A rigorous method to compare interpretability of rule-based algorithms.

Vincent Margot

Advestis
69 Boulevard Haussmann, 75008 Paris, France
vmargot@advestis.com

April 7, 2020

Abstract

Interpretability is becoming increasingly important in predictive model analysis. Unfortunately, as mentioned by many authors, there is still no consensus on that idea. The aim of this article is to propose a rigorous mathematical definition of the concept of interpretability, allowing fair comparisons between any rule-based algorithms. This definition is built from three notions, each of which being quantitatively measured by a simple formula: predictivity, stability and simplicity. While predictivity has been widely studied to measure the accuracy of predictive algorithms, stability is based on the Dice-Sorensen index to compare two sets of rules generated by an algorithm using two independent samples. Simplicity is based on the sum of the length of the rules deriving from the generated model. The final objective measure of the interpretability of any rule-based algorithm ends up as a weighted sum of the three aforementioned concepts. This paper concludes with the comparison of the interpretability between four rule-based algorithms.

Keywords: Interpretability, Transparency, Explainability, Predictivity, Stability, Simplicity, Rule-based algorithms, Machine Learning.

1 Introduction

The widespread use of machine learning (ML) in many sensible areas such as healthcare, justice, asset management has underlined the importance of interpretability in the decision-making process. In recent years, the number of publications on interpretability has increased exponentially. Usually, two main ways can be distinguished for the production of interpretable predictive models. The first one relies on the use of an uninterpretable machine learning algorithm to create predictive models, and then to take them up again to create a so-called post-hoc interpretable model, for example LIME [1], DeepLIFT [2], SHAP [3].

These explanatory models try to measure the importance of a feature on the prediction process (see [4] for an overview of existing methods). However, as outlined in [5], the explanations may not be sufficient for a sensitive decision-making process.

The other way is to use an intrinsically interpretable algorithm to directly generate an interpretable model such as decision tree algorithms CART [6], ID3 [7], C4.5 [8], RIPPER [9] or rule-based algorithms FORS [10], M5 Rules [11], RuleFit [12], Ender [13], Node Harvest [14] or more recently SIRUS [15] and RICE [16].

These algorithms are based on the notion of rule. A rule is a If-Then statement of the form

$$\begin{aligned} \text{IF} \quad & c_1 \text{ And } c_2 \text{ And } \dots \text{ And } c_k \\ \text{THEN} \quad & \text{Prediction} = p, \end{aligned} \tag{1}$$

The condition part If is a logical conjunction, where c_i's are tests that check whether the observation has the specified properties or not The number k is called the length of the rule. If all c_i's are fulfilled the rule is said activated. And the conclusion part Then is prediction of the rule if it is activated. Usually, if the feature space is $\mathbb{R}^d$, each c_i checks if one specific feature is in an interval (e.g $x \in [a, b]$).

In [17], author emphasizes that there is no rigorous mathematical foundation for the concept of interpretability. In this paper, a rigorous, quantitative and objective measure of the interpretability is proposed as a comparison criterion for any rule-based algorithms. This measure is based on the triptych predictability, computability, stability presented in [18]: Predictability measures the accuracy of the predictive model. Stability quantifies the noise sensitivity of an algorithm. Finally, the notion of computability has been replaced by a notion of simplicity. Computability is important in practice, but it does not reflect the model's ability to be interpreted, whereas the simplicity of the model proves to be more efficient at this task.

2 Predictivity

The aim of a predictive model is to predict the value of a variable of interest $Y \in \mathbb{R}$, given features $X \in \mathbb{R}^d$. Formally, we set the standard regression setting as follows: Let (X, Y) be a couple of random variable in $\mathbb{R}^d \times \mathbb{R}$ of unknown distribution $\mathbb{Q}$ such that

$$Y = g^*(X) + Z, \tag{2}$$

where $\mathbb{E}[Z] = 0$ and $\mathbb{V}(Z) - \sigma^2$ and g^* is a measurable function from $\mathbb{R}^d$ to $\mathbb{R}$.

