Entropic Variable Projection for Model Explainability and Intepretability

François Bachoc[1], Fabrice Gamboa[12], Max Halford[13], Jean-Michel Loubes[14] and Laurent Risser[12]

[1]Institut de Mathématiques de Toulouse
[2] ANITI, Artificial and Natural Intelligence Toulouse Institute
[3] Institut de recherche en informatique de Toulouse
[4] DEEL (DEpendable Explainable Learning)

Abstract

In this paper, we present a new explainability formalism designed to explain how the input variables of the testing set impact the predictions of black-box decision rules. Hence we propose a group explainability frame for machine learning algorithms based on the variability of the distribution of the input variables. Our formalism is based on an information theory framework that quantifies the influence of all input-output observations when emphasizing the impact of each input variable, based on entropic projections. This formalism is thus the first unified and model agnostic framework enabling us to interpret the dependence between the input variables, their impact on the prediction errors, and their influence on the output predictions. In addition and most importantly, we prove that computing an explanation in our framework has a low algorithmic complexity making it scalable to real-life large datasets. We illustrate our strategy by explaining complex decision rules learned using XGBoost, Random Forest or Deep Neural Network classifiers. We finally make clear its differences with explainability strategies based on single observations, such as those of LIME or SHAP.

1 Introduction

Machine learning algorithms build predictive models which are nowadays used for a large variety of tasks. Over the last decades, the complexity of such algorithms has grown, going from simple and interpretable prediction models based on regression rules to very complex models such as random forests, gradient boosting, and models using deep neural networks. We refer to Hastie et al. [2009] for a description of these methods. Such models are designed to maximize the accuracy of their predictions at the expense of the interpretability of the decision rule. Little is also known about how the information is processed in order to obtain a prediction, which explains why such models are widely considered as black boxes.

This lack of interpretability gives rise to several issues. When an empirical risk is minimized, the training procedure may be unstable or highly dependent

on the optimization procedure due to *e.g.* non-convexity and multimodality. Another subtle, though critical, issue is that the optimal decision rules learned by a machine learning algorithm highly depend on the properties of the learning sample. If a learning sample presents unwanted trends or a bias, then the learned decision rules will reproduce these trends or bias, even if there is no intention of doing so. As a consequence, many users express a lack of trust in these algorithms. The European Parliament even adopted a law called GDPR (General Data Protection Regulation) to protect the citizens from decisions made without the possibility of explaining why they were taken, introducing a right for explanation in the civil code. Hence, building intelligible models is nowadays an important challenge for data scientists.

Different methods have been proposed to make understandable the reasons leading to a prediction, each author using a different notion of explainability and interpretability. We mention early works by Herlocker et al. [2000] for recommender systems, Craven and Shavlik [1995] for neural networks, Dzindolet et al. [2003] or Lou et al. [2012] for generalized additive models. Another generic solution, described in Baehrens et al. [2010] and Caruana et al. [2015], focused on medical applications. In Lipton [2016], a discussion was recently opened to refine the discourse on interpretability. Recently, a special attention has also been given to deep neural systems. We refer for instance to Montavon et al. [2017], Selvaraju et al. [2016], Hooker et al. [2019] and references therein. Clues for real-world applications are given in Hall et al. [2018]. Ribeiro et al. [2016] (LIME) recently proposed to locally mimic a black-box model and then to give a feature importance analysis of the variables at the core of the prediction rule. A counterfactual model Wachter et al. [2018] was also proposed in Goyal et al. [2019] to explain how the prediction made by a classifier on a query image can be changed by transforming a region of this image. In the same vein, a method called integrated gradients was specifically designed in Sundararajan et al. [2017] for the interpretability of single predictions using neural networks. In the *Fair learning* community, counterfactual models are also used to assess whether the predictions of machine learning models are fair Kusner et al. [2017], Black et al. [2020]. An individual explanation method on images through adversarial examples was also presented in Ignatiev et al. [2019]. In Koh and Liang [2017] the authors finally proposed a strategy to understand black-box models, as we do, but in a parametric setting.

Our conception of the notion of interpretability for machine learning algorithms is the ability to quantify the specific influence of each of the $p \geq 1$ variables in a test set. We specifically determine the global effect of each variable in the learning rule and how a particular variation of this variable affects the accuracy of the prediction. This allows to understand how the predictions evolve when a characteristic of the observations is modified. To achieve this, we propose in this paper a sensitivity analysis strategy for machine learning algorithms. It is directly inspired by the field of sensitivity analysis for computer code experiments (see *e.g.* Lemaître et al. [2015]), where the relative importance of the input variables involved in an abstract input-output relationship modeling a computer code is computed Saltelli et al. [2008].

We emphasize that contrary to *e.g.* Ribeiro et al. [2016], Sundararajan et al. [2017] or Lundberg and Lee [2017] (SHAP), our method deals with global explainability since it quantifies the global effect of the variables for all the test samples instead of individual observations. We also highlight that our point of view is different from previous works where the importance of each variable was also considered. Sparse models (see Bühlmann and Van De Geer [2011] for a general introduction on the importance of sparsity) enable to identify important variables. Importance indicators have also been developed in machine learning to detect which variables play a key role in the algorithm. For instance, importance of variables is often computed using feature importance or Gini indices (see in Raileanu and Stoffel [2004] or Hastie et al. [2009]). Yet such indexes are computed without investigating the particular effects of each variable and without explaining their particular role in the decision process. We also strongly believe that running the algorithm over observations which are created artificially by increasing stepwise the value of a particular variable is not a desirable solution. By doing so, the correlations between variables are indeed not taken into account. Moreover, newly generated observations may be outliers with respect to the learning and test samples.

The paper falls into the following parts: Methodology is explained in Sections 2 and 3. Results are given in Section 4 and the discussions are finally developed in Section 5.

2 Optimal perturbation of Machine Learning datasets

We consider a test set $\{(X_i, Y_i)\}_{i=1,\dots,n}$ ($X_i = (X_i^1, \dots, X_i^p)$ are input observations while Y_i is the true output), on which we consider the outcome of black box decision rules $f : \mathbb{R}^p \to \mathbb{R}$. We consider throughout this paper that f has been learned by a training set and is fixed. We set $\hat{Y}_i = f(X_i)$ the predicted values. Hence we have at hand values $\{(X_i, \hat{Y}_i, Y_i)\}$ for $i = 1, \dots, n$.

Our goal is to explain the global behaviour of f. For this, we propose to study the response of f to distributional perturbations of the input variables. Since f has been learnt using data following a given distribution, the domain of validity of the algorithm should not deviate too much from this initial distribution. Hence we propose to build perturbed observations with a distribution as close as possible to the initial distribution using an information theory method. We will show that this amounts to reweighing the observations by proper weights calibrated to incorporate the chosen perturbation on the input variables as explained in Sections 2.2 and 2.3. The methodology to make this problem well posed and to quickly compute the optimal weights is the core of this paper.

2.1 General optimal perturbations under moment constraints in machine learning

In order to experience and to explore the behavior of a predictive model, a natural idea is to study its response to stressed inputs. There exist different

solutions to create modifications of a probability measure. In this paper, we consider an information theory framework in which we modify the distribution of the original test set $Q_n = \sum_{i=1}^n \frac{1}{n}\delta_{X_i,\hat{Y}_i,Y_i}$, by stressing the mean value of a function Φ of its variables (or simply by stressing the mean value of given variables), while minimizing the Kullback-Leibler information (also called mutual entropy) between the modified distribution Q_t and Q_n, making the problem well posed.

First, let us recall the definition of the Kulback-Leibler information. Let $(E, \mathcal{B}(E))$ be a measurable space and Q a probability measure on E. If P is another probability measure on $(E, \mathcal{B}(E))$, then the Kullback-Leibler information $\mathrm{KL}(P, Q)$ is defined as equal to $\int_E \log \frac{\mathrm{d}P}{\mathrm{d}Q}\,\mathrm{d}P$, if $P \ll Q$ and $\log \frac{\mathrm{d}P}{\mathrm{d}Q} \in L^1(P)$, and equal to $+\infty$ otherwise.

For a given $k \geq 1$, let $\Phi : \mathbb{R}^{p+2} \to \mathbb{R}^k$ be a measurable function representing the shape of the stress deformation. We write

$$t_0 = \int_{\mathbb{R}^k} \Phi(x)\,\mathrm{d}Q_n(x) = \frac{1}{n}\sum_{i=1}^n \Phi(X_i, \hat{Y}_i, Y_i)$$

as the empirical mean of Φ with the distribution Q_n.

