A Generic Framework for Black-box Explanations

Clement Henin, Daniel Le Métayer

A Generic Framework for Black-box Explanations

Clément Henin
Univ Lyon, Inria, INSA Lyon, CITI
École des Ponts ParisTech
Champs-sur-Marne, France
clement.henin@inria.fr

Daniel Le Métayer
Univ Lyon, Inria, INSA Lyon, CITI
Villeurbanne, France
daniel.le-metayer@inria.fr

***Abstract*—Beyond their differences, most black-box explanation methods share a number of features and can be framed in a common structure. We identify these features and propose a generic and parameterized framework which makes it possible to combine them in different ways. This framework has been implemented in a proof of concept system called IBEX (for "Interactive Black-box EXplanation system"). IBEX makes it possible to address a variety of needs of different types of explainees (e.g. local or global explanations, detailed or simple explanations, explanations in the form of counterfactuals, rules, plots, etc.). We illustrate the benefit of the approach in terms of versatility through several case studies corresponding to different types of explainees.**

***Index Terms*—Algorithmic decision system, explainability, explanation, transparency, black-box model, machine-learning, artificial intelligence, AI**

I. Introduction

Algorithmic Decision Systems (hereafter "ADS") are increasingly used in many areas, sometimes with a major impact on the lives of the people affected by the decisions. Some of these systems make automatic decisions, for example to reduce or to increase the speed of an autonomous car, while others only make suggestions that a human user is free to follow or to dismiss. In some cases, the user is a professional, for example a medical practitioner or a judge, while in other cases he is an individual, for example an internet user or a consumer. Some ADS rely on traditional algorithms, while others are based on machine learning (hereafter "ML") and may involve complex models such as neural networks, support vector machines or random forests. Regardless of these considerations, when an ADS can have a significant impact on people, it must provide a minimum level of intelligibility. Indeed, understanding the result of an ADS is necessary both for the human agent taking the decision (or responsibility for the decision) and for people affected by the decision (for example, to accept it or to contest it). For this reason, explainability has generated increased interest during the last decade and many papers have been published on this topic, with sharp increase in recent years [1]. Most of these papers propose new methods or enhancements of existing methods to produce a specific type of explanation targeting a particular type of user (and sometimes a given class of ADS). In practice, different users (hereafter "explainees") may have different motivations and levels of expertise and therefore different needs in terms of explanations. A first option to address this variety of needs consists in implementing a toolkit including a set of methods that can be selected by the users. For example, AIX360 ("AI Explainability 360") [2] contains eight explainability algorithms and allows users to choose among them based on a taxonomy of pre-defined criteria. In the same spirit, the What-If tool includes visualization components and facilities to generate counterfactual examples and partial dependence plots. We believe that this approach is very promising but it is necessary to go further in at least two ways:

- The interaction facilities should allow explainees to express their needs in the most convenient way for them, depending on their level of expertise.
- Explanation methods should not just be put alongside each other in a toolbox. It should rather be possible to combine their features to generate the most suitable explanation for each explainee.

In this paper, we address the second issue and focus on black-box explanation methods[1]. In contrast with white-box explanation methods, which make use of the code of the ADS[2], black-box methods only assume that the ADS can be accessed through queries. Therefore, they can be used in a wider variety of situations. Moreover, they do not provide the same type of information as white-box methods. We discuss further the advantages and drawbacks of each approach in Section V. For the sake of conciseness, we use the expression "explanation method" for "black-box explanation method" in the sequel. Our objective is to go one step further in the integration of explanation methods through the identification of their fine-grain components and parameters. Indeed, our study shows that, beyond their differences, explanation methods share a number of features and can be framed in a common structure. We identify these features and use them to design a generic and parameterized framework which makes it possible to combine the components of different methods in different ways. This framework has been implemented in a proof of concept system called IBEX (for "Interactive Black-box EXplanation system"). The goal of IBEX is to address a variety of needs of different types of explainees (e.g. local or global explanations, detailed or simple explanations, explanations in the form of counterfactuals, rules, plots, etc.). We illustrate the benefit of

[1] Interested readers can find more details on interaction issues in a companion paper [3].

[2] Including all parameters and model coefficients, if need be.

the approach in terms of versatility through several case studies corresponding to different types of explainees.

We first present the general framework in Section II. The framework relies on two main components, namely *Sampling* and *Generation*, and their parameters. In Section III we illustrate the approach with the application of our proof-of-concept system IBEX to several case studies. Section IV discusses related work and Section V concludes with prospects for future work.

II. Generic framework

In the following, we call *explainer*, a system producing explanations. We first introduce our generic explainer architecture in Section II-A. Then we describe the two main elements of this architecture, the Sampling and the Generation components, with their parameters, in Section II-B and Section II-C respectively. From these elements, we derive the set of technical options available in the framework in Section II-D.

A. Generic explainer architecture

The first condition to be able to produce a range of explanations meeting the needs of different types of explainees is to be able to translate these needs in technical terms. To this aim, we introduce in this section a generic parameterized explainer architecture. The architecture is generic in the sense that many black-box explanation methods (including existing methods [4]) correspond to specific choices for its components and parameters. Furthermore, these components can be composed and parameterized in many other ways than in the implementations of existing "on the shelves" methods. The wealth of this combinatory is critical to match the variety of explainees' needs.

