Interpretable Random Forests via Rule Extraction

Clément Bénard, Gérard Biau, Sébastien da Veiga, Erwan Scornet

Interpretable Random Forests via Rule Extraction

Bénard, Clément [1 2] **Biau, Gérard** [2] **Da Veiga, Sébastien** [1] **Scornet, Erwan** [3]

Abstract

We introduce SIRUS (**S**table and **I**nterpretable **RU**le **S**et) for regression, a stable rule learning algorithm which takes the form of a short and simple list of rules. State-of-the-art learning algorithms are often referred to as "black boxes" because of the high number of operations involved in their prediction process. Despite their powerful predictivity, this lack of interpretability may be highly restrictive for applications with critical decisions at stake. On the other hand, algorithms with a simple structure—typically decision trees, rule algorithms, or sparse linear models—are well known for their instability. This undesirable feature makes the conclusions of the data analysis unreliable and turns out to be a strong operational limitation. This motivates the design of SIRUS, which combines a simple structure with a remarkable stable behavior when data is perturbed. The algorithm is based on random forests, the predictive accuracy of which is preserved. We demonstrate the efficiency of the method both empirically (through experiments) and theoretically (with the proof of its asymptotic stability). Our `R/C++` software implementation `sirus` is available from `CRAN`.

1. Introduction

State-of-the-art learning algorithms, such as random forests or neural networks, are often criticized for their "black-box" nature. This criticism essentially results from the high number of operations involved in their prediction mechanism, as it prevents to grasp how inputs are combined to generate predictions. Interpretability of machine learning algorithms is receiving an increasing amount of attention since the lack of transparency is a strong limitation for many applications, in particular those involving critical decisions. The analysis of production processes in the manufacturing industry typically falls into this category. Indeed, such processes involve complex physical and chemical phenomena that can often be successfully modeled by black-box learning algorithms. However, any modification of a production process has deep and long-term consequences, and therefore cannot simply result from a blind stochastic modelling. In this domain, algorithms have to be interpretable, i.e., provide a sound understanding of the relation between inputs and outputs, in order to leverage insights to guide physical analysis and improve efficiency of the production.

Although there is no agreement in the machine learning litterature about a precise definition of interpretability (Lipton, 2016; Murdoch et al., 2019), it is yet possible to define simplicity, stability, and predictivity as minimum requirements for interpretable models (Bénard et al., 2019; Yu & Kumbier, 2019). Simplicity of the model structure can be assessed by the number of operations performed in the prediction mechanism. In particular, Murdoch et al. (2019) introduce the notion of *simulatable models* when a human is able to reproduce the prediction process by hand. Secondly, Yu (2013) argues that "interpretability needs stability", as the conclusions of a statistical analysis have to be robust to small data perturbations to be meaningful. Instability is the symptom of a partial and arbitrary modelling of the data, also known as the *Rashomon effect* (Breiman, 2001b). Finally, as also explained in Breiman (2001b), if the decrease of predictive accuracy is significant compared to a state-of-the-art black-box algorithm, the interpretable model misses some patterns in the data and is therefore misleading.

Decision trees (Breiman et al., 1984) can model nonlinear patterns while having a simple structure. They are therefore often presented as interpretable. However, the structure of trees is highly sensitive to small data perturbation (Breiman, 2001b), which violates the stability principle and is thus a strong limitation to their practical use. Rule algorithms are another type of nonlinear methods with a simple structure, defined as a collection of elementary rules. An elementary rule is a set of constraints on input variables, which forms a hyperrectangle in the input space and on which the associated prediction is constant. As an example, such a rule typically takes the following simple form:

$$\textbf{If } \begin{cases} X^{(1)} < 1.12 \\ \textbf{\&}\ X^{(3)} \geq 0.74 \end{cases} \textbf{ then } \hat{Y} = 0.18 \textbf{ else } \hat{Y} = 4.1 \textbf{ .}$$

[1]Safran Tech [2]Sorbonne Université [3]Ecole Polytechnique. Correspondence to: Bénard, Clément <clement.benard@safrangroup.com>.

A large number of rule algorithms have been developed, among which the most influential Decision List (Rivest, 1987), CN2 (Clark & Niblett, 1989), C4.5 (Quinlan, 1992), IREP (Incremental Reduced Error Pruning, Fürnkranz & Widmer, 1994), RIPPER (Repeated Incremental Pruning to Produce Error Reduction, Cohen, 1995), PART (Partial Decision Trees, Frank & Witten, 1998), SLIPPER (Simple Learner with Iterative Pruning to Produce Error Reduction, Cohen & Singer, 1999), LRI (Leightweight Rule Induction, Weiss & Indurkhya, 2000), RuleFit (Friedman & Popescu, 2008), Node harvest (Meinshausen, 2010), ENDER (Ensemble of Decision Rules, Dembczyński et al., 2010), BRL (Bayesian Rule Lists, Letham et al., 2015), RIPE (Rule Induction Partitioning Estimator, Margot et al., 2018; 2019), and Wei et al. (2019, Generalized Linear Rule Models). It turns out, however, that among the hundreds of existing rule algorithms, most of them are designed for supervised classification and very few have the ability to handle regression problems, with the notable exception of RuleFit and Node harvest. Yet, despite their powerful predictive skills, these two methods tend to produce long, complex, and unstable list of rules (typically of the order of 50), which makes their interpretability questionable.

The purpose of this article is to propose a new stable rule algorithm for regression, SIRUS (**S**table and **I**nterpretable **RU**le **S**et), and therefore demonstrate that rule methods can address regression problems efficiently while producing compact and stable list of rules. To this aim, we build on Bénard et al. (2019), who have introduced SIRUS for classification problems. Our algorithm is based on random forests (Breiman, 2001a), and its general principle is as follows: since each node of each tree of a random forest can be turned into an elementary rule, the core idea is to extract rules from a tree ensemble based on their frequency of appearance. The most frequent rules, which represent robust and strong patterns in the data, are ultimately linearly combined to form predictions. Just as a preliminary illustration, Table 1 shows that SIRUS outperforms its main competitors, Rulefit and Node harvest, in terms of stability, with a comparable predictive accuracy as we will see. The stability measure is designed using a 10-fold cross-validation to simulate data perturbation (the metric is the average proportion of rules shared by two models of two distinct folds of the cross-validation). Besides, our algorithm inherits the computational complexity of random forests.

We present SIRUS algorithm in Section 2. In Section 3, experiments illustrate the good performance of our algorithm in various settings. Section 4 is devoted to studying the theoretical properties of the method, with, in particular, a proof of its asymptotic stability. Finally, Section 5 summarizes the main results and discusses research directions for future work. Additional details are gathered in the Supplementary Material.

Dataset	RuleFit	Node Harvest	SIRUS
Ozone	0.22	0.30	**0.62**
Mpg	0.25	0.43	**0.83**
Prostate	0.32	0.23	**0.48**
Housing	0.19	0.40	**0.80**
Diabetes	0.18	0.39	**0.66**
Machine	0.23	0.29	**0.88**
Galaxy	0.40	0.39	**0.77**
Abalone	0.31	0.38	**0.82**
Bones	0.59	0.52	**0.89**

Table 1. Mean stability over a 10-fold cross-validation for various public datasets.