For $t \in \mathbb{R}^k$, $t \neq t_0$, we now aim at finding a new distribution Q_t satisfying $\int_{\mathbb{R}^k} \Phi(x)\,\mathrm{d}Q_t(x) = t$, and with $\mathrm{KL}(Q_t, Q_n)$ as small as possible. Different functions make it possible to quantify the differences between Q_t and Q_n as a function of $t - t_0$. Our choice of the Kullback-Leibler information as a measure of similarity between Q_t and Q_n is motivated by our next theorem, that states that the corresponding optimization problem is favorable, both theoretically and numerically.

We set for two vectors $x, y \in \mathbb{R}^k$ the scalar product as $\langle x, y\rangle = x^\top y$.

Theorem 2.1. *Let $t \in \mathbb{R}^k$ and $\Phi : \mathbb{R}^{p+2} \to \mathbb{R}^k$ be measurable. Assume that t can be written as a convex combination of $\Phi(X_1, \hat{Y}_1, Y_1), \ldots, \Phi(X_n, \hat{Y}_n, Y_n)$, with positive weights. Assume also that* $\mathrm{span}(\Phi(X_1, \hat{Y}_1, Y_1), \ldots, \Phi(X_n, \hat{Y}_n, Y_n)) = \mathbb{R}^k$.

Let $\mathbb{P}_{\Phi,t}$ be the set of all probability measures P on $\mathbb{R}^{p+2}$ such that $\int_{\mathbb{R}^{p+2}} \Phi(x)\,\mathrm{d}P(x) = t$. For a vector $\xi \in \mathbb{R}^k$, let $Z(\xi) := (1/n)\sum_{i=1}^n e^{\langle \Phi(X_i,\hat{Y}_i,Y_i),\xi\rangle}$. Define now $\xi(t)$ as the unique minimizer of the strictly convex function $H(\xi) := \log Z(\xi) - \langle \xi, t\rangle$. Then,

$$Q_t := \mathrm{arginf}_{P \in \mathbb{P}_{\Phi,t}} \mathrm{KL}(P, Q_n) \tag{1}$$

exists and is unique. Furthermore, we have

$$Q_t = \frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \delta_{X_i,\hat{Y}_i,Y_i}, \tag{2}$$

with, for $i = 1, \ldots, n$,

$$\lambda_i^{(t)} = \exp\Big(\langle \xi(t), \Phi(X_i, \hat{Y}_i, Y_i)\rangle - \log Z(\xi(t))\Big). \tag{3}$$

A particularly appealing aspect of Theorem 2.1 is that Q_t is supported by the same observations as Q_n, the mean change of Φ only leading to different weights for the observations. As a consequence, sampling new stressed test sets does not require to create new input-output observations $(X_i, \hat{Y}_i, Y_i)$ but only to compute the weights $\lambda_i^{(t)}$. This can be solved very quickly using Eq. (3). The optimization problem in Theorem 2.1 is convex and can be addressed very efficiently. The gradient of its objective function is provided in Appendix B. This makes it possible to deal with very large databases without computing new values for new observations. This differs from existing techniques based on perturbed observations as *e.g.* in LIME Ribeiro et al. [2016], where the data used for testing are created by changing randomly the labels or by bootstrapping the observations.

We remark that, in Theorem 2.1, the condition that t can be written as a convex combination of $\Phi(X_1, \hat{Y}_1, Y_1), \ldots, \Phi(X_n, \hat{Y}_n, Y_n)$, with positive weights, is almost minimal. Indeed, it is necessary that t can be written as a convex combination of $\Phi(X_1, \hat{Y}_1, Y_1), \ldots, \Phi(X_n, \hat{Y}_n, Y_n)$ (otherwise the set of distributions that are absolutely continuous with respect to Q_n and yield expectation t for Φ is empty). For all the examples we have considered, this condition in Theorem 2.1 was not restrictive.

Hereafter, we show how to choose Φ specifically, to shed light on the impact of various features of the input variables.

2.2 Application to variable importance by stress of the mean

We now apply Theorem 2.1 to the special case of perturbing the mean of one of the p input variables, meaning that Φ is valued in $\mathbb{R}$ (*i.e.* $k = 1$).

Theorem 2.2. *Let $t \in \mathbb{R}$ and $j_0 \in \{1, \ldots, p\}$. Assume that $\min_{i=1}^n X_i^{j_0} < t < \max_{i=1}^n X_i^{j_0}$.*

Let $\mathbb{P}_{j_0,t}$ be the set of probability measures on $\mathbb{R}^{p+2}$ such that, when $(X, \hat{Y}, Y)$ follows a distribution $P \in \mathbb{P}_{j_0,t}$, we have $\mathbb{E}(X^{j_0}) = t$. For $\xi \in \mathbb{R}$, let $Z(\xi) := (1/n)\sum_{i=1}^n e^{\xi X_i^{j_0}}$. Define now $\xi(t)$ as the unique minimizer of the strictly convex function $H(\xi) := \log Z(\xi) - \xi t$. Then,

$$Q_{j_0,\tau} := \operatorname{arginf}_{P \in \mathbb{P}_{j_0,\tau}} \mathrm{KL}(P, Q_n)$$

exists and is unique. Furthermore, we have

$$Q_{j_0,t} = \frac{1}{n}\sum_{i=1}^n \lambda_i^{(j_0,t)} \delta_{X_i, \hat{Y}_i, Y_i},$$

with, for $i = 1, \ldots, n$,

$$\lambda_i^{(j_0,t)} = \exp\Big(\xi(t) X_i^{j_0} - \log Z(\xi(t))\Big).$$

This theorem enables to re-weight the observations of a variable so that its mean increases or decreases. This is then used in Section 3 to understand the particular role played by each variable.

2.3 Stressing several means, variances and covariances

The next proposition enables to stress the means of several variables at the same time.

Proposition 2.3. *Let $1 \leq c \leq p$ and let $j_1, \ldots, j_c$ be two-by-two distinct in $\{1, \ldots, p\}$. Let $t_1, \ldots, t_c \in \mathbb{R}$. Assume that there exists a convex combination of $(X_i^{j_1}, \ldots, X_i^{j_c})_{i=1,\ldots,n}$ with positive weights that is equal to $(t_1, \ldots, t_c)$. Assume also that* $\operatorname{span}\Big((X_i^{j_1}, \ldots, X_i^{j_c})_{i=1,\ldots,n}\Big) = \mathbb{R}^c$. *Then, there exists a unique distribution $Q_{t_1,\ldots,t_c}$ on $\mathbb{R}^{p+2}$ such that for $(X^1, \ldots, X^p, \hat{Y}, Y) \sim Q_{t_1,\ldots,t_c}$ we have $\mathbb{E}(X^{j_a}) = t_a$ for $a = 1, \ldots, c$ and such that $\mathrm{KL}(Q_{t_1,\ldots,t_c}, Q_n)$ is minimal. This distribution is obtained by the distribution Q_t in Theorem 2.1, in the case where $k = c$ and $\Phi(X^1, \ldots, X^p, \hat{Y}, Y) = (X^{j_1}, \ldots, X^{j_c})$.*

The next proposition enables to stress the variance of a variable while preserving its mean $m_{j_0} = (1/n)\sum_{i=1}^n X_i^{j_0}$.

Proposition 2.4. *Let $j_0 \in \{1, \ldots, p\}$. Let $v \in [0, \infty)$. Assume that there exists a convex combination of $(X_i^{j_0}, (X_i^{j_0})^2)_{i=1,\ldots,n}$ with positive weights that is equal to $(m_{j_0}, m_{j_0}^2 + v)$. Assume also that* $\operatorname{span}\Big((X_i^{j_0}, (X_i^{j_0})^2)_{i=1,\ldots,n}\Big) = \mathbb{R}^2$. *Then, there exists a unique distribution $Q_{j_0,v}$ on $\mathbb{R}^{p+2}$ such that for $(X^1, \ldots, X^p, \hat{Y}, Y) \sim Q_{j_0,v}$ we have $\mathbb{E}(X^{j_0}) = m_{j_0}$ and $\mathrm{Var}(X^{j_0}) = v$ and such that $\mathrm{KL}(Q_{j_0,v}, Q_n)$ is minimal. This distribution is obtained by the distribution Q_t in Theorem 2.1, in the case where $k = 2$, $\Phi(X^1, \ldots, X^p, \hat{Y}, Y) = (X^{j_0}, (X^{j_0})^2)$ and $t = (m_{j_0}, m_{j_0}^2 + v)$.*

Finally, we next show how to stress the covariance between two variables while preserving their means $m_{j_1} = (1/n)\sum_{i=1}^n X_i^{j_1}$ and $m_{j_2} = (1/n)\sum_{i=1}^n X_i^{j_2}$.