We denote by $\mathcal{G}$ the set of all measurable functions from $\mathbb{R}^d$ to $\mathbb{R}$. The accuracy of a regression function $g \in \mathcal{G}$ is measured by its risk, defined as

$$\mathrm{L}(g) := \mathbb{E}_{\mathbb{Q}} \left[\gamma \left(g; (X, Y) \right) \right], \tag{3}$$

where $\gamma : \mathcal{G} \times (\mathbb{R}^d \times \mathbb{R}) \to [0, \infty[$ is called a contrast function. The risk measures the average discrepancy, given a new observation (X, Y) from the distribution $\mathbb{Q}$, between $g(X)$ and Y.

Given a sample $D_n = ((X_1, Y_1), \ldots, (X_n, Y_n))$, we aim at predicting Y conditionally on X. The observations (X_i, Y_i) are independent and identically distributed (i.i.d) from the distribution $\mathbb{Q}$.

To do so, we consider a statistical algorithm $\mathcal{A}$ which is a measurable mapping defined by

$$\begin{array}{rccc} \mathcal{A} & : & (\mathbb{R}^d \times \mathbb{R})^n & \to & \mathcal{G}_n^{\mathcal{A}} \\ & & D_n & \mapsto & g_n^{\mathcal{A}}, \end{array} \tag{4}$$

where $\mathcal{G}_n^{\mathcal{A}} \subseteq \mathcal{G}$.

The purpose of an algorithm $\mathcal{A}$, is to generate a measurable function $g_n^{\mathcal{A}}$ that minimizes the risk (3). To carry out this minimization, the algorithms use the Empirical Risk Minimization principle (ERM) [19], meaning that

$$g_n^{\mathcal{A}} = \underset{g \in \mathcal{G}_n^{\mathcal{A}}}{\arg\min}\, \mathrm{L}_n(g), \tag{5}$$

where $\mathrm{L}_n(g) = \frac{1}{n} \sum_{i=1}^{n} \gamma(g, (X_i, Y_i))$ is the empirical risk.

The choice of γ depends on the nature of Y. For example, if $Y \in \mathbb{R}$, one generally uses the quadratic contrast with $\gamma\left(g; (X, Y)\right) = \left(g(X) - Y\right)^2$. The minimizer of the risk (3) with the quadratic contrast is the called the regression function $\eta \in \mathcal{G}$, defined by

$$\eta(x) := \mathbb{E}_{\mathbb{Q}}\left[Y \mid X = x\right].$$

If $Y \in \{0, 1\}$, one uses the $0-1$ contrast function $\gamma\left(g; (X, Y)\right) := \mathbf{1}_{g(X) \neq Y}$. The minimizer of the risk (3) with the $0-1$ contrast function is called the Bayes classifier $s \in \mathcal{G}$ defined by

$$\eta(x) := \mathbf{1}_{\mathbb{Q}(Y=1|X=x) \geq 1/2}.$$

Hence, according to the ERM principle, the choice of γ determines the function that an algorithm, $\mathcal{A}$ tries to estimate, and thus the function $g_n^{\mathcal{A}}$.

The notion of predictivity is based on the ability of an algorithm to provide an accurate predictive model. This notion has been well studied since years. In this paper, the predictivity is defined as follows:

$$P(g_n^{\mathcal{A}}, \gamma) := \frac{\mathrm{L}(g_n^{\mathcal{A}})}{\mathrm{L}(h_\gamma)}, \tag{6}$$

where h_γ is the trivial constant predictor according to γ. For example, for the quadratic contrast $h_\gamma = \mathbb{E}[Y]$ and for the $0-1$ contrast $h_\gamma = median(Y)$.

This quantity as a measure of the accuracy is independent from the range of Y. We may assume that it is a positive number between 0 and 1. Indeed, the risk (3) is a positive function and if $P(g_n^{\mathcal{A}}, \gamma) > 1$, it means that the predictor $g_n^{\mathcal{A}}$ is worse than the trivial constant predictor.

3 q-Stability

In [15], authors have proposed a measure of the stability of a rule-based algorithm built upon the following definition:

> *A rule learning algorithm is stable if two independent estimations based on two independent samples result in two similar lists of rules.*

The notion of q-stability is based on the same definition. This notion appears to be fairer for algorithms that do not use features discretisation and operate on real rather than integer values. In fact, the probability that a decision tree algorithm cuts on the same exact value for the same rule, given two independent samples is null. For this reason, the pure stability appears too penalizing in this case.