To introduce our explainer architecture, let us consider a simple spam classifier. This ADS takes as input the text of an email and outputs the probability that this email is a spam. Since we assume that the code of the ADS is not available, the method can only build emails, submit them to the ADS and analyze the results. For example, to assess the role of the signature part in the classification of a specific email, the explainer can create different versions with and without the signature part, or with different pieces of text in the signature part. The explainer has then to compute the answer based on the results of the ADS and to present it to the explainee.

This simple example highlights the two main components of an explainer architecture: (i) the selection of inputs to submit to the ADS to be explained, which is called the *Sampling* component; and (ii) the analysis of the links between the selected inputs and the corresponding outputs of the ADS to generate the content of the explanation, which is called the *Generation* component. If the input data are not meaningful for humans, as the pixels of an image for example, a preliminary component is required to extract an interpretable representation, as done in LIME [5]. Because the representation step is not essential to the description of the technical options, we focus now on the two other components. We propose formal characterizations of the sampling and generation components which are generic enough to encompass existing black-box explanation methods[3] and to serve as a basis for the production of explanations meeting different user needs, as shown in II-D. The main notations used in this section and the followings are sketched in Table I.

Name	Description	Example
F	Black-box model	The spam classifier
X	Input space of F	Set of all possible emails
Y	Output space of F	$[0,1]$
E	Scope of the explanation	Email x_e
S	Samples (product of the sampling step)	Emails with modified signature
Θ	Parameters of the sampling	Part of the email
D	Dataset describing the overall population	Training set of F

TABLE I: Main notations for the generic explainer

B. Sampling

The role of the *Sampling* component is to select appropriate inputs (or "samples") to answer a question about a model F. The choice of the samples may depend on a number of factors. The first aspect to take into consideration is whether the question concerns the whole model or specific inputs. We call E the scope of the explanation. If the question concerns a single input x_e, then $E = \{x_e\}$; if the question is about the whole model F, then $E = D$ with D a multiset[4] representation of the population (possible inputs to F) available to the explainer. In general, E and D could be any (multi)subset of possible input values. We call X the set of input values, which can be seen as the support set (or type set) of multiset D. In the spam filter example, X is the set of all possible emails (i.e. the set of all texts of a given format) and D represents the actual data set of emails available to the explainer, which can be used, for example, to estimate the distributions of the features. Typical examples of D would be the training or testing sets used during the learning process, or simply historical data accumulated during the use of the model. When the explainer does not have any information about this distribution, D is the empty set ($D = \emptyset$).

The result of the *Sampling* component is a set of samples $S = \{x_1, ..., x_n\} \in X^n$. For example, to address the first question about the impact of the signature on the classification of x_e, a possible option is to select a single sample obtained by removing the signature part of x_e. This strategy does not require any information about the actual distribution of the population and can therefore be applied even if $D = \emptyset$. However, the answer may not be realistic or precise enough. A more elaborate strategy would be to replace the original signature of x_e by real signatures obtained from many other

[3]Interested readers can find in [4] an analysis of existing black-box explanation methods and their expression in terms of the components and parameters of our generic architecture.

[4]D is a multiset because it can contain multiple occurrences of the same value to reflect the distribution of the values in the real population.