Proposition 2.5. *Let $j_1, j_2 \in \{1, \ldots, p\}$ be distinct. Let $c \in \mathbb{R}$. Assume that there exists a convex combination of $(X_i^{j_1}, X_i^{j_2}, X_i^{j_1} X_i^{j_2})$ with positive weights that is equal to $(m_{j_1}, m_{j_2}, m_{j_1} m_{j_2} + c)$. Assume also that* $\operatorname{span}\Big((X_i^{j_1}, X_i^{j_2}, X_i^{j_1} X_i^{j_2})_{i=1,\ldots,n}\Big) = \mathbb{R}^3$. *Then, there exists a unique distribution $Q_{j_1,j_2,c}$ on $\mathbb{R}^{p+2}$ such that for $(X^1, \ldots, X^p, \hat{Y}, Y) \sim Q_{j_1,j_2,c}$ we have $\mathbb{E}(X^{j_1}) = m_{j_1}$, $\mathbb{E}(X^{j_2}) = m_{j_2}$ and $\mathrm{Cov}(X^{j_1}, X^{j_2}) = c$ and such that $\mathrm{KL}(Q_{j_1,j_2,c}, Q_n)$ is minimal. This distribution is obtained by the distribution Q_t in Theorem 2.1, in the case where $k = 3$, $\Phi(X^1, \ldots, X^p, \hat{Y}, Y) = (X^{j_1}, X^{j_2}, X^{j_1} X^{j_2})$ and $t = (m_{j_1}, m_{j_2}, m_{j_1} m_{j_2} + c)$.*

2.4 Asymptotic consistency

While, in this paper, the test set $(X_i, \hat{Y}_i, Y_i)_{i=1,\ldots,n}$ with empirical distribution Q_n is considered fixed, in Section 2.4 (and only in Section 2.4), we assume that

this test set $(X_i, \hat{Y}_i, Y_i)_{i=1,\dots,n}$ is random, composed of i.i.d. realizations of a distribution $Q^\star$.

The following proposition proves the statistical consistency of our methodology. We show that the optimally perturbed distribution Q_t of Theorem 2.1 (defined w.r.t. Q_n, Φ and t) converges to the corresponding optimally perturbed distribution $Q_t^\star$, (defined w.r.t. $Q^\star$, Φ and t).

Proposition 2.6. *Let $\Phi : \mathbb{R}^{p+2} \to \mathbb{R}^k$ and $t \in \mathbb{R}^k$ be fixed. Assume that the support of $Q^\star$ is bounded and that Φ is bounded in absolute value on the support of $Q^\star$. Assume also that for $v \in \mathbb{R}^k$, $Q^\star(\{x \in \mathbb{R}^{p+2}; \langle v, \Phi(x)\rangle = 0\}) = 1$ if and only if $v = 0$. Assume finally that there exists a distribution Q, absolutely continuous w.r.t. $Q^\star$, such that $\int_{\mathbb{R}^{p+2}} \Phi(x)\,\mathrm{d}Q(x) = t$.*

Then there exists a unique measure $Q_t^\star$ on $\mathbb{R}^{p+2}$ such that $\int_{\mathbb{R}^{p+2}} \Phi(x)\,\mathrm{d}Q_t^\star(x) = t$ and $\mathrm{KL}(Q_t^\star, Q^\star)$ is minimal. Furthermore, almost surely as $n \to \infty$, Q_t (given in Theorem 2.1) converges to $Q_t^\star$ weakly.

For the sake of concision, Proposition 2.6 is stated under boundedness assumptions and with independent realizations from $Q^\star$. These conditions could be weakened.

3 Explainable models using optimally perturbed data sets

In this section we consider that the transformation $\Phi : \mathbb{R}^{p+2} \to \mathbb{R}^k$ and the target multidimensional moment $t \in \mathbb{R}^k$ have been selected and that the conditions of Theorem 2.1 are satisfied. This theorem provides the optimally perturbed distribution Q_t, given by the weights $(\lambda_i^{(t)})_{i=1,\dots,n}$. We now suggest various quantitative properties of Q_t (that we call quantities of interest), that can quantify the behavior of the studied black box decision rule.

We shall focus on two classical situations encountered in machine learning: binary classification and multi-class classification. The regression case is also explained in Appendix C.

3.1 The case of binary classification

Consider that Y_i and $\hat{Y}_i = f(X_i)$ belong to $\{0, 1\}$ for all $i = 1 \dots, n$. This corresponds to the binary classification problem for which the usual loss function is $\ell(Y, f(X)) = \mathbf{1}\{Y \neq f(X)\}$. We suggest to use the indicators described hereafter for the perturbed distributions $Q_t = \frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \delta_{(X_i, \hat{Y}_i, Y_i)}$. Explaining the decision rules may first consist in quantifying the evolution of the error rate as a function of $t - t_0$, hence the first indicator is the error rate, *i.e.*

$$\mathrm{ER}_t = \frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \mathbf{1}\{f(X_i) \neq Y_i\}.$$

In terms of interpretation, when Φ is given as in Theorem 2.2, with $\Phi(X^1, \ldots, X^p, \hat{Y}, Y) = X^{j_0}$, t corresponds to the new (stressed) mean of the variable X^{j_0} while the former (unstressed) mean is t_0. In this case, plotting ER_t as a function of $t - t_0$ highlights the variables which produce the largest amount of confusion in the error, *i.e.* those for which small or large values provide the most variability among the two predicted classes, hampering the prediction error rate. The False and True Positive Rates may alternatively be represented using

$$\mathrm{FPR}_t = \frac{\frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \mathbf{1}\{Y_i \neq 1\} \mathbf{1}\{f(X_i) = 1\}}{\frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \mathbf{1}\{f(X_i) = 1\}}$$

and

$$\mathrm{TPR}_t = \frac{\frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \mathbf{1}\{f(X_i) = 1\} \mathbf{1}\{Y_i = 1\}}{\frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \mathbf{1}\{Y_i = 1\}}.$$

Again with Φ as in Theorem 2.2, a ROC curve corresponding to perturbations of the variable j_0 can then be obtained by plotting pairs $(\mathrm{FPR}_t, \mathrm{TPR}_t)$ for a large number of $t \in \mathbb{R}$. We then obtain the evolution of both errors when t evolves, which allows a sharper analysis of the error evolution (see *e.g.* Appendix D.4). Finally, the variables influence on the predictions may be quantified by computing the proportion of predicted 1s

$$\mathrm{P1}_t = \frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} f(X_i)$$

which we suggest to plot similarly as ER_t, with Φ as in Theorem 2.2 (see Figure 1-(top)). The figures representing $\mathrm{P1}_t$ make it possible to simply understand the particular influence of the variables to obtain a decision $Y = 1$, whatever the veracity of the prediction. Importantly, they point out which variables should be positively or negatively modified in order to change a given decision.

3.2 The case of multi-class classification

We now consider the case of a classification into k different categories, *i.e.* where Y_i and $f(X_i)$ belong to $\{1, \ldots, k\}$ for all $i = 1, \ldots, n$, and where $k \in \mathbb{N}$ is fixed. In this case, the strategy described for the binary classification can naturally be extended using

$$\mathrm{Pj}_t = \frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \mathbf{1}\{f(X_i) = j\},$$

which denotes the portion of individuals assigned to the class j.

3.3 Using quantiles to compare multiple mean changes

Consider the case where Φ is given as in Theorem 2.2, with $\Phi(X^1, \ldots, X^p, \hat{Y}, Y) = X^{j_0}$, and where we want to plot the quantities of interest of Sections 3.1 and

3.2, as a function of $t - t_0$, for all the values of $j_0 = 1, \ldots, p$. In this case, an issue is that the interpretation of $t - t_0$ depends on the order of magnitude of the variable j_0, and thus changes with j_0.