2. SIRUS

We consider a standard regression setting where we observe an i.i.d. sample $\mathscr{D}_n = \{(\mathbf{X}_i, Y_i), i = 1, \dots, n\}$, with each $(\mathbf{X}_i, Y_i)$ distributed as a generic pair $(\mathbf{X}, Y)$ independent of $\mathscr{D}_n$. The p-tuple $\mathbf{X} = (X^{(1)}, \dots, X^{(p)})$ is a random vector taking values in $\mathbb{R}^p$, and $Y \in \mathbb{R}$ is the response. Our objective is to estimate the regression function $m(\mathbf{x}) = \mathbb{E}(Y|\mathbf{X} = \mathbf{x})$ with a small and stable set of rules.

2.1. Algorithm

Rule generation The **first step** of SIRUS is to grow a random forest with a large number M of trees based on the available sample $\mathscr{D}_n$. Two critical features of our approach to stabilize the forest structure are (i) to restrict node splits to the q-empirical quantiles of the marginals $X^{(1)}, \dots, X^{(p)}$ (with, typically, $q = 10$), and (ii) to grow shallow trees of depth 2. As the experiments will show, these modifications to Breiman's original algorithm are harmless for predictive accuracy. Next, the obtained forest is broken down in a large collection of rules in the following process. First, observe that each node of each tree of the resulting ensemble defines a hyperrectangle in the input space $\mathbb{R}^p$. Such a node can therefore be turned into an elementary regression rule, by defining a piecewise constant estimate whose value only depends on whether the query point falls in the hyperrectangle or not. Formally, a tree node is represented by a path, say $\mathscr{P}$, which describes how to reach the node from the root of the tree. In the sequel, we denote by Π the finite list of all possible paths, and insist that each path $\mathscr{P} \in \Pi$ defines a regression rule. Based on this principle, in the first step of the algorithm, both internal and external nodes are extracted from the trees of depth 2 of the random forest to generate a large collection of rules, typically 10^4.

Rule selection The **second step** of SIRUS is to select the relevant rules from this large collection. Despite the tree randomization in the forest construction, there are some redundancy in the extracted rules. Indeed those with a

high frequency of appearance represent strong and robust patterns in the data, and are therefore good candidates to be included in a compact, stable, and predictive rule ensemble. This occurrence frequency is denoted by $\hat{p}_{M,n}(\mathscr{P})$ for each possible path $\mathscr{P} \in \Pi$. Then a threshold $p_0 \in (0,1)$ is simply used to select the relevant rules, that is

$$\hat{\mathscr{P}}_{M,n,p_0} = \{\mathscr{P} \in \Pi : \hat{p}_{M,n}(\mathscr{P}) > p_0\}.$$

The threshold p_0 is a tuning parameter, whose influence is illustrated later in the experiments in Figures 2 and 3. In a word, SIRUS uses the principle of randomized bagging, but aggregates the forest structure itself instead of predictions in order to stabilize the rule selection.

Rule set post-treatment The rules associated with the set of distinct paths $\hat{\mathscr{P}}_{M,n,p_0}$ are dependent by definition of the path extraction mechanism. In particular, let us consider the 6 rules extracted from a random tree of depth 2. Since the tree structure is recursive, 2 rules are made of one split and 4 rules of two splits. Those 6 rules are linearly dependent because their associated hyperrectangles overlap. Consequently, to properly settle a linear aggregation of the rules, the **third step** of SIRUS filters $\hat{\mathscr{P}}_{M,n,p_0}$ with the following post-treatment procedure: if the rule induced by the path $\mathscr{P} \in \hat{\mathscr{P}}_{M,n,p_0}$ is a linear combination of rules associated with paths with a higher frequency of appearance, then $\mathscr{P}$ is simply removed from $\hat{\mathscr{P}}_{M,n,p_0}$.

Rule aggregation By following the previous steps, we finally obtain a small set of regression rules. As such, a rule $\hat{g}_{n,\mathscr{P}}$ associated with a path $\mathscr{P}$ is a piecewise constant estimate: if a query point $\mathbf{x}$ falls into the corresponding hyperrectangle $H_{\mathscr{P}} \subset \mathbb{R}^p$, the rule returns the average of the Y_i's for the training points $\mathbf{X}_i$'s that belong to $H_{\mathscr{P}}$; symmetrically, if $\mathbf{x}$ falls outside of $H_{\mathscr{P}}$, the average of the Y_i's for training points outside of $H_{\mathscr{P}}$ is returned. A non-negative weight is assigned to each of the selected rule, in order to combine them into a single estimate of $m(\mathbf{x})$. These weights are defined as the ridge regression solution, where each predictor is a rule $\hat{g}_{n,\mathscr{P}}$ for $\mathscr{P} \in \hat{\mathscr{P}}_{M,n,p_0}$ and weights are constrained to be non-negative. Thus, the aggregated estimate $\hat{m}_{M,n,p_0}(\mathbf{x})$ of $m(\mathbf{x})$ computed in the **fourth step** of SIRUS has the form

$$\hat{m}_{M,n,p_0}(\mathbf{x}) = \hat{\beta}_0 + \sum_{\mathscr{P} \in \hat{\mathscr{P}}_{M,n,p_0}} \hat{\beta}_{n,\mathscr{P}} \hat{g}_{n,\mathscr{P}}(\mathbf{x}), \qquad (2.1)$$

where $\hat{\beta}_0$ and $\hat{\beta}_{n,\mathscr{P}}$ are the solutions of the ridge regression problem. More precisely, denoting by $\hat{\boldsymbol{\beta}}_{n,p_0}$ the column vector whose components are the coefficients $\hat{\beta}_{n,\mathscr{P}}$ for $\mathscr{P} \in \hat{\mathscr{P}}_{M,n,p_0}$, and letting $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ and $\boldsymbol{\Gamma}_{n,p_0}$ the matrix whose rows are the rule values $\hat{g}_{n,\mathscr{P}}(\mathbf{X}_i)$ for $i \in \{1, \ldots, n\}$, we have

$$(\hat{\boldsymbol{\beta}}_{n,p_0}, \hat{\beta}_0) = \underset{\boldsymbol{\beta} \geq 0, \beta_0}{\operatorname{argmin}} \frac{1}{n} ||\mathbf{Y} - \beta_0 \mathbf{1}_{\boldsymbol{n}} - \boldsymbol{\Gamma}_{n,p_0} \boldsymbol{\beta}||_2^2 + \lambda ||\boldsymbol{\beta}||_2^2,$$

where $\mathbf{1}_{\boldsymbol{n}} = (1, \ldots, 1)^T$ is the n-vector with all components equal to 1, and λ is a positive parameter tuned by cross-validation that controls the penalization severity. The mininum is taken over $\beta_0 \in \mathbb{R}$ and all the vectors $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_{c_n}\} \in \mathbb{R}_+^{c_n}$ where $c_n = |\hat{\mathscr{P}}_{M,n,p_0}|$ is the number of selected rules.

