Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-022-00878-5

Stable and actionable explanations of black-box models through factual and counterfactual rules

Riccardo Guidotti[1] · Anna Monreale[1] · Salvatore Ruggieri[1] · Francesca Naretto[2] · Franco Turini[1] · Dino Pedreschi[1] · Fosca Giannotti[2]

Received: 28 March 2021 / Accepted: 9 September 2022

Abstract

Recent years have witnessed the rise of accurate but obscure classification models that hide the logic of their internal decision processes. Explaining the decision taken by a black-box classifier on a specific input instance is therefore of striking interest. We propose a local rule-based model-agnostic explanation method providing stable and actionable explanations. An explanation consists of a factual logic rule, stating the reasons for the black-box decision, and a set of actionable counterfactual logic rules, proactively suggesting the changes in the instance that lead to a different outcome. Explanations are computed from a decision tree that mimics the behavior of the black-box locally to the instance to explain. The decision tree is obtained through a bagging-like approach that favors stability and fidelity: first, an ensemble of decision trees is learned from neighborhoods of the instance under investigation; then, the ensemble is merged into a single decision tree. Neighbor instances are synthetically generated through a genetic algorithm whose fitness function is driven by the black-box behavior. Experiments show that the proposed method advances the state-of-the-art towards a comprehensive approach that successfully covers stability and actionability of factual and counterfactual explanations.

Keywords Explainable AI · Local explanations · Model-agnostic explanations · Rule-based explanations · Counterfactuals

Responsible editor: Martin Atzmüller.

✉ Riccardo Guidotti
riccardo.guidotti@unipi.it

[1] Department of Computer Science, University of Pisa, Largo B. Pontecorvo, 3, 56127 Pisa, PI, Italy

[2] Scuola Normale Superiore, Piazza dei Cavalieri, 7, 56126 Pisa, PI, Italy

Published online: 14 November 2022

1 Introduction

Explaining the decisions of black-box classifiers is one of the the principal obstacles to the acceptance and trust of applications based on Artificial Intelligence (AI) (Li et al. 2022; Miller 2019). Magazines and newspapers are full of commentaries about AI systems taking critical decisions that heavily impact on our life and society, from loan concession in bank systems to pedestrian detection in self-driving cars. The worry is not only due to the increasing automation of AI decision making, but mostly to the fact that the underlying algorithms are opaque and their logic unexplained (Pasquale 2015). The leading cause for this lack of transparency is that the process of inferring a classification model from examples cannot be fully controlled because the size of the training data and the complexity of such a process are too big for humans (Freitas 2013). It is a paradoxical situation in which, on one side, the legislator defines new regulations requiring that automated decisions should be explained[1] while, on the other side, more and more sophisticated and obscure algorithms for decision making are designed (Malgieri and Comandé 2017; Pedreschi et al. 2019). The lack of transparency in machine learning models grants to them the power to perpetuate or reinforce forms of injustice by learning bad habits from the data. In fact, if the training data contains biased decision records, it is likely that the resulting model inherits the biases and recommends discriminatory or simply wrong decisions (Ntoutsi et al. 2020; Berk et al. 2018). For these reasons, there has recently been a flourishing of proposals for explaining classification models (Li et al. 2022; Guidotti et al. 2019d). The spectrum of approaches ranges from explaining the whole decision logic of a model (*global approaches*), to explaining its decision on a specific input instance (*local approaches*), and from assuming no information on the model (*model-agnostic approaches*) to assuming the model is of a specific type (*model-specific approaches*). A radically different direction aims at developing new models and new inference algorithms that are interpretable by-design (Rudin 2019). This last line of research is very promising and aims at redefining the entire panorama of machine learning methods making them natively transparent; however, it is still in its infancy, while opaque AI systems are already in usage. For this reason, we firmly believe that it is urgent to have stable post-hoc “explanators” covering current machine learning technology.

The objective of this paper is to explain the decisions taken by an obscure black-box classifier on specific input instances by providing *meaningful* and *stable* explanations of the logic involved. We aim at a *model-agnostic* method, disregarding the black-box internals and learning process, that works analyzing the input-output behavior of the black-box *locally*, i.e., in the neighborhood of the instance to explain. We perform our research under some specific assumptions. First, we assume that the vehicle for offering explanations should be as close as possible to a language of formal reasoning, such as propositional logic. Thus, we are also assuming that the user can understand the semantics of elementary logic rules, as taught in secondary schools or undergraduate courses. Second, we assume that an explanation is interesting if it answers: (i) the factual question of *why* a specific decision concerning to a user has been made; (ii)

[1] We refer to the “right to explanation" established in the European General Data Protection Regulation (GDPR), entered into force in 2018.

as well as the counterfactual question of *what* conditions would change the black-box decision. Third, we assume that the black-box system can be queried as many times as necessary, to probe its decision behavior to the scope of reconstructing its logic.[2]

Resorting to logic rules is a step towards *comprehensibility* of the explanations, but it is not enough for achieving meaningful explanations. First, the reconstruction logic of the black-box in the neighborhood of the instance to explain should be consistent with the black-box decisions, a property known as *fidelity* (Freitas 2013). In particular, the factual rules should have high precision in characterizing conditions for a specific black-box decision. Second, the counterfactual answer should consists of a minimal number of changes to the feature values of the instance to explain (*minimality*), and such changes should allow for actionable recourse, a property known as *actionability* (Venkatasubramanian and Alfano 2020; Karimi et al. 2020). Third, the generation of explanations should guarantee *stability* of its output against possible local perturbations of the input (Alvarez-Melis and Jaakkola 2018). This is crucial for local approaches, which rely on some randomness in neighborhood generation. Fourth, the approach should be general enough to encompass not only tabular data but also images, texts and multi-label data (*generality*).

We aim at advancing state-of-the-art approaches, including our previous work LORE (Guidotti et al. 2019b), to a comprehensive proposal that is able to extend the coverage of comprehensibility, fidelity, minimality and generality, by also dealing with stability and actionability. We propose LORE$_{sa}$, a Stable and Actionable LOcal Rule-based Explanation method extending LORE. Given a black-box predictor b and a specific instance x labeled with outcome y by b, LORE$_{sa}$ builds a simple, interpretable local decision tree predictor by first generating an ensemble of balanced sets of neighbor instances of x through an ad-hoc genetic algorithm, then extracting from each set a decision tree classifier, and finally merging the ensemble of decision trees in a single decision tree classifier. A *(counter)factual explanation* is then extracted from the obtained decision tree which locally approximates the behavior of the black-box around x. The (counter)factual explanation is a pair composed by (i) a—factual—*logic rule*, corresponding to the path in the tree that explains why x has been labelled as y by b, and (ii) a set of *counterfactual rules*, explaining which changes in x would invert the class y assigned by b. For example, for an instance from the `compas` dataset (Berk et al. 2018) we may have as explanation the rule $\{age \leq 39, race = African\text{-}American, is\_recid\} \rightarrow High\text{-}Risk$, and the counterfactual rules $\{age > 40\} \rightarrow Low\text{-}Risk$ and $\{race = Caucasian\} \rightarrow Low\text{-}Risk$. The factual explanation is that the high risk of recidivism is predicted for a black younger than 40 with prior recidivism; the counterfactuals explain that a lower risk would be predicted if the person were older than 40 or white.

LORE$_{sa}$ largely improves on stability compared to LORE by adopting a bagging-like approach. Guided by the statistical principle that “averages vary less", we first build an ensemble of decision trees from several local neighborhoods. Differently from pure bagging methods, where aggregation of the ensemble is obtained at prediction

[2] This is the case in a legal argumentation in court, or in an industrial setting where a company wants to stress-test a machine learning component of a manufactured product, to minimize the risk of failures and consequent industrial liability (Bhatt et al. 2020).

time, we (have to) aggregate the decision trees by merging them into a single decision tree (Strecht et al. 2014), from which explanations are then extracted.

LORE$_{sa}$ deals with actionability of counterfactuals by assuming a set U of constraints on features that the rule must satisfy. A constraint is an equality or an inequality over features involving the values of the instance under analysis. For example, *race* = *African-American* constraints the value of the feature *race*. Thanks to the choice of formal logic as the language of the explanations, checking for actionability boils down to test for constraint satisfaction, namely that the premise of a counterfactual rule implies the constraints in U. Indeed, the meaningfulness and usefulness of the explanation depends on the stakeholder (Bhatt et al. 2020), for which we assume the set of constraints U to be given. For instance, the counterfactual $\{race = Caucasian\} \rightarrow$ *Low-Risk* may make sense to a judge that wants to double-check the suggestion of the decision support system. However, the same counterfactual is not useful to the prisoner that cannot change the reality of being black.

We present an extensive experimentation comparing LORE$_{sa}$ with state-of-the-art explanation methods. The experimental setting covers datasets of different type (tabular data, images, texts, and multi-labelled data) and four black-box models. Evaluation methods include a qualitative analysis, a ground-truth validation, and quantitative metrics of the expected properties of the compared methods (fidelity, comprehensibility, stability, minimality).

The rest of the paper is organized as follows. Related work is reviewed in Sect. 2. (Counter)factual explanations as logic rules are introduced in Sect. 3. LORE$_{sa}$ is presented in Sect. 4 and experimented with in Sect. 5. Conclusions summarize contributions and open directions. Appendices report further experiments supporting the design choices of LORE$_{sa}$.

2 Related work

The research of methods for explaining black-box decision systems has recently caught much attention (Li et al. 2022; Miller 2019; Minh et al. 2020; Adadi and Berrada 2018; Molnar 2019). The aim is to couple effective machine learning classifiers with explainers of their logic. Explanation methods can be categorized with respect to two aspects (Guidotti et al. 2019d). One contrasts model-specific *vs* model-agnostic approaches, depending on whether the explanation method exploits knowledge about the internals of the black-box or not. The other contrasts local *vs* global approaches, depending on whether the explanation is provided for any specific instance or for the logic of the black-box as a whole. The proposed explanation method LORE$_{sa}$ fits the line of research of *local*, *model-agnostic* methods originated with (Ribeiro et al. 2016) and extended in several directions in the last few years.

In Ribeiro et al. (2016) the authors introduce LIME, a local model-agnostic explainer. LIME randomly generates instances "around" the instance to explain creating a local neighborhood. Then, it learns a linear model on the neighborhood instances labeled with the black-box decision, and it returns as explanation the feature importance of the most relevant features in the linear model. The number of such features has to be specified by the user. This can be a limitation since users may have no clue about the correct number of features. Besides being model-agnostic, LIME is also not tied to a specific type of data. However, it employs conceptually different neighborhood generation strategies (Guidotti et al. 2019a) for tabular data, images, and texts.[3] A further limitation of LIME is that the random neighborhood generation does not take into account the *density* of black-box outcomes. These drawbacks of LIME are addressed in the literature (Guidotti et al. 2019a; Jia et al. 2019; Zhang et al. 2019; Laugel et al. 2018). A stream of research is based on evolutionary approaches (Sharma et al. 2019; Virgolin et al. 2020; Evans et al. 2019). Our proposal fits that line by adopting a genetic algorithm for the generation of the neighborhood to overcome the deficiencies above.

Explanations in forms of feature importance are also produced by SHAP and MAPLE. SHAP (Lundberg and Lee 2017) connects game theory with local explanations and overcomes the LIME limitation related to the user-provided number of features. SHAP exploits the Shapely values of a conditional expectation function of the black-box by providing the unique additive feature importance. MAPLE (Plumb et al. 2018) provides explanations as features importance of a linear model by exploiting random forests for the supervised selection of the features.

The aforementioned approaches base their explanation on features importance. We advocate instead for the use of formal logic languages, and in particular for explanations as logic rules[4] (Yang et al. 2017; Lakkaraju et al. 2016; Angelino et al. 2017). ANCHOR (Ribeiro et al. 2018) adopts decision rules (called anchors) as explanations. ANCHOR needs to discretize continuous features, while LORE$_{sa}$ does not require this preprocessing step that can affect the quality of explanations. The BRL approach in Ming et al. (2019) provides a rule-based representation describing the local behavior of the black-box though a Bayesian rule list (Yang et al. 2017).

A further expected property of explanation methods regards their stability. For local approaches, the generation of the neighborhood introduces randomness in the process, leading to different explanations for a same instance in different runs of the method (Zafar and Khan 2019), or disproportionately different explanations for two close instances (Alvarez-Melis and Jaakkola 2018). Instability of interpretable shadow models in global approaches has been also pointed out (Guidotti and Ruggieri 2019), and some model-specific approaches have been proposed (Bénard et al. 2019).

The concept of counterfactuals, i.e., instances similar to those to explain but with different labels assigned by the black-box, is a key element in causal approaches to interpretability (Chou et al. 2022; Moraffah et al. 2020; Verma et al. 2020), and it is supported by human thinking (Byrne 2016). In Wachter (2017) a counterfactual is computed by solving an optimization problem. Other notions of counterfactuals can be also obtained with different objective functions (Lucic et al. 2020; Sharma et al. 2019; Mothilal et al. 2020). LORE$_{sa}$ provides a more abstract notion of counterfactual,

[3] For images, LIME randomly replaces real super-pixels with super-pixels containing a fixed color. For texts, it randomly removes words. For tabular data, it assumes uniform distributions for categorical attributes and normal distributions for the continuous ones.

[4] Formal logic as a theory of human reasoning is questioned in the psychology literature, even in the simple case of if-then rules (Byrne and Johnson-Laird 2009). On the other side, formal logic is extensively adopted in mathematics, computer science, linguistics, etc., to unambiguously state arguments (e.g., theories, specifications) and to reason over them. See e.g., Calegari et al. (2020) for a survey on types and applications of logics in AI.

consisting of logic rules rather than flips of feature values. Thus, the user is given not only a specific example of how to obtain actionable recourse (Venkatasubramanian and Alfano 2020; Karimi et al. 2020), but also an abstract characterization of its neighborhood instances with reversed black-box outcome.

Finally, LORE$_{sa}$ largely improves over our previous work LORE (Guidotti et al. 2019b) with regard to the following aspects: (i) LORE$_{sa}$ accounts for counterfactual explanations that are actionable by satisfying user-provided constraints on unmodifiable attributes; (ii) LORE$_{sa}$ accounts for stability by generating multiple local decision trees and merging them to average their instabilities; (iii) LORE$_{sa}$ is able to explain multi-class and multi-label black-boxes, while LORE works only with binary black-boxes; (iv) LORE$_{sa}$ can be applied also to images and texts, while LORE works only with tabular data.

3 Problem formulation and explanation definition

We first set the basic notation for classification models. Afterwards, we define the *black-box outcome explanation problem*, and the notion of *explanation* that our method will be able to provide.