In order to compare values of $t - t_0$ accross different variables, we suggest a common parametrization of $t - t_0$ for $j_0 = 1, \ldots, p$. For $j_0 = 1, \ldots, p$, we consider the empirical quantile function q_{j_0} associated to the variable X^{j_0}, so $q_{j_0}(\rho) = X^{j_0}_{\sigma([n\rho])}$ for $0 \leq \rho < 1$ and where $\sigma(.)$ is a function ordering the sample, *i.e.* $X^{j_0}_{\sigma(0)} \leq X^{j_0}_{\sigma(1)} \leq \ldots \leq X^{j_0}_{\sigma(n-1)}$. Then, the range of the stressed mean t will be in $[q_{j_0}(\alpha), q_{j_0}(1-\alpha)]$, where $\alpha \in (0, 1/2)$ (a typical value is 0.05).

We then tune $t - t_0$ as equal to $\epsilon_{j_0,\tau}$, where $\epsilon_{j_0,\tau} = \tau(t_0 - q_{j_0}(\alpha))$ if $\tau \in [-1, 0]$, and $\epsilon_{j_0,\tau} = \tau(q_{j_0}(1-\alpha) - t_0)$ if $\tau \in [0, 1]$. Parameter τ therefore allows to intuitively parametrize the level of stress whatever the distribution of the $\{X^{j_0}_1, \ldots, X^{j_0}_n\}$. More precisely, $\tau = 0$ yields no change of mean, $\tau = -1$ changes the mean from t_0 to the (small) quantile $q_{j_0}(\alpha)$ and $\tau = 1$ changes the mean from t_0 to the (large) quantile $q_{j_0}(1-\alpha)$.

For $j_0 = 1, \ldots, p$ and $\tau \in [-1, 1]$, we thus naturally suggest to compute

$$\mathrm{ER}_{j_0,\tau},\ \mathrm{FPR}_{j_0,\tau},\ \mathrm{TPR}_{j_0,\tau},\ \mathrm{P1}_{j_0,\tau},\ \mathrm{Pj}_{j_0,\tau} \tag{4}$$

that are defined as $\mathrm{ER}_t, \mathrm{FPR}_t, \mathrm{TPR}_t, \mathrm{P1}_t, \mathrm{Pj}_t$ in Sections 3.1 and 3.2, with $\Phi(X^1, \ldots, X^p, \hat{Y}, Y) = X^{j_0}$ and $t = t_0 + \epsilon_{j_0,\tau}$ as explained above. For a given τ, it makes sense to compare the indicators in (4) accros $j_0 = 1, \ldots, p$. For instance, one can plot $\mathrm{ER}_{j_0,\tau}$ as a function of τ for $\tau \in [-1, 1]$ and for each $j_0 \in \{1, \ldots, p\}$, as shown in Figure 1-(bottom). Remark that $\tau = 0$ corresponds to the algorithm performance baseline, without any perturbation of the test sample.

4 Results

In this section, we illustrate the use of the indices obtained using the entropic projection of samples on two classification cases: In subsection 4.1, the *Adult income* dataset[1] is considered, where X represents $n = 32000$ observations of dimension $p = 14$ and Y has 2 classes. Results of subsection 4.2 are obtained on the *MNIST* dataset[2], where X represents $n = 60000$ images of $p = 784$ pixels and Y has 10 classes. Note that the method accuracy is also assessed on synthetic data in Appendix D.2 and that the effect of 4 variables on the classification of the well known Iris dataset is shown in Appendix D.3. Importantly, the Python code to reproduce these experiments is freely available on GitHub[3].

4.1 Two class classification

In order to illustrate the performance of our procedure, we first consider the *Adult Income* dataset. It is made of about $n = 32000$ observations represented by

[1] https://archive.ics.uci.edu/ml/datasets/adult
[2] http://yann.lecun.com/exdb/mnist/
[3] https://github.com/XAI-ANITI/ethik

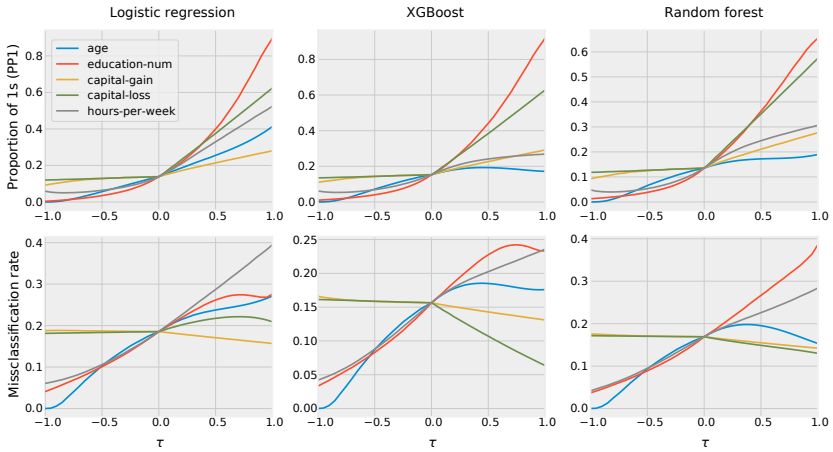

Figure 1: Results of Section 4.1 on the *Adult income* dataset. **(Top - PP1s)** Portion of predicted ones (*i.e.* High Incomes) with respect to the explanatory variable perturbation τ. **(Bottom - Mis. Rate)** Classification errors with respect to τ. There is no perturbation if $\tau = 0$. The larger (resp. the lower) τ, the larger (resp. the lower) the values of the selected explanatory variable.

$p = 14$ attributes (6 numeric and 9 categorical), each of them describing an adult. We specifically interpret the influence of 5 numeric variables on the categorical variable representing whether each adult has an income higher ($Y_i = 1$) or lower ($Y_i = 0$) than 50000$ per year.

We first trained three different classifiers (Logit Regression, XGboost and Random Forest[4]) on 25600 randomly chosen observations (80% of the whole dataset). We then performed the proposed sensitivity analysis strategy for each learned classifier on a test set made of the 6400 remaining observations. Detailed results are shown in Figure 1. Instead of only quantifying a score for each variable, we display in this figure the evolution of the algorithm confronted at gradually lower or higher values of τ (see Section 3.3) for each variable. The curves were computed using 21 regularly sampled values of τ between -1 and 1 with $\alpha = 0.05$. For each variable, the weighted observations were therefore stressed so that their mean is contained between the 0.05 and 0.95 quantiles of the original (non-weighted) values distribution in the test set. Note that for a quick and quantified overview of the variables response to a positive (resp. negative) stress, the user can simply interpret the difference of the response for $\tau = 1.$ and $\tau = 0$ (resp. $\tau = 0$ and $\tau = -1.$), as illustrated in Section 4.2 in the image case.

[4] *R* command *glm* and packages *xgboost* and *ranger*.

Influence of the variables in the decision rule We present in Figure 1 *(Top)* the role played by each variable in the portion of predicted ones (*i.e.* high incomes) for the *Logit Regression*, *XGboost* and *Random Forest* classifiers. The curves in Figure 1 *(Top)* highlight the role played by the variable *education-num*. The more educated the adult, the higher his/her income will be. The two variables *capital-gain* and *capital-loss* are also testimonial of high incomes since the adults having large incomes can obviously have more money than others on their bank accounts, or may easily contract debts, although the contrary is not true. It is worth pointing out the role played by the age variable which appears clearly in the figure: young adults earn increasingly large incomes with time, which is well captured by the decision rules (left part of the red curves). For τ higher than about 0.25 (corresponding to 34 years old), being increasingly old is however not captured by the three tested decision rules to be related to higher incomes.

We emphasize that these curves enable to intuitively interpret the complex trends captured by *black-box* decision rules. They indeed quantify non-linear effects of the variables and very different behaviors depending on whether the variables increase or decrease. In other strategies designed to explain black-box decision rules, explainability is obtained by using the global feature importance indexes included in the decision rules. The *Feature Weights* count the number of times a feature appears in a classifier obtained by combining several classifiers (*e.g.* an ensemble of trees). For tree based methods, the *Gain* counts the average gain of splits using the feature, while the *Coverage* is based on the average coverage (number of samples affected) of splits using the feature. Implemented algorithms in Python or R enable to view these features importance, but they often contradict themselves as already quoted by several authors (see for instance in LIME Ribeiro et al. [2016]). Contrary to algorithms that study the influence of a variable by computing information theory criterion between different outputs of the algorithm for changes in the variables (see in Skater Kramer and Choudhary [2018]), the variable changes we use are plausible since the stressed variables have distribution that are as close as possible to the initial distribution of true variables. Finally, we work directly on the real black-box model and do not approximate it by any surrogate model, as in Ribeiro et al. [2016].