Features discretization is a common solution for controlling the complexity of a rule generator. In [21], for example, the authors use the entropy minimization heuristic to discretize the features and for the algorithms BRL (Bayesian Rule Lists) [22], SIRUS [15] and RICE [16], authors have used the empirical quantiles of features to discretize them. See [23] for an overview of usual discretization methods.

Let $q \in \{1, \dots, n\}$ be the number of quantiles considered for the discretization and let X be a feature. An integer $p \in \{1, \dots, q\}$, named bin, is associated to each interval $[x_{(p-1)/q}, x_{p/q}]$, where $x_{p/q}$ is p-th q-quantile of X. A discrete version of a feature X, denoted $Q_q(X)$, is designed by replacing each value by its corresponding bins. in other words, a value p_a is associated for all $a \in X$ such that $a \in [x_{(p_a-1)/q}, x_{p_a/q}]$.

This discretization process can be extended to a rule set by replacing for all rules, the intervals' bound of each test c_i by their corresponding bins. For example, the test $X \in [a, b]$ becomes $Q_q(X) \in [\![p_a, p_b]\!]$, where p_a and p_b are such that $a \in [x_{(p_a-1)/q}, x_{p_a/q}]$ and $b \in [x_{(p_b-1)/q}, x_{p_b/q}]$.

The formula of the q-stability is based on the so-called Dice-Sorensen index. Let $\mathcal{A}$ be a rule-based algorithm and let D_n and D'_n two independant samples of n i.i.d observations drawn from the same distribution $\mathbb{Q}$. And let $R_n^{\mathcal{A}}$ and $R_n'^{\mathcal{A}}$ be the sets of rules generated by $\mathcal{A}$, given D_n and D'_n respectively. Then, the q-stability is calculated by

$$\mathcal{S}_n^q(\mathcal{A}) := 1 - \frac{2\left|Q_q(R_n^{\mathcal{A}}) \cap Q_q(R_n'^{\mathcal{A}})\right|}{|Q_q(R_n^{\mathcal{A}})| + |Q_q(R_n'^{\mathcal{A}})|}, \tag{7}$$

where $Q_q(R)$ is the discretized version of the rule-set R and with the convention that $0/0 = 0$. The discretization process is performed using D_n and D'_n respectively.

This quantity is a positive number between 0 and 1: If $Q_q(R_n^{\mathcal{A}})$ and $Q_q(R_n'^{\mathcal{A}})$ have no common rules, then $\mathcal{S}_n^q(\mathcal{A}) = 1$ while if $Q_q(R_n^{\mathcal{A}})$ and $Q_q(R_n'^{\mathcal{A}})$ are the same, then $\mathcal{S}_n^q(\mathcal{A}) = 0$.

4 Simplicity

In [20], authors have introduced a notion of interpretability score based on the sum of the length of all the rules constituting the predictive model.

Definition 4.1. *The interpretability score of an estimator g_n generated by a set of rules R_n is defined by*

$$Int(g_n) := \sum_{r \in R_n} length(r). \tag{8}$$

Furthermore, the value (8), which is a positive number, cannot be directly compared to the values from (6) and (7), which are between 0 and 1.

The measure of the simplicity is based on the definition 4.1. The idea is to compare (8) relatively to a set of algorithms $\mathcal{A}_1^m = \{\mathcal{A}_1, \ldots, \mathcal{A}_m\}$. Hence the simplicity of an algorithm $\mathcal{A}_i \in \mathcal{A}_1^m$ is defined in relative terms as follows:

$$S_n(\mathcal{A}_i, \mathcal{A}_1^m) = 1 - \frac{min\{Int(g_n^A : A \in \mathcal{A}_1^m)\}}{Int(g_n^{\mathcal{A}_i})}. \tag{9}$$

Like the previous ones, this quantity is a positive number between 0 and 1: If $\mathcal{A}_i$ generates the simplest predictor among the set of algorithms $\mathcal{A}_1^m$ then $S_n(\mathcal{A}_i, \mathcal{A}_1^m) = 0$. Then, the simplicity of other algorithms in $\mathcal{A}_1^m$ are evaluated relatively to $\mathcal{A}_i$.