A *predictor* or classifier, is a function $b : \mathcal{X}^{(m)} \rightarrow \mathcal{Y}$ which maps data instances (tuples) x from a feature space $\mathcal{X}^{(m)}$ with m input features to a decision y in a target space $\mathcal{Y}$ of size $L = |\mathcal{Y}|$, i.e., y can assume one of the L different labels ($L = 2$ is binary classification, $L > 2$ is multi-class classification). We write $b(x) = y$ to denote the decision y taken by b, and $b(X) = Y$ as a shorthand for $\{b(x) \mid x \in X\} = Y$. If b is a probabilistic classifier, we denote with $b_p(x)$ the vector of probabilities for the different labels. Hence, we have that $b(x) = y$ is the label with the largest probability among the L values in $b_p(x)$. An instance x consists of a set of m attribute-value pairs (a_i, v_i), where a_i is a feature (or attribute) and v_i is a value from the domain of a_i. The domain of a feature can be continuous or categorical. We assume that a predictor is available as a function that can be queried at will. In the following, b will be a *black-box* predictor, whose internals are either unknown to the observer or they are known but uninterpretable by humans. Examples include neural networks, SVMs, ensemble classifiers (Freitas 2013; Guidotti et al. 2019d). Instead, we denote with c an *interpretable* (comprehensible) predictor, whose internal processing leading to a decision $c(x) = y$ can be given a symbolic interpretation which is understandable by a human. Examples of such predictors include rule-based classifiers, decision trees, decision sets, and rational functions (Freitas 2013; Guidotti et al. 2019d).

Given a black-box b and an instance x, the *black-box outcome explanation problem* consists of providing an explanation e for the decision $b(x) = y$. We approach the problem by learning an interpretable predictor c that reproduces and accurately mimes the *local* behavior of the black-box. An explanation of the decision is then derived from c. By *local*, we mean focusing on the behavior of the black-box in the neighborhood of the specific instance x, without aiming at providing a description of the logic of the black-box for all possible instances. The neighborhood of x is not given, but rather it has to be generated as part of the explanation process. However, we assume that some knowledge is available about the characteristics of the feature space $\mathcal{X}^{(m)}$, in particular the ranges of admissible values for the domains of features and, possibly, the (empirical) distribution of features. Nothing is instead assumed about the training data/process of the black-box.

Definition 1 (*Black-Box Outcome Explanation*) Let b be a black-box, and x an instance whose decision $y = b(x)$ has to be explained. The *black-box outcome explanation problem* consists of finding an explanation $e \in E$ belonging to a human-interpretable domain E.

Interpretable predictors are specific of the black-box and of the instance to explain and they must agree with the black-box decision.

Definition 2 (*Explanation through Interpretable Model*) Let $c = \zeta(b, x)$ be an interpretable predictor derived from the black-box b and the instance x using some procedure ζ, and s.t. $c(x) = b(x)$. An *explanation* $e \in E$ is obtained through c, if $e = \varepsilon(c, x)$ for some explanation logic ε over c and x.

These definitions are parametric in the domain E of explanations, which has to be instantiated. We define it by adopting a combination of factual and counterfactual rules. Formally, we define an explanation e as:

$$e = \langle r = p \to y, \Phi \rangle$$

The first component $r = p \to y$ is a *factual* decision rule describing the reason for the decision value $y = b(x) = c(x)$. The second component Φ is a set of *counterfactual* rules, namely rules describing a (minimal) number of changes in the feature values of x that would change the decision of the predictor to $y' \neq y$. As an example, the following is an explanation for the decision to reject the loan application of instance $x_0 = \{age = 22, sex = male, income = 800, car = no\}$:

$$\begin{aligned} e{=}\langle r{=}\{age{\leq}25,\, sex{=}male,\, income{\leq}900\}{\to} deny,\\ \Phi{=}\{\{income{>}900\}{\to} grant,\, \{sex{=}female\}{\to} grant\}\rangle \end{aligned}$$

In this example, the decision *deny* is due to the age lower or equal than 25, the sex that is male, and an income lower or equal than 900 (see component r). In order to obtain a different decision, the applicant should have a greater income, or be a female (see component Φ).

In a *factual rule* r of the form $p \to y$, the decision y is the *consequence* of the rule, while the *premise* p is a boolean condition on feature values. We assume that p is a conjunction of *split conditions* of the form $a_i{\in}[v_i^{(l)}, v_i^{(u)}]$, where a_i is a feature and $v_i^{(l)}, v_i^{(u)}$ are lower and upper bound values in the domain of a_i extended with[5] $\pm\infty$. An instance x *satisfies* r, or r *covers* x, if the boolean condition p evaluates to true for

[5] Using $\pm\infty$ we can model with a single notation typical univariate split conditions, i.e., equality ($a = v$ as $a \in [v, v]$), upper bounds ($a \leq v$ as $a \in [-\infty, v]$), strict lower bounds ($a > v$ as $a \in [v + \epsilon, \infty]$ for a sufficiently small ϵ). However, since our method is parametric to a decision tree induction algorithm, split conditions can also be multivariate, e.g, $a \leq b + v$ for a, b features (as in oblique decision trees (Murthy et al. 1994)).

x, i.e., if $sc(x)$ is true for every $sc \in p$. The rule r in the example above is satisfied by x_0, and not satisfied by $x_1 = \{age = 22, sex = male, income = 1000, car = no\}$. We say that r *is consistent* with the interpretable predictor c, if $c(x) = y$ for every instance x that satisfies r. Consistency means that the rule provides a sufficient condition for which the predictor outputs y. If the instance x to explain satisfies p, the rule $p \rightarrow y$ represents then a candidate explanation of the decision $c(x) = y$. Moreover, if the interpretable predictor mimics the behavior of the black-box in the neighborhood of x, we further conclude that the rule is a candidate local explanation of $b(x) = c(x) = y$.

Consider now a set δ of split conditions. We denote the update of p by δ as $p[\delta] = \delta \cup \{(a \in [v_i^{(l)}, v_i^{(u)}]) \in p \mid \nexists w_i^{(l)}, w_i^{(u)}.(a \in [w_i^{(l)}, w_i^{(u)}]) \in \delta\}$. Intuitively, $p[\delta]$ is the logical condition p with ranges for attributes overwritten as stated in δ, e.g., $\{age{\leq}25, sex = male\}[age{>}25]$ is $\{age{>}25, sex = male\}$. A *counterfactual rule* for p is a rule of the form $p[\delta] \rightarrow y'$, for $y' \neq y$. We call δ a *counterfactual*. Consistency w.r.t. c is meaningful also for counterfactual rules, denoting now a sufficient condition for a reverse decision y' of the predictor c. A counterfactual δ describes *which* features to change and *how* to change them to get an outcome different from y. Continuing the loan example, changing the income to any value > 900 will change the predicted outcome of b from *deny* to *grant*. A desirable property of a consistent counterfactual rule $p[\delta] \rightarrow y'$ is that it should be *minimal* (Lucic et al. 2019; Wachter 2017) with respect to x. Minimality can be measured (see Guidotti et al. (2019b)) with respect to the number of split conditions in $p[\delta]$ not satisfied by x. Formally, we define $nf(p[\delta], x) = |\{sc \in p[\delta] \mid \neg sc(x)\}|$ (where $nf(\cdot, \cdot)$ stands for the number of falsified split conditions[6]). In the loan example, $\{age{>}25, income{>}1500\} \rightarrow grant$ is a counterfactual with two conditions falsified. It is not minimal as the counterfactual $r = \{age{\leq}25, sex{=}male, income{>}900\} \rightarrow grant$ has only one falsified condition. In summary, a counterfactual rule $p[\delta] \rightarrow y'$ is a (minimal) *motivation for reversing* the decision outcome of the predictor b.

In this work, we add to the properties of consistency and minimality of counterfactual rules, the one of *actionability* (also called *feasibility*), which is intended to prevent generating invalid or unrealistic rules. E.g., a counterfactual split condition $age \leq 25$ is not actionable for a loan applicant of age 30 because she cannot change her age. Formally, we assume a set U of *constraints on features* of the form: $a = x[a]$, meaning that the attribute a cannot be changed (e.g., $age = 30$ or $sex = male$); or, $a \leq x[a]$ (resp., $a \geq x[a]$), meaning that the attribute a cannot be increased (resp., decreased). Actionability requires that the premise $p[\delta]$ of a counterfactual rule must satisfy the conditions specified in U, i.e., $p[\delta] \rightarrow U|_{p[\delta]}$ is a true formula, where $U|_{p[\delta]}$ are the constraints in U involving attributes occurring in $p[\delta]$. Going back to our example if $U = \{age = 22\}$, then the counterfactual $\{age{>}25, income{>}1500\} \rightarrow grant$ is not actionable.

We can now formally introduce our notion of explanation.

Definition 3 (*Explanation*) Let $c = \zeta(b, x)$ be an interpretable predictor such that $c(x) = b(x)$, and U a set of constraints. A local (counter)factual explanation $e = \langle r, \Phi \rangle$ is a pair of: a rule $r = (p \rightarrow y)$ consistent with c and satisfied by x; and, a set

[6] When clear we write nf as shorthand of $nf(p[\delta], x)$.

Algorithm 1: LORE$_{sa}(x, b, K, U)$

```
Input  : $x$ - instance to explain, $b$ - black-box, $K$ knowledge, $U$ constr.
Output: $e$ - (counter)factual explanation of $x$
1  $\mathcal{D} \leftarrow \emptyset$;                                                // init. empty set of decision trees
2  for $i \in \{1, \dots, N\}$ do
3      $Z_=^{(i)} \leftarrow genetic(x, fitness_=^x, b, K)$;                  // neighborhood generation
4      $Z_{\neq}^{(i)} \leftarrow genetic(x, fitness_{\neq}^x, b, K)$;        // neighborhood generation
5      $Z^{(i)} \leftarrow Z_= \cup Z_{\neq}$;                                   // merge neighborhoods
6      $Y^{(i)} \leftarrow b(Z^{(i)})$;                                          // apply black-box
7      $d^{(i)} \leftarrow buildDecisionTree(Z^{(i)}, Y^{(i)})$;                //build decision tree
8      $\mathcal{D} \leftarrow D \cup \{d^{(i)}\}$;                           // add decision tree to list
9  $c \leftarrow mergeDecisionTrees(\mathcal{D})$;                            // merge decision trees
10 $r = (p \rightarrow y) \leftarrow extractDecisionRule(c, x)$;            // factual rule
11 $\Phi \leftarrow extractCounterfactuals(c, r, x, U)$;                   // extract counterfactual
12 return $e \leftarrow \langle r, \Phi \rangle$;
```

$\Phi = \{p[\delta_1] \rightarrow y', \dots, p[\delta_v] \rightarrow y'\}$ of counterfactual rules for p consistent with c such that $p[\delta_i]$ satisfies U, for $i = 1, \dots, v$.

Unless otherwise stated, in the rest of the paper we will simply write “an explanation" instead of “a local (counter)factual explanation". According to Definition 2, we will design a solution to the outcome explanation problem by defining: (i) the function ζ that computes an interpretable predictor c for a given black-box b and an instance x, and (ii) the explanation logic ε that derives a (counter)factual explanation e from c and x as in Definition 3.

4 Local rule-based explanation

We propose LORE$_{sa}$, a Stable and Actionable LOcal Rule-based Explanation method, described in Algorithm 1 as extension of LORE (Guidotti et al. 2019b). LORE$_{sa}$ takes in input a black-box b, an instance x to explain, a set of constraints U, and a knowledge base K which contains information about feature distributions (domain of admissible values, mean, variance, probability distribution, etc.). LORE$_{sa}$ first generates N sets of neighbor instances $Z = \{Z^{(1)}, \dots, Z^{(N)}\}$ of x through a *genetic algorithm*. The knowledge base K is exploited in genetic mutation to be consistent with the distributions of the features. Next, LORE$_{sa}$ labels the generated instances with the black-box decision. For each labelled neighborhood $Z^{(i)}$ a decision tree $d^{(i)}$ is built, and all such trees are merged into a single interpretable predictor c still in the form of a *decision tree*. Rules and counterfactual rules are extracted from c, satisfying the constraints in U.

LORE$_{sa}$ fits the definitions of the previous section as follows: lines 1–9 in Algorithm 1 implement the ζ function for extracting the interpretable decision tree c, which approximates locally the behavior of the black-box b; and lines 10–11 implement the function ε to extract the (counter)factual explanation e from the logic of the decision tree.

Stability of the explanation process follows from the “bagging-like" approach of building and aggregating several decision trees. In fact, it is well-known that decision trees are unstable to small data perturbations (Breiman 2001). Bagging is a widespread method to stabilize decision trees (Breiman 1996). Experiments will confirm this by contrasting stability metrics of LORE$_{sa}$ with its “single-tree" version LORE. Resorting to bagging, however, produces a collection of interpretable explainers. We need then to aggregate them at symbolic level—which is different from standard bagging, where aggregation is at prediction time. For this, we have a merging procedure in line 9 of Algorithm 1.

The actionability of the counterfactuals follows from taking into account the constraint set U on admissible feature changes (Algorithm 1, line 11). The search for counterfactuals will also consider the minimality requirement.

In the following, we discuss the details of LORE$_{sa}$ by motivating the design choices by the expected properties of the explanation process: locality, fidelity and stability, comprehensibility, actionability, and generality.

4.1 Locality: neighborhood generation

The goal of this phase is to identify sets of instances $Z^{(i)}$, whose feature are close to the ones of x, in order to be able to reproduce the behavior of the black-box b locally to x. Since the aim is to learn a predictor from $Z^{(i)}$, such a neighborhood should be flexible enough to include instances with decision values equal and different from $b(x)$. In Algorithm 1, first we extract balanced subsets $Z^{(i)}_{=}$ and $Z^{(i)}_{\neq}$ (lines 2–3), where instances $z \in Z^{(i)}_{=}$ are such that $b(z) = b(x)$, and instances $z \in Z^{(i)}_{\neq}$ are such that $b(z) \neq b(x)$, and then we define $Z^{(i)} = Z^{(i)}_{=} \cup Z^{(i)}_{\neq}$ (line 4). We depart from instance *selection* approaches (Olvera-López et al. 2010), and in particular the ones based on genetic algorithms (Tsai et al. 2013), in that their objective is to select a subset of instances from an given training set. In our case, instead we cannot assume that the training set used to learn b is available, or not even that b is a supervised machine learning predictor for which a training set exists. Instead, our task is similar to instance *generation* in active learning (Fu et al. 2013), which also includes evolutionary approaches (Derrac et al. 2010).