Since rules are correlated by construction, a simple optimal least square solution leads to negative values for some of the coefficients $\hat{\beta}_{n,\mathscr{P}}$. Such behavior drastically undermines the interpretability of the algorithm. This is why, as in Node harvest (Meinshausen, 2010), the constraint $\boldsymbol{\beta} \geq 0$ is added to ensure that all coefficients are non-negative. In this correlated setting, the constraint $\boldsymbol{\beta} \geq 0$ enforces sparsity and then instability. For this reason a ridge penalty is added to the loss function to stabilize the estimate $\hat{\boldsymbol{\beta}}_{n,p_0}$ and mitigate sparsity.

2.2. Interpretability

As stated in the introduction, despite the lack of a precise definition of interpretable models, there are three minimum requirements to be taken into account: simplicity, stability, and predictivity. These notions need to be formally defined and quantified to enable comparison between algorithms.

Simplicity Simplicity refers to the model complexity, in particular the number of operations involved in the prediction mechanism. In the case of rule algorithms, a measure of simplicity is naturally given by the number of rules.

Stability Intuitively, a rule algorithm is stable when two independent estimations based on two independent samples return similar lists of rules. Formally, let $\hat{\mathscr{P}}'_{M,n,p_0}$ be the list of rules output by SIRUS fit on an independent sample $\mathscr{D}'_n$. Then the proportion of rules shared by $\hat{\mathscr{P}}_{M,n,p_0}$ and $\hat{\mathscr{P}}'_{M,n,p_0}$ gives a stability measure. Such a metric is known as the Dice-Sorensen index, and is often used to assess variable selection procedures (Chao et al., 2006; Zucknick et al., 2008; Boulesteix & Slawski, 2009; He & Yu, 2010; Alelyani et al., 2011). In our case, the Dice-Sorensen index is then defined as

$$\hat{S}_{M,n,p_0} = \frac{2\left|\hat{\mathscr{P}}_{M,n,p_0} \cap \hat{\mathscr{P}}'_{M,n,p_0}\right|}{\left|\hat{\mathscr{P}}_{M,n,p_0}\right| + \left|\hat{\mathscr{P}}'_{M,n,p_0}\right|}.$$

However, in practice one rarely has access to an additional sample $\mathscr{D}'_n$. Therefore, to circumvent this problem, we use a 10-fold cross-validation to simulate data perturbation. The

stability metric is thus empirically defined as the average proportion of rules shared by two models of two distinct folds of the cross-validation. A stability of 1 means that the exact same list of rules is selected over the 10 folds, whereas a stability of 0 means that all rules are distinct between any 2 folds.

Predictivity For regression problems, the proportion of unexplained variance is a natural measure of the prediction error. The estimation is performed by 10-fold cross-validation.

3. Experiments

Experiments are run over 9 diverse public datasets to demonstrate the improvement of SIRUS over state-of-the-art methods. Table 1 in Section 2 of the Supplementary Material provides details about the datasets.

SIRUS rule set Our algorithm is illustrated on the "LA Ozone" dataset from Friedman et al. (2001), which records the level of atmospheric ozone concentration from eight daily meteorological measurements made in Los Angeles in 1976: wind speed ("wind"), humidity ("humidity"), temperature ("temp"), inversion base height ("ibh"), daggot pressure gradient ("dpg"), inversion base temperature ("ibt"), visibility ("vis"), and day of the year ("doy"). The response "Ozone" is the log of the daily maximum of ozone concentration.

The list of rules output for this dataset is presented in Table 2. The column "Frequency" refers to $\hat{p}_{M,n}(\mathscr{P})$, the occurrence frequency of each rule in the forest, used for rule selection. It enables to grasp how weather conditions impact the ozone concentration. In particular, a temperature larger than $65°$F or a high inversion base temperature result in high ozone concentrations. The third rule tells us that the interaction of a high temperature with a visibility lower than 150 miles generates even higher levels of ozone concentration. Interestingly, according to the ninth rule, especially low ozone concentrations are reached when a low temperature and and a low inversion base temperature are combined.

Recall that to generate a prediction for a given query point $\mathbf{x}$, for each rule the corresponding ozone concentration is retrieved depending on whether $\mathbf{x}$ satisfies the rule conditions. Then all rule outputs for $\mathbf{x}$ are multiplied by their associated weight and added together. One can observe that rule importances and weights are not related. For example, the third rule has a higher weight than the most two important ones. It is clear that rule 3 has multiple constraints and is therefore more sensitive to data perturbation—hence a smaller frequency of appearance in the forest. On the other hand, its associated variance decrease in CART is more important than for the first two rules, leading to a higher weight in the linear combination. Since rules 5 and 6 are strongly correlated, their weights are diluted.

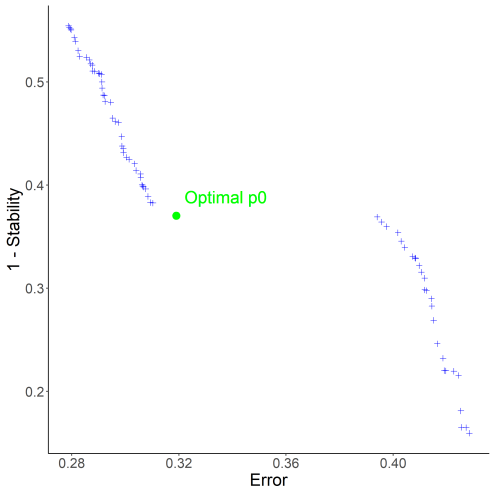

Figure 1. Pareto front of stability versus error (unexplained variance) when p_0 varies, with the optimal value in green for the "Ozone" dataset. The optimal point is the closest one to the ideal point $(0, 0.1)$ of 0 unexplained variance and 90% stability.

Tuning SIRUS has two hyperparameters: the number of trees M and the threshold p_0 to select the most frequent rules in the forest. Clearly, the stability, predictivity, and computation time increase with the number of trees. Thus a stopping criterion is designed to grow the minimum number of trees that ensure stability and predictivity to be close to their maximum—see Section 3 of the Supplementary Material for a detailed definition of this criterion. On the other hand, p_0 is tuned by cross-validation to maximize both stability and predictivity. To find a tradeoff between these two properties, we follow a standard bi-objective optimization procedure illustrated in Figure 1: the optimal value of p_0 is defined as the p_0 associated to stability and predictivity values that minimize the Euclidean distance to the ideal point of 0 unexplained variance and 90% stability. This ideal point is chosen for its empirical efficiency: stability never reaches values higher than 90%, whereas unexplained variance can be arbitrarily close to 0, depending on the data. Additionally, we consider that long lists of rules are not interpretable and set an arbitrary limit of 25 rules. Besides, we set $\texttt{mtry} = \lfloor \frac{p}{3} \rfloor$, $q = 10$ quantiles, and transform categorical variables into multiple binary variables.