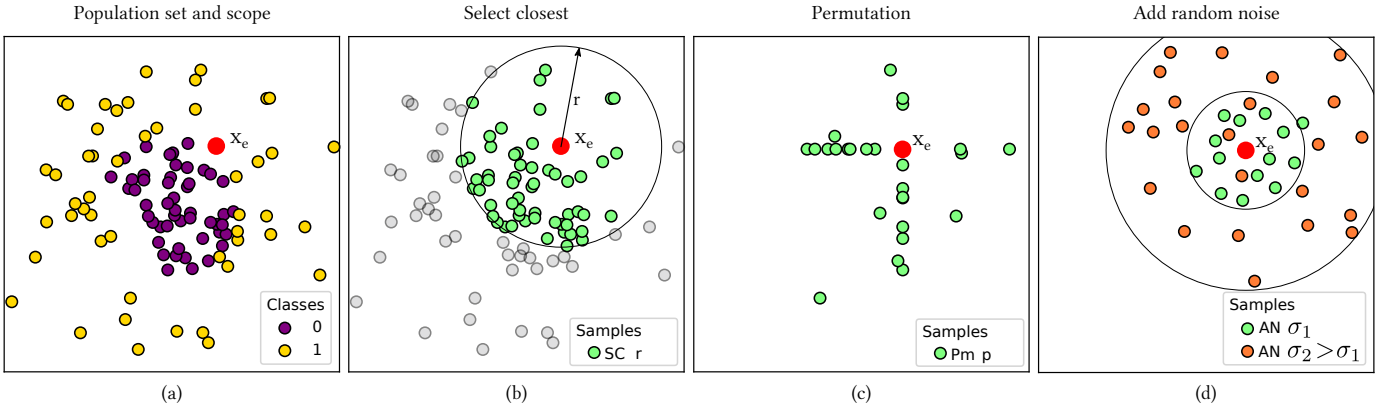


Fig. 1: Schematic view of local sampling processes with two dimensional continuous variables. The point of interest of the explanation (or scope) is the red circle x_e. (a) Population set and point of interest for a binary classification problem. Classes are depicted with different colors. (b) "Select closest" sampling with threshold r. Samples are inputs of the population set within a circle of size r centered at the point of interest. (c) "Permutation" sampling with probability p. Samples are altered versions of the point of interest with one or two features drawn from the empirical distribution. (d) "Add random Noise" sampling with σ_1 and $\sigma_2 > \sigma_1$. Samples are noisy versions of the point of interest.

emails. This strategy requires information about the actual distribution of the population ($D \neq \emptyset$) in order to ensure that the sample set reflects the reality. We can now define the sampling procedure as follows [5]:

$$S = \{h_\theta(x_e, x_p) \mid (\theta, x_e, x_p) \in \Theta \times E \times D, Z(\theta, x_e, x_p)\} \quad (1)$$

with

$$h_\theta : E \times D \to X \quad (2)$$

Θ is the set of parameters for the sampling, Z is a filter function and h_θ defines how samples are generated. In a nutshell, the θ parameter makes it possible to generate several samples for each pair (x_e, x_p) while Z restricts the generation of samples to a selection of pairs (x_e, x_p). In our spam filter example, E is limited to a single email to be explained ($E = \{x_e\}$). The email is represented by the content of its different parts (header, body, signature...) and $h_\theta(x_e, x_p)$ is a version of x_e that is obtained by replacing a part of x_e by the corresponding part of x_p. The part that is replaced is specified by θ. For instance, taking $\Theta = \{(SIG)\}$ and assuming that D contains 1000 emails, the sampling procedure generates 1000 perturbed versions of x_e with signatures (since $\theta = (SIG)$) extracted from the emails in D. The role of the θ parameter is therefore to customize the sampling function. For instance, if both the header and the signature of the email are taken into consideration, θ could specify which parts of the email are replaced (header, signature or both). With $\Theta = \{(HDR), (SIG), (HDR, SIG)\}$, the sampling procedure would generate 3000 versions of x_e with header, signature or both replaced by the corresponding parts of other emails in D. Another possibility provided by Definition (1) is to use a filter function Z, for example selecting only emails from the same sender as x_e, or relying on a notion of distance to select only emails close to x_e. To make the presentation more concrete, let us present three examples of sampling strategies which are instances of Definition (1). These strategies are available, among others, in the proof-of-concept system IBEX illustrated in Section III. We focus on local explanations here since global explanations rely on the whole population set. In the first example, called "Select Closest" (SC), Z is used to select from the population set D inputs that are close to x_e by comparing the distance $d(x_e, x_p)$ to a predefined threshold r. In this case, h_θ^{close} simply returns the unmodified input from the population set (cf Figure 1(b)).

$$\begin{aligned} S = \{&h_\theta^{close}(x_e, x_p) = x_p \mid \\ &(\theta, x_e, x_p) \in \{r\} \times E \times D,\ d(x_e, x_p) < \theta\} \end{aligned} \quad (3)$$

Since h^{close} returns samples from the population set D, it may be suitable to generate realistic explanations. The number of samples and their closeness to x_e can be tuned using r.

Another strategy, called "Permutation" (P) swaps features among samples to account for the underlying distribution of X. The following sampling function:

$$\begin{aligned} h_\theta^{perm}(x_e, x_p) = (&x_e[i] \text{ if } i \in \theta \text{ else } x_p[i]) \\ &\text{with } x_e \in E, x_p \in D \end{aligned} \quad (4)$$

combines the features of x_e with the features of x_p ($x[i]$ denotes the i^{th} feature of x) and the parameter θ defines the origin of a feature : the scope or an input of the population set (cf Figure 1(c)). θ is drawn randomly in such a way that each feature comes from x_p with probability p, which is a parameter of the sampling. The computation of Shapley values in [6] or the generation of local rule based models in Anchors [7] are based on similar sampling strategies. Each feature is independently drawn from the empirical distribution of X and only features included in the same θ are correlated. "Permutation" sampling is an intermediate level of realism.

[5] If Θ, D or E are empty, they are set to $\{0\}$ in (1), otherwise the product space would also be empty.

Finally, "Add random Noise" (AN), generates samples by adding a certain amount of noise to x_e. Samples are then noisy versions of x_e, with the noise drawn from a normal distribution with 0 mean (cf. Figure 1 (d)).

$$h_\theta^{noise}(x_e, x_p) = x_e + \theta, \text{ with } \theta \sim \mathcal{N}(0, \sigma^2) \tag{5}$$

The distribution of samples obtained with AN does not use the information of the population set ($D = \emptyset$), and features are independent. Variable σ represents the standard deviation of the added noise: small values of σ generate samples that are close to x_e while bigger values generate samples in a wider space (as depicted with the two circles in Figure 1 (d)). "Add random noise" provides non-realistic samples.