We adopt an approach based on a *genetic algorithm* which generates $Z^{(i)}_{=}$ and $Z^{(i)}_{\neq}$ by minimizing the following fitness functions:

$$\mathit{fitness}^{x}_{=}(z) = I_{x \neq z} + d(x, z) + l(b_p(x), b_p(z))$$
$$\mathit{fitness}^{x}_{\neq}(z) = I_{x \neq z} + d(x, z) + (1 - l(b_p(x), b_p(z)))$$

where $d : \mathcal{X}^{(m)} \rightarrow [0, 1]$ is a distance function in the feature space (hence $d(x, z)$ is close to zero when two instances are similar with respect to their features), $l : \mathcal{R} \rightarrow [0, 1]$ is a distance function in the label space with respect to the prediction probability b_p (hence $l(b_p(x), b_p(z))$ is close to zero when two instances are similar with respect to their label probabilities), and the function $I_{x \neq z}$ returns zero if z is not equal to x, and ∞ otherwise. The genetic neighborhood process tries to minimize these fitness

Algorithm 2: *genetic*$(x, \mathit{fitness}, b, K)$

```
Input  : $x$ - instance to explain, $\mathit{fitness}$ - fitness function,
         $b$ - black-box, $K$ knowledge base
Params: $n$ - population size, $g$ - nbr of generations,
         $p_c$ - prob crossover, $p_m$ - prob mutation
Output : $Z$ - neighbors of $x$

$P_0 \leftarrow (x \mid \forall 1, \ldots, n); i \leftarrow 0$;        // population init.
while $i < g$ do
    $P' \leftarrow \mathit{crossover}(P_i, p_c)$;                      // mix records
    $P'' \leftarrow \mathit{mutate}(P', p_m, K)$;                      // perform mutations
    $S \leftarrow \mathit{evaluate}(P'', \mathit{fitness}, b)$;        // evaluate population
    $P_{i+1} \leftarrow \mathit{select}(P'', S)$;                      // select sub-population
    $i \leftarrow i + 1$                                               // update population
$Z \leftarrow P_i$
return $Z$;
```

functions. Therefore, $\mathit{fitness}^x_=(z)$ looks for instances z similar to x (term $d(x, z)$), but not equal to x (term $I_{x \neq z}$), for which the black-box b has a similar behavior (term $l(b_p(x), b_p(z))$). On the other hand, $\mathit{fitness}^x_{\neq}(z)$ leads to the generation of instances z similar to x, but not equal to it, for which b returns a different decision. We underline that $\mathit{fitness}^x_=(x)$=$\mathit{fitness}^x_{\neq}(x)$=$\infty$. Hence, the minimization occurs for $z \neq x$.

A key element for the fitness functions are the distances $d(x, z)$ and $l(b_p(x), b_p(z))$. Concerning $d(x, z)$, we account for mixed types of features by a weighted sum of Simple Matching distance (SM) for categorical features, and of the normalized Euclidean distance (NE)[7] for continuous features. Assuming h categorical features and $m - h$ continuous ones, we use:

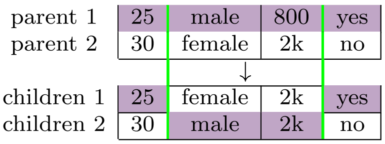
$$d(x, z) = \frac{h}{m} \cdot SM(x, z) + \frac{m-h}{m} \cdot NE(x, z).$$

Our approach is parametric[8] to d, and it can readily be applied to improved heterogeneous distance functions (McCane and Albert 2008). With regard to $l(b_p(x), b_p(z))$, we account for sparse numeric vectors by adopting the cosine distance. If b is not a probabilistic classifier, then $l(b_p(x), b_p(z))$ is replaced by identity checking, namely $l(b(x), b(z)) = 0$ if $b(x) = b(z)$, and 1 otherwise.

Genetic algorithms (Holland 1992) are inspired by the biological metaphor of evolution and are based on three distinct aspects. (i) The potential solutions of the problem are encoded into representations that support the *variation* and *selection* operations. In our case, these representations, generally called chromosomes, correspond to instances in the feature space $\mathcal{X}^m$. (ii) A fitness function evaluates which chromosomes are the "best life forms", that is, most appropriate for the result. These are then favored in *survival* and *reproduction*, thus shaping the next generation according to the fitness function. In our case, these instances correspond to those similar to x, according to $d(\cdot, \cdot)$, and those similar/different to the outcome returned by the black-box $b_p(x)$,

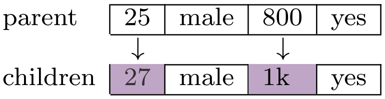
[7] See *NormalizedSquaredEuclideanDistance* at Wolfram.

[8] See "Appendix B" for a comparison of a few distance functions.



parent 1	25	male	800	yes
parent 2	30	female	2k	no
		↓		
children 1	25	female	2k	yes
children 2	30	male	2k	no

Fig. 1 Crossover



parent	25	male	800	yes
	↓		↓	
children	27	male	1k	yes

Fig. 2 Mutation

according to $l(\cdot, \cdot)$, for the fitness function $\mathit{fitness}^x_=$ and $\mathit{fitness}^x_{\neq}$ respectively. (iii) Mating (called crossover) and mutation produce a new generation of chromosomes by recombining features of their parents. The final generation of chromosomes, according to a stopping criterion, is the one that best fits the solution.

Algorithm 2 generates the neighborhoods $Z^{(i)}_=$ and $Z^{(i)}_{\neq}$ of x by instantiating the evolutionary approach described in Bäck et al. (2000). Using the terminology of the survey (Derrac et al. 2010), it is an instance of generational genetic algorithms for evolutionary prototype generation. However, prototypes are a condensed subset of a training set that enable some optimization in training predictors. We aim instead at generating new instances that separate well the decision boundary of the black-box b. The usage of classifiers within fitness functions of genetic algorithms can be found in Wu and Olafsson (2006). However, the classifier they use is always the one for which the population must be selected or generated from and not another one (the black-box) like in our case. Algorithm 2 first initializes the population P_0 with n copies of the instance x to explain. Then it enters the evolution loop that begins with the crossover operator applied to a proportion p_c of P_i: the resulting and the untouched instances are inserted in P'. We use a *two-point crossover* which selects two parents and two crossover features and swap the crossover feature values of the parents (see Fig. 1). Next, a proportion of P', determined by the p_m probability, is mutated (see Fig. 2) by exploiting the feature distributions given by the knowledge[9] base K. Mutated and unmutated instances are added in P''. Instances in P'' are evaluated according to the fitness function, and the top n of them w.r.t. the fitness score are selected to become P_{i+1}—the next generation. The evolution loop continues until g generations are completed.[10] The best individuals are returned. LORE$_{sa}$ runs Algorithm. 2 twice, once using the fitness function $\mathit{fitness}^x_=$ to derive neighbor instances $Z^{(i)}_=$, and once

[9] K is assumed to include the probability mass functions of discrete features and the density function of continuous features. In experiments, K is empirically estimated from the set of instances to explain (not used for training the black-box) by taking the frequencies of values for discrete features, and by selecting the best fit of the empirical density of continuous features with one of the following families of distributions: uniform, normal, exponential, gamma, beta, alpha, chi-square, Laplace, log-normal, power law. We also assume that features are independent, hence, we do not infer the joint distribution.

[10] In the implementation of LORE$_{sa}$, we set the number of instances $n = 500$, the number of generations $g = 20$, the probabilities of crossover $p_c = 0.7$ and of mutation $p_m = 0.5$. Experiments showing the effect of varying these parameters are reported in C.

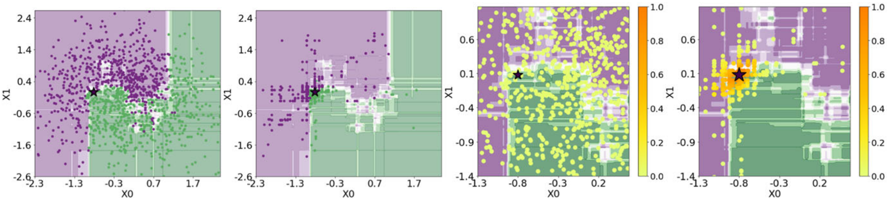


Fig. 3 Black-box boundary: purple versus green. Starred instance x. Uniformly random (1st) and genetic (2nd) neighborhoods. In the (3rd) and (4th) plot is reported the density with levels in the bar (best view in color)

using the function $\mathit{fitness}^x_{\neq}$ to derive $Z^{(i)}_{\neq}$. Finally, setting $Z^{(i)}{=}Z^{(i)}_{=}{\cup}Z^{(i)}_{\neq}$ guarantees that $Z^{(i)}$ is balanced.

Figure 3 shows an example of neighborhood generation for a black-box consisting of a random forest model on a bi-dimensional feature space. The figure contrasts uniform random generation (1st, 3rd plots) around a specific instance x (starred) to our genetic approach (2nd, 4th plots). The latter yields a neighborhood that is denser in the boundary region of the predictor. The density of the generated instances is a key factor in extracting correct and faithful local interpretable predictors and explanations. For instance, a purely random procedure like the one adopted in LIME (Ribeiro et al. 2016) does not account for sources of variability, like the randomness of the sampling procedure in the neighborhood of the instance to explain (Zhang et al. 2019). On the contrary, the genetic approach of LORE$_{sa}$ is driven by minimization of the fitness functions, hence less variable neighborhoods are generated. As a further issue, simply centering the neighborhood generation on the instance to explain may not be the best strategy to approximate the black-box decision boundary. Jia et al. (2019) and Laugel et al. (2018) propose neighborhood generation approaches that enhance locally important features with respect to globally important ones by moving the center of the generation towards the decision boundary. The two fitness functions in the genetic generation procedure of LORE$_{sa}$ enforce the same effect. An experimental comparison of various neighborhood generation techniques is reported in Appendix A.

4.2 Fidelity and stability: bagging of interpretable predictors

We tackle the issue of instability of the local predictor trained on a random neighborhood of x by adopting an approach which exploits the multiple generation of random neighborhoods $Z = \{Z^{(1)}, Z^{(2)}, \ldots, Z^{(N)}\}$ that then, can be used for learning a single decision tree c, i.e., the local interpretable predictor, by following a bagging-like approach. Bagging, boosting, and random forests achieve high predictive performances, which, in our context means high fidelity (accuracy w.r.t. black-box decisions). Moreover, they achieve stability of predictions by averaging the decisions of several trees (Sagi and Rokach 2018). For each neighborhood $Z^{(i)}$ of x, LORE$_{sa}$ builds a decision tree classifier $d^{(i)}$ trained on the instances in $Z^{(i)}$ labeled with the black-box decisions $Y^{(i)} = b(Z^{(i)})$. We adopt CART (Breiman et al. 1984) for the tree building function *buildDecisionTree* of Algorithm 1. The N decision trees[11] are then

[11] In the experiments, we set $N = 5$. Details in Sect. 5.

merged into a single decision tree c, the local interpretable predictor. Such a classifier is intended to mimic the behavior of b in the neighborhoods of x. The requirement that $c(x) = b(x)$ from Definition 3, is tested on the merged decision tree, and, if it is not met, the algorithm is restarted.[12] The choice of decision trees as interpretable predictors allows for symbolic reasoning: (i) factual decision rules can readily be derived from the root-to-leaf path in a tree; and, (ii) counterfactual rules can be extracted by symbolic reasoning over a decision tree (Breiman et al. 1984; Sokol and Flach 2019). However, the decision logic of ensembles cannot be directly turned into rules.

For this, we first merge the N decision trees into a single decision tree c. A stream of research focuses on this problem (Assche and Blockeel 2007; Vidal and Schiffer 2020; Sagi and Rokach 2020). In this paper we propose to adopt the method introduced by Fan et al. (2020) which implements the schema of merging multiple decision trees described in Strecht et al. (2014). The procedure for merging a set of trees $d^{(1)}, d^{(2)}, \ldots, d^{(N)}$, trained on various subsets of a given dataset, into a unique decision tree c is composed of two main phases. In the *first phase*, the decision regions of the different tree models are *merged* using a recursive approach which allows for their simultaneously. It uses the notion of *condition tree*. Given a decision tree d and a condition Cd, let S_j denote the condition set of node j in d, which is composed of conditions from root to node j, then a condition tree $d^{(Cd)}$ is composed of those nodes in d such that all the conditions in S_j satisfies Cd. Hence, if an inner node in d is not included in $d^{(Cd)}$, then all its branches are not included in $d^{(Cd)}$. Once computing the condition tree for each decision tree $d^{(i)}$ to be merged, they are recursively merged to obtain one branch of the root with condition Cd. After merging all the models, the *second phase*, called “pruning”, tries to reduce the number of decision regions involved. In particular, the merged decision tree c is pruned by removing inner nodes having as leaves the same class. This procedure returns a final decision tree with multi-way splits even though the input decision trees are trees with binary splits. One of the most important advantages of this approach is that the merging method is *lossless* as it maintains for every instance the class label assigned by the tree ensembles. Also, Fan et al. (2020) show that their approach is more efficient with respect to the state of art approaches because requires less memory than others.

The idea behind this procedure is that we want to exploit: (i) the multiple neighborhood generation for increasing the probability of covering the whole local domain around the instance to be explained, and (ii) the ability of learning from the decisions made by different decision trees tailored on their training data; and (iii) the ability of the merging procedure to derive a single model that generalizes the knowledge contained in the multiple original decision trees. These three characteristics help the *local* interpretable predictor to be more *stable* because they mitigate the possible effect of the randomness introduced in the neighborhood generation, which could lead to have for the same instance a slightly different explanation. Moreover, the generalized representation of the knowledge contained in the multiple decision trees helps in reducing the probability that small changes in the data may result in very different explanations.

[12] Notice that, since the genetic generation starts from a dataset with all instances equal to x (P_0 in Algorithm 2), the case $c(x) \neq b(x)$ is rather infrequent. In our experiments (not reported here), this occurred only in 0.4% of cases.

Algorithm 3: *extractCounterfactuals*(c, r, x, U)

```
   Input  : $c$ - decision tree, $r$ - rule, $x$ - instance to explain, $U$ - constraints
   Output: $\Phi$ - set of counterfactual rules for $p$
1  $Q \leftarrow getPathsWithDifferentLabel(c, y)$;                 // get paths with $y' \neq y$
2  $\Phi \leftarrow \emptyset$; $min \leftarrow +\infty$;            // initialize counterfactual set
3  for $q \in Q$ do
4      if not $q \rightarrow U|_q$ then
5          continue;                                              // skip rule if constraints not satisfied
6      $qlen \leftarrow nf(q, x) = |\{sc \in q \mid \neg sc(x)\}|$
7      if $qlen < min$ then
8      $\Phi \leftarrow \{q \rightarrow y'\}$; $min \leftarrow qlen$
9      else if $qlen = min$ then
10     $\Phi \leftarrow \Phi \cup \{q \rightarrow y'\}$
11 return $\Phi$;
```

4.3 Comprehensibility: extracting (counter-)factual rules

We achieve high-level comprehensibility of explanations by extracting them in the form of factual rules and sets of counterfactual rules. Given the decision tree c, we derive an explanation $e = \langle r, \Phi \rangle$ as follows. The factual rule $r = p \rightarrow y$ is formed by including in p the split conditions on the path[13] from the root to the leaf satisfied by x, and setting $y = c(x) = b(x)$. By construction, r is consistent with c and satisfied by x. Consider now the counterfactual rules in Φ. Algorithm 3 looks for all paths in the decision tree c leading to a decision $y' \neq y$ (line 1). For one of such paths, let q be the conjunction of split conditions in it. By construction, $q \rightarrow y'$ is a counterfactual rule consistent with c. Notice that the counterfactual δ for which $q = p[\delta]$ has not to be explicitly computed.[14] All such q's can be ranked by the number of split conditions not satisfied by x, a.k.a. the number of features to be changed in x. The $q \rightarrow y'$'s with minimal number of changes are returned as counterfactuals (lines 6-8).