5 Interpretability

The main idea underlying the definition of interpretability of a rule-based algorithm, is the use of a weighted sum of the predictivity (6) the stability (7) and the simplicity (9). Let $\mathcal{A}_1^m$ be a set of rule-based algorithms, the interpretability of any algorithm $\mathcal{A}_i \in \mathcal{A}_1^m$ is defined by:

$$\mathcal{I}(\mathcal{A}_i, D_n, D'_n, \gamma, q) = \alpha_1 P(g_n^{\mathcal{A}_i}, \gamma) + \alpha_2 \mathcal{S}_n^q(\mathcal{A}_i) + \alpha_3 S_n(\mathcal{A}_i, \mathcal{A}_1^m), \tag{10}$$

where the coefficients α_1, α_2 and α_3 have been chosen according to the statistician's desiderata such that $\alpha_1 + \alpha_2 + \alpha_3 = 1$. If a statistician tries to understand and to describe a phenomenon then simplicity and predictivity are more important than stability.

It is important to notice that the definition of interpretability (10) depends on the set of rule-based algorithms and a regression setting. Therefore, the interpretable value only makes sense within that set of algorithms and for a specific regression setting.

6 Application

The aim of this application is to compare four rule-based algorithms: the Decision Tree algorithm (DT) [6], RuleFit (RF) [12], the Covering Algorithm (CA)

[20] and RICE [16]. Their parametrization is summarized in Table 1. For this application the same model as in [12] is considered. Two samples D_1 and D_2 of $n = 5000$ data are generated following the regression setting:

$$Y = g^*(X) + Z,$$

where $d = 10$ (the dimension of X) and

$$g^*(X) = 9 \prod_{j=1}^{3} \exp\left(-3\left(1 - X_j\right)^2\right) - 0.8 \exp\left(-2\left(X_4 - X_5\right)\right) + 2\sin^2(\pi \cdot X_6) - 2.5\left(X_7 - X_8\right), \quad (11)$$

where X_j is the j-st component of X and $Z \sim \mathcal{N}(0, \sigma^2)$. The value of $\sigma > 0$ was chosen to produce a two-to-one signal-to-noise ratio. The variables were generated from a uniform distribution on $[0, 1]$.

Algorithm	**Parameters**
DT	Number maximal of rules 2000
RF	Maximal number of rules 2000 Cross validation 3
CA	Number of rules by tree 200 Number of trees 100
RICE	Number of candidates 150 Maximal length 3

Table 1: Algorithm parameters.

Predictivity (6) is approximated using 50000 test observations and by averaging error of predictors generated from D_1 and D_2. The q-stability (7) is measured by setting $q = 10$ and discretizing with respect to D_1 and D_2. Simplicity (9) of an algorithm is computed by averaging the measure on predictors generated from D_1 and D_2 respectively. Finally, the interpretability (10) is calculated with $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$. The results are summarized in Table 2.

Remark: For the sake of simplification, linear relationships generated by RuleFit have been considered as rules for the evaluation of the q-stability. This algorithm generates four linear relationships from D_1 using variables X_3, X_6, X_7, X_8 whereas a single relationship is produced from D_2 using X_6. Regarding the linear relationships generated by RuleFit on the two datasets D_1 and D_2 they only show one "rule" in common.

RICE and Covering Algorithm seem to be the most interpretable algorithms for this setting. However, the predictivity value of RICE is very poor compared to the other algorithms. Therefore, the Covering Algorithm is the most interpretable algorithm in this panel for the setting (11). Even if RuleFit is the best algorithm of this panel in predictivity and q-stability it generated too many rules and has therefore a weaker simplicity.

Algorithm	Predictivity	q-Stability	Simplicity	Interpretability
DT	0.51	1	0.39	0.63
RF	0.26	0.89	0.61	0.59
CA	0.35	0.97	0.30	0.54
RICE	0.67	0.95	0	0.54

Table 2: Details of the interpretability value for each algorithm.