Influence of the variables in the accuracy of the classifier Besides the influence of the variables on the algorithm outcome, it is worth studying their influence on the accuracy or veracity of the model. We then present in Figure 1 *(Bottom)* the evolution of the classification error when each variable is stressed by τ. The three sub-figures (one for each prediction model) represents the evolution of the error confronted to the same modification of each variables. The error of the method on the original data is obtained for $\tau = 0$. Such curves point out which variables are the most sensitive to increasing prediction flaws. Such result may be used to temper the trust in the forecast depending on the values of the variables.

As previously, the curves appear as more informative than single scores: The

three models enable to select the same couple of variables that are important for the accuracy of the prediction when they increase *i.e.* education number and numbers of hours worked pro week. The latter makes the prediction task the most difficult when it is increased. Indeed, the people working a large number of hours per week may not always increase their income, since it relies on different factors. People with high income however usually work a large number of weekly hours. Hence, these two variables play an important role in the prediction and their changes impact the prediction error. In the same flavor, more insight on the error terms could be obtained by dealing with the evolution of the False Positive Rate and True Positive Rate as presented in Appendix D.4.

4.2 Image classification

We now measure the influence of pixel intensities in image recognition tasks. Each pixel intensity is treated as a variable and the stress is used to saturate the intensities towards one side of their spectrum (red if $\tau = 1$ or blue if $\tau = -1$). We specifically trained a CNN on the MNIST dataset using a typical architecture that can be found on the Keras documentation[5]. The CNN was trained on a set of 60,000 images whilst the predictions were made on another 10,000 images. It achieved a test set accuracy north of 99%. For each of the 784 pixels j_0, we computed the $\{\lambda_i^{(j_0,\tau)}\}_{i=1,\dots,10000}$ in the cases where τ equals -1 as well as 1, using the method of Section 2. The prediction proportions of each of the 10 digits was then computed using the method of Section 3.2. The whole process took around 9 seconds to run on a modern laptop (Intel Core i7-8550U CPU @ 1.80GHz, 24GB RAM) running Linux. The results are presented in Figure 2-(top).

The color of each pixel in Figure 2-(top) represents its contribution towards the prediction of each digit. For example, a value of 0.15 means that the CNN predicts on average this digit 15% more often when the associated pixel is activated ($\tau = 1$) instead of having the background intensity ($\tau = -1$). Although our method is pixel-based, it is still able to uncover regions which the CNN uses to predict each digit. Likewise, redder regions contain pixels that are positively correlated with each digit. Note that the edges of each image don't change color because the corresponding pixels have no impact whatsoever on the predictions. The left part of number 5 has pixels in common with number 6. However, we are able to see that the CNN identifies 6s by using the bottom part of the 6, more so than the top stroke which it uses to recognize 5s. Likewise, according to the CNN, the most distinguishing part of number 9 is the part that links the top ring with the bottom stroke.

We finally emphasize the main difference between our strategy and the two popular interpretability solutions LIME (Ribeiro et al. [2016])[6] and SHAP (Lundberg and Lee [2017])[7]: we work on whole test sets while these solutions interpret the variables (here pixels) influence when predicting specific labels in *individual observations*. As an illustration, Figure 2-(bottom-left) represents the

[5] https://keras.io/examples/mnist_cnn
[6] https://github.com/marcotcr/lime
[7] https://github.com/slundberg/shap

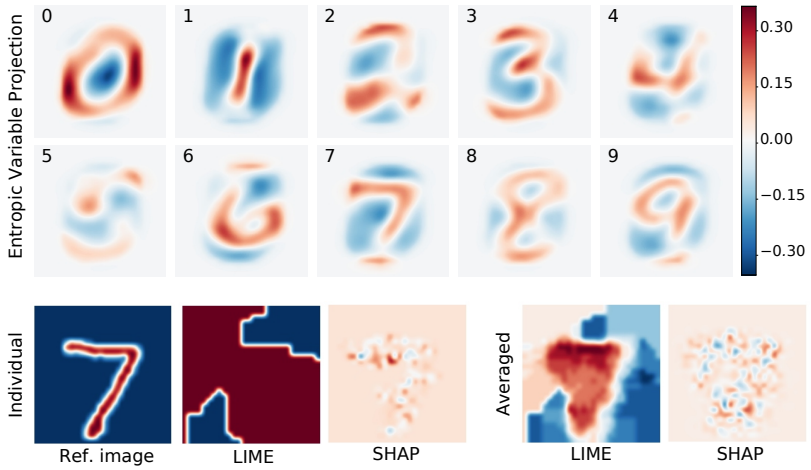

Figure 2: **(top)** Pixel contributions towards each digit according to our Entropic Variable Projection method. **(bottom-left)** Pixel contributions to predict seven in an individual image representing a seven, using the LIME and SHAP packages. **(bottom-right)** Average pixel contributions to predict seven in all images of the MNIST test set representing a seven, using the LIME and SHAP packages.

most influential pixels found with LIME and SHAP to predict a seven in an image of the MNIST test set representing a seven. To draw similar interpretations as those made on Figure 2-(top), one can represent the average results obtained by using LIME or SHAP over all images of the MNIST test set representing a seven, as represented in Figure 2-(bottom-left). Note that the computations required take about 7 and 10 hours using LIME and SHAP, respectively, which is much longer than when using our method (10 seconds). Averaged results are also less resolved for LIME and harder to interpret for SHAP. Our method can also natively compute other properties of the black-box decision rules with a negligible additional computational cost, as described in Section 3.

5 Conclusion

Explainability of *black-box* decision rules in the machine learning paradigm has many interpretations and has been tackled in a large variety of contributions. Here, we focused on the analysis of the variables importance and sensitivity and their impact on a decision rule, inspired by the field of computer code experiments. When building a surface response in computer code experiments, the prediction algorithm is applied to new entries to explore its possible outcomes. In a Machine Learning problem, the context is quite different since the test input variables must follow the distribution of the learning sample. Therefore, evaluating the

decision rule at all possible points does not make any sense. To cope with this issue, we have proposed an information theory procedure to stress the original variables without losing the information conveyed by the initial distribution. We proved that this solution amounts in re-weighting the observations of the test sample, leading to very fast computations and the construction of new indices to make clear the role played by each variable.

Remark that our strategy can be seen as a *what if* tool, as counterfactual methods Wachter et al. [2018]. It indeed explains model decisions by quantifying how their outputs change when the machine learning data are transformed. Nevertheless, existing counterfactual methods substitute individual counterfactual observations for individual baseline observations. In contrast, our strategy substitutes counterfactual data sets, with new variable distribution characteristics, for the original data set.

The first key advantage of this strategy is to preserve as much as possible the distribution of the test set $(X_i^1, \ldots, X_i^p, \hat{Y}_i, Y_i)$, $i = 1, \ldots, n$ and thus preserving the correlations of the input variables that have then an impact on the indicators computed by using our procedure. In contrasts with other interpretability paradigms such as the PAC learning framework Valiant [2013], we do not create artificial outliers. Its second key advantage is that, for a given perturbation, the weights are obtained by minimizing a convex function, for which the evaluation cost is $\mathcal{O}(n)$. The total cost is then $\mathcal{O}(np)$ for studying the impact of each of the p variables (see Appendix D.1) and there is no need to generate new data, nor even to compute new predictions from the black box algorithm, which is particularly costly if n or p is large. Our procedure therefore scales particularly well to large datasets as *e.g.* real-world image databases. Finally, the flexibility of the entropic variable projection procedure enables to study the response to various types of stress on the input variables (not only their mean but also their variability, joint correlations, ...) and thus to interpret the decision rules encountered in a wide range of applications encountered in the field of Machine Learning.

References

Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. The elements of statistical learning: data mining, inference, and prediction, 2009.

Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 241–250. ACM, 2000.