4.4 Actionability: constraint satisfaction testing

The counterfactuals provided by LORE$_{sa}$ support actionable recourse. This is implemented in Algorithm 3 by filtering from the candidate counterfactuals $q \rightarrow y'$ those not satisfying the constraints U on features (lines 4-5). Since both the premise q and the constraints U are logic formulae, the test amounts at checking validity of the implication $q \rightarrow U|_q$. For the basic form of constraints that we have considered (conjunction of equality/comparison conditions) the test is straightforward. In principle, however,

[13] The set of split conditions in the path is also called a direct reason, and it is not necessarily minimal. Minimal sets (called sufficient conditions, or prime implicant explanations) are considered in Shih et al. (2018) and Darwiche and Hirth (2020). We do not further purse minimizing the factual explanation as experiments shows LORE$_{sa}$ returns very small rules.

[14] However, it can be done as follows. Consider the path from the leaf of p to the leaf of q. When moving from a child to a father node, we retract the split condition. E.g., $a_i \leq v_i^{(u)}$ is retracted from $\{a_j \in [v_j^{(l)}, v_j^{(u)}]\}$ by adding $a_i \in [v_i^{(l)}, +\infty]$ to δ. When moving from a father node to a child, we add the split condition to δ.

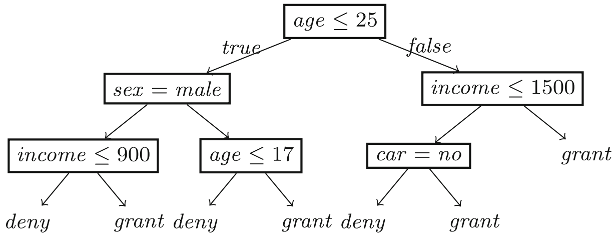


Fig. 4 Example of decision tree locally mimicking the black-box behavior

more complex premises (e.g., multivariate) can be dealt with by resorting to automatic theorem proving.

Let assume that the decision tree in Fig. 4 is the merged decision tree c. Let x=\{*age*=22, *sex*=*male*, *income*=800, *car*=*no*\} be the instance for which the decision *deny* (e.g., of a loan) has to be explained. The path followed by x is the leftmost in the tree. The decision rule extracted from the path is \{*age*≤25, *sex*=*male*, *income*≤900\}→*deny*. There are four paths leading to *grant*: q_1=\{*age*≤25, *sex*=*male*, *income*>900\}, q_2=\{17<*age*≤25, *sex*=*female*\}, q_3=\{*age*>25, *income*≤1500, *car*=*yes*\}, and q_4 =\{*age*>25, *income*>1500\}. The number of changes for the q_i's are as follow: $nf(q_1, x)$=1, $nf(q_2, x)$=1, $nf(q_3, x)$ =2, $nf(q_4, x)$=2. Therefore, the set of minimal counterfactuals is Φ=\{q_1→*grant*, q_2→*grant*\}. Assuming that U=\{*sex*=*male*\}, then q_2→*grant* is not actionable, hence the set of actionable counterfactuals is Φ=\{q_1→*grant*\}.

Finally, we point out that an actionable counterfactual rule $q \rightarrow y'$ can be used to generate an *actionable counterfactual instance*. Among all possible instances that satisfy $q \rightarrow y'$, we choose the one that differ minimally from x. This is done by looking at the split conditions falsified by x: $\{sc \in q \mid \neg sc(x)\}$, and selecting for features appearing in an sc the lower/upper bound that is closer to the value of the feature in x. For instance, the q_1→*grant* counterfactual instance of x is x' = \{*age*=22, *sex*=*male*, *income*=900+ϵ)\}. We also check that x' constructed in this way is a valid counterfactual, i.e., $b(x')$=*grant*. If this does not occur, x' is not returned as a counterfactual instance.

4.5 Generality: explanations for images, texts and multi-label data

Following the approach of LIME (Ribeiro et al. 2016), LORE$_{sa}$ can be adapted to work on images and texts. Moreover, inspired by Panigutti et al. (2020), we show how it deals with multi-label data.

Image and Text Data In the pre-processing strategy of LIME, an instance in the form of an image or a text is mapped to a vector of binary values. For images, each element in the vector indicates the presence/absence of a contiguous patch of similar pixels (called super-pixels). For words, it indicates the presence/absence of a specific word in the text. This reduces the problem to the analysis of tabular data, and we can reuse LORE$_{sa}$ as introduced so far. Due to the binary nature of data involved, the genetic neighborhood approach boils down to generate instances by suppressing super-pixels or words from the instance to explain. This is close to the way that LIME works, but with a fitness optimizing approach instead of a purely random suppression. As for LIME, the generated instances may not be realistic images or texts.

Multi-labelled Data The formulation of LORE$_{sa}$ admits so far binary and multi-class black-boxes. Multi-labelled classifiers return, for an input instance x, one or more class labels. This case is common, for instance, in health data, where more than one disease may be associated with a same list of symptoms. In particular, probabilistic multi-labelled classifiers return a vector of probabilities $b_p(x)$ whose sum is not necessarily 1, as in the multi-class case. Rather, the ith element in $b_p(x)$ is the probability that the ith label is included in the output (with a typical cut-off at 0.5). LORE$_{sa}$ can be extended to (probabilistic) multi-labelled black-boxes by adopting multi-class decision trees in the function *buildDecisionTree*() of Algorithm 1. Factual rules will be of the form $p \rightarrow y_1, \ldots, y_k$, with $k \geq 1$. Counterfactual rules will be of the form $p[\delta] \rightarrow y'_1, \ldots, y'_{k'}$, with $k \geq 1$ and such that $\{y_1, \ldots, y_k\} \neq \{y'_1, \ldots, y'_{k'}\}$ (but possibly with proper inclusion).

5 Experiments

After presenting the experimental setting and the evaluation metrics, we compare LORE$_{sa}$ against the competitors through: (i) a qualitative comparison of explanations provided, (ii) a quantitative validation of the explanations based on synthetically generated ground truth, and (iii) a quantitative assessment of the proposed method and comparison with state-of-the-art approaches in terms of several metrics.[15] Moreover, the Appendices report further experiments: (iv) comparing different neighborhood generation methods, (v) showing the impact of different distance functions in genetic neighbor generation, (vii) illustrating the effect of the parameters on the genetic neighbor generation, (vii) providing statistical evidence of the differences among LORE$_{sa}$ and its competitors, and (viii) reporting on running times.

5.1 Experimental setup

We experimented with ten tabular datasets, one image dataset, one text dataset, and one multi-labelled dataset. Table 1 reports the dataset details. Almost all tabular datasets have both categorical[16] and continuous features. For most of the datasets, instances regard attributes of an individual person, and the decisions taken by a black-box target socially sensitive tasks.

A random subset of each dataset, denoted by X_{bb}, was used to train the black-box classifiers while the remaining part, denoted by X, was used as instances to explain—

[15] LORE$_{sa}$ has been developed in Python, using *deap* (Fortin et al. 2012, https://github.com/DEAP/deap) for genetic neighborhood generation, and the optimized version of CART (Tan et al. 2005) offered by *scikit-learn* (http://scikit-learn.org/stable/modules/tree.html) for decision tree induction. The source code of LORE$_{sa}$, the datasets, and the scripts for reproducing the experiments are publicly available at https://github.com/francescanaretto/LORE_sa. Experiments were performed on Ubuntu 20.04 LTS, 252 GB RAM, 3.30 GHz × 36 Intel Core i9.

[16] The number of features is calculated prior to one hot encoding.

Table 1 Top: datasets summaries

#	Tabular data										images	text	multi-l.
	adult	bank	churn	compas	compas-m	fico	german	iris	wine-r	wine-w	mnist	20news	medical
n	32,561	150k	3333	7214	7214	10,459	1000	150	1599	54,898	70,000	18,846	978
m	10	13	19	11	11	23	20	4	11	11	28×28	37,096	1449
L	2	2	2	2	3	2	2	3	6	7	10	2	45
X_{bb}	.83	.89	.85	.81	.64	.72	.76	.91	.71	.55	.99	.91	.52
X	.80	.89	.83	.75	.61	.69	.70	.87	.55	.41	.98	.75	.98

n: instances, m: fetures, L: labels. Bottom: average accuracy of all black-boxes (DNN, NN, RF and SVM) on training X_{bb} and test X

Table 2 Average accuracy and stddev of the black-box classifiers

	DNN	NN	RF	SVM
X	.69 ± .24	.75 ± .17	.78 ± .12	.68 ± .16
X_{bb}	.72 ± .26	.76 ± .17	.88 ± .11	.77 ± .13

in brief, the *explanation set*. For tabular data, the split was 70%-30% and stratified w.r.t. the class attribute. For `mnist`, `20news`, and `medical` we followed the split custom in the relevant literature.[17] We denote with $\hat{Y} = b(X)$ the decisions of b on X, and with $Y = c(X)$ the decisions of c on X. We assume that the dataset used to train the black-box is unknown at the time of explanation. Hence, we can only rely on the set X of instances to explain. Indeed, the knowledge base K is derived from the explanation set as stated in Footnote 9. Similarly, information about features' domains required by the competitor methods is computed from X.

We trained and explained the following black-box models: Random Forest (RF), Support Vector Machine (SVM) and Neural Network (NN) as implemented by *scikit-learn*, and Deep Neural Networks (DNN) implemented by *keras*.[18] For each black-box, for each dataset, we performed a random search for the best parameter setting.[19] Average classification accuracies are shown in Table 1 (bottom) and in Table 2. We compare LORE$_{sa}$ against LIME (Ribeiro et al. 2016), MAPLE (Plumb et al. 2018), SHAP (Lundberg and Lee 2017), ANCHOR (Ribeiro et al. 2018) and BRL (Ming et al. 2019). We also compare the counterfactuals of LORE$_{sa}$ with the stochastic optimized counterfactuals SOC (Russell 2019) as implemented by the `alibi` library,[20] and against the brute force coutnerfactual explainer (BF) as implemented by the `fat-forensics` library.[21] Unless stated otherwise, default parameters are used for LORE$_{sa}$ and all the other methods.[22]

5.2 Evaluation metrics

We evaluate the performances of explanation methods under various perspectives. The measures reported in the following are stated for a single instance to be explained. The metrics obtained as the mean value of the measures over all the instances in the explanation set X, can then be used to evaluate the performances of the explanation methods. Let $x \in X$ be an instance to explain.

[17] http://qwone.com/~jason/20Newsgroups/, http://yann.lecun.com/exdb/mnist/.

[18] Black-boxes: https://scikit-learn.org/, https://keras.io/.

[19] Details of the parameters can be found in LORE$_{sa}$ repository.

[20] https://github.com/SeldonIO/alibi.

[21] https://fat-forensics.org/.

[22] We highlight that for SHAP we used the KernelSHAP explainer that can be adopted for any black-box model. Also, as background knowledge for SHAP we used the medoid of the training set. We highlight that, different choices of the background knowledge can significantly impact on the outcome as illustrated in Gosiewska and Biecek (2020) and Sundararajan and Najmi (2020). However, we relied on the medoid because as illustrated in the tutorial for KernelSHAP on tabular data provides the best trade-off between reliability and efficiency. We did not compare against other counterfactual explainers as this is out from the purpose of the paper. We refer to Guidotti (2022) for a comprehensive survey and benchmarking.

Correctness We will evaluate the correctness of explanations under controlled situations where ground truth is available. Let e and $\widetilde{e}$ be the binary vectors indicating the presence/absence (1/0) of a feature in the explanation for x of a given method, and in the ground truth respectively. For rule-based explanations, presence means that the feature appears in the premise of the rule. For feature importance vectors, presence means that the feature has non-zero magnitude. We measure the *correctness* of an explanation w.r.t. the ground-truth using the *f1-score*:

$$\textit{f1-score}(e, \widetilde{e}) = 2 \cdot \frac{\textit{recall}(e, \widetilde{e}) \cdot \textit{precision}(e, \widetilde{e})}{\textit{recall}(e, \widetilde{e}) + \textit{precision}(e, \widetilde{e})}$$

where the *precision* is the percentage of features present in e that are also in $\widetilde{e}$, and the *recall* is the percentage of features in $\widetilde{e}$ that are also in e.

When ground truth is not available, we will consider the following measures to evaluate specific properties of an explanation process.

Silhouette We measure the quality the neighborhood[23] in a local approach by measuring how similar is x to instances in $Z_{=}$ compared to instances in $Z_{\neq}$. Let $d(x, S)$ denote the mean Euclidean distance between x and instances in S. Inspired by clustering validation (Tan et al. 2005), we define:

$$\textit{silhouette}(x) = \frac{d(x, Z_{\neq}) - d(x, Z_{=})}{\textit{max}\{d(x, Z_{\neq}), d(x, Z_{=})\}}$$

High silhouette results from accurate neighborhood generation (Sect. 4.1).

Fidelity It answers the question: how good is the interpretable predictor c at mimicking the black-box b? Fidelity can be measured in terms of accuracy (Doshi-Velez and Kim 2017) of the predictions $Y = c(Z)$ of the interpretable predictor c w.r.t. the predictions $\hat{Y} = b(Z)$ of the black-box b, where Z is the neighborhood of x generated by the local method. High fidelity of c results from both accurate neighborhood generation (Sect. 4.1) and predictive performance of the learning algorithm (Sect. 4.2).

Complexity It is a proxy of the comprehensibility of an explanation, with larger values of complexity denoting harder to understand explanations (Freitas 2013). For rule-based explanations, as complexity we adopt the size of the rule premise (for LORE$_{sa}$ we consider only the factual rule). Low complexity results from general (non-overfitting, stable) local interpretable surrogate predictors (Sect. 4.2) and a direct method to extract the rule (Sect. 4.3). For feature importance vectors, as complexity we adopt the number of non-zero features. For instance in LIME are those of the local surrogate linear regressor.

Stability It measures the ability to provide similar explanations to similar instances. Also named *robustness* or *coherence*, it is a crucial requirement for gaining trust by the users (Guidotti and Ruggieri 2019). We measure it through the local Lipschitz condition (Alvarez-Melis and Jaakkola 2018):

[23] In order to evaluate the neighborhood generated by an explainer, it must be available. BRL, MAPLE and SHAP do not use a notion of neighborhood to return the explanation. However, the SHAP library allows access to the permutation of x tested to determine the Shapely value approximations. We used this set of instances as the neighborhood for SHAP.

$$instability(x) = max_{x_i \in \mathcal{N}_k(x)} \frac{\|e_i - e\|_2}{\|x_i - x\|_2} \tag{1}$$

where $\mathcal{N}_k(x)$ is the set of the $k = 5$ instances in $X \setminus \{x\}$ closest to x w.r.t. Euclidean distance, e is the binary vector of the explanation of x, and e_i is the binary vector of the explanation of $x_i \in \mathcal{N}_k(x)$. Intuitively, the larger is the ratio the more different are the explanations for instances close to x. Low instability (or, high stability) results from general (non-overfitting, stable) local interpretable surrogate predictors (Sect. 4.2). While low instability could be the result of under-fitting, this is not the case of *local* explanation methods which, being local and being based on random components, are not prone to exhibit the same explanation for different instances. In addition, we consider also sensitivity of a local explanation method to randomness introduced in the neighborhood generation. This is measured by the distance of explanations generated for a same instance over multiple calls to the explanation method:

$$instability_{si}(x) = max_{e_i, e_j \in \mathcal{E}_k(x)} \|e_i - e_j\|_2 \tag{2}$$

where $\mathcal{E}_k(x)$ is the set of the explanations obtained by calling the method $k = 5$ times on the same input instance x. A low same-instance instability is obtained when similar explanations are returned over multiple runs. Instances and explanations are normalized before calculating the instability measure.