Performance We compare SIRUS with its two main competitors RuleFit (with rule predictors only) and Node harvest. Both methods also extract large collection of rules from tree ensembles: RuleFit uses Importance Sampled Learning Ensemble (ISLE, Friedman & Popescu, 2003) whereas

Average Ozone $= 12$				**Intercept** $= -7.8$			
Frequency		**Rule**					**Weight**
0.29	**if**	temp < 65	**then**	Ozone $= 7$	**else**	Ozone $= 19$	0.12
0.17	**if**	ibt < 189	**then**	Ozone $= 7$	**else**	Ozone $= 18$	0.07
0.063	**if**	temp ≥ 65 **&** vis < 150	**then**	Ozone $= 20$	**else**	Ozone $= 7$	0.31
0.061	**if**	vh < 5840	**then**	Ozone $= 10$	**else**	Ozone $= 20$	0.072
0.060	**if**	ibh < 2110	**then**	Ozone $= 16$	**else**	Ozone $= 7$	0.14
0.058	**if**	ibh < 2960	**then**	Ozone $= 15$	**else**	Ozone $= 6$	0.10
0.051	**if**	temp ≥ 65 **&** ibh < 2110	**then**	Ozone $= 21$	**else**	Ozone $= 8$	0.16
0.048	**if**	vis < 150	**then**	Ozone $= 14$	**else**	Ozone $= 7$	0.18
0.043	**if**	temp < 65 **&** ibt < 120	**then**	Ozone $= 5$	**else**	Ozone $= 15$	0.15
0.040	**if**	temp < 70	**then**	Ozone $= 8$	**else**	Ozone $= 20$	0.14
0.039	**if**	ibt < 227	**then**	Ozone $= 9$	**else**	Ozone $= 22$	0.21

Table 2. SIRUS rule list for the "LA Ozone" dataset.

Node harvest is based on random forests. Rule selection is performed by a sparse linear aggregation, respectively the Lasso (Tibshirani, 1996) for Rulefit and a constrained quadratic program for Node harvest. For predictive accuracy, we ran random forests and (pruned) CART to provide the baseline. To enable stability comparison, data is binned using 10 quantiles to fit Rulefit and Node harvest. Our `R`/`C++` package `sirus` is adapted from `ranger`, a fast random forests implementation (Wright & Ziegler, 2017). We also use available `R` implementations `pre` (Fokkema, 2017, RuleFit) and `nodeharvest` (Meinshausen, 2015).

While the predictive accuracy of SIRUS is comparable to Node harvest and slightly below RuleFit, the stability is considerably improved with much smaller rule lists. Experimental results are gathered in Table 3 for model sizes, Table 1 for stability, and Table 4 for predictive accuracy. All results are averaged over 10 repetitions of the cross-validation procedure. Since standard deviations are negligible, they are not displayed to increase readability.

To illustrate the typical behavior of our method, we comment the results for two specific datasets: "Diabetes" (Efron et al., 2004) and "Machine" (Dua & Graff, 2017). The "Diabetes" data contains $n = 442$ diabetic patients and the response of interest Y is a measure of disease progression over one year. A total of 10 variables are collected for each patient: age, sex, body mass index, average blood pressure, and six blood serum measurements $s1, s2, \ldots, s6$. For this dataset, SIRUS is as predictive as a random forest, with only 12 rules when the forest performs about 10^4 operations: the unexplained variance is 0.56 for SIRUS and 0.55 for random forest. Notice that CART performs considerably worse

Dataset	**RuleFit**	**Node Harvest**	**SIRUS**
Ozone	21	46	**11**
Mpg	40	43	**9**
Prostate	**14**	41	23
Housing	54	40	**6**
Diabetes	25	42	**12**
Machine	44	42	**9**
Galaxy	34	36	**4**
Abalone	58	35	**6**
Bones	5	13	**1**

Table 3. Mean model size over a 10-fold cross-validation for various public datasets.

with 0.67 unexplained variance. For the second dataset, "Machine", the output Y of interest is the CPU performance of computer hardware. For $n = 209$ machines, 7 variables are collected about the machine characteristics. For this dataset, SIRUS, RuleFit and Node harvest have a similar predictivity, in-between CART and random forests. Our algorithm achieves such performance with a readable list of only 9 rules stable at 88%, while RuleFit and Node harvest incorporate respectively 44 and 42 rules with stability levels of 23% and 29%. Stability and predictivity are represented as p_0 varies for "Diabetes" and "Machine" datasets in Figures 2 and 3, respectively.

Remark 1. *In the above experiments, the parameter p_0 controlling the number of rules in the final model is tuned to achieve a tradeoff between stability and accuracy, which often leads to small rule lists, typically of the order of* 10*. If we drop the tuning of p_0 and set its value to extract, say,* 100

Dataset	Random Forest	CART	RuleFit	Node Harvest	SIRUS	SIRUS 100 Rules	SIRUS d=3
Ozone	0.25	0.36	**0.27**	0.31	0.32	**0.26**	**0.28**
Mpg	0.13	0.20	**0.15**	0.20	0.21	**0.15**	**0.14**
Prostate	0.48	0.60	**0.53**	**0.52**	**0.48**	0.55	0.59
Housing	0.12	0.28	**0.16**	0.24	0.31	0.21	0.20
Diabetes	0.55	0.67	**0.55**	**0.58**	**0.56**	**0.54**	**0.55**
Machine	0.12	0.39	**0.26**	**0.29**	**0.29**	**0.27**	**0.26**
Galaxy	0.027	0.089	**0.031**	0.066	0.20	0.042	0.040
Abalone	0.44	0.56	**0.46**	0.61	0.66	0.64	0.63
Bones	0.64	0.67	**0.70**	**0.70**	**0.74**	**0.72**	**0.71**

Table 4. Proportion of unexplained variance estimated over a 10-fold cross-validation for various public datasets. For rule algorithms only, i.e., RuleFit, Node harvest, and SIRUS, maximum values are displayed in bold, as well as values within 10% of the maximum for each dataset.

*rules from the forest, SIRUS predictivity overcomes Node harvest and is similar to RuleFit as shown in the column "**SIRUS 100 Rules**" of Table 4. On the other hand, stability drops to around* 50% *(*$70-80\%$ *when* p_0 *is tuned). Also notice that the final number of rules is around* 50 *because of the additional selection performed in the final rule aggregation: the model size is then comparable to Rulefit and Node harvest. We also observe that increasing the depth of trees to* 3 *(with* 100 *extracted rules) does not significantly improve predictivity—see the last column "**SIRUS d=3**" of Table 4.*

4. Theoretical Analysis

Among the three minimum requirements for interpretable models, stability is the critical one. In SIRUS, simplicity is explicitly controlled by the hyperparameter p_0 and the limit of 25 rules. The wide literature on rule learning provides many experiments to show that rule algorithms have an accuracy comparable to tree ensembles. On the other hand, designing a stable rule procedure is more challenging (Letham et al., 2015; Murdoch et al., 2019). For this reason, we therefore focus our theoretical analysis on the asymptotic stability of SIRUS.

