p-1}{|S|}} \Delta\mathbb{E}_S\end{aligned}$$

Now let us write the subsets $S \subseteq \{2, \ldots, p\}$ containing p as $S = Z \cup \{p\}$. For these subsets, $|S| = 1 + |Z|$. Furthermore, since F does not depend on x_p, we know that $\Delta\mathbb{E}_S = \Delta\mathbb{E}_Z$.

$$\phi_1(x^*) = \sum_{S \subseteq \{2,\ldots,p-1\}} \frac{1}{p\binom{p-1}{|S|}} \Delta\mathbb{E}_S + \sum_{Z \subseteq \{2,\ldots,p-1\}} \frac{1}{p\binom{p-1}{1+|Z|}} \Delta\mathbb{E}_Z.$$

Now both summations are over the same subsets, so we can join them:

$$\begin{aligned}
\phi_j(x^*) &= \sum_{S\subseteq\{2,\dots,p-1\}} \frac{1}{p}\Big(\frac{1}{\binom{p-1}{|S|}} + \frac{1}{\binom{p-1}{1+|S|}}\Big)\Delta\mathbb{E}_S \\
&= \sum_{S\subseteq\{2,\dots,p-1\}} \frac{1}{p!}\Big(|S|!(p-1-|S|)! + (|S|+1)!(p-2-|S|)!\Big)\Delta\mathbb{E}_S \\
&= \sum_{S\subseteq\{2,\dots,p-1\}} \frac{|S|!(p-2-|S|)!}{p!}\Big((p-1-|S|) + (|S|+1)\Big)\Delta\mathbb{E}_S \\
&= \sum_{S\subseteq\{2,\dots,p-1\}} \frac{|S|!(p-2-|S|)!\cdot p}{p!}\Delta\mathbb{E}_S \\
&= \sum_{S\subseteq\{2,\dots,p-1\}} \frac{|S|!(p-2-|S|)!}{(p-1)!}\Delta\mathbb{E}_S \\
&= \sum_{S\subseteq\{2,\dots,p-1\}} \frac{1}{(p-1)\binom{p-1}{|S|}}\Delta\mathbb{E}_S.
\end{aligned}$$

This concludes the proof, as the last step is precisely the formula for Shapley values for the predictors $x_1, \dots, x_{p-1}$. □

Proof of Lemma 2

Lemma. *For any $a, b, c \in \mathbb{N}$ such that $c \leqslant b$, the following equality holds:*

$$\sum_{l=0}^{c}\binom{a+l}{l}\binom{b-l}{c-l} = \binom{a+b+1}{c}.$$

Proof. Consider the setting where you have $a+b$ elements and you want to count all possible subsets of size c that may be taken from these $a+b$ elements. We know that this number is, in total, $\binom{a+b}{c}$. Assume to partition these $a+b$ elements into two groups, G_1 and G_2.

A more cumbersome way to compute this number of possible subsets would be to sum over all possible numbers l of elements that you take from the group G_1, and then consider in how many ways you may take l elements from G_1 and the remaining $c-l$ from G_2. In other words, if we know that we're taking exactly l elements from G_1 and the remaining $c-l$ elements from G_2, then in total we have $\binom{|G_1|}{l}\binom{|G_2|}{c-l}$ ways of having such subsets of the $a+b$ elements. If we sum over all possible numbers of elements that we get from G_1, we obtain a very similar summation to the one above, and it is equal to $\binom{a+b}{c}$.

The only difference is: in the summation above G_1 is not fixed: in the summation above, we may interpret $\binom{a+l}{l}\binom{b-l}{c-l}$ as the number of groups that can be taken if $G_1 =: G_1^{(l)}$ is made of the first $a+l$ elements and $G_2 =: G_2^{(l)}$ is made of the last $b-l$ elements. So, as l increases, $G_1^{(l)}$ also "steals" an elements from $G_2^{(l)}$. This complicates calculations because it means that some subsets are being counted twice, and therefore need to be subtracted in order to have the total number of subsets be $\binom{a+b}{c}$.