Coverage and Precision These measures apply to rule-based explanations $p \rightarrow y$ only (for LORE$_{sa}$ we consider only the factual rule). Let Z be the neighborhood of x generated by the local method. The coverage of the explanation is the proportion of instances in Z that satisfy p. The precision is the proportion of instances $z \in Z$ satisfying p such that $b(z) = y$. Coverage and precision are competing metrics which respectively estimate the generality of the rule and the probability it correctly models the black-box behavior locally to the instance to explain. They depend both on the characteristics of the neighborhood generation (Sect. 4.1) and on the predictive performance of the learning algorithm (Sect. 4.2).

Changes An indicator of the quality of a counterfactual is the number of changes w.r.t. the instance x. For a set of counterfactual instances, such as those provided by SOC, we count the mean number of features whose value is different from x. For a set of counterfactual rules $p[\delta] \rightarrow y$, provided by LORE$_{sa}$, we count the mean number of falsified split conditions $nf(p[\delta], x)$. For LORE$_{sa}$, we expect a small number of changes thanks to the selection of counterfactual paths in the surrogate predictor with minimum number of changes (Sect. 4.3). However, actionability of counterfactuals maybe achieved at the cost of a larger number of changes (Sect. 4.4).

Dissimilarity We measures the proximity between x and the counterfactual x' generated as the distance between x and the counterfactual instance x' that we obtain by applying to x the changes described by $p[\delta]$. We calculate the distance using the same function described in Sect. 4.1. The lower the better.

Plausibility We evaluate the plausibility of the explanations in terms of the goodness of the counterfactuals returned by using the following metrics based on distance and outlierness Guidotti and Monreale (2020).

$x =$ {age = 20, priors_cnt = 3, days_b_screen_arrest = 0,
is_recid = 1, is_violent_recid = 0, two_year_recid = 1,
length_of_stay = 1, age_cat = Less than 25, sex = Male,
race = Afr.-Am., charge_degree = F}
$b(x) =$ High

LORE$_{sa}$

$r =$ {age ≤ 20.5, days_b_screen_arrest ≤ 0.50,
priors_cnt > 2.5, 0.5 < length_of_stay ≤ 3.5,
sex = Male, race ≠ Asian } → High
Φ ={ { age > 28.5 } → Low, {age > 23.5} → Medium,
{sex = Female} → Medium, {race = Asian} → Medium
{days_b_screen_arrest > 7.5} → Medium,
{priors_cnt ≤ 2.50} → Medium } }
Φ^* ={ { days_b_screen_arrest > 7.5 } → Medium,
{ priors_cnt ≤ 2.50 } → Medium }

ANCHOR

$e =$ {age_cat=Less than 25 > 0.0, age_cat=25 - 45 ≤ 0.0,
age ≤ 25.0, days_b_screen_arrest ≤ 1.0,
0.0 < race=Afr.-Am. ≤ 1.0, race=Caucasian ≤ 0.0,
race=Asian ≤ 0.0, race=Other ≤ 0.0,
priors_cnt > 2.0, 0.0 < length_of_stay ≤ 1.0,
0.0 < two_year_recid ≤ 1.0, 0.0 < is_recid ≤ 1.0,
0.0 < charge_degree=F ≤ 1.0, charge_degree=M ≤ 0.0,
sex=Female ≤ 0.0, sex=Male ≤ 1.0 } → High

BRL

$e =$ {12.04 < age ≤ 34.41} → High

LIME

$e =$ [(age, 0.02), (priors_cnt, -0.01),
(is_violent_recid, 0.002),
(days_b_screen_arrest, 0.002), (sex=Male, -0.002),
(charge_degree=M, 0.002), (is_recid, 0.001),
(age_cat=25 - 45, 0.001),
(charge_degree=F, -0.001)]

SHAP

$e =$ [(age, -0.7)]

MAPLE

$e =$ [(age, -1.84), (priors_cnt, 0.19),
(two_year_recid, 0.04),
(days_b_screen_arrest, -0.28), (sex=Female, 0.06),
...
(charge_degree=F, -0.02),
(charge_degree=M, 0.02)]

SOC

$e =$ {[(age, 17.31)], [(priors_cnt, 4.34)],
[(age_cat=25 - 45, 1.0), (race=Other, 1.0)],
[(is_violent_recid, 10), (charge_degree=M, 10.)] }
$b(x[e]) =$ Medium

Fig. 5 Explanations for an instance x of the `compas-m` dataset classified as *High* risk of recidivism by a NN black-box

Minimum Distance Metric As a straightforward but effective evaluation measure, we adopt proximity. Given the counterfactual x' returned for instance x, x' is plausible if it is not too much different from the most similar instance in a given reference dataset X. Hence, for a given explained instance x, we calculate the plausibility in terms of Minimum Distance $MDM = \min_{\bar{x} \in X/\{x\}} d(x', \bar{x})$ where the lower the MDM, the more plausible is x' the more reliable is the explanation, because x' resembles a real instance in X.

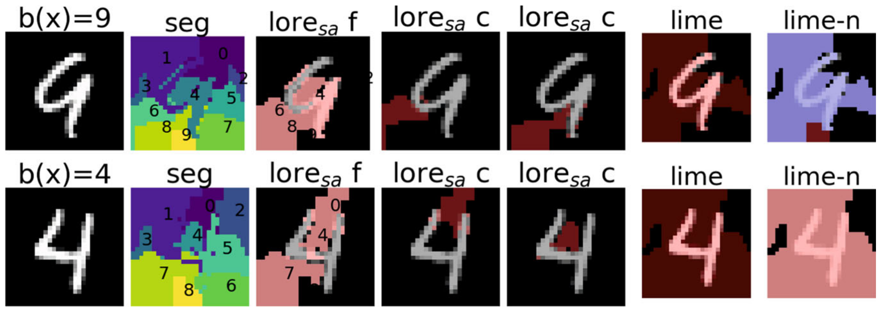
Outlier Detection Metrics We also evaluate the plausibility of the counterfactuals by judging how much they appears as outliers. The lower the scores the more plausible they are. In particular, we estimate the degree of outlierness of a counterfactual x' returned for an instance x by employing the outlier detection technique Isolation Forest (IsoFor) Liu et al. (2008).

5.3 Qualitative evaluation

We qualitatively compare LORE$_{sa}$ explanations with those returned by competitors on an instance x of the `compas-m` dataset, assuming a NN as the black-box. The instance and the explanations are shown in Fig. 5.

The factual rule r of LORE$_{sa}$ clarifies that x is considered at high risk of recidivism because of his young *age* and of the number of *previous detections*. The counterfactuals Φ show that the risk would have been lowered to *Low* for an older individual, or *Medium* for various reasons some of which are not actionable, e.g., different age, sex or race. The counterfactuals Φ^* are obtained by considering the set of constraints U={ *age=20, age_cat=Less than 25, race=Afr.-Am., sex=Male*}. In this case, the decision $b(x)$ would have been different only with a lower number of prior arrests or with a larger number of days between the screening and the arrest.

The competitor rule-based explainers suffer from a few weaknesses. ANCHOR returns various conditions, involving many features, in order to guarantee high precision. Thus, its explanation result hard to read and unnecessarily complex. BRL bases



LORE$_{sa}$

$$
\begin{aligned}
r_1 &= \{s_8,\, s_6,\, s_4\} \rightarrow \{9\}\\
\Phi_1 &= \{\ \{\neg s_6\} \rightarrow \{4\},\ \{\neg s_8\} \rightarrow \{6\}\ \}\\
r_2 &= \{s_7,\, s_0,\, s_4\} \rightarrow \{4\}\\
\Phi_2 &= \{\ \{\neg s_0\} \rightarrow \{9\},\ \{\neg s_4\} \rightarrow \{9\}\ \}
\end{aligned}
$$

Fig. 6 Explanations of LORE$_{sa}$ and LIME for two instances x (one per row) of the mnist dataset classified as *9* and *4* by a RF black-box. Meaning of columns is 1st: instance x, 2nd: superpixel segmentation, 3rd: LORE$_{sa}$ factual rule, 4–5th: LORE$_{sa}$ counterfactuals, 6^{th}: LIME explanation, 7th: LIME counterfactuals (towars unspecified class)

its explanation on a rule with a single feature, which on the example instance is *age*. Even though it is (partly) correct, the user can hardly trust such a simple and minimal justification. We will show next that BRL is indeed not particularly good in mimicking black-boxes' behaviors. The feature importance-based explainers LIME, SHAP and MAPLE provide a list of features with a score of their relevance in the decision. The most important features for LIME, i.e., *age* and *priors_cnt*, are in line with the factual rule of LORE$_{sa}$. SHAP attributes the decision of the black-box only to *age*. MAPLE provides a (unnecessarily long) list of features (shortened for space reasons) with scores in agreement with the other explainers. Regarding counterfactuals, SOC suggests a set of changes to x's feature values turning the risk prediction to *Medium*. Compared to $\Phi*$, the changes are either non-actionable (e.g., *age*=17.31) or less informative or impossible (e.g., *priors_cnt*=4.34).

Explanations on Images, Texts & Multi-label Data We compare LORE$_{sa}$ explanations for images and texts with LIME explanations.

Figure 6 shows such comparison on two images of Fig.mnist. Both methods adopt the same segmentation shown in the second column of the figure. The factual explanations of LORE$_{sa}$, shown visually in the 3rd column of Fig. 6, clearly attribute the classifications for *9* and *4* to the presence of super-pixels s_8, s_6, s_4 and s_7, s_0, s_4, respectively. The absence of some of such super-pixels (4th column), would have changed the black-box decision as shown in Φ_1 and Φ_2. For instance, the image of *9* would have been classified as *4* if the area of the super-pixel s_6 would have been white. The explanation returned by LIME are less intuitive both when considering only the super-pixels pushing the classification towards a class (5th column), or pushing the classification towards another (unspecified) class (6th column).

Figure 7 reports the explanations of LORE$_{sa}$, LIME and ANCHOR for a text from the 20news dataset. All methods adopt the same document vectorization. LORE$_{sa}$ shows that the text is classified as *atheism* because of the simultaneous presence of some

x = {*Could an atheist accept a usage in which religious literature or tradition is viewed in a metaphorical way? [...] It's also entirely unclear, and to me quite unlikely, that one could take a contemporary religion like that and divorce the metaphoric potential from the literalism and absolutism it carries now in many cases.*}
$b(x)$ = *atheism*

LORE$_{sa}$
r = {*Christianity, com, religion, edu, religious, atheist, believes, cons*} → *atheism*
Φ ={ {¬ *religion*} → *christian*, {¬ *com*} → *christian*,
{¬ *religious*} → *christian*, {¬ *edu*} → *christian* }

LIME
e = *[(Christianity, 0.05), (want, 0.04), (edu, -0.02), (com, -0.02), (good, 0.02), (religion, -0.02)]*

ANCHOR
e = {*religion, religious, set, an* } → *atheism*

Fig. 7 Explanations of LORE$_{sa}$ and LIME for an instance x of the `20news` dataset classified as *atheism* by a NN black-box

words in the factual rule. The absence of specific words in the counterfactual rules would change the classification to *christian*. LIME explanation is in agreement with the one of LORE$_{sa}$ as the words *edu*, *com* and *religion* have negative weight on the classification towards *atheism*. The explanation of ANCHOR highlights the presence of *religion* and *religious*, but it also includes less meaningful words.

Figure 8 reports an example of explanation derived for multi-labelled classification using the `medical` dataset. The instance x is labelled with the diseases corresponding to *Class 12* and *Class 38*. The explanation is the conjunction of symptoms in the factual rule r. A single label would have been returned by the black-box if *cough* were absent and, either *pneumonia* were absent or *hypertrophy* were present. We cannot compare with SOC, because it is not able to deal with multi-labelled classification.

In conclusion, we believe that the reported examples of factual, counterfactual, and actionable explanations of LORE$_{sa}$ offer to the user a clearer and more trustable understanding than what is offered by the other explainers.

x = {*15-month, chest, cough, fever, focal, male, normal, pneumonia, x-ray*}
$b(x)$ = {*Class 12, Class 38*}
r = {¬ *hydronephrosis, cough, fever, minimal* } → { *Class 12, Class 38* }
Φ ={ {¬ *cough*, ¬ *pneumonia*} → {*Class 12*},
{¬ *cough, hypertrophy*} → {*Class 38*} }

Fig. 8 LORE$_{sa}$ explanations for an instance x of the `medical` dataset classified as *Class 12* and *Class 38* by a RF black-box

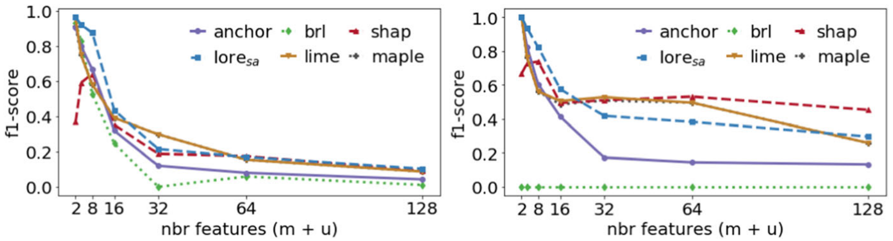


Fig. 9 Correctness metric by varying the total number of features $m+u$. *Left:* synthetic rule-based classifiers. *Right:* synthetic linear regressors

5.4 Ground truth validation

By synthetically generating transparent classifiers and using them as black-boxes, we can compare the explanations provided by an explainer with the ground-truth decision logic of the black-box (Guidotti 2021). In particular, the *f1-score*() metric accounts for the correctness of the explanations.

In order to have a comparison as fair as possible among methods returning different types of explanations, we build two types of black-boxes: rule-based classifiers and linear regressor-based. The former are closer to rule-based explainers, the latter to feature importance explainers. In both cases, we start from datasets of m binary informative features and u Gaussian-noise uninformative features. The total number of features $m+u$ varies over $\{2, 4, 8, 16, 32, 64, 128\}$ and, for a fixed $m+u$, we generate 100+100 such datasets where $m < \min\{32, m+u\}$.[24] The informative features are generated following the approach of Guyon (2003) implemented in `scikit-learn`.[25] Thus, we have 700 synthetic datasets for training rule-based classifiers and 700 for training linear regressors. Each dataset contains 10,000 instances, 1000 of which are used as explanation set.

Rule-based black-boxes are obtained by training a decision tree from a synthetic dataset, and then extracting rules from such a decision tree. The ground-truth explanation for an instance x is the rule satisfied by x in the black-box. Linear regressors black-boxes are obtained by an adaption of the approach of Klimke (2003). The ground-truth explanation for an instance x is the gradient of the instance in the decision boundary closest to x. Additional details[26] can be found in Guidotti (2021).