To get started, we need a rigorous definition of the rule extraction procedure. To this aim, we introduce a symbolic representation of a path in a tree, which describes how to reach a given node from the root. We insist that such path encoding can be used in both the empirical and theoretical algorithms to define rules. A path $\mathscr{P}$ is defined as

$$\mathscr{P} = \{(j_k, r_k, s_k),\, k = 1, \dots, d\},$$

where, for $k \in \{1, \dots, d\}$ ($d \in \{1, 2\}$), the triplet (j_k, r_k, s_k) describes how to move from level $(k-1)$ to level k, with a split using the coordinate $j_k \in \{1, \dots, p\}$, the index $r_k \in \{1, \dots, q-1\}$ of the corresponding quantile, and a side $s_k = L$ if we go to the left and $s_k = R$ if we go to the right. The set of all possible such paths is denoted by Π. An example is given in Figure 4. Each tree of the forest is randomized in two ways: (i) the sample $\mathscr{D}_n$ is bootstrapped prior to the construction of the tree, and (ii) a subset of coordinates is randomly selected to find the best split at each node. This randomization mechanism is governed by a random variable that we call Θ. We define $T(\Theta, \mathscr{D}_n)$, a random subset of Π, as the collection of the 6 extracted paths from the random tree of depth $d = 2$ built with Θ and $\mathscr{D}_n$. Now, let $\Theta_1, \dots, \Theta_\ell, \dots, \Theta_M$ be the independent randomizations of the M trees of the forest. With this notation, the empirical frequency of occurrence of a path $\mathscr{P} \in \Pi$ in the forest takes the form

$$\hat{p}_{M,n}(\mathscr{P}) = \frac{1}{M} \sum_{\ell=1}^{M} \mathbb{1}_{\mathscr{P} \in T(\Theta_\ell, \mathscr{D}_n)},$$

which is simply the proportion of trees that contain $\mathscr{P}$. By definition, $\hat{p}_{M,n}(\mathscr{P})$ is the Monte Carlo estimate of the probability $p_n(\mathscr{P})$ that a Θ-random tree contains a particular path $\mathscr{P} \in \Pi$, that is,

$$p_n(\mathscr{P}) = \mathbb{P}(\mathscr{P} \in T(\Theta, \mathscr{D}_n) | \mathscr{D}_n).$$

Next, we introduce all theoretical counterparts of the empirical quantities involved in SIRUS, which do not depend on the sample $\mathscr{D}_n$ but only on the unknown distribution of $(\mathbf{X}, Y)$. We let $T^\star(\Theta)$ be the list of all 6 paths contained in the theoretical tree built with randomness Θ, in which splits are chosen to maximize the theoretical CART-splitting criterion instead of the empirical one. The probability $p^\star(\mathscr{P})$ that a given path $\mathscr{P}$ belongs to a theoretical randomized tree (the theoretical counterpart of $p_n(\mathscr{P})$) is

$$p^\star(\mathscr{P}) = \mathbb{P}(\mathscr{P} \in T^\star(\Theta)).$$

We finally define the theoretical set of selected paths $\mathscr{P}^\star_{p_0} = \{\mathscr{P} \in \Pi : p^\star(\mathscr{P}) > p_0\}$ (with the same post-treatment as

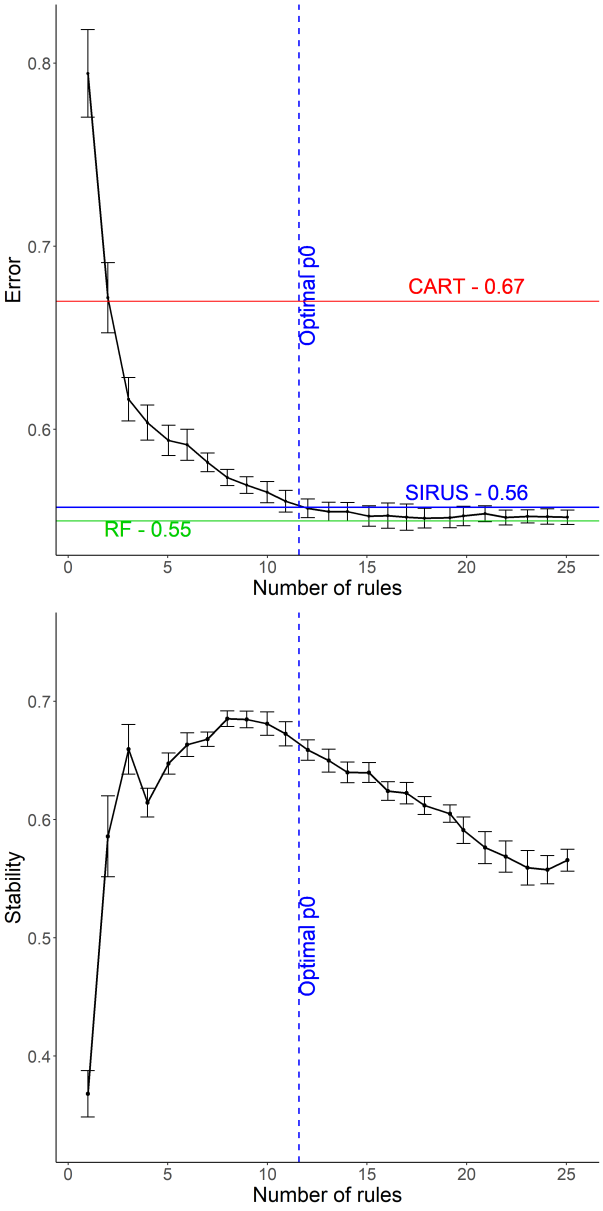

Figure 2. For the dataset "Diabetes", unexplained variance (top panel) and stability (bottom panel) versus the number of rules when p_0 varies, estimated via 10-fold cross-validation (results are averaged over 10 repetitions).