More specifically, for a fixed l, we know that the sets that will also be counted again at the next iteration with $l+1$, and that therefore need to be subtracted from the count, are the ones that have exactly l elements in $G_1^{(l)}$ but then will also have exactly $l+1$ elements in $G_1^{(l+1)}$. This happens exactly when the $(a+l+1)$-th element (the one element by which $G_1^{(l+1)}$ and $G_1^{(l)}$ differ) is in the subset. Therefore, such type of double-counted subset contains:

- the $(a+l+1)$-th element
- any l elements in $G_1^{(l)}$
- any $c-l-1$ elements in $G_2^{(l+1)}$

Figure 6 shows an example.

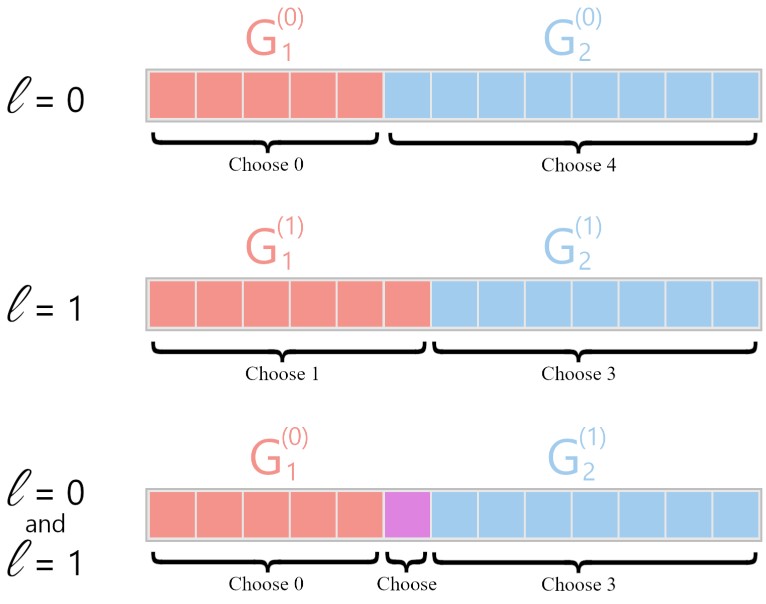

Figure 6.11: Example of subsets being counted twice for $a=5, b=8, c=4$ and $l \in \{0,1\}$. A subset here is counted twice if it has no elements in $G_1^{(0)}$ and one element in $G_1^{(1)}$, four elements in $G_2^{(0)}$ and three elements in $G_2^{(1)}$. This happens when the purple element is in the set, then no elements are on its left and $3 = c-1$ elements are on the right.

This means that at every l-th iteration there is $\binom{a+l}{l}\binom{b-l-1}{c-l-1}$ subsets that need to be removed from the count because they will also be counted at the next iteration. This quantity is not subtracted at the last iteration, as there will be no next iteration to double-count the subsets.

This means that we can write:

$$\binom{a+b}{c} = \sum_{l=0}^{c} \binom{a+l}{l}\binom{b-l}{c-l} - \sum_{l=0}^{c-1} \binom{a+l}{l}\binom{b-l-1}{c-l-1}$$

Now that we have this formula, we can prove the equation above by induction on c. For $c = 0$, we have:

$$\sum_{l=0}^{0} \binom{a+l}{l}\binom{b-l}{c-l} = \binom{a}{0}\binom{b}{0} = 1 = \binom{a+b+1}{0} = \binom{a+b+1}{c}$$

So equality holds. For the induction step, we use the formula above:

$$\sum_{l=0}^{c} \binom{a+l}{l}\binom{b-l}{c-l} = \binom{a+b}{c} + \sum_{l=0}^{c-1} \binom{a+l}{l}\binom{b-l-1}{c-l-1} = \binom{a+b}{c} + \binom{a+b}{c-1}$$