Figure 9 reports the *f1-score* metric at the variation of the total number of features $m+u$ in synthetic datasets. Each point shows the mean *f1-score* over the explanation sets of such datasets. LORE$_{sa}$ outperforms the other explainers when $m+u \leq 16$. For larger values of $m+u$, LORE$_{sa}$ performance is comparable to those of LIME and

[24] We specified 32 as maximum number of features m because typically tabular datasets with columns having clear and interpretable semantics have less than 30 features (like those used in the experiments). Thus, since our purpose is not to perform a scalability test but a correctness test, we selected this upper limits.

[25] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_classification.html.

[26] We highlight that the transformation of features importance and of rules into binary vectors indicating the presence of a feature is a simplification adopted to make possible the comparison of explainers returning different types of explanations using the same metric.

Table 3 Aggregated evaluation metrics over experimental datasets and black-boxes

	silhouette	*fidelity*	*complexity*	*instability*	$instability_{si}$
ANCHOR	.116 ± .51	.912 ± .21	4.950 ± 8.20	.174 ± 0.29	.651 ± .949
BRL	.019 ± .30	.869 ± .09	**1.998 ± 1.23**	.889 ± 0.45	n.a.
LIME	.444 ± .49	.904 ± .23	9.733 ± 1.47	.787 ± 1.58	.159 ± .142
LORE	.408 ± .49	.996 ± .01	4.917 ± 3.69	.123 ± 0.22	.259 ± .847
MAPLE	.127 ± .56	.949 ± .09	29.014 ± 3.25	.651 ± 1.66	n.a.
SHAP	.463 ± .56	n.a.	6.070 ± 3.84	.608 ± 0.58	**.017 ± .052**
LORE$_{sa}$	**.569 ± .46**	.992 ± .20	3.986 ± 3.93	**.073 ± 0.07**	.107 ± .081
LORE$^{d}_{sa}$	**.569 ± .46**	**.999 ± .01**	5.105 ± 4.29	.083 ± 0.08	.107 ± .066

	ANCHOR	LORE	BRL	LORE$_{sa}$	LORE$^{d}_{sa}$
coverage	.284 ± .32	.492 ± .27	.344 ± .30	**.742 ± .27**	.485 ± .26
precision	.912 ± .21	.993 ± .07	.732 ± .22	.772 ± .26	**.998 ± .02**
h-mean	.433 ± .25	.657 ± .11	.468 ± .25	**.694 ± .25**	.615 ± .22

Bold value indicates the best perfomance

SHAP for rule-based classifiers, and slightly lower than their performance for linear regressors.

5.5 Quantitative evaluation

We quantitatively assess the quality of LORE$_{sa}$ and of the competitor explainers through the other evaluation metrics of Sect. 5.2.

In order to evaluate the importance of the trees merging strategy employed by LORE$_{sa}$ for deriving the single local decision tree, we implemented a variant that avoids the merging operation. We call it LORE$^{d}_{sa}$ and works as follows. After learning the decision trees $d^{(1)}, d^{(2)}, \ldots, d^{(N)}$ on their corresponding local neighborhood $Z^{(1)}, Z^{(2)}, \ldots, Z^{(N)}$ labeled by the back-box b, we use each tree $d^{(i)}$ for labeling its training neighborhoods, $Z^{(i)}$, i.e., $Y^{(i)}_d = d^{(i)}(Z^{(i)})$. Then, we compute the union of the new labeled neighbors, i.e., $D^Z = \bigcup_{\forall i \in [1,N]} (Z^{(i)}, Y^{(i)}_d)$ and we use D^Z to learn the final decision tree c.

For sake of compactness, to quantitatively compare all the explanation methods we report only aggregate results, i.e., mean and standard deviation of the metrics over all datasets and black-boxes. Table 3 (top) reports the *silhouette*, *fidelity*, *complexity*, *instability*, and $instability_{si}$ metrics. LORE$_{sa}$ overcomes all the other explainers on 3 metrics, and it is runner-up on the other 2 metrics. As expected, LORE$_{sa}$ considerably improves the *complexity* and the two *instability* metrics with respect to its predecessor LORE while maintaining the same level of *fidelity*. In terms of *complexity*, LORE$_{sa}$ is the second best performer after the BRL approach which, on the other hand, has lower performance on the other metrics and is one of the most stable. The only competitors with lower *instability* are SHAP and MAPLE which provide more complex explanations. Moreover, our experimental results show that LORE$_{sa}$ has also lower complexity and

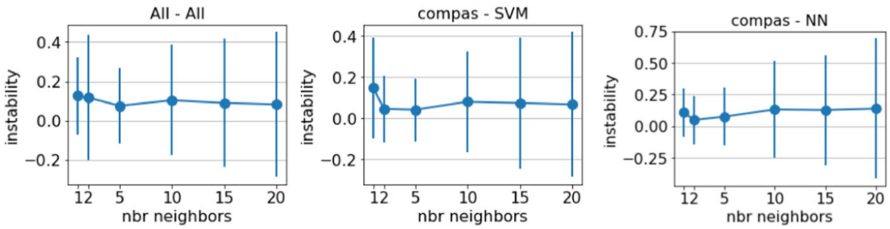


Fig. 10 Instability metric by varying the number N of decision trees in LORE$_{sa}$

instability with respect to LORE$_{sa}^{d}$ highlighting the importance of the merging procedure for the stability. The better performance of LORE$_{sa}$ is paid with a slightly higher runtime required to get an explanation due to the merging procedure that is on average 315.59 ± 185.74 seconds among all datasets and black-box models 315.59, while it is on average 285.23 ± 179.83 seconds for LORE$_{sa}^{d}$. We underline that having an efficient implementation is out form the purpose of this paper and that, how specified in the "Appendix", several possibilities are available to speed up the calculus through parallelization of the explanation process. Figure 10 shows how instability behaves varying the number N of local neighborhoods/decision trees generated by LORE$_{sa}$. Similar results are obtained for LORE$_{sa}^{d}$. There is a (local) minimum at $N = 5$, which is the value set by default in LORE$_{sa}$. Finally, with respect to the *instability*$_{si}$ metric,[27] we point out that BRL and MAPLE are deterministic methods, hence the metric does not apply to them. SHAP, which has the best performances, bases its explanation process on permutations of x with respect to a set of *base values*. Using a single background value as the medoid of the training set, as suggested in SHAP tutorials' can markedly limit the variability of the permutations of x. This explains the low *instability*$_{si}$ value. On the other hand, different background values could lead to different explanations (Gosiewska and Biecek 2020; Sundararajan and Najmi 2020).

In Table 3 (bottom) we report the *coverage* and *precision* metrics for the rule-based explainers under analysis. Furthermore, to capture both measures with a single value, we also report the *harmonic mean (h-mean)* of coverage and precision. We notice that, LORE$_{sa}$, LORE$_{sa}^{d}$ and LORE overcome ANCHOR and BRL for both indicators. LORE$_{sa}$ considerably improves the rule coverage paying something in precision; however, looking at the *h-mean* LORE$_{sa}$ is the best performer. This is another beneficial effect of the bagging-like approach, which improves on generality (less overfitting) of the interpretable predictor. A Friedman test (Demsar 2006) on each of the metrics rejects the null hypothesis of zero difference among the methods (p *value* $< 10^{-5}$). Further evidence is reported in "Appendix D".

Table 4 compares LORE$_{sa}$ with the merging variant and with two competitors with respect to the counterfactual part of the explanation. We highlight that LORE$_{sa}$ is not an explainer directly returning counterfactual instances on its own. However, counterfactual instances can be created by modifying the instance under analysis x according to the counterfactual rules in Φ. We notice that the brute force approach BF has the lowest

[27] Differently from *instability*, the *instability*$_{si}$ metric is not normalized—see (1), (2). Hence, the columns for the two metrics in Table 3 (top) cannot be compared to each other.

Table 4 Aggregated evaluation metrics estimating the proximity of the counterfactual explanations in terms of dissimilarity and the plausibility as MDM and IF scores

	SOC	BF	LORE$_{sa}$	LORE$^{d}_{sa}$
dissimilarity	0.170 ± 0.27	**0.056 ± 0.07**	*0.093* ± **0.03**	0.111 ± 0.00
MDM	0.166 ± 0.28	0.067 ± 0.08	*0.026 ± 0.00*	**0.019 ± 0.01**
IsoFor	1.074 ± 0.09	1.221 ± 0.36	**1.007 ± 0.00**	*1.060 ± 0.07*

The lower the better for all the measures: in **bold** the best performer, in *italic* the runner up

Table 5 Aggregated evaluation metrics for counterfactuals over experimental datasets and black-boxes

X	Explainer	*no. cf.*	*changes*	b	Explainer	*no. cf.*	*changes*
`adult`	SOC	9.6 ± 1.0	3.8 ± 2.7	DNN	SOC	9.9 ± 0.7	2.6 ± 1.7
	LORE$_{sa}$	2.9 ± 2.7	1.3 ± 0.5		LORE$_{sa}$	2.4 ± 1.6	1.2 ± 0.5
	LORE$_{s\underline{a}}$	1.8 ± 1.7	2.2 ± 0.4		LORE$_{s\underline{a}}$	1.2 ± 0.4	2.5 ± 0.5
`bank`	SOC	3.5 ± 2.4	1.6 ± 0.6	NN	SOC	7.2 ± 2.4	5.0 ± 3.3
	LORE$_{sa}$	1.4 ± 0.9	1.3 ± 0.5		LORE$_{sa}$	2.5 ± 2.4	1.3 ± 0.5
	LORE$_{s\underline{a}}$	1.6 ± 0.8	1.5 ± 0.2		LORE$_{s\underline{a}}$	1.4 ± 0.9	2.2 ± 0.4
`churn`	SOC	8.4 ± 1.8	5.8 ± 3.7	RF	SOC	7.5 ± 2.3	3.6 ± 2.7
	LORE$_{sa}$	2.0 ± 1.9	1.5 ± 0.7		LORE$_{sa}$	2.4 ± 2.1	1.3 ± 0.6
	LORE$_{s\underline{a}}$	1.5 ± 0.9	2.3 ± 0.5		LORE$_{s\underline{a}}$	1.9 ± 1.2	2.2 ± 0.5
`cps-m`	SOC	5.2 ± 1.8	2.9 ± 1.4	SVM	SOC	6.9 ± 3.5	3.2 ± 2.7
	LORE$_{sa}$	3.5 ± 2.2	1.1 ± 0.3		LORE$_{sa}$	3.0 ± 2.3	1.2 ± 0.5
	LORE$_{s\underline{a}}$	1.8 ± 1.1	1.3 ± 0.2		LORE$_{s\underline{a}}$	1.6 ± 1.1	2.2 ± 0.4

LORE$_{s\underline{a}}$ is LORE$_{sa}$ with constraints U in input

dissimilarity but LORE$_{sa}$ and LORE$^{d}_{sa}$ achieve closer results. SOC is the worst performer with respect to this metric, meaning that the counterfactual instances returned by SOC are not highlighting minimal changes with respect to x to change decision. Furthermore, LORE$_{sa}$ alternatives return the most plausible counterfactuals with respect to the the MDM and IsoFor metrics. There is not a clear winner but overall the plausibility scores of LORE$_{sa}$ are better being always the best performer or the runner up, i.e., lower, than those of BF and SOC, enabling it to be used also as a possible counterfactual explainer.

In Table 5, we compare LORE$_{sa}$ with the counterfactual explainer SOC that is typically used as a baseline (Guidotti 2022). Mean and standard deviations are reported for the number of counterfactual instances (SOC) or counterfactual rules (LORE$_{sa}$) produced, and the changes metrics (number of changes to instance x to revert the black-box outcome). For all the reported datasets and black-boxes, LORE$_{sa}$ produce less changes than SOC. On the other hand, SOC returns more counterfactuals. The number of counterfactuals returned by LORE$_{sa}$ could be increased trading off with changes, simply by relaxing the requirement of minimality in Algorithm 3. Let us now denote with LORE$_{s\underline{a}}$ with underlined $\underline{a}$ the execution of LORE$_{sa}$ with in input dataset-specific constraints U stating features that cannot be changed: *age*, *race*, *sex*, *native-country*, *marital-status* for adult; *age* for bank; *state*, *state-area*, *state* for churn; *age*, *age-cat*, *race*, *sex* for compas-m (shown as cps-m in the table). As expected, it turns out that LORE$_{s\underline{a}}$ produces less counterfactuals its counterpart ignoring the actionability. This is due to the filtering of the counterfactual rules that do not satisfy the feature constraints. On average, such counterfactual require more changes to the instance x to explain, but still less than SOC.

6 Conclusion

We have proposed LORE$_{sa}$, a black-box agnostic method for local explanations providing informative factual decision rules and actionable counterfactual rules. An ample experimental evaluation with state-of-the-art methods has shown that LORE$_{sa}$ largely improves as per stability of explanations, while ranking top or runner-up in several other quantitative metrics. Stability of the provided explanations is achieved by adopting a novel bagging-like approach in generating and aggregating several local decision trees.

A few directions can be mentioned as future work to expand the applicability of LORE$_{sa}$. *First*, synthetically generated instances may not respect correlations among attributes (e.g., age and education level). Hence, it is worth extending the approach by integrating domain knowledge (dependencies or causal relationships) among attributes in the neighborhood generation and/or in the inference of the interpretable predictor. *Second*, in multi-class problems, alternative definitions of $fitness_{\neq}$ could be implemented to drive the selection of counterfactual rules towards some specific class value. E.g., in a credit risk rating context, to provide counterfactuals toward a lower risk label. *Third*, the adaptation of LORE$_{sa}$ to images and texts with a simple binary encoding, modeling presence/absence of a super-pixel/word, suffers from the same problems as LIME, namely the generation of unrealistic synthetic instances. More complex encoding using autoencoders can be used to overcome these limitations and to produce neighborhoods of realistic images and texts (Guidotti et al. 2019c). *Finally*, LORE$_{sa}$ assumes that the black-box can be queried as many times as necessary. When this is not the case, the neighborhood generation phase must take into account constraints on the number of admissible queries, e.g., by adopting an active learning variant of the genetic approach.

Author Contributions RG: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing—Original Draft, Writing—Review and Editing, Visualization. AM: Conceptualization, Methodology, Resources, Writing—Original Draft, Writing—Review and Editing. SR: Conceptualization, Methodology, Resources, Writing—Original Draft, Writing—Review and Editing. FN: Software, Validation, Experiments. FT: Conceptualization, Supervision, Resources, Writing—Review and Editing. DP: Conceptualization, Supervision, Project administration, Funding acquisition. FG: Conceptualization, Supervision, Project administration, Funding acquisition.

Funding Open access funding provided by Università di Pisa within the CRUI-CARE Agreement. This work has been partially supported by the European Community Horizon 2020 programme under the funding schemes: G.A. 871042 *SoBigData++* (sobigdata), G.A. 952026 *HumanE AI Net* (humane-ai), G.A. 834756 *XAI* (xai), G.A. 871042 *NoBIAS* (nobias), G.A. 952215 *TAILOR* (tailor), and CHIST-ERA grant *CHIST-ERA-19-XAI-010*, by MUR (N. not yet available), FWF (N. I 5205), EPSRC (N.