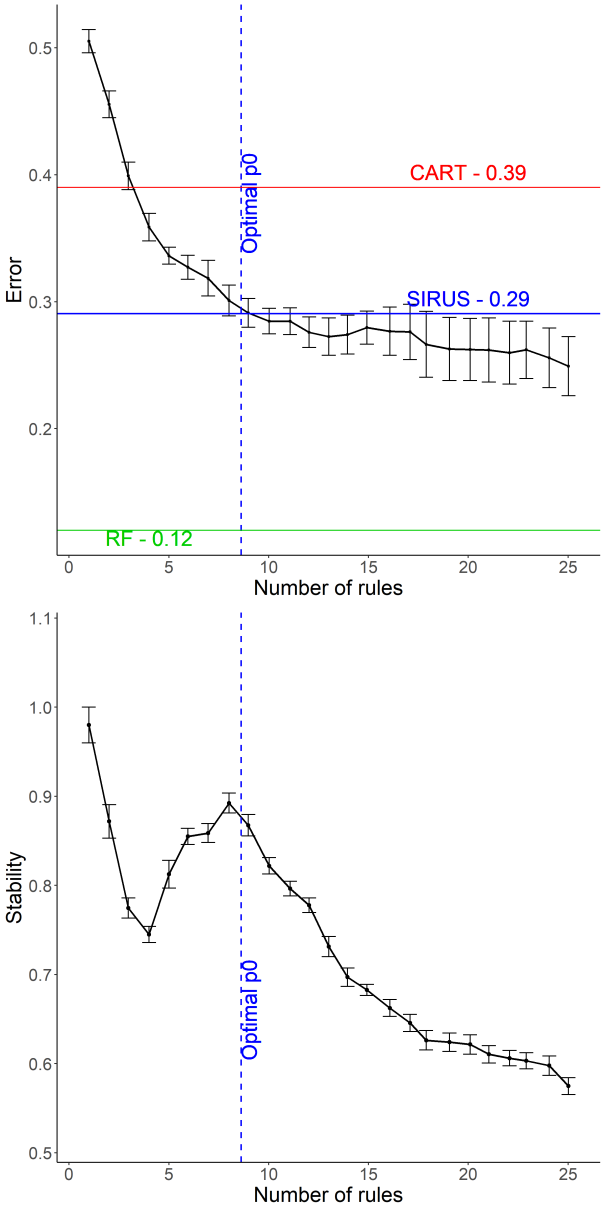

Figure 3. For the dataset "Machine", unexplained variance (top panel) and stability (bottom panel) versus the number of rules when p_0 varies, estimated via 10-fold cross-validation (results are averaged over 10 repetitions).

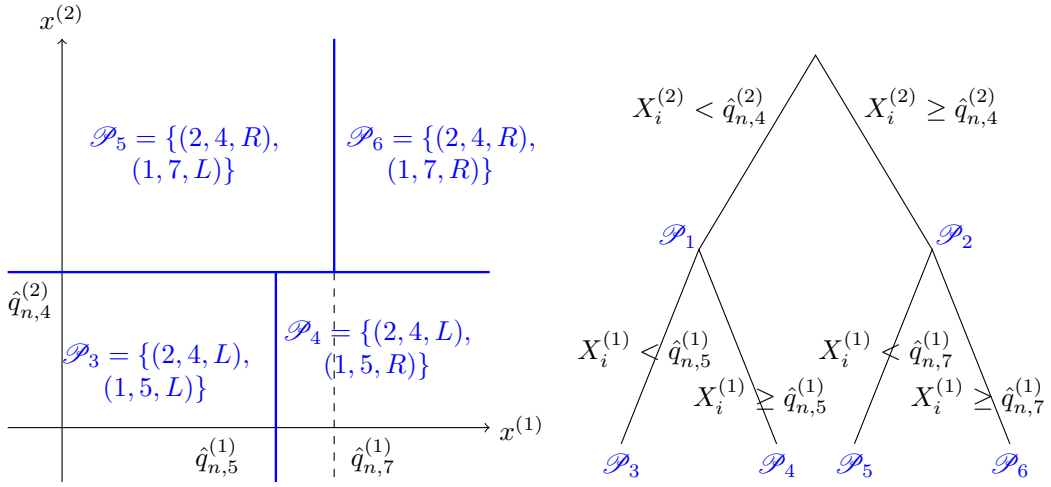

Figure 4. Example of a root node $\mathbb{R}^2$ partitioned by a randomized tree of depth 2: the tree on the right, the associated paths and hyperrectangles of length $d = 2$ on the left.

for the data-based procedure—see Section 2—to remove linear dependence between rules, and keeping only paths with a strictly positive coefficient in the final linear aggregation of the rules).

As it is often the case in the theoretical analysis of random forests, e.g. Scornet et al. (2015); Mentch & Hooker (2016), we assume throughout this section that the subsampling of a_n observations prior to each tree construction is done without replacement to alleviate the mathematical analysis. Our stability result holds under the following mild assumptions:

(A1) The subsampling rate a_n satisfies $\lim_{n\to\infty} a_n = \infty$ and $\lim_{n\to\infty} \frac{a_n}{n} = 0$.

(A2) The number of trees M_n satisfies $\lim_{n\to\infty} M_n = \infty$.

(A3) The random variable $\mathbf{X}$ has a strictly positive density f with respect to the Lebesgue measure on $\mathbb{R}^p$. Furthermore, for all $j \in \{1, \ldots, p\}$, the marginal density $f^{(j)}$ of $X^{(j)}$ is continuous, bounded, and strictly positive. Finally, the random variable Y is bounded.

We recall that stability is assessed by the Dice-Sorensen index as

$$\hat{S}_{M,n,p_0} = \frac{2\left|\hat{\mathscr{P}}_{M,n,p_0} \cap \hat{\mathscr{P}}'_{M,n,p_0}\right|}{\left|\hat{\mathscr{P}}_{M,n,p_0}\right| + \left|\hat{\mathscr{P}}'_{M,n,p_0}\right|},$$

where $\hat{\mathscr{P}}'_{M,n,p_0}$ stands for the list of rules output by SIRUS fit with an independent sample $\mathscr{D}'_n$ and where the random forest is parameterized by independent copies $\Theta'_1, \ldots, \Theta'_M$. The following theorem states that SIRUS is asymptotically stable, i.e., provided that the sample size is large enough, the same list of rules is systematically output across several fit on independent samples. The proof can be found in Section 1 of the Supplementary Material.

Theorem 1. *Assume that Assumptions (A1)-(A3) are satisfied, and let* $\mathcal{U}^\star = \{p^\star(\mathscr{P}) : \mathscr{P} \in \Pi\}$ *be the set of all theoretical probabilities of appearance for each path* $\mathscr{P}$. *Then, provided* $p_0 \in [0,1] \setminus \mathcal{U}^\star$ *and* $\lambda > 0$, *we have*

$$\lim_{n\to\infty} \hat{S}_{M_n,n,p_0} = 1 \quad \textit{in probability}.$$

5. Conclusion

Interpretability of machine learning algorithms is required whenever the targeted applications involve critical decisions. This is in particular the case for the analysis of production processes in the manufacturing industry. Although interpretability does not have a precise definition, we argued that simplicity, stability, and predictivity are minimum requirements for interpretable models. In this context, rule algorithms are well known for their good predictivity and simple structures. On the other hand these methods are also often highly unstable and almost exclusively dedicated to supervised classification.

Therefore we proposed a new regression rule algorithm called SIRUS (**S**table and **I**nterpretable **RU**le **S**et), whose general principle is to extract rules from random forests. Our algorithm exhibits an accuracy comparable to state-of-the-art rule algorithms, in particular RuleFit and Node harvest, while producing much more stable and shorter list of rules. This remarkably stable behavior is theoretically understood since the rule selection is consistent. A `R/C++` software `sirus` is available as an `R` package from `CRAN`.