C. Generation

The set S of samples and the model F are the inputs of the explanation generation process. Even if explanations can take many different forms, the generation process can be broadly defined as the computation of a *proxy* of the model F followed by the construction of an explanation based on this proxy. In some cases, the proxy model is considered as the explanation itself, and the second phase is therefore just the identity function.

Coming back to the spam classifier example, an option for the *Generation* component is to train a rule-based model on the samples to predict the output of the classifier. An example of rule generated this way could be: "If the signature of the email is less than 60 characters long, then the classifier will consider that it is a spam; otherwise it will consider that it as an acceptable email". Because such rules are easily interpretable, they can be used directly as explanations. In other situations, either because the type of model used is too complex or the model is too big to be understandable (for example if it involves a large number of rules), simpler explanations have to be generated from the proxy model. This phase can return, for example, the most important feature(s) of the input. For the spam classifier, the generated explanation could then be: "The length of the signature part and the number of typos are the two most important features used by the ADS to decide if an email is a spam".

Technically speaking, the proxy model is denoted by f_w (the rule-based model in the example), which is a function of the same type as the model F, parameterized by w:

$$f_w : X \to Y \tag{6}$$

The core of the *Generation* component is to find the best proxy f_w to answer the question of the explainee, which amounts to find the optimal values of w. Optimality can be defined formally using constraints $o_i(w, S) \in \mathbb{B}$ and criteria $c_i(w, S) \in \mathbb{R}$ where $\mathbb{R}$ and $\mathbb{B}$ are the sets of real numbers and booleans respectively. The global objective takes the following form:

$$\begin{aligned} w^* = \underset{w}{\text{argmin}} \quad & \sum_i \lambda_i c_i(w, S) \\ \text{subject to} \quad & o_i(w, S) \end{aligned} \tag{7}$$

where $\lambda_i \in \mathbb{R}$ are used to weight the criteria.

In many methods, the objective is to find the parameters w such that the proxy f_w is as close as possible to F on the elements of S (samples). Indeed, finding a good explanation is often a matter of trade-off. A typical example is finding the right balance between precision and complexity – often used as a rough approximation of understandability. For example, a simple explanation of the spam classifier that would be accurate (i.e. predicting the actual result of the classifier) on only seventy percent of its inputs would not be acceptable; on the other hand, an accurate explanation that would take the form of several pages of rules would provide little insight to the user. Using both criteria and constraints offers flexibility. This distinction is already used in some existing methods. For instance Anchors [7], sets a *constraint* on the precision of the rule-based model and advocates that explanations should be highly precise, while BETA [8] sets the precision of the rule-based model as a *criterion* and advocates that explanations should first be interpretable.

To make the presentation more concrete, let us consider three examples of generation strategies, which are instances of Definition (7). These strategies are available, among others, in the proof-of-concept system IBEX presented in Section III. The first example is the "Rule-Based model" (RB) generation f_w with w the set of rules. A possible instance is to use as criterion the number of rules and as constraint the precision of the model, as done in [7], which can be expressed using the following minimization:

$$\begin{aligned} w^* = \underset{w}{\text{argmin}} \quad & ||w|| \\ \text{subject to} \quad & \#\{x \in S, f_w(x) = F(x)\}/\#S > a \end{aligned} \tag{8}$$

with $||w||$ the number of rules, $\#$ denoting the cardinality and a, the minimum accuracy.

Another example is to use a "Local linear Approximation" (LA) of the model, as done in [5]. In this case, f_w can be defined as $f_w(x) = \sum_i w_i x[i]$ with the following minimization:

$$w^* = \underset{w}{\text{argmin}} \quad \lambda||w|| + \sum_{x \in S} (f_w(x) - F(x))^2 \tag{9}$$

which amounts to a classical Lasso regression. The derived coefficients of the Lasso regression provide information about the local behaviour of the ADS. More precisely, by comparing their values, the explainee can estimate what would be the impact of the modification of a variable on the model output. In many cases, it approximates the importance of a feature for a specific output.

Finally, as proposed in [9], the generation step may be used to find a counterfactual example, which can be expressed as follows:

$$\begin{aligned} w^* = \underset{w \in \{x - x_e, x \in S\}}{\text{argmin}} \quad & ||w|| \\ \text{subject to} \quad & f_w(x_e) \neq F(x_e) \end{aligned} \tag{10}$$

with $f_w(x) = F(x + w)$ and $||w||$ denoting the distance between $x + w$ and x. A counterfactual example is the input closest to the point of interest for which the ADS returns an

Name	Component	Focus	Parameters	Short description
Add random Noise (AN)	Sampling	Local	σ	Adds Gaussian noise to the point of interest
Permutation (Pm)	Sampling	Local	p	Swaps of values between the scope and population inputs
Select Closest (SC)	Sampling	Local	r	Selects inputs from the population closest to the point of interest
Identity (Id)	Sampling	Global	$\emptyset$	Returns the population set
Replace with Constant (RC)	Sampling	Global	α	Replaces all values of one feature with constant α
Rule-Based model (RB)	Generation	Local	a	Accurate and simple RBM
Local linear Approximation (LA)	Generation	Local	λ	Lasso regression
CounterFactual (CF)	Generation	Local	$\emptyset$	Finds the closest sample to the point of interest leading to a different output
Decision tree (DT)	Generation	Global	a_{DT}	Decision tree (sampling: Id)
Pearson Correlation (PC)	Generation	Global	$\emptyset$	Global linear importance of features (sampling: Id)
Partial Dependence (PD)	Generation	Global	$n^{(i)}$	Computes average output of each features value (sampling: RC)

TABLE II: Technical options: components and their parameters. (i) Variable n denotes the number of bins used for continuous variables.