Mark Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, November 27-30, 1995*, pages 24–30, 1995.

Mary T Dzindolet, Scott A Peterson, Regina A Pomranky, Linda G Pierce, and Hall P Beck. The role of trust in automation reliance. *International journal of human-computer studies*, 58(6):697–718, 2003.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 150–158, 2012. ISBN 978-1-4503-1462-6.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11(Jun):1803–1831, 2010.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 1721–1730, 2015.

Zachary C. Lipton. The mythos of model interpretability. In *Proceedings of the ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, pages 96–100, 2016.

Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 2017.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.

Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. A benchmark for interpretability methods in deep neural networks. In *NeurIPS*, 2019.

Patrick Hall, Navdeep Gill, and Mark Chan. Practical techniques for interpreting machine learning models: Introductory open source examples using python, h2o, and xgboost, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos 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.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard journal of law & technology*, 31:841–887, 04 2018.

Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2376–2384, 2019.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML17, page 33193328, 2017.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. In *Advances in Neural Information Processing Systems 30*, pages 4066–4076. 2017.

Emily Black, Samuel Yeom, and Matt Fredrikson. Fliptest: fairness testing via optimal transport. In *FAT* '20: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 111–121. 2020.

Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. On relating explanations and adversarial examples. In *Advances in Neural Information Processing Systems 32*, pages 15857–15867. 2019.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, 2017.

Paul Lemaître, Ekatarina Sergienko, Aurélie Arnaud, Nicolas Bousquet, Fabrice Gamboa, and Bertrand Iooss. Density modification-based reliability sensitivity analysis. *Journal of Statistical Computation and Simulation*, 85(6):1200–1223, 2015.

Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4765–4774, 2017.

Peter Bühlmann and Sara Van De Geer. *Statistics for High-Dimensional Data*. Springer, 2011.

Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41 (1):77–93, 2004.

Aaron Kramer and Pramit Choudhary. Model Interpretation with Skater. `https://oracle.github.io/Skater/`, 2018. [Online; accessed 28-Jan-2020].

Leslie Valiant. *Probably Approximately Correct: Nature's Algorithms for Learning and Prospering in a Complex World*. Basic Books (AZ), 2013.

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, pages 146–158, 1975.

Imre Csiszár. Sanov property, generalized I-projection and a conditional limit theorem. *The Annals of Probability*, pages 768–793, 1984.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.

Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

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

Appendix

A Proofs of the main results

The proofs rely on the following theorem, that is a simplified version of the Theorems in Csiszár [1975] and in Csiszár [1984].

Theorem A.1. *Let $(E, \mathcal{B}(E))$ be a measurable space and Q a probability measure on E. Consider $t \in \mathbb{R}^k$ and a measurable function $\Phi : E \to \mathbb{R}^k$. We assume that, for $v \in \mathbb{R}^k$, $Q(\{x \in E; \langle v, \Phi(x)\rangle = 0\}) = 1$ if and only if $v = 0$. Let $\mathbb{P}_{\Phi,t}$ be the set of all probability measures P on $(E, \mathcal{B}(E))$ such that $\int_E \Phi(x)\,\mathrm{d}P(x) = t$. Assume that $\mathbb{P}_{\Phi,t}$ contains a probability measure that is mutually absolutely continuous with respect to Q.*

For a vector $\xi \in \mathbb{R}^k$, let $Z(\xi) := \int_E e^{\langle \xi, \Phi(x)\rangle}\,\mathrm{d}Q(x)$. We assume that the set on which Z is finite is open. Define now $\xi(t)$ as the unique minimizer of the strictly convex function $H(\xi) := \log Z(\xi) - \langle \xi, t\rangle$. Then,

$$Q_t := \operatorname*{arginf}_{P \in \mathbb{P}_{\Phi,t}} \mathrm{KL}(P, Q) \tag{5}$$

exists and is unique. Furthermore it can be computed as

$$Q_t = \frac{\exp\langle \xi(t), \Phi\rangle}{Z(\xi(t))} Q.$$

Proof of Theorem 2.1 We will apply Theorem A.1 with $E = \mathbb{R}^{p+2}$ and $Q = Q_n$. Because of the assumption that t can be written as a convex combination of $\Phi(X_1, \hat{Y}_1, Y_1), \ldots, \Phi(X_n, \hat{Y}_n, Y_n)$, with positive weights, we have that $\mathbb{P}_{\Phi,t}$ in Theorem A.1 contains a probability measure that is mutually absolutely continuous with respect to Q. Furthermore, we have assumed that $\operatorname{span}\Big(\Phi(X_1, \hat{Y}_1, Y_1), \ldots, \Phi(X_n, \hat{Y}_n, Y_n)\Big) = \mathbb{R}^k$, which means that for any $v \in \mathbb{R}^k$, $\langle v, \Phi(X_1, \hat{Y}_1, Y_1)\rangle, \ldots, \langle v, \Phi(X_n, \hat{Y}_n, Y_n)\rangle$ are not all equal to zero. This implies that $Q_n(\{x \in \mathbb{R}^{p+2}; \langle v, \Phi(x)\rangle = 0\}) = 1$ if and only if $v = 0$. Hence, all the assumptions of Theorem A.1 are satisfied.

We have, starting from the notation of Theorem A.1,

$$\int_E e^{\langle \xi, \Phi\rangle}\,\mathrm{d}Q(x) = \frac{1}{n}\sum_{i=1}^n e^{\langle \xi, \Phi(X_i, \hat{Y}_i, Y_i)\rangle}$$

and thus the definitions of $Z(\xi)$ in Theorems A.1 and 2.1 indeed coincide. Hence, also the definitions of $\xi(t)$ in Theorems A.1 and 2.1 coincide. Hence, we have

$$\begin{aligned} Q_t &= \frac{\exp\langle \xi(t), \Phi\rangle}{Z(\xi(t))} Q \\ &= \frac{1}{\frac{1}{n}\sum_{i=1}^n e^{\langle \xi(t), \Phi(X_i, \hat{Y}_i, Y_i)\rangle}} \\ &\quad \frac{1}{n}\sum_{i=1}^n \exp\Big(\langle \xi(t), \Phi(X_i, \hat{Y}_i, Y_i)\rangle\Big)\,\delta_{X_i, \hat{Y}_i, Y_i} \\ &= \frac{1}{n}\sum_{i=1}^n \lambda_i^{(t)} \delta_{X_i, \hat{Y}_i, Y_i}. \end{aligned}$$

This concludes the proof. □

Proof of Theorem 2.2 The proof of this theorem comes from Theorem 2.1, by considering $\Phi : \mathbb{R}^{p+2} \to \mathbb{R}$ defined by $\Phi(X^1, \ldots, X^p, \hat{Y}, Y) = X^{j_0}$ and by considering the same $t \in \mathbb{R}$ in Theorems 2.1 and 2.2. We have assumed that $\min_{i=1}^n X_i^{j_0} < t < \max_{i=1}^n X_i^{j_0}$. Hence, t can be written as a convex combination of $X_1^{j_0}, \ldots, X_n^{j_0}$ with positive weights. Furthermore, $X_1^{j_0}, \ldots, X_n^{j_0}$ are not all equal and thus span $\left(X_1^{j_0}, \ldots, X_n^{j_0}\right) = \mathbb{R}$. Hence the conditions of Theorem 2.1 hold and the conclusion of this theorem directly provides Theorem 2.2. □

Proof of Proposition 2.3 With the choice of Φ of the proposition and with the assumptions there, the conditions of Theorem 2.1 hold. Hence the conclusion of this theorem proves the proposition. □

Proof of Proposition 2.4 With the choice of Φ of the proposition and with the assumptions there, the conditions of Theorem 2.1 hold. Hence the conclusion of this theorem proves the proposition, since $\left(\mathbb{E}(X^{j_0}) = m_{j_0}, \mathrm{Var}(X^{j_0}) = v\right)$ is equivalent to $\left(\mathbb{E}(X^{j_0}) = m_{j_0}, \mathbb{E}((X^{j_0})^2) = m_{j_0}^2 + v\right)$. □