The last equality holds by induction, since $\sum_{l=0}^{c-1}\binom{a+l}{l}\binom{b-l-1}{c-l-1}$ is exactly the summation we are inductively calculating, but with $b-1$ instead of b and with $c-1$ instead of c. Using Pascal's rule, we conclude that $\sum_{l=0}^{c}\binom{l+r-1}{l}\binom{b-l}{c-l} = \binom{a+b+1}{c}$ □

Proof of Theorem 3

Theorem. *Assume to have a dataset $\mathcal{T}$ of size n. Consider a 0-1 coded rule of the form $R(x_1, \ldots, x_p) = \prod_{j=1}^{p} R_j(x_j)$, with $R_j : \mathbb{R} \to \{0, 1\}$. Then an unbiased estimator of the marginal Shapley value of $F(x) = \beta R(x)$ for the j-th predictor and the datapoint x^* is:*

$$\widehat{\phi}_j(x^*) = \beta \cdot \left(\frac{1}{n(p - q_j(x^*))} \sum_{\substack{t \in \mathcal{T} \ s.t. \\ \Omega(t) \supseteq \Omega(x^*)'}} \frac{R_j(x_j^*) - R_j(t_j)}{\binom{2p-q_j(x^*)-q_j(t)-1}{p-q_j(x^*)}} \right),$$

where $\Omega(t)$ and $\Omega(x^*)$ are sets of predictor indices defined as:

$$\Omega(t) = \{k \in \{1, \ldots, p\} | R_k(t_k) = 1\}, \qquad \Omega(x^*) = \{k \in \{1, \ldots, p\} | R_k(x_k^*) = 1\},$$

and $q_j(t)$ and $q_j(x^*)$ are set sizes defined as:

$$q_j(t) = |\Omega(t) \backslash \{j\}|, \qquad q_j(x^*) = |\Omega(x^*) \backslash \{j\}|.$$

Note that $\Omega(x^*)'$ denotes the complementary subset of $\Omega(x^*)$ with respect to $\{1, \ldots, p\}$.

Proof. By linearity of marginal Shapley values, we can assume $\beta = 1$. For simplicity, let us consider $j = p$ and write $S' := \{1, \ldots, p-1\} \backslash S$, for any subset $S \subseteq \{1, \ldots, p-1\}$. For any subset of active players S, we have:

$$\begin{aligned}
\mathbb{E}[R(x)|x_S = x_S^*] &= \mathbb{E}[\prod_{j=1}^{p} R_j(x_j)|x_S = x_S^*] \\
&= \prod_{j \in S} R_j(x_j^*) \mathbb{E}[\prod_{j \in \{1,\ldots,p\}\backslash S} R_j(x_j)|x_S = x_S^*] \\
&= \prod_{j \in S} R_j(x_j^*) \mathbb{E}[\prod_{j \in \{1,\ldots,p\}\backslash S} R_j(x_j)].
\end{aligned}$$

Analogously, we have:

$$\mathbb{E}[R(x)|x_S = x_S^*, x_p = x_p^*] = \prod_{j \in S \cup \{p\}} R_j(x_j^*) \mathbb{E}[\prod_{j \in S'} R_j(x_j)]$$

We write these products more compactly by grouping up the indices: define $R_S(x_S^*) := \prod_{j \in S} R_j(x_j^*)$ and, consequently, $R_{S'}(x_{S'}) := \prod_{j \in S'} R_j(x_j)$. Then $\phi_p(x^*)$ may be rewritten as:

$$\begin{aligned}
\phi_1(x^*) &= \sum_{S \subseteq \{1,\ldots,p-1\}} w(S) \Big(\mathbb{E}[R(x)|x_S = x_S^*, x_p = x_p^*] - \mathbb{E}[R(x)|x_S = x_S^*] \Big) \\
&= \sum_{S \subseteq \{1,\ldots,p-1\}} w(S) \mathbb{E}[R(x)|x_S = x_S^*, x_p = x_p^*] \\
&\qquad - \sum_{S \subseteq \{1,\ldots,p-1\}} w(S) \mathbb{E}[R(x)|x_S = x_S^*] \\
&= \sum_{S \subseteq \{1,\ldots,p-1\}} w(S) R_S(x_S^*) R_p(x_p^*) \mathbb{E}[R_{S'}(x_{S'})] \\
&\quad - \sum_{S \subseteq \{1,\ldots,p-1\}} w(S) R_S(x_S^*) \mathbb{E}[R_{S' \cup \{p\}}(x_{S' \cup \{p\}})].
\end{aligned}$$

Let us treat the two summations separately; define the following:

$$\mathcal{A} = \sum_{S \subseteq \{1,\ldots,p-1\}} w(S) R_S(x_S^*) R_p(x_p^*) \mathbb{E}[R_{S'}(x_{S'})],$$

$$\mathcal{B} = \sum_{S \subseteq \{1,\ldots,p-1\}} w(S) R_S(x_S^*) \mathbb{E}[R_{S' \cup \{p\}}(x_{S' \cup \{p\}})].$$

Note that the expectations that appear in these formulas can be unbiasedly estimated by their sample means over the set $\mathcal{T}$. Let us focus on the first summation $\mathcal{A}$, which is therefore approximated by:

$$\begin{aligned}
\widehat{\mathcal{A}} &= \sum_{S \not\ni p} w(S) R_S(x_S^*) R_p(x_p^*) \widehat{\mathbb{E}}[R_{S'}(x_{S'})] \\
&= \sum_{S \not\ni p} w(S) R_S(x_S^*) R_p(x_p^*) \frac{1}{n} \sum_{t \in \mathcal{T}} R_{S'}(t_{S'}) \\
&= R_p(x_p^*) \cdot \frac{1}{n} \sum_{t \in \mathcal{T}} \sum_{S \not\ni p} w(S) R_S(x_S^*) R_{S'}(t_{S'}).
\end{aligned}$$

Now let us focus on $\mathcal{C} := \sum_{S \not\ni p} w(S) R_S(x_S^*) R_{S'}(t_{S'})$, for a fixed datapoint t. Define

$\Omega_p(t) := \Omega(t)\backslash\{p\}$. By definition of Ω_p, we have:

$$\begin{aligned} R_S(x^*_S)R_{S'}(t_{S'}) \neq 0 &\iff \begin{cases} R_S(x^*_S) = 1 \\ R_{S'}(t_{S'}) = 1 \end{cases} \\ &\iff \begin{cases} S \subseteq \Omega_p(x^*) \\ S' \subseteq \Omega_p(t) \end{cases} \\ &\iff \begin{cases} S \subseteq \Omega_p(x^*) \\ \Omega_p(t)' \subseteq S \end{cases} \quad , \end{aligned}$$

where $\Omega_p(t)'$ is meant as the complementary set with respect to $\{1, \ldots, p-1\}$.
This means that $w(S)R_S(x^*_S)R_{S'}(z_{S'})$ only gives a non-zero contribution for the datapoints t such that $\Omega_p(t)' \subseteq \Omega_p(x^*)$, in which case the only subsets S that contribute are the ones such that $\Omega_p(t)' \subseteq S \subseteq \Omega_p(x^*)$. Since all such sets S contain $\Omega_p(t)'$, they can all be uniquely identified by the indices that they have *besides* those in $\Omega_p(t)'$. In other words, each S can be (uniquely) re-written as $S = \Omega_p(t)' \cup Z$, with $Z \subseteq \Omega_p(x^*)\backslash\Omega_p(t)'$. For every size $|Z| = l$, there are exactly $\binom{|\Omega_p(x^*)\backslash\Omega_p(t)'|}{l} = \binom{q_p(x^*)+q_p(t)-p+1}{l}$ possible choices of Z, and they all have the same contribution $w(S) = \frac{1}{p\binom{p-1}{|S|}} = \frac{1}{p\binom{p-1}{p-1-q_p(t)+l}}$.