Component	Parameter	Short description
Add random noise	σ	Standard deviation of noise
Permutation	p	Probability to change feature value
Select closest	r	Distance to farthest sample
Rule-based model	a	Minimum accuracy
Local linear approximation	λ	Lasso penalization weight
Decision tree	a_{DT}	Minimum accuracy
Partial dependence plot	n	Number of bins

TABLE III: List of technical parameters

output different from the output returned for the point of interest. Our formulation of counterfactuals involves the differences between the point of interest and the counterfactual, named w, that should be as small as possible. Equation (10) involves the norm of w, which is the distance between the counterfactual and the point of interest, and a constraint on the output of the ADS for the counterfactual, which should differ from the output of the ADS for the point of interest.

D. Set of technical options

In the previous two sections, we have presented the two main components of the generic explainer architecture, the *Sampling* component and the *Generation* component. These components can be instantiated and parameterized in different ways. The instantiations and parameterization options together make up the technical options available to produce explanations. In this section, we review this set of technical options based on the notions introduced in Section II-B and Section II-C.

The instantiations of the *Sampling* and the *Generation* components currently available in the implementation of IBEX are presented with their parameters in Table II. Considering that, for local explanations, the two phases (*Sampling* and *Generation*) are independent, there are nine possible combinations of instantiations. For global explanations, three additional options are possible, making a total of twelve options for the instantiation of components. The second part of the technical options, the choice of the parameters, mostly depends on the instantiation of the component, as shown in Table II. Table III provides an overview of the parameters used by the sampling and generation components.

As an illustration of the possibilities of combinations of different instantiations for the sampling and generation components, let us consider the example of counterfactuals (CF). When a counterfactual example is obtained using a realistic sampling strategy, the final explanation looks like a real email, very similar to the point of interest (with a small number of words modified). In this case, a realistic counterfactual could an altered version of the point of interest with longer words in the signature such that its length exceeds sixty characters. This type of explanation is useful for a regular user of the system who wants to understand the ADS, for instance to assess its reliability. On the other hand, a counterfactual obtained from a non-realistic sampling does not necessarily look like a real email. For instance, one of the non-realistic sampling strategies that could be used would consist in a random addition of characters to the original email. The additional wording would

look like typos for the ADS. This type of counterfactual is more suited for technical experts trying to improve the model as such (for any input data, disregarding their actual "real life" distribution).

The choice of the parameters associated with each component further multiplies the number of technical options available. For example, the value of σ (Definition (5)) has an impact on the average distance between samples and the point of interest, which we call the range of the sampling. So explanations obtained with greater values of σ are likely to be more general than explanations with small values of σ (this impact of the choice of σ is confirmed experimentally on the case study presented in Section III-B). As another example, the value of parameter a (Definition 8) represents the minimum accuracy imposed during the search for the rule-based model. This parameter can be used to control the simplicity of the resulting explanation.

III. IBEX AT WORK: APPLICATION TO CASE STUDIES

In this section, we illustrate our approach with the application of our proof-of-concept system IBEX to several case studies. The code of the system is publicly available[6]. We focus on the combination of components and parameters to produce different types of explanations here. The expression of the objectives of the users (technical experts, domain experts, auditors and lay users) and the translation of these objectives into specific combinations of components and parameters are presented in a companion paper [3].

In order to illustrate the benefits of the approach in terms of versatility and the diversity of explanations generated by the system, we present three case studies showing respectively:

1) The impact of the type of sampling on local explanations (Section III-A).
2) The impact of the sampling range on the generality of the explanations (Section III-B).
3) The impact of the generation method on explanations (Section III-C).

A. Impact of the type of sampling on local explanations

This case study involves the adult census data set[7]. This data set, which has been extracted from the 1994 US census, contains personal information about American citizens such as their age, education level or marital status. The goal of the ADS is to predict, based on these features, if the individual earns more or less than $50,000\$$ per year. The system is based on a 2-layer neural network model.

For this case study, we show how different sampling strategies may result in different explanations, we provide an interpretation for these differences and show how it can be used to foster interactive capabilities [3]. Both explanations shown in Figure 2 were generated by IBEX using local linear approximation. The leftmost explanation (Figure 2(a)) was generated using the *Add random noise* sampling, which does not take into consideration the distribution of input data, while the rightmost explanation (Figure 2(b)) was generated using the *Select closest* sampling, which takes into consideration the distribution of input data. Conceptually, the main difference between the two explanations is that the latter takes into account the probability of observing a change in the input scope together with the effect of this change on the ADS output. In contrast, the leftmost explanation describes the effect of a change on the output disregarding the actual distribution of the input data.