Proof of Proposition 2.5 With the choice of Φ of the proposition and with the assumptions there, the conditions of Theorem 2.1 hold. Hence the conclusion of this theorem proves the proposition, since $\left(\mathbb{E}(X^{j_1}) = m_{j_1}, \mathbb{E}(X^{j_2}) = m_{j_2}, \mathrm{Cov}(X^{j_1}, X^{j_2}) = c\right)$ is equivalent to $\mathbb{E}(X^{j_1}) = m_{j_1}, \mathbb{E}(X^{j_2}) = m_{j_2}, \mathbb{E}(X^{j_1} X^{j_2}) = m_{j_1} m_{j_2} + c$. □

Proof of Proposition 2.6 The existence and unicity of $Q_t^\star$ follows from Theorem A.1. Also from this theorem, $Q_t^\star$ is of the form

$$\mathrm{d}Q_t^\star(x) = e^{\langle \xi^\star(t), \Phi(x) \rangle - \log(Z^\star(\xi^\star(t)))} \, \mathrm{d}Q^\star(x),$$

where $\xi^\star(t)$ is the minimizer of the strictly convex function

$$\xi \mapsto \log(Z^\star(\xi)) - \langle \xi, t \rangle$$

with

$$Z^\star(\xi) = \int_{\mathbb{R}^{p+2}} e^{\langle \xi, \Phi(x) \rangle} \, \mathrm{d}Q^\star(x).$$

We also recall from Theorem 2.1 that Q_t is of the form

$$\mathrm{d}Q_t(x) = e^{\langle \xi(t), \Phi(x) \rangle - \log(Z(\xi(t)))} \, \mathrm{d}Q_n(x),$$

where $\xi(t)$ is the minimizer of the strictly convex function

$$\xi \mapsto \log(Z(\xi)) - \langle \xi, t \rangle$$

with

$$Z(\xi) = \int_{\mathbb{R}^{p+2}} e^{\langle \xi, \Phi(x) \rangle} \, \mathrm{d}Q_n(x).$$

For $\epsilon > 0$, let $F_\epsilon = \{\xi \in \mathbb{R}^k; ||\xi - \xi^\star(t)|| = \epsilon\}$. Then $\inf_{\xi \in F_\epsilon} \log(Z^\star(\xi)) - \langle \xi, t \rangle > \log(Z^\star(\xi^\star(t))) - \langle \xi^\star(t), t \rangle$ by strict convexity. Furthermore, from the boundedness assumptions the quantities $\log(Z^\star(\xi)) - \langle \xi, t \rangle$, $\log(Z(\xi)) - \langle \xi, t \rangle$ and their gradients w.r.t. ξ are bounded in absolute value uniformly over $\xi \in F_\epsilon$. It follows that almost surely as $n \to \infty$, $\inf_{\xi \in F_\epsilon} \log(Z(\xi)) - \langle \xi, t \rangle > \log(Z(\xi^\star(t))) - \langle \xi^\star(t), t \rangle$. Hence by strict convexity, almost surely $||\xi(t) - \xi^\star(t)|| < \epsilon$ for n large enough. Hence $\xi(t) \to \xi^\star(t)$ almost surely.

Let now $\mathcal{F} = \{f : \mathbb{R}^{p+2} \to \mathbb{R}; \text{f is 1-Lipshitz}, f(0) = 0\}$. We have

$$\sup_{f\in\mathcal{F}} \left| \int_{\mathbb{R}^{p+2}} f(x)\, \mathrm{d}Q_t^\star(x) - \int_{\mathbb{R}^{p+2}} f(x)\, \mathrm{d}Q_t(x) \right|$$

$$\leq \sup_{f\in\mathcal{F}} \int_{\mathbb{R}^{p+2}} f(x)$$

$$\left| e^{\langle \xi^\star(t), \Phi(x)\rangle - \log(Z^\star(\xi^\star(t)))} - e^{\langle \xi(t), \Phi(x)\rangle - \log(Z(\xi(t)))} \right| \mathrm{d}Q^\star(x) \tag{6}$$

$$+ \sup_{f\in\mathcal{F}} \left| \int_{\mathbb{R}^{p+2}} f(x) e^{\langle \xi(t), \Phi(x)\rangle - \log(Z(\xi(t)))} \left(\mathrm{d}Q^\star(x) - \mathrm{d}Q_n(x)\right) \right|. \tag{7}$$

In (6) the term in the absolute value goes to zero almost surely and uniformly over the support of $Q^\star$, since $\xi(t) \to \xi^\star(t)$ and from our boundedness assumptions. Hence, the supremum over $f \in \mathcal{F}$ of the integral in (6) goes to zero almost surely, since also the functions in $\mathcal{F}$ are bounded uniformly on the support of $Q^\star$. In (7), the function that is integrated is uniformly bounded, with uniformly bounded Lipshitz norm, from our boundedness asumptions and since $f \in \mathcal{F}$. Also, from for instance Fournier and Guillin [2015], the L^1 Wasserstein distance between Q_n and $Q^\star$ goes to zero almost surely. This implies that the supremum over $f \in \mathcal{F}$ in (7) also goes to zero almost surely (see for instance Villani [2008] for the link between the Wasserstein distance and differences of expectations of Lipshitz functions). Hence we have proved that almost surely as $n \to \infty$,

$$\sup_{f\in\mathcal{F}} \left| \int_{\mathbb{R}^{p+2}} f(x)\, \mathrm{d}Q_t^\star(x) - \int_{\mathbb{R}^{p+2}} f(x)\, \mathrm{d}Q_t(x) \right| \to 0.$$

From for instance Villani [2008], this implies that Q_t converges weakly to $Q_t^\star$ almost surely as $n \to \infty$. □

B Gradient of the objective function in Theorem 2.1

Let us denote $v_i = (X_i, \hat{Y}_i, Y_i)$ for $i = 1, \ldots, n$. In Theorem 2.1 we want to minimize, over $\xi \in \mathbb{R}^k$,

$$H(\xi) = \log\left(\frac{1}{n}\sum_{i=1}^{n} \exp\left(\langle \xi, \Phi(v_i)\rangle\right)\right) - \langle \xi, t\rangle. \tag{8}$$

The gradient of Eq. (8) is:

$$\nabla_\xi H(\xi) = \frac{\sum_{i=1}^n \Phi(v_i) \exp\langle \xi, \Phi(v_i)\rangle}{\sum_{i=1}^n \exp\langle \xi, \Phi(v_i)\rangle} - t\,, \tag{9}$$

which makes it possible to compute $\xi(t)$ in Theorem 2.1 using gradient based optimization methods.

C Extension to the regression case

C.1 Methodology

As an extension to Section 3, we consider now the case of a real valued regression where $Y_i, f(X_i) \in \mathbb{R}$ for $i = 1 \ldots, n$. In order to understand the effects of each variable, first

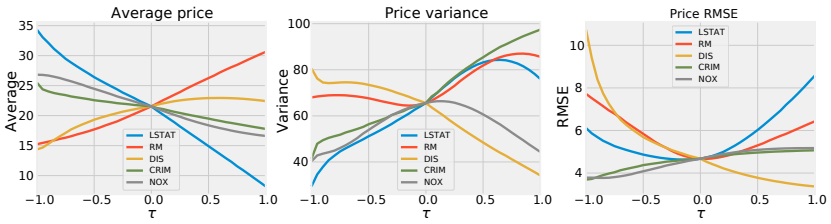

Figure 3: Results obtained on the *Boston Housing* dataset with Random Forests. The explanatory variable perturbation τ has the same signification as in Figure 1.

we consider the mean criterion

$$\mathrm{M}_{i_0,\tau} = \frac{1}{n}\sum_{i=1}^{n} \lambda_i^{(i_0,\tau)} f(X_i),$$

which will indicate how a change in the variable will modify the output of the learned regression (τ is explained in Section 3.3). Second the variance criterion

$$\mathrm{V}_{i_0,\tau} = \frac{1}{n}\sum_{i=1}^{n} \lambda_i^{(i_0,\tau)} \left(f(X_i) - \mathrm{M}_{i_0,\tau}\right)^2$$

is meant to study the stability of the regression with respect to the perturbation of the variables. Finally the root mean square error (RMSE) criterion

$$\mathrm{RMSE}_{i_0,\tau} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \lambda_i^{(i_0,\tau)} \left(f(X_i) - Y_i\right)^2}$$

is analogous to the classification error criterion since it enables to detect possibly misleading or confusing variables when learning the regression.
For each $i_0 \in \{1, \ldots, p\}$, these three criteria can be plotted as a function of τ for $\tau \in [-1, 1]$.