References

Alelyani, S., Zhao, Z., and Liu, H. A dilemma in assessing stability of feature selection algorithms. In *13th IEEE International Conference on High Performance Computing & Communication*, pp. 701–707, Piscataway, 2011. IEEE.

Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. SIRUS: Making random forests interpretable. *arXiv:1908.06852*, 2019.

Boulesteix, A.-L. and Slawski, M. Stability and aggregation of ranked gene lists. *Briefings in Bioinformatics*, 10: 556–568, 2009.

Breiman, L. Random forests. *Machine Learning*, 45:5–32, 2001a.

Breiman, L. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science*, 16:199–231, 2001b.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. *Classification and Regression Trees*. Chapman & Hall/CRC, Boca Raton, 1984.

Chao, A., Chazdon, R., Colwell, R., and Shen, T.-J. Abundance-based similarity indices and their estimation when there are unseen species in samples. *Biometrics*, 62: 361–371, 2006.

Clark, P. and Niblett, T. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

Cohen, W. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning*, pp. 115–123. Morgan Kaufmann Publishers Inc., San Francisco, 1995.

Cohen, W. and Singer, Y. A simple, fast, and effective rule learner. In *Proceedings of the 16th National Conference on Artificial Intelligence and 11th Conference on Innovative Applications of Artificial Intelligence*, pp. 335–342, Palo Alto, 1999. AAAI Press.

Dembczyński, K., Kotłowski, W., and Słowiński, R. ENDER: A statistical framework for boosting decision rules. *Data Mining and Knowledge Discovery*, 21:52–90, 2010.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of statistics*, 32:407–499, 2004.

Fokkema, M. PRE: An R package for fitting prediction rule ensembles. *arXiv:1707.07149*, 2017.

Frank, E. and Witten, I. H. Generating accurate rule sets without global optimization. In *Proceedings of the 15th International Conference on Machine Learning*, pp. 144–151, San Francisco, 1998. Morgan Kaufmann Publishers Inc.

Friedman, J. and Popescu, B. Importance sampled learning ensembles. *Journal of Machine Learning Research*, 94305:1–32, 2003.

Friedman, J. and Popescu, B. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2:916–954, 2008.

Friedman, J., Hastie, T., and Tibshirani, R. *The Elements of Statistical Learning*, volume 1. Springer, New York, 2001.

Fürnkranz, J. and Widmer, G. Incremental reduced error pruning. In *Proceedings of the 11th International Conference on Machine Learning*, pp. 70–77, San Francisco, 1994. Morgan Kaufmann Publishers Inc.

He, Z. and Yu, W. Stable feature selection for biomarker discovery. *Computational Biology and Chemistry*, 34: 215–225, 2010.

Letham, B., Rudin, C., McCormick, T., and Madigan, D. Interpretable classifiers using rules and Bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9:1350–1371, 2015.

Lipton, Z. The mythos of model interpretability. *arXiv:1606.03490*, 2016.

Margot, V., Baudry, J.-P., Guilloux, F., and Wintenberger, O. Rule induction partitioning estimator, 2018.

Margot, V., Baudry, J.-P., Guilloux, F., and Wintenberger, O. Consistent regression using data-dependent coverings. *arXiv:1907.02306*, 2019.

Meinshausen, N. Node harvest. *The Annals of Applied Statistics*, 4:2049–2072, 2010.

Meinshausen, N. Package nodeharvest, 2015. URL https://cran.r-project.org/web/packages/nodeHarvest/.

Mentch, L. and Hooker, G. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *Journal of Machine Learning Research*, 17:841–881, 2016.

Murdoch, W., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. Interpretable machine learning: Definitions, methods, and applications. *arXiv:1901.04592*, 2019.

Quinlan, J. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, 1992.

Rivest, R. Learning decision lists. *Machine Learning*, 2: 229–246, 1987.

Scornet, E., Biau, G., and Vert, J.-P. Consistency of random forests. *The Annals of Statistics*, 43(4):1716–1741, 2015.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, pp. 267–288, 1996.

Van der Vaart, A. W. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Wei, D., Dash, S., Gao, T., and Günlük, O. Generalized linear rule models. *arXiv preprint arXiv:1906.01761*, 2019.

Weiss, S. and Indurkhya, N. Lightweight rule induction. In *Proceedings of the 17th International Conference on Machine Learning*, pp. 1135–1142, San Francisco, 2000. Morgan Kaufmann Publishers Inc.

Wright, M. N. and Ziegler, A. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software*, 77:1–17, 2017.

Yu, B. Stability. *Bernoulli*, 19:1484–1500, 2013.

Yu, B. and Kumbier, K. Three principles of data science: Predictability, computability, and stability (PCS). *arXiv:1901.08152*, 2019.

Zucknick, M., Richardson, S., and Stronach, E. Comparing the characteristics of gene expression profiles derived by univariate and multivariate classification methods. *Statistical Applications in Genetics and Molecular Biology*, 7: 1–34, 2008.

Supplementary Material For: Interpretable Random Forests via Rule Extraction

1. Proof of Theorem 1

Proof of Theorem 1. We consider $p_0 \in [0,1] \setminus \mathcal{U}^\star$ and $\lambda > 0$. There are two sources of randomness in the estimation of the final set of selected paths: (i) the path extraction from the random forest based on $\hat{p}_{M,n}(\mathscr{P})$ for $\mathscr{P} \in \Pi$, and (ii) the final sparse linear aggregation of the rules through the estimate $\hat{\boldsymbol{\beta}}_{n,p_0}$. To show that the stability converges to 1, these estimates have to converge towards theoretical quantities that are independent of $\mathscr{D}_n$.

Note that, throughout the paper, the final set of selected paths is denoted $\hat{\mathscr{P}}_{M_n,n,p_0}$. Here, for the sake of clarity, $\hat{\mathscr{P}}_{M_n,n,p_0}$ is now the post-treated set of paths extracted from the random forest, and $\hat{\mathscr{P}}_{M_n,n,p_0,\lambda}$ the final set of selected paths in the ridge regression.

(i) **Path extraction** The first step of the proof is to show that the post-treated path extraction from the forest is consistent, i.e., in probability

$$\lim_{n\to\infty} \mathbb{P}(\hat{\mathscr{P}}_{M_n,n,p_0} = \mathscr{P}^\star_{p_0}) = 1. \tag{1.1}$$

Using the continuous mapping theorem, it is easy to see that this result is a consequence of the consistency of $\hat{p}_{M,n}(\mathscr{P})$, i.e.,

$$\lim_{n\to\infty} \hat{p}_{M_n,n}(\mathscr{P}) = p^\star(\mathscr{P}) \quad \text{in probability.}$$

Since the output Y is bounded (by Assumption (A3)), the consistency of $\hat{p}_{M,n}(\mathscr{P})$ can be easily adapted from Theorem 1 of Bénard et al. (2019).

Finally, the result still holds for the post-treated rule set because the post-treatment is a deterministic procedure.