EP/V055712/1), NCN (N. 2020/02/Y/ST6/00064), ETAg (N. SLTAT21096), BNSF (N. KP-06-AOO2/5) SAI (sai).

Availability of data and material The datasets adopted in this work are open source and available at https://archive.ics.uci.edu/ml/datasets.php for adult, bank, churn, german, iris, wine-r, and wine-w https://www.kaggle.com/datasets for compas, compas-m, and medical, https://tinyurl.com/d43jymx2 for fico, http://qwone.com/\protect\unhbox\voidb@x\penalty\@M\jason/20Newsgroups/ for 20news, and http://yann.lecun.com/exdb/mnist/ for mnist.

Code availability The code is open source, and can be downloaded at https://github.com/francescanaretto/LORE_sa.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication The authors declare that they all provide consent for publication.

A Neighborhood generation

In Sect. 5, we contrasted LORE$_{sa}$ with LORE, which adopts a simplified fitness function where the distance between class probability vectors (namely, $l(b_p(x), b_p(z))$ in $fitness^x_=(z)$) is replaced by identity of the predictions (namely, $I_{b(x)=b(z)}$), and which consider a single neighborhood/decision tree. Here, we conduct an ablation study by contrasting LORE$_{sa}$ with a version of it including a purely random neighborhood generation, called RND$_{sa}$. Table 6 reports the evaluation metrics described in Sect. 5.2 for the experimental datasets and black-box classifiers. The best results are highlighted in bold. LORE$_{sa}$ has nearly always the best *silhouette*, *fidelity*, *precision*, and *instability*. The best performing silhouette of LORE$_{sa}$ confirms that the genetic neighborhood generation strategy leads to dense neighborhoods with instances that separate the local decision boundary and they are still similar to the one to explain. Hence, it is not just a marginal improvement over random generation but a key feature to achieve compactness and separation of the neighborhoods $Z_=$ and $Z_{\neq}$. Other basic techniques of neighborhood generation, such as oversampling and instance selection, are also overcome by the genetic approach, as shown in Guidotti et al. (2019b). For black-box aggregations, RND$_{sa}$ shows the best *coverage* and *complexity*. This can be attributed

Table 6 Aggregated evaluation metrics over datasets (top) and black-boxes (bottom)

	Method	*silhouette*	*fidelity*	*coverage*	*precision*	*complexity*	*instability*
X							
adu.	LORE$_{sa}$	**.16 ± .91**	**.99 ± .02**	**.57 ± .28**	**.96 ± .16**	**3.74 ± 3.17**	**.35 ± 1.03**
	RND$_{sa}$	.12 ± .90	.87 ± .04	.56 ± .11	.87 ± .04	4.67 ± .59	.40 ± .42
comp.	LORE$_{sa}$	**.54 ± .22**	**.99 ± .00**	**.44 ± .16**	**1.00 ± .03**	4.97 ± 2.15	.19 ± .27
	RND$_{sa}$	.14 ± .22	.93 ± .09	.39 ± .22	.85 ± .16	**3.84 ± 1.25**	**.19 ± .22**
ger.	LORE$_{sa}$	**.70 ± .57**	**1.00 ± .00**	.87 ± .11	**1.00 ± .00**	**.98 ± .84**	**.80 ± 1.97**
	RND$_{sa}$	.56 ± .71	.89 ± .17	**.88 ± .24**	.89 ± .17	1.29 ± .57	1.26 ± .54
b							
DNN	LORE$_{sa}$	**.61 ± .17**	**.99 ± .01**	.55 ± .20	**1.00 ± .02**	6.96 ± 3.97	.12 ± .38
	RND$_{sa}$	.25 ± .78	.85 ± .13	**.83 ± .26**	.85 ± .14	**3.46 ± .80**	**.12 ± .24**
NN	LORE$_{sa}$	.50 ± .27	**.99 ± .01**	.32 ± .18	**.99 ± .10**	6.93 ± 3.79	**.84 ± .99**
	RND$_{sa}$	.40 ± .70	.88 ± .11	**.83 ± .27**	.87 ± .15	**3.09 ± 1.24**	.85 ± 1.41
RF	LORE$_{sa}$	**.69 ± .13**	**.97 ± .02**	.36 ± .12	**.98 ± .06**	3.36 ± 1.86	**.76 ± .13**
	RND$_{sa}$	.57 ± .51	.76 ± .13	**.67 ± .29**	.78 ± .15	**3.26 ± 1.00**	.79 ± .31

Bold value indicates the best perfomance

to a weaker, hence simpler, decision boundary characterization of the purely random strategy.

B Impact of distance functions

A key element of the neighborhood generation is the distance function used by the genetic algorithm. In this section we show how the explanations of LORE$_{sa}$ are affected by different distance functions. For example, Wachter (2017) shows that considerable differences of the counterfactual instances occur at the variation of the distance function adopted by their stochastic optimization approach. As alternative distances to the normalized euclidean distance (*neucliden*) adopted by LORE$_{sa}$, we report results using the *cosine* distance and the normalized mean deviation (*nmeandev*) distance. Experiments over the `compas`, `fico` and `german` datasets, and over DNN, NN, and SVM black-boxes are reported in Table 7. There is no major difference in terms of fidelity and precision, whilst *neucliden* has the best performance or is a close runner up for all other metrics.

C Genetic algorithm parameters

We investigate on the impact of the parameters of the genetic Algorithm 2: (i) the neighborhood size n, (ii) the crossover probability p_c, (iii) the mutation probability p_m, and (iv) the number of generations g. We vary one parameter at a time while keeping the others fixed at their default value, i.e., $n = 500$, $p_c = 0.7$, $p_m = 0.5$,

Table 7 Aggregated evaluation metrics over datasets (top) and black-boxes (bottom) w.r.t. distance functions in the neighborhood generation of LORE$_{sa}$

	Distance	*silhouette*	*fidelity*	*coverage*	*precision*	*complexity*	*instability*
X							
compas	*neuclidean*	**.54 ± .22**	**.99 ± .00**	**.44 ± .16**	1.00 ± .03	**4.97 ± 2.15**	**.21 ± .32**
	cosine	.50 ± .24	**.99 ± .00**	.43 ± .16	**1.00 ± .02**	5.00 ± 2.11	.24 ± .39
	nmeandev	.27 ± .26	**.99 ± .00**	.29 ± .18	.99 ± .11	5.10 ± 1.86	.24 ± .44
fico	*neuclidean*	.52 ± .17	**.98 ± .01**	**.40 ± .21**	.98 ± .10	**9.49 ± 3.77**	**.07 ± .04**
	cosine	**.54 ± .12**	**.98 ± .01**	.39 ± .19	**.99 ± .07**	9.88 ± 3.66	.27 ± .31
	nmeandev	.14 ± .17	**.98 ± .01**	.19 ± .19	.94 ± .21	9.78 ± 3.52	.18 ± .16
german	*neuclidean*	**.70 ± .57**	**1.00 ± .00**	**.87 ± .11**	**1.00 ± .00**	.98 ± .84	**.80 ± 1.97**
	cosine	.66 ± .57	**1.00 ± .00**	.78 ± .18	**1.00 ± .00**	1.09 ± .90	.97 ± 1.33
	nmeandev	.61 ± .60	**1.00 ± .00**	.85 ± .15	**1.00 ± .00**	**.73 ± .66**	.90 ± 1.27
b							
DNN	*neuclidean*	.61 ± .17	**.99 ± .01**	.55 ± .20	1.00 ± .02	6.96 ± 3.97	**.12 ± .38**
	cosine	**.62 ± .14**	**.99 ± .01**	**.56 ± .19**	**1.00 ± .01**	6.54 ± 3.82	.13 ± .38
	nmeandev	.12 ± .23	**.99 ± .01**	.21 ± .24	.96 ± .19	**6.22 ± 3.04**	.13 ± .45
NN	*neuclidean*	.50 ± .27	**.99 ± .01**	.32 ± .18	.99 ± .10	6.93 ± 3.79	**.84 ± .99**
	cosine	**.50 ± .24**	**.99 ± .00**	.32 ± .15	**.99 ± .07**	**6.88 ± 3.76**	1.08 ± 1.26
	nmeandev	.31 ± .31	**.99 ± .01**	**.37 ± .23**	.99 ± .09	6.93 ± 3.76	1.00 ± 1.17
SVM	*neuclidean*	**.48 ± .25**	**.99 ± .01**	.39 ± .17	.99 ± .10	**7.28 ± 4.18**	**.18 ± .09**
	cosine	.46 ± .27	**.99 ± .01**	**.39 ± .15**	**.99 ± .06**	7.32 ± 4.22	.56 ± .15
	nmeandev	.28 ± .27	**.99 ± .01**	.28 ± .21	.96 ± .18	7.56 ± 4.40	.22 ± .32

Bold value indicates the best perfomance

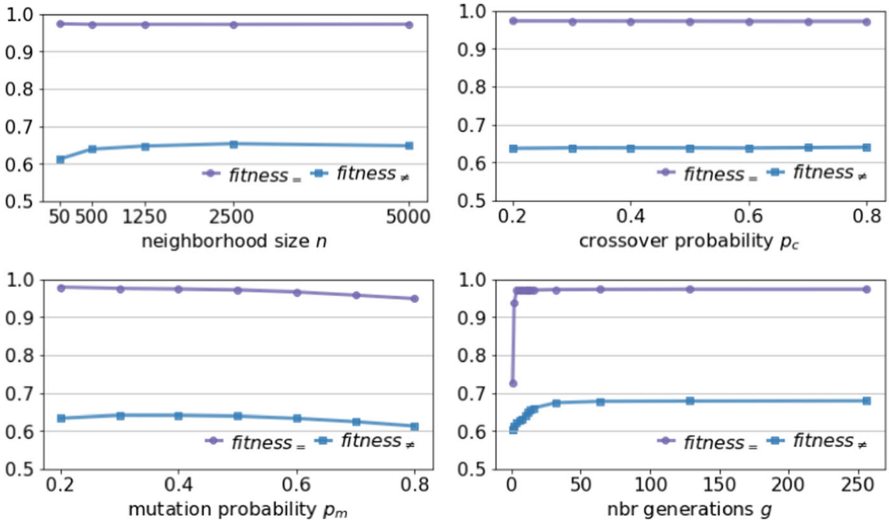


Fig. 11 Mean values of fitness functions at the last generation of Algorithm 2 on the compas dataset and DNN black-box by varying parameters

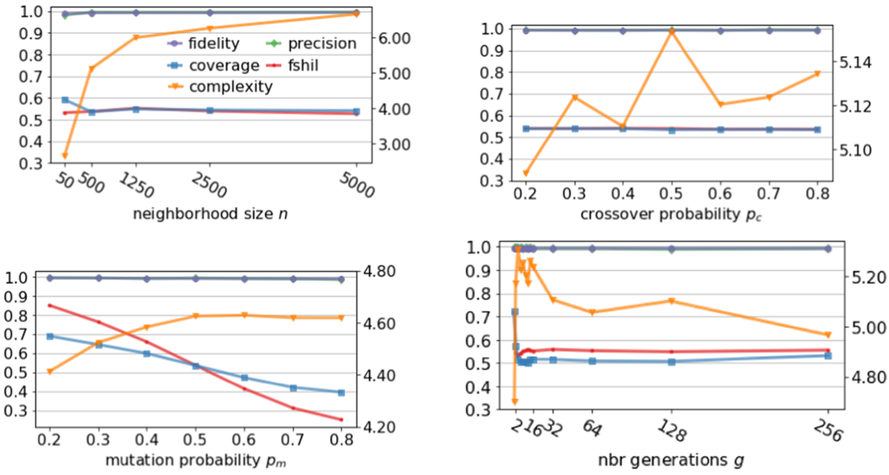


Fig. 12 Aggregated evaluation metrics on the `compas` dataset and DNN black-box by varying the parameters of Algorithm 2. Scale on the right y axis is for complexity

$g = 20$, $N = 5$ (recall that $n = 500$ implies that the neighborhood $Z^{(i)} = Z_{=}^{(i)} \cup Z_{\neq}^{(i)}$ consists of $2n = 1000$ instances). Experiments are performed on the `compas` dataset and using the RF as black-box. Figure 11 shows the average values of the two fitness functions at the last generation for the two components of the neighborhood: $Z_{=}$ and $Z_{\neq}$. Regarding n, values greater than 1000 do not lead to a net increase of the fitness functions. A similar behavior is observed for $g \approx 20$–50. On the other hand, varying crossover and mutation probabilities does not make appreciable impact on the fitness functions. To better understand their impact, we analyze in Fig. 12 some evaluation metrics while varying the parameters as before[28]. With respect to the neighborhood size n, for $n > 1000$, complexity grows remarkably, whist all other metrics become stable. Similarly,the crossover probability p_c does not affect any metric but the complexity. The mutation probability p_m appears negatively correlated to coverage, silhouette, and complexity. Finally, a very low number of generations g lead to bad results, while for $g > 10$, all the metrics become stable. In summary, the default values for the parameters $n = 1000$, $g = 20$, and $p_m = 0.5$ were chosen experimentally based on the previous discussion. Regarding the cross-over parameter $p_c = 0.7$, we departed from a 50–50 choice in favor of the diversity of generated instances.

D Statistical tests

Tables 8 and 9 report the mean rank values (ranging from 1 to 6) among the different explainers for a given dataset (resp., black-box) over all combinations of black-boxes (resp., datasets), and of the evaluation metrics of *silhouette*, *fidelity*, *complexity*, and *instability*. The first column of Table 8 reports the overall mean rank. It is readily checked that LORE$_{sa}$ ranks the best in general (p value < 0.001 using a Wilcoxon

[28] The plots show fidelity, precision, coverage, complexity, and silhouette labeled as fshil in the legend.