As we can see, the resulting explanations are very different. Explanation (a) shows that *capital_gain* and *capital_loss* have the highest impact on the decisions (respectively positive and negative), while Explanation (b) emphasizes *education_num* and *hours_per_week*. None of these explanations is better than the other, they just explain the behaviour of the ADS from different perspectives. Our analysis of the population of the data set shows that less than 9% of all inputs have capital gains and they were all classified as '$> 50k$'. This corroborates the fact that, even if *capital_gain* is a very good indicator of the class '$> 50k$', it does not appear in the explanation on the right because it is a rare event. Not taking into account the distribution of input data allows IBEX to identify aspects of a model that could be used to achieve specific goals of the user. For instance, if it is not considered acceptable that a feature like *capital_gain* plays a role in a decision, then this explanation could be used to challenge it. This type of explanation can also be useful to technical experts who want to improve the ADS.

B. Impact of the sampling range on generality

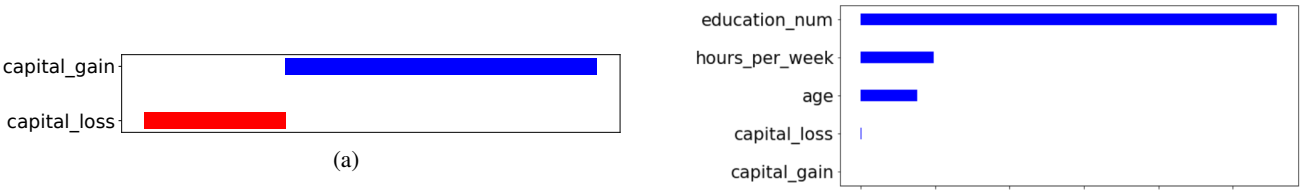
In this section, we show the results of an experiment conducted to assess the impact of the parameter σ (AN sampling) on the generality of the final explanation. Higher values of σ can be used to sample in a wider space around the point of interest x_e as shown in Figure 1(d). The intuition, that we would like to test here, is that a wider sample space results in explanations that are more general, that is, applying to more data points.

We conducted simulations on the Iris flower dataset[8] using local linear approximation for the generation task with $\lambda = 8.10^{-3}$. We compared the explanation for x_e with explanations for n closest neighbors of x_e in the dataset (n taking values from 2 to 149) and measured the proportion of neighbors for which the explanation is valid. The explanation generated for x_e is considered valid on an arbitrary point x if all coefficients of the regression are equal up to an error margin set to the empirical standard deviation of the coefficients. Using the standard deviation as error margin allows to account for the shrinking of coefficients due to the addition of noise (see 3.5 p. 174 [10]). The experiment has been conducted 150 times and the averaged results shown in Figure 3 confirm our intuition: increasing σ tends to flatten the curve. As an illustration, we can compare the situation for extreme values of σ: when

[6]https://gitlab.inria.fr/chenin/ibex

[7]https://archive.ics.uci.edu/ml/datasets/Adult

[8]https://archive.ics.uci.edu/ml/datasets/Iris



(a) (b)

Fig. 2: Two *LA* explanations generated by IBEX for the same adult input predicted as '$< 50k$'. The positive (resp. negative) values shown by blue bars (resp. red bars) indicate a positive (resp. negative) impact of the feature on the output '$> 50k$'. (a) LA explanation reflecting the logic of the ADS, disregarding the actual distribution of the input data. (b) LA explanation, based on samples accounting for the distribution of the input data set, reflecting the behaviour of the ADS on real data.

$\sigma = 2$, the explanation for x_e is valid for 74 % of the 10 closest neighbors and for 24 % of the 149 closest neighbors; when $\sigma = 10$ (corresponding to a wider sampling), the explanation for x_e is valid for 58 % of the 10 closest neighbors and for 35 % of the 149 closest neighbors. This experiment shows one example among others of how the sampling parameters can be finely tuned to obtain desired explanation properties.

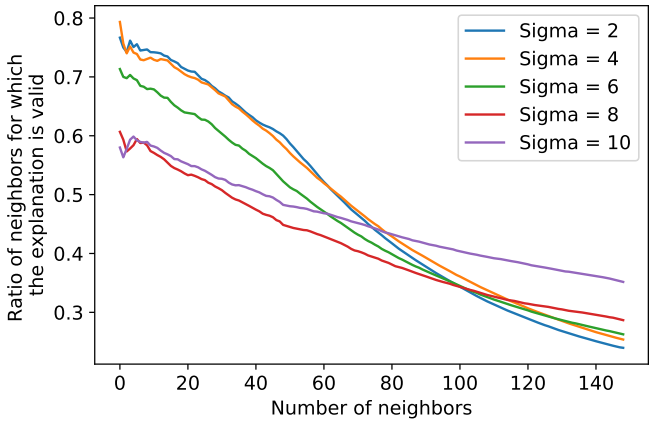


Fig. 3: Impact of σ on the generality of local linear approximations. The x-axis shows the number of neighbors on which the explanation is tested and the y-axis shows the proportion of these neighbors for which the explanation is valid. Flat curves correspond to general explanations (explanations valid for a large number of neighbors) and sharply decreasing curves are not general explanations (explanations valid for the closest neighbors only).