This means that we can write:

$$\begin{aligned}
\mathcal{C} &= \sum_{S \not\ni p} w(S) R_S(x^*_S) R_{S'}(t_{S'}) \\
&= \sum_{l=0}^{|\Omega_p(x^*) \setminus \Omega_p(t)|} \frac{\binom{q_p(x^*)+q_p(t)-p+1}{l}}{p\binom{p-1}{l+p-1-q_p(t)}} \\
&= \sum_{l=0}^{q_p(x^*)+q_p(t)-p+1} \frac{\binom{q_p(x^*)+q_p(t)-p+1}{l}}{p\binom{p-1}{l+p-1-q_p(t)}} \\
&= \frac{(q_p(x^*)+q_p(t)-p+1)!}{p!} \\
&\quad \cdot \sum_{l=0}^{q_p(x^*)+q_p(t)-p+1} \frac{(q_p(t)-l)!(p-1-q_p(t)+l)!}{(q_p(x^*)+q_p(t)-p+1-l)!l!} \\
&= \frac{(q_p(x^*)+q_p(t)-p+1)!}{p!} \\
&\quad \cdot \sum_{l=0}^{q_p(x^*)+q_p(t)-p+1} \left[\binom{q_p(t)-l}{q_p(x^*)+q_p(t)-p+1-l} (p-1-q_p(x^*))! \right. \\
&\qquad\qquad \left. \cdot \binom{p-1-q_p(t)+l}{l} (p-1-q_p(t))! \right] \\
&= \frac{(q_p(x^*)+q_p(t)-p+1)!(p-1-q_p(t))!(p-1-q_p(x^*))!}{p!} \\
&\quad \cdot \sum_{l=0}^{q_p(x^*)+q_p(t)-p+1} \binom{q_p(t)-l}{q_p(x^*)+q_p(t)-p+1-l} \binom{p-1-q_p(t)+l}{l}.
\end{aligned}$$

Using Lemma 2 with $a = p-1-q_p(t), b = q_p(t), c = q_p(x^*)+q_p(t)-p+1$, we conclude:

$$\begin{aligned}
\mathcal{C} &= \sum_{S \not\ni p} w(S) R_S(x^*_S) R_{S'}(t_{S'}) \\
&= \frac{(q_p(x^*)+q_p(t)-p+1)!(p-1-q_p(t))!(p-1-q_p(x^*))!}{p!} \\
&\qquad \cdot \binom{p}{q_p(x^*)+q_p(t)-p+1} \\
&= \frac{(p-1-q_p(t))!(p-1-q_p(x^*))!}{(2p-q_p(x^*)-q_p(t)-1)!} \\
&= \frac{1}{(p-q_p(x^*))} \cdot \frac{1}{\binom{2p-q_p(x^*)-q_p(t)-1}{p-q_p(x^*)}}.
\end{aligned}$$

This allows us to conclude that:

$$\widehat{\mathcal{A}} = R_p(x_p^*) \cdot \frac{1}{n(p - q_p(x^*))} \sum_{\substack{t \in \mathcal{T} \text{ s.t.} \\ \Omega_p(t)' \subseteq \Omega_p(x^*)}} \frac{1}{\binom{2p - q_p(x^*) - q_p(t) - 1}{p - q_p(x^*)}}.$$