(ii) **Sparse linear aggregation** Recall that the estimate $(\hat{\boldsymbol{\beta}}_{n,p_0}, \hat{\beta}_0)$ is defined as

$$(\hat{\boldsymbol{\beta}}_{n,p_0}, \hat{\beta}_0) = \underset{\boldsymbol{\beta}\geq 0, \beta_0}{\operatorname{argmin}}\, \ell_n(\boldsymbol{\beta}, \beta_0), \tag{1.2}$$

where $\ell_n(\boldsymbol{\beta}, \beta_0) = \frac{1}{n}||\mathbf{Y} - \beta_0 \mathbf{1_n} - \mathbf{\Gamma}_{n,p_0}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2$. The dimension of β is stochastic since it is equal to the number of extracted rules. To get rid of this technical issue in the following of the proof, we rewrite $\ell_n(\boldsymbol{\beta}, \beta_0)$ to have $\boldsymbol{\beta}$ a parameter of fixed dimension $|\Pi|$, the total number of possible rules:

$$\ell_n(\boldsymbol{\beta}, \beta_0) = \frac{1}{n}\sum_{i=1}^{n} \big(Y_i - \beta_0 - \sum_{\mathscr{P}\in\Pi} \beta_{\mathscr{P}} g_{n,\mathscr{P}}(\mathbf{X}_i)\mathbb{1}_{\mathscr{P}\in\hat{\mathscr{P}}_{M_n,n,p_0}}\big)^2 + \lambda||\boldsymbol{\beta}||_2^2.$$

By the law of large numbers and the previous result (1.1), we have in probability

$$\begin{aligned}\lim_{n\to\infty} \ell_n(\boldsymbol{\beta}, \beta_0) &= \mathbb{E}\big[\big(Y - \beta_0 - \sum_{\mathscr{P}\in\mathscr{P}^\star_{p_0}} \beta_{\mathscr{P}} g^\star_{\mathscr{P}}(\mathbf{X})\big)^2\big] + \lambda||\boldsymbol{\beta}||_2^2 \\ &\overset{\text{def}}{=} \ell^\star(\boldsymbol{\beta}, \beta_0),\end{aligned} \tag{1.3}$$

where $g^\star_{\mathscr{P}}$ is the theoretical rule based on the path $\mathscr{P}$ and the theoretical quantiles.

Since Y is bounded, it is easy to see that each component of $\hat{\boldsymbol{\beta}}_{n,p_0}$ is bounded from the following inequalities:

$$\begin{aligned}\lambda||\hat{\boldsymbol{\beta}}_{n,p_0}||_2^2 \leq& \ell_n(\hat{\boldsymbol{\beta}}_{n,p_0}, \hat{\beta}_0) \\ \leq& \ell_n(\mathbf{0}, 0) \leq \frac{||Y||_2^2}{n} \leq \max_i Y_i^2.\end{aligned}$$

Consequently, the optimization problem (1.2) can be equivalently written with $(\boldsymbol{\beta}, \beta_0)$ constrained to belong to a compact and convex set K. Since ℓ_n is convex and converges pointwise to $\ell^\star$ according to (1.3), the uniform convergence over the compact set K also holds, i.e., in probability

$$\lim_{n\to\infty} \sup_{(\boldsymbol{\beta},\beta_0)\in K} |\ell_n(\boldsymbol{\beta}, \beta_0) - \ell^\star(\boldsymbol{\beta}, \beta_0)| = 0. \tag{1.4}$$

Additionnally, since $\ell^\star$ is a quadratic convex function and the constraint domain K is convex, $\ell^\star$ has a unique minimum that we denote $\boldsymbol{\beta}^\star_{p_0,\lambda}$.

Finally, since the maximum of $\ell^\star$ is unique and ℓ_n uniformly converges to $\ell^\star$, we can apply theorem 5.7 from Van der Vaart (2000, page 45) to deduce that $(\hat{\boldsymbol{\beta}}_{n,p_0}, \hat{\beta}_0)$ is a consistent estimate of $\boldsymbol{\beta}^\star_{p_0,\lambda}$.

We can conclude that, in probability,

$$\lim_{n\to\infty} \mathbb{P}\big(\hat{\mathscr{P}}_{M_n,n,p_0,\lambda} = \{\mathscr{P} \in \mathscr{P}^\star_{p_0} : \beta^\star_{\mathscr{P},p_0,\lambda} > 0\}\big) = 1,$$

and the final stability result follows from the continuous mapping theorem. □

2. Dataset Descriptions

Dataset	Sample Size	Total Number of Variables	Number of Categorical Variables
Ozone	203	12	0
Mpg	392	7	0
Prostate	97	8	0
Housing	506	13	0
Diabetes	442	10	0
Machine	209	7	1
Galaxy	323	4	0
Abalone	4177	8	1
Bones	485	3	2

Table 5. Description of datasets

3. Number of Trees

The stability, predictivity, and computation time of SIRUS increase with the number of trees. Thus a stopping criterion is designed to grow the minimum number of trees that ensures stability and predictivity to be close to their maximum. It happens in practice that stabilizing the rule list is computationally more demanding in the number of trees than reaching a high predictivity. Therefore the stopping criterion is only based on stability, and defined as the minimum number of trees such that when SIRUS is fit twice on the same given dataset, 95% of the rules are shared by the two models in average.

To this aim, we introduce $1 - \varepsilon_{M,n,p_0}$, an estimate of the mean stability $\mathbb{E}[\hat{S}_{M_n,n,p_0}|\mathscr{D}_n]$ when SIRUS is fit twice on the same dataset $\mathscr{D}_n$. ε_{M,n,p_0} is defined by

$$\varepsilon_{M,n,p_0} = \frac{\sum_{\mathscr{P}\in\Pi} z_{M,n,p_0}(\mathscr{P})(1 - z_{M,n,p_0}(\mathscr{P}))}{\sum_{\mathscr{P}\in\Pi}(1 - z_{M,n,p_0}(\mathscr{P}))},$$

where $z_{M,n,p_0}(\mathscr{P}) = \Phi(Mp_0, M, p_n(\mathscr{P}))$, the cdf of a binomial distribution with parameter $p_n(\mathscr{P}) = \mathbb{E}[\hat{p}_{M_n,n}(\mathscr{P})|\mathscr{D}_n]$, M trials, evaluated at Mp_0. It happens that ε_{M,n,p_0} is quite insensitive to p_0. Consequently it is simply averaged over a grid $\hat{V}_{M,n}$ of many possible values of p_0. Therefore, the number of trees is set, for $\alpha = 0.05$, by

$$\underset{M}{\text{argmin}}\Big\{\frac{1}{|\hat{V}_{M,n}|}\sum_{p_0\in\hat{V}_{M,n}} \varepsilon_{M,n,p_0} < \alpha\Big\},$$

to ensure that 95% of the rules are shared by the two models in average. See Section 4 from Bénard et al. (2019) for a thorough explanation of this stopping criterion.