C. Impact of the generation method on explanations

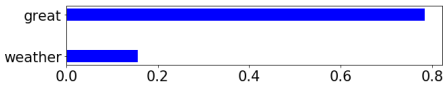
The last case study involves the airline sentiment analysis database[9]. This data set contains tweets about airline companies and the objective of the ADS is to classify them into three categories: negative, neutral or positive. Negative tweets are supposed to express negative emotions (anger, irritation, etc.), positive tweets are supposed to express positive emotions (happiness, gratitude, etc.) and neutral tweets show no or little emotion. The model takes as input the text of tweet (sequence of characters/words) and outputs the predicted sentiment class. For this experiment, a long-short term memory neural network was used to classify the tweets.

(a)

Rule-based explanation:
If the words **"great"** and **"weather"** appear then the tweet is positive

(b)

Local linear approximation explanation:



(c)

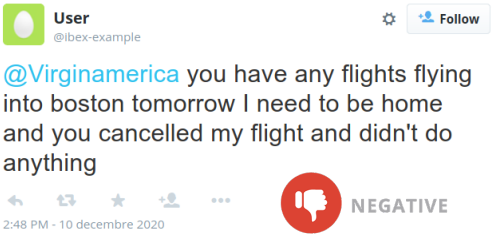
Counterfactual explanation:
Adding the word **cancelled** changes the output from positive to negative

(d)

Fig. 4: Tweet sentiment classification and its explanations generated with IBEX. (a) Tweet classified as positive (scope of the explanation). (b) Rule describing the local behavior of the model. (c) Local linear approximation of the model. Blue bars are for positive coefficients towards the "positive" class. (d) Counterfactual explanation.

Figures 4(a) and 5(a) show two examples of tweets classified respectively as positive and negative by the ADS. For each of them, we use IBEX to provide explanations. In both cases, three explanations are presented (denoted (b), (c) and (d)).

[9]https://www.kaggle.com/crowdflower/twitter-airline-sentiment

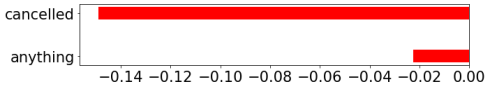


(a)

Rule-based explanation:
If the word **"cancelled"**
appears then the tweet is negative

(b)

Local linear approximation explanation:



(c)

Counterfactual explanation:
Removing the word **cancelled** changes the output from negative to positive

(d)

Fig. 5: Tweet sentiment classification and its explanations generated with IBEX. (a) Tweet classified as negative (scope of the explanation). (b) Rule describing the local behavior of the model. (c) Local linear approximation of the model. Red bars are for negative coefficients towards the "positive" class. (d) Counterfactual explanation.

They were generated with the same sampling[10] method but with three different generation methods: respectively local linear approximation, rule-based model and counterfactuals. This case study is useful to understand the impact of the generation method on the form of the explanation.

First, we see that all explanations are reasonable and easy to understand. They provide some information about the behavior of the model. For instance, they show that the model relies on specific words to classify tweets and the explanations help to identify them. Also, it is interesting to see that explanation contents vary only slightly with respect to the generation method. Except for the counterfactual (Figure 4(d)), the same words are used: "great" and "weather" for the tweet of Figure 4 and the word "cancelled" for the tweet of Figure 5. This observation is not surprising as information from the black-box model comes solely from the samples. However, each generation method provides explanations of a different form, which can be useful to best match the preferences of the user.

[10]Permutation sampling with $p = 0.05$ was used. In the representation of tweets, it amounts to randomly replacing with probability p each word of the tweet with a word from a randomly selected tweet.

IV. Related works

To the best of our knowledge, no existing explanation system provides the diversity of explanations and level of genericity offered by IBEX. As far as the generation of explanations is concerned, a plethora of methods already exist. In most cases, explanations are seen as static objects that are provided without taking into consideration the explainees' profiles or specific needs [5], [7], [8], [11].

Among the systems that provide a certain level of genericity, SHAP [12] proposes a unified approach to describe four explanation methods. It is related to our generic explainer, in the sense that it uses a unique theoretical description to describe several explanation methods. However, SHAP is restricted to explanations under the form of feature importance and it is not interactive. Some systems, such as LIME [5], make it possible to choose between different forms of explanations but all other options are fixed. Another effort in the same direction is described in [13], which allows the user to fix the values of certain features to restrict the choice of counterfactuals. Another example is the bLIMEy system (for "build LIME yourself"), which is a generic explainer relying on the framework proposed in [4]. As in IBEX, several sampling strategies are available to produce a variety of explanations. However, bLIMEy does not provide the wide range of choices available in IBEX.

Many projects have also been initiated recently to provide implementations of existing methods [14]–[16]. The goal of these projects is to integrate a variety of existing methods, but they do not include a comprehensive interaction module and a fine-grain decomposition of components as done in IBEX.