7 Conclusion and perspectives

In this paper, a quantitative criterion for interpretability of rule-based algorihms was presented. This measure is based on the triptych: Predictivity (6), Stability (7) and Simplicity (9). This new concept of interpretability has been thought to be fair and rigorous. It can be adapted to the various desiderata of the statistician by choosing appropriate the coefficients in the interpretability formula (10). An application on four rule-based algorithm: Decision Tree algorithm [6], RuleFit [12], Covering Algorithm [20] and RICE [16], shows how to use and analyse the interpretability value. This application will be extended to others well-known rule-based algorithms such as C4.5 [8], RIPPER [9], Ender [13] and SIRUS [15] in a further work.

This methodology seems to make the interpretability comparison of rule-based algorithms quite fair. However, according to Definition 4.1, 100 rules of length 1 have the same simplicity that one single rule of length 100, which is debatable. Moreover, the stability measure is purely syntactic and rather restrictive. Indeed, if some features are duplicated, two rules may have two different syntactic conditions but by otherwise identical based in their activations. One way of relaxing this stability criterion could be to compare the rules, based on their activation sets (i.e. by looking to observations where conditions are met simultaneously). Finally, this comparison of interpretability between a set of algorithms makes only sense for rule-based algorithm. It could be interesting to extend it to other types of algorithms.

References

[1] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1135–1144. ACM, 2016.

[2] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. arXiv preprint arXiv:1704.02685, 2017.

[3] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems, pages 4765–4774, 2017.

[4] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti. A survey of methods for explaining black box models. arXiv preprint arXiv:1802.01933, 2018.

[5] C. Rudin Please stop explaining black box models for high stakes decisions. arXiv preprint arXiv:1811.10154, 2018.

[6] L. Breiman and J.H. Friedman and R. Olshen and C. Stone Classification and Regression Trees. In CRC press 1984.

[7] J. R. Quinlan Induction of decision trees. In Springer pages 81–106. 1986.

[8] J. R. Quinlan C4.5: Programs for Machine Learning. In Morgan Kaufmann, 1993.

[9] W.W. Cohen Fast effective rule induction. In Machine Learning Proceedings, pages 115-123, 1995.

[10] A. Karalič and I. Bratko First order regression. InSpringer, pages 147–176, 1997.

[11] G. Holmes, M. Hall and E. Prank Generating rule sets from model trees. In Springer, pages 1–12, 1999.

[12] J.H. Friedman and B.E. Popescu Predective Learning via Rule Ensembles. In The Annals of Applied Statistic, pages 916–954, 2008.

[13] K. Dembczyński, W. Kotłowski and R. Słowiński Solving Regression by Learning an Ensemble of Decision Rules. In International Conference on Artificial Intelligence and Soft Computing, pages 533–544, 2008.

[14] N. Meinshausen Node harvest. In Institute of Mathematical Statistics, pages 2049–2072, 2010.

[15] C. Bénard and G. Biau, S. Da Veiga and E. Scornet SIRUS: making random forests interpretable. arXiv preprint arXiv:1908.06852, 2019.

[16] V. Margot, J.P. Baudry, F. Guilloux and O. Wintenberger Rule Induction Covering Estimator : A New Data Dependent Covering Algorithm, , 2020.

[17] Z.C. Lipton, The mythos of model interpretability. arXiv preprint arXiv:1606.03490, 2017.

[18] Y.Bin and K.Kumbier Three principles of data science: predictability, computability, and stability (PCS). aarXiv preprint arXiv:1901.08152, 2019.

[19] V. Vapnik and S. Kotz Estimation of Dependences Based on Empirical Data. Springer-Verlag New York, 1982.

[20] V. Margot, J.P. Baudry, F. Guilloux and O. Wintenberger Consistent Regression using Data-Dependent Coverings arXiv preprint arXiv:1907.02306, 2019.

[21] U.M Fayyad and K.B Irani Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In Proc. 13th Int. Joint Conf. on Artificial Intelligence, pages 1022–1027, 1993.

[22] B. Letham, C. Rudin, T.H. McCormick, D. Madigan Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. In The Annals of Applied Statistics, volume 9, number 3, pages 1350–1371, 2015.

[23] J. Dougherty, R. Kohavi and M. Sahami Supervised and unsupervised discretization of continuous features. In Machine Learning Proceedings, pages 194–202, 1995.