C.2 Application

We use now our strategy on the Boston Housing dataset[8]. This dataset deals with houses prices in Boston. It contains 506 observations with 13 variables that can be used to predict the price of the house to be sold. When considering an optimized Random Forest algorithm, the importance calculated as described in Breiman [2001], enables to select the 5 most important variables as follows: *lstat* (15227), *rm* (14852), *dis* (2413), *crim* (2144) and *nox* (2042). Remark that the coefficients obtained using a linear model would lead to similar interpretations, with the the 5 most important variables as follows: *lstat* (-3.74), *dis* (-3.10), *rm* (2.67), *rad* (2.66), *tax* (-2.07)

As shown in Figure 3, our analysis goes further than this score. In particular we point out the non linear influence of the variables depending whether they are high or low. For instance the average number of rooms in a house (variable *rm*) is an

[8] https://www.kaggle.com/c/boston-housing

$Mean_0 - Mean_{-0.5}$
black (4.1)
rm (3.0)
dis (1.7)
zn (0.85)
...
age (-2.78)
indus (-3.2)
ptratio (-3.8)
lstat (-5.1)

$Mean_{0.5} - Mean_0$
rm (6.80)
zn (4.60)
chas (2.74)
dis (1.64)
...
rad (-2.99)
indus (-3.05)
tax (-3.18)
lstat (-5.26)

Table 1: Most responsive variables to a positive or negative stress τ when estimating House prices. Scores are shown between brackets and computed as the difference of the *Mean* curves of Figure 3 for **(left)** $\tau = -0.5$ and $\tau = 0$, and **(right)** $\tau = 0$ and $\tau = 0.5$.

important factor that makes the price increase in the case of large houses ($\tau > 0.$ in Figure 3 *(Average)*). Interestingly, this is far less the case for smaller houses ($\tau < 0.$ in Figure 3 *(Average)*) since there are other arguments than the number of rooms to keep a high price in this case.

Note that when the number of variables is large, the presence of too many curves may make the graph difficult to understand. In this cases, scores that represent average individual evolutions on given ranges of τ values for each variables can be computed. Then the highest and lowest scores can be represented as the most influential variables on the predictions. For instance, we represent in Table. 1 the evolution of the Mean curves in Figure 3 between $\tau = -0.5$ and $\tau = 0$, as well as between $\tau = 0$ and $\tau = 0.5$, which makes clearly understandable which are the most influential variables. It is important to remark that our methodology still allows that the learned decision rules won't be mainly influenced by the same variables depending on whether it the variable increases ($\tau > 0$) or decreases ($\tau < 0$). In Table. 1, the more influential variables are indeed *rm*, *lstat* and *zn* in the positive direction, while in the negative direction, the variables are *lstat*, *black* and *pratio*. Note that such variables are also cited in studies that relies on LIME Ribeiro et al. [2016] or SHAP Lundberg and Lee [2017] packages, but the curves we present are more informative and relies on the same distributional input.

D Additional results in the Classification case

D.1 Evaluation of the computational burden

We explained in Section 5 that our strategy only optimizes, for each of the p variables, a function which evaluation cost is $\mathcal{O}(n)$ with no additional outputs predictions out of the *black box* machine learning algorithm. To quantify this, we show in Table 2 the computational times dedicated to the analysis of synthetic datasets having a different amount of variables p and observations n. The variables interpretation was made using

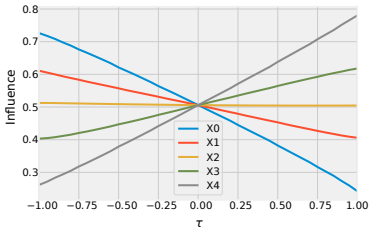

Figure 4: Proportion of ones found on synthetic data generated using a logistic regression model

21 values of τ, leading to curves as *e.g.* in Figure 1. Computations were run with Python on a standard Intel Core i7 laptop with 24GB memory and no parallelization. It appears that our strategy indeed has a $\mathcal{O}(np)$ cost, so we then believe it may have a high impact to study the rules learned by black-box machine learning algorithms on large real-life datasets. Remark that when interpreting the influence of the pixel intensities on image test sets, as in Figure 2, only 3 values of τ are used. The computations are therefore about 7 times faster. This coherent with the 10 seconds required on 10000 MNIST images of 28×28 pixels in Section 4.2. Note finally that a preliminary implementation of our method in R has lead to very similar results.

p	n	time (sec)
10	10000	0.76
100	10000	7.79
1000	10000	82.5
10	100000	7.93
10	1000000	86.0

Table 2: Computational times required on synthetic datasets, where 21 levels of stress (τ) were computed on each of the p variables.

D.2 Results on simulated data

In order to further show that our procedure is able to properly recover the characteristics of machine learning algorithms, we again tested it on synthetic data. We have run an experiment with $p = 5$ variables and $n = 10^6$ observations, where synthetic data are generated using a logistic regression model, with independent regressors and coefficient vector equal to $(-4, 2, 0, 2, 4)$. Figure 4 clearly shows that our method enables to recover the signs and the hierarchy of the coefficients.

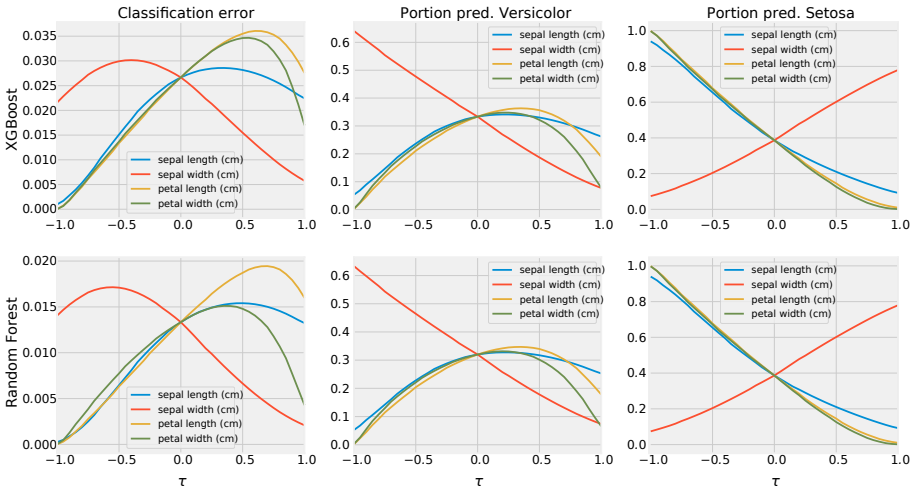

Figure 5: Evaluation of the classification error and the prediction with respect to the explanatory variable perturbation τ, on the *Iris* dataset. The quantity τ and the lines have the same signification as in the main part. **(Top)** XGBoost Model. The sepal width enables to differenciate the *Setosa* class. **(Bottom)** Random Forest Model. The sepal width again enables to differenciate the *Setosa* class.

D.3 Results on the Iris dataset

As an additional assesment of the method on very well known and simple data, we now consider the *Iris* dataset[9]. This dataset is composed of 150 observations with 4 variables used to predict a label into three categories: *setosa*, *versicolor*, *virginica*. To predict the labels, we used an *Extreme Gradient Boosting* model and a *Random Forest* classifier. Results are show in Figure 5. We first present for both models the Classification error. Then the two other subfigures show the effects of increasing or decreasing the 4 parameters, i.e the width or the length of the sepal or petal is shown for all classes. As expected, we recover the well known result that the width of the sepal is the main parameter which enables to differentiate the class *Setosa* while the differentiation between the two other remaining classes is less obvious.

D.4 Other indices: ROC Curves

In the case of two class classification on the Adult Income dataset (Section 4.1), we have shown the evolution of the classification error when the stress parameter τ increases. Such results can straightforwardly be extended to True and False Positive Rates, which are commonly represented in ROC curves, that we display in Figure 6. Each point of these curves corresponds to the False Positive Rate and the True Positive Rate, for a sample drawn for each τ and each variable. All curves cross at the same point which

[9] https://archive.ics.uci.edu/ml/datasets/iris

Figure 6: Evolution of Roc Curves in the *Adult income* dataset (Section 4.1). As for the classification errors, we observe that large values of the variable *hoursWeek* make the classification difficult.

corresponds to $\tau = 0$. It therefore becomes possible to study the evolution of each criterion.