The sum is only over the datapoints with $\Omega_p(t)' \subseteq \Omega_p(x^*)$, as the other datapoints have a null contribution. In a similar fashion, let us compute $\widehat{\mathcal{B}}$:

$$\begin{aligned}
\widehat{\mathcal{B}} &= \sum_{S \not\ni p} w(S) R_S(x_S^*) \mathbb{E}[R_{S' \cup \{p\}}(x_{S' \cup \{p\}})] \\
&= \sum_{S \not\ni p} w(S) R_S(x_S^*) \frac{1}{n} \sum_{t \in \mathcal{T}} R_{S' \cup \{p\}}(t_{S' \cup \{p\}}) \\
&= \sum_{S \not\ni p} w(S) R_S(x_S^*) \frac{1}{n} \sum_{t \in \mathcal{T}} R_S(x_S^*) R_{S'}(t_{S'}) R_p(t_p) \\
&= \frac{1}{n} \sum_{t \in \mathcal{T}} R_p(t_p) \sum_{S \not\ni p} w(S) R_S(x_S^*) R_{S'}(t_{S'}).
\end{aligned}$$

As we can see, the expression now contains the exact same summation as before, which immediately allows us to conclude that:

$$\widehat{\mathcal{B}} = \frac{1}{n(p - q_p(x^*))} \sum_{\substack{t \in \mathcal{T} \text{ s.t.} \\ \Omega_p(t)' \subseteq \Omega_p(x^*)}} \frac{R_p(t_p)}{\binom{2p - q_p(x^*) - q_p(t) - 1}{p - q_p(x^*)}}.$$

Combining the two expressions together gives us the formula:

$$\widehat{\phi_p}(x^*) = \widehat{\mathcal{A}} - \widehat{\mathcal{B}} = \frac{1}{n(p - q_p(x^*))} \sum_{\substack{t \in \mathcal{T} \text{ s.t.} \\ \Omega_p(t)' \subseteq \Omega_p(x^*)}} \frac{R_p(x_p^*) - R_p(t_p)}{\binom{2p - q_p(x^*) - q_p(t) - 1}{p - q_p(x^*)}}.$$

This expression coincides with the formula except for the summation condition: here, it states $\Omega_p(t)' \subseteq \Omega_p(x^*)$ (or, equivalently, $\Omega_p(t) \supseteq \Omega_p(x^*)'$) instead of $\Omega(t) \supseteq \Omega(x^*)$. However, these two conditions are only different for the points t, x^* such that $R_p(t_p) = R_p(x_p^*) = 0$, for which the contributions would be null anyway. It is therefore possible to just add these points to the summation and obtain the equality presented in the theorem. □

Bibliography

[1] Jerome H Friedman and Bogdan E Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3):916–954, 2008.

[2] Malte Nalenz and Mattias Villani. Tree ensembles with rule structured horseshoe regularization. *The Annals of Applied Statistics*, 12(4):2379–2408, 2018.

[3] Marieke B Snijder, Henrike Galenkamp, Maria Prins, Eske M Derks, Ron JG Peters, Aeilko H Zwinderman, and Karien Stronks. Cohort profile: the healthy life in an urban setting (helius) study in amsterdam, the netherlands. *BMJ open*, 7(12):e017873, 2017.

[4] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE, 2018.

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pages 254–264. Auerbach Publications, 2022.

[6] Zachary C Lipton. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *ACM Queue*, 16(3):31–57, 2018.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[8] Fuyong Zhang, Yi Wang, Shigang Liu, and Hua Wang. Decision-based evasion attacks on tree ensemble classifiers. *World Wide Web*, 23:2957–2977, 2020.

[9] Claudio Busatto and Mark van de Wiel. Informative co-data learning for high-dimensional horseshoe regression. *arXiv preprint arXiv:2303.05898*, 2023.

[10] Robert Tibshirani. The lasso method for variable selection in the cox model. *Statistics in Medicine*, 16(4):385–395, 1997.

[11] Jelle Goeman, Rosa Meijer, and Nimisha Chaturvedi. L1 and l2 penalized regression models. http://cran.r-project.org, 2012.

[12] Ismaël Castillo, Johannes Schmidt-Hieber, and Aad Van der Vaart. Bayesian linear regression with sparse priors. *The Annals of Statistics*, 43(5):1986–2018, 2015.