Table 8 Mean rank of explainers by dataset over all combinations of black-boxes and metrics (*silhouette*, *fidelity*, *complexity*, *instability*)

Method	ovr	adult	bank	churn	compas	compas-m	fico	german	iris	wine-r	wine-w
ANCHOR	3.12	3.08	3.24	3.24	3.05	3.11	3.11	3.80	4.14	3.02	2.82
BRL	3.72	3.65	3.87	3.53	4.36	3.79	3.38	3.56	4.01	3.91	3.50
LIME	3.74	3.10	4.08	3.88	4.53	4.15	3.54	2.81	4.15	3.62	3.90
MAPLE	3.46	3.54	3.21	2.70	3.36	4.26	3.85	2.11	3.48	4.41	3.92
SHAP	3.49	3.27	2.85	3.57	3.45	3.89	4.62	3.31	3.52	3.67	3.51
LORE$_{sa}$	**2.19**	**2.12**	**2.27**	**2.32**	**2.12**	**2.02**	**2.21**	**2.22**	**2.82**	**2.35**	**2.17**

Bold value indicates the best perfomance

Table 9 Mean rank over all combinations of datasets and evaluation metrics

Method	RF	SVM	NN	DNN
ANCHOR	3.02	3.04	3.33	3.41
BRL	3.72	3.77	3.96	1.72
LIME	3.93	3.03	4.25	3.33
MAPLE	3.67	4.00	3.61	**1.41**
SHAP	3.15	3.45	3.71	3.19
LORE$_{sa}$	**2.09**	**2.36**	**2.61**	**1.41**

Bold value indicates the best perfomance

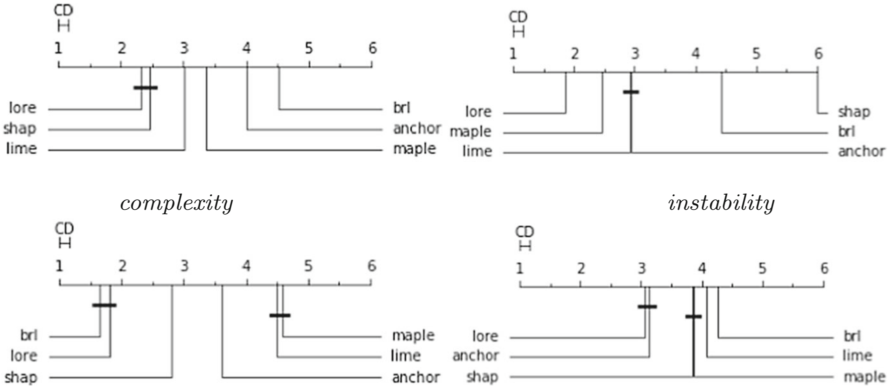


Fig. 13 Critical difference diagrams using the Nemenyi test at $\alpha = 0.05$. The name LORE in the plots indicate the LORE$_{sa}$ method

signed rank test), for each dataset, and for each black-box. For the compas-m, bank and fico datasets and for the RF and SVM black-boxes, LORE$_{sa}$ ranks markedly higher than the competitors. Figure 13 shows four Critical Difference (CD) diagrams (Demsar 2006). They display the statistical significance of the observed paired differences in performances between pairs of the explanation methods. Two methods are tied if the null hypothesis that their performances are the same cannot be rejected

Table 10 Running time (mean ± stdev) in secs for SVM

Method	compas	adult	german
ANCHOR	4.48 ± 6.43	101.60 ± 203.72	2.81 ± 0.84
LIME	1.49 ± 0.24	3.10 ± 0.83	**0.20 ± .03**
MAPLE	743.62 ± 0.02	34643.04 ± 0.01	273.08 ± 0.02
BRL	53.20 ± 0.01	621.20 ± 0.68	33.10 ± 0.02
SHAP	**0.29 ± 0.31**	**0.46 ± 0.60**	0.86 ± 0.15
SOC	3.52 ± 0.02	39.18 ± 2.80	4.72 ± 0.07
LORE$_{sa}$	8.02 ± 0.36	62.48 ± 4.55	7.76 ± 0.19

Bold value indicates the best perfomance

using the Nemenyi test at α=0.05. LORE$_{sa}$ performs better than the compared methods with regards to fidelity, and the differences are statistically significant. For each of the other metrics, the method tied to LORE$_{sa}$ is always a different one. Hence, LORE$_{sa}$ wins over any other method in at least 3 out of the 4 metrics.

E Running time

Table 10 reports the running time (in secs) of producing an explanation for three experimental datasets and for the SVM black-box. LORE$_{sa}$ performances are in line with ANCHOR and SOC, and better than MAPLE and BRL. They are instead worse than LIME and SHAP. The vast majority of running time ($> 90\%$) of LORE$_{sa}$ is used by the genetic neighborhood generation. The implementation, however, can be readily sped up by parallelising the generation of $Z_{=}^{(1)}, Z_{\neq}^{(1)}, \ldots, Z_{=}^{(N)}, Z_{\neq}^{(N)}$ ($2 \cdot N$ independent calls to Algorithm 2).

References

Adadi A, Berrada M (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access 6:52138–52160

Alvarez-Melis D, Jaakkola TS (2018) Towards robust interpretability with self-explaining neural networks. In: NeurIPS, pp 7786–7795

Angelino E, Larus-Stone N, Alabi D, Seltzer MI, Rudin C (2017) Learning certifiably optimal rule lists for categorical data. J Mach Learn Res 18:234:1-234:78

Assche AV, Blockeel H (2007) Seeing the forest through the trees: learning a comprehensible model from an ensemble. In: ECML. Lecture notes in computer science, vol 4701. Springer, pp 418–429

Bäck T, Fogel DB, Michalewicz Z (2000) Evolutionary computation 1: basic algorithms and operators, vol 1. CRC Press, Boca Raton

Bénard C, Biau G, Veiga SD, Scornet E (2019) SIRUS: making random forests interpretable. CoRR arXiv:1908.06852

Berk R, Heidari H, Jabbari S, Kearns M, Roth A (2018) Fairness in criminal justice risk assessments: the state of the art. Sociol Methods Res 50(1):3–44

Bhatt U, Xiang A, Sharma S, Weller A, Taly A, Jia Y, Ghosh J, Puri R, Moura JMF, Eckersley P (2020) Explainable machine learning in deployment. In: FAT*, ACM, pp 648–657

Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140

Breiman L (2001) Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat Sci 16:199–231

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and regression trees. Wadsworth, Belmont

Byrne RM (2016) Counterfactual thought. Annu Rev Psychol 67(1):135–157

Byrne RMJ, Johnson-Laird P (2009) "If" and the problems of conditional reasoning. Trends Cogn Sci 13(9):282–287

Calegari R, Ciatto G, Denti E, Omicini A (2020) Logic-based technologies for intelligent systems: state of the art and perspectives. Information 11(3):167

Chou Y, Moreira C, Bruza P, Ouyang C, Jorge JA (2022) Counterfactuals and causability in explainable artificial intelligence: theory, algorithms, and applications. Inf Fusion 81:59–83

Darwiche A, Hirth A (2020) On the reasons behind decisions. In: ECAI, IOS Press, frontiers in artificial intelligence and applications, vol 325, pp 712–720

Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30

Derrac J, García S, Herrera F (2010) A survey on evolutionary instance selection and generation. Int J Appl Metaheuristic Comput 1(1):60–92

Doshi-Velez F, Kim B (2017) A roadmap for a rigorous science of interpretability. CoRR arXiv:1702.08608

Evans BP, Xue B, Zhang M (2019) What's inside the black-box? A genetic programming method for interpreting complex machine learning models. In: GECCO, ACM, pp 1012–1020

Fan C et al (2020) Classification acceleration via merging decision trees. In: FODS, ACM, pp 13–22

Fortin F, Rainville FD, Gardner M, Parizeau M, Gagné C (2012) DEAP: evolutionary algorithms made easy. J Mach Learn Res 13:2171–2175

Freitas AA (2013) Comprehensible classification models: a position paper. SIGKDD Explor 15(1):1–10

Fu Y, Zhu X, Li B (2013) A survey on instance selection for active learning. Knowl Inf Syst 35(2):249–283

Gosiewska A, Biecek P (2020) Do not trust additive explanations. CoRR arXiv:1903.11420

Guidotti R (2021) Evaluating local explanation methods on ground truth. Artif Intell 291:103428

Guidotti R (2022) Counterfactual explanations and how to find them: literature review and benchmarking. Data Min Knowl Discov. https://doi.org/10.1007/s10618-022-00831-6

Guidotti R, Monreale A (2020) Data-agnostic local neighborhood generation. In: ICDM, IEEE, pp 1040–1045

Guidotti R, Ruggieri S (2019) On the stability of interpretable models. In: IJCNN, IEEE, pp 1–8

Guidotti R, Monreale A, Cariaggi L (2019a) Investigating neighborhood generation methods for explanations of obscure image classifiers. In: PAKDD (1). Lecture notes in computer science, vol 11439. Springer, pp 55–68

Guidotti R, Monreale A, Giannotti F, Pedreschi D, Ruggieri S, Turini F (2019b) Factual and counterfactual explanations for black box decision making. IEEE Intell Syst 34(6):14–23

Guidotti R, Monreale A, Matwin S, Pedreschi D (2019c) Black box explanation by learning image exemplars in the latent feature space. In: ECML/PKDD (1). Lecture notes in computer science, vol 11906. Springer, pp 189–205

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2019d) A survey of methods for explaining black box models. ACM Comput Surv 51(5):93:1-93:42

Guyon I (2003) Design of experiments of the NIPS 2003 variable selection benchmark. In: NIPS workshops

Holland JH (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge

Jia Y, Bailey J, Ramamohanarao K, Leckie C, Houle ME (2019) Improving the quality of explanations with local embedding perturbations. In: KDD, ACM, pp 875–884

Karimi A, Barthe G, Schölkopf B, Valera I (2020) A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. CoRR arXiv:2010.04050

Klimke A (2003) RANDEXPR: a random symbolic expression generator. Technical report 4, Universitat Stuttgart

Lakkaraju H, Bach SH, Leskovec J (2016) Interpretable decision sets: a joint framework for description and prediction. In: KDD, ACM, pp 1675–1684

Laugel T, Renard X, Lesot M, Marsala C, Detyniecki M (2018) Defining locality for surrogates in post-hoc interpretablity. CoRR arXiv:1806.07498

Li X, Cao CC, Shi Y, Bai W, Gao H, Qiu L, Wang C, Gao Y, Zhang S, Xue X, Chen L (2022) A survey of data-driven and knowledge-aware explainable AI. IEEE Trans Knowl Data Eng 34(1):29–49

Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: 2008 eighth IEEE international conference on data mining, IEEE, pp 413–422

Lucic A, Oosterhuis H, Haned H, de Rijke M (2019) Actionable interpretability through optimizable counterfactual explanations for tree ensembles. CoRR arXiv:1911.12199

Lucic A, Haned H, de Rijke M (2020) Why does my model fail? Contrastive local explanations for retail forecasting. In: FAT*, ACM, pp 90–98

Lundberg SM, Lee S (2017) A unified approach to interpreting model predictions. In: NIPS, pp 4765–4774

Malgieri G, Comandé G (2017) Why a right to legibility of automated decision-making exists in the GDPR. Int Data Privacy Law 7(4):243–265

McCane B, Albert M (2008) Distance functions for categorical and mixed variables. Pattern Recognit Lett 29(7):986–993

Miller T (2019) Explanation in artificial intelligence: insights from the social sciences. Artif Intell 267:1–38

Ming Y, Qu H, Bertini E (2019) Rulematrix: visualizing and understanding classifiers with rules. IEEE Trans Vis Comput Graph 25(1):342–352

Minh D, Wang HX, Li YF, Nguyen TN (2022) Explainable artificial intelligence: a comprehensive review. Artif Intell. Review To appear

Molnar C (2019) Interpretable machine learning. Lulu Press, Morrisville

Moraffah R, Karami M, Guo R, Raglin A, Liu H (2020) Causal interpretability for machine learning: problems, methods and evaluation. SIGKDD Explor 22(1):18–33

Mothilal RK, Sharma A, Tan C (2020) Explaining machine learning classifiers through diverse counterfactual explanations. In: FAT*, ACM, pp 607–617

Murthy SK, Kasif S, Salzberg S (1994) A system for induction of oblique decision trees. J Artif Intell Res 2:1–32

Ntoutsi E et al (2020) Bias in data-driven artificial intelligence systems: an introductory survey. WIREs Data Min Knowl Discov 10(3):e1356

Olvera-López JA, Carrasco-Ochoa JA, Trinidad JFM, Kittler J (2010) A review of instance selection methods. Artif Intell Rev 34(2):133–143

Panigutti C, Guidotti R, Monreale A, Pedreschi D (2020) Explaining multi-label black-box classifiers for health applications. In: Precision health and medicine, studies in computational intelligence, vol 843. Springer, pp 97–110

Pasquale F (2015) The black box society: the secret algorithms that control money and information. Harvard University Press, Cambridge

Pedreschi D, Giannotti F, Guidotti R, Monreale A, Ruggieri S, Turini F (2019) Meaningful explanations of black box AI decision systems. In: AAAI, AAAI Press, pp 9780–9784

Plumb G, Molitor D, Talwalkar AS (2018) Model agnostic supervised local explanations. In: NeurIPS, pp 2520–2529

Ribeiro MT, Singh S, Guestrin C (2016) "Why should I trust you?": explaining the predictions of any classifier. In: KDD, ACM, pp 1135–1144

Ribeiro MT, Singh S, Guestrin C (2018) Anchors: high-precision model-agnostic explanations. In: AAAI, AAAI Press, pp 1527–1535

Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. NMI 1:206

Russell C (2019) Efficient search for diverse coherent explanations. In: FAT, ACM, pp 20–28

Sagi O, Rokach L (2018) Ensemble learning: a survey. WIREs Data Min Knowl Discov 8(4):e1249

Sagi O, Rokach L (2020) Explainable decision forest: transforming a decision forest into an interpretable tree. Inf Fusion 61:124–138

Sharma S, Henderson J, Ghosh J (2019) CERTIFAI: counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models. CoRR arXiv:1905.07857

Shih A, Choi A, Darwiche A (2018) A symbolic approach to explaining Bayesian network classifiers. In: IJCAI, ijcai.org, pp 5103–5111

Sokol K, Flach PA (2019) Desiderata for interpretability: explaining decision tree predictions with counterfactuals. In: AAAI, AAAI Press, pp 10035–10036

Strecht P, Mendes-Moreira J, Soares C (2014) Merging decision trees: a case study in predicting student performance. In: ADMA. Lecture notes in computer science, vol 8933. Springer, 535–548

Sundararajan M, Najmi A (2020) The many Shapley values for model explanation. In: ICML, PMLR, proceedings of machine learning research, vol 119, pp 9269–9278

Tan P, Steinbach MS, Kumar V (2005) Introduction to data mining. Addison-Wesley, Reading

Tsai C, Eberle W, Chu C (2013) Genetic algorithms in feature and instance selection. Knowl Based Syst 39:240–247

Venkatasubramanian S, Alfano M (2020) The philosophical basis of algorithmic recourse. In: FAT*, ACM, pp 284–293

Verma S, Dickerson JP, Hines K (2020) Counterfactual explanations for machine learning: a review. CoRR arXiv:2010.10596

Vidal T, Schiffer M (2020) Born-again tree ensembles. In: ICML, PMLR, proceedings of machine learning research, vol 119, pp 9743–9753

Virgolin M, Alderliesten T, Bosman PAN (2020) On explaining machine learning models by evolving crucial and compact features. Swarm Evol Comput 53:100640

Wachter S et al (2017) Counterfactual explanations without opening the black box. Harv JL Technol 31:841

Wu S, Olafsson S (2006) Optimal instance selection for improved decision tree induction. In: IIE, IISE, p 1

Yang H, Rudin C, Seltzer MI (2017) Scalable Bayesian rule lists. In: ICML, PMLR, proceedings of machine learning research, vol 70, pp 3921–3930

Zafar MR, Khan NM (2019) DLIME: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. CoRR arXiv:1906.10263

Zhang Y, Song K, Sun Y, Tan S, Udell M (2019) Why should you trust my explanation? Understanding uncertainty in LIME. arXiv:1904:12991

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.