[13] Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(1):187–193, 2011.

[14] Carlos M. Carvalho, Nicholas G. Polson, and James G. Scott. Handling sparsity via the horseshoe. In David van Dyk and Max Welling, editors, *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 73–80, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR.

[15] Beat Neuenschwander, Satrajit Roychoudhury, and Heinz Schmidli. On the use of co-data in clinical trials. *Statistics in Biopharmaceutical Research*, 8(3):345–354, 2016.

[16] Mark A Van De Wiel, Tonje G Lien, Wina Verlaat, Wessel N van Wieringen, and Saskia M Wilting. Better prediction by use of co-data: adaptive group-regularized ridge regression. *Statistics in Medicine*, 35(3):368–381, 2016.

[17] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(1):49–67, 2006.

[18] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[19] Britta Velten and Wolfgang Huber. Adaptive penalization in high-dimensional regression and classification with external covariates using variational Bayes. *Biostatistics*, 22(2):348–364, 2021.

[20] Magnus M Münch, Carel FW Peeters, Aad W Van Der Vaart, and Mark A Van De Wiel. Adaptive group-regularized logistic elastic net regression. *Biostatistics*, 22(4):723–737, 2021.

[21] Mirrelijn M van Nee, Tim van de Brug, and Mark A van de Wiel. Fast marginal likelihood estimation of penalties for group-adaptive elastic net. *Journal of Computational and Graphical Statistics*, 32(3):1–11, 2022.

[22] Mirrelijn M van Nee, Lodewyk FA Wessels, and Mark A van de Wiel. Flexible co-data learning for high-dimensional prediction. *Statistics in Medicine*, 40(26):5910–5925, 2021.

[23] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[24] Erik Strumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[25] Erik Štrumbelj and Igor Kononenko. Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems*, 41:647–665, 2014.

[26] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 2017.

[27] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to shapley values. *Artificial Intelligence*, 298:103502, 2021.

[28] Mukund Sundararajan and Amir Najmi. The many shapley values for model explanation. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9269–9278. PMLR, 13–18 Jul 2020.

[29] Khashayar Filom, Alexey Miroshnikov, Konstandinos Kotsiopoulos, and Arjun Ravi Kannan. On marginal feature attributions of tree-based models. *arXiv preprint arXiv:2302.08434*, 2023.

[30] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. *arXiv preprint arXiv:1802.03888*, 2018.

[31] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.

[32] Christoph Molnar. *Interpretable Machine Learning*. 2 edition, 2022.

[33] Christopher Frye, Damien de Mijolla, Tom Begley, Laurence Cowton, Megan Stanley, and Ilya Feige. Shapley explainability on the data manifold. *arXiv preprint arXiv:2006.01272*, 2020.

[34] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15(3):651–674, 2006.

[35] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, Yutian Li, and Jiaming Yuan. *xgboost: Extreme Gradient Boosting*, 2023. R package version 1.7.5.1.

[36] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[37] Jerome Friedman, Robert Tibshirani, and Trevor Hastie. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[38] Stephanie van der Pas, James Scott, Antik Chakraborty, and Anirban Bhattacharya. *horseshoe: Implementation of the Horseshoe Prior*, 2019. R package version 0.2.0.

[39] Malte Nalenz and Mattias Villani. R package for the horserule model. `https://github.com/mattiasvillani/horserule/`, 2017.

[40] Jerome H Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

[41] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[42] Marjolein Fokkema. Fitting prediction rule ensembles with R package pre. *Journal of Statistical Software*, 92:1–30, 2020.

[43] David Dyk and Max Welling, editors. *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*. PMLR.

[44] Hal Daumé III and Aarti Singh, editors. *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*. PMLR.

[45] Claudio Busatto and Mark van de Wiel. infhs. `https://github.com/cbusatto/infHS`, 2023.