Several surveys and taxonomies of explanation methods have been published in recent years [17]–[19] . Some of them focus on the theoretical underpinnings of explanations while others are general overviews of existing methods. On the theoretical side, [12] introduces a formal framework to unify four black-box explanation methods. The framework is restricted to methods that compute the contribution of each feature to a given prediction. Moreover, it does not attempt to identify the common components shared by different methods. The scope of explanation methods considered in [17] is particularly wide, including black-box, white-box and constructive methods. This very comprehensive survey introduces a glossary and a taxonomy for interpretable and explainable AI. It then identifies a wide range of publications in this field and classifies them according to the taxonomy. Our approach differs from [17] in several ways. First, we start from a mathematical definition of the explanation tasks[11] and we derive our classification from the parameters of the formal framework. In addition, our framework makes it possible to compare existing methods in a very precise way. The range of methods considered in [17] is broader however, as it goes beyond black-box methods. In the same vein, [18]

[11]Rather than the explanation *problem* as in [17], which amounts to characterize explanation tasks by their types rather than their functional definitions as done in this paper.

is a high-level survey of explainable AI along four axes: explainability strategies, evaluation of explanations, interaction with humans and more general considerations about on the role of explanations. Finally, [1] provides an exhaustive meta-review of the explainable AI literature. Its scope is very wide as it includes references to explanation methods and relevant work from social sciences. It is a very comprehensive source of references on the topic. However, unlike our paper, it does not attempt to provide a common view on explanation techniques.

Another approach is followed in [20] which proposes a high-level taxonomy of interpretable and interactive machine learning composed of six elements[12] that are characterized in a very abstract way. Some papers focus on explanations for specific types of machine learning techniques. We do not discuss them in detail in this paper since our focus is black-box methods but still mention [21] which considers three types of explanation methods for deep learning[13] and discusses in a general way desirable properties of explainers and technical challenges.

A different perspective is provided in [22] which provides a taxonomy of interpretability in Human-Agent Systems. The interest of the authors is more general as it also includes the motivations of the explainee and the expected form of interaction with the explainer. However, [22] does not compare or analyze explanation methods as it refers to a single method. [23], [24] and [25] analyze more generally the needs for explainability and transparency considering social and technical aspects.

Finally, [26] provides a formal definition of explanations with a focus on the criteria to evaluate them. The evaluation of explanations, which is a critical issue, is not covered by this paper. We come back to this issue in the conclusion.

The above papers provide very useful overviews of the field but, to our best knowledge, none of them aims to define precise technical criteria to compare on a rigorous basis existing black-box explanation methods and to combine fine-grained components, as presented in this paper.

V. Conclusion

The main goal of the work described in this paper is to address the variety of needs in terms of explanations of ADS and to design an explanation system that can be used by a wide range of explainees, including lay users. To this aim, we have proposed a framework relying on a fine-grain decomposition of explanation tasks that can be combined in different ways. As a byproduct of this work, it is possible to use this decomposition into atomic components to compare and classify existing black-box explanation methods more precisely than presented in the various surveys published on this topic. Interested readers can find in [4] a table and discussion showing that existing black-box methods can be seen as particular instantiations of this framework, i.e. particular choices of the technical options presented in this paper. This analysis shows the generality of the framework and the benefits of the fine-grain approach to devise new combinations of options.

In this paper, we focused on black-box explanation methods. As mentioned in the introduction, in contrast with white-box methods, they can be used in the many instances where the code of the ADS is not available. Furthermore, the two types of explanations address different needs: white-box explanations tend to focus on operational aspects (how the ADS produces a given result) whereas black-box explanations provide information about the relationships between inputs and outputs (independently of the means used by the ADS to produce these outputs). Therefore, they are more logical than operational in nature. Operational explanations are usually more useful for developers (e.g. to improve the system) than for laymen (who may find them difficult to understand).

In this paper, we have shown, through the IBEX prototype, the feasibility of a versatile explanation system based on our approach. However, as stated above, IBEX is a proof of concept implementation and it can be improved and extended in several directions. It should be noted that the architecture of IBEX is highly modular. For example, new sampling strategies or data representations can easily be added without major modification of the system. In order to prove its usability as an explanation system in real life, it should be tested through a randomized user study involving different types of explainees, which we plan to do in the near future with applications in the health care and the judicial sectors.

In the current version of IBEX, the interactions with the system rely on a multi-layered interface allowing users to express their requests for explanations at different levels of abstraction [3]. It would be interesting to provide a richer and higher-level language to interact with explainees, for instance a restricted version of natural language that could be used by explainees to express questions and by the system to provide explanations. Dialogue specifications could rely on models such as [27].

A second improvement of IBEX could be the use of more elaborate sampling strategies that would provide further advantages in term of flexibility and efficiency of the computation, especially for high-dimensional data. The use of genetic algorithms, as presented in [28], is a promising approach to achieve this goal.

Another important area for further research is the design of new types of explanations and interactions to make it easier for the users of an ADS (or people affected by its decisions) to challenge its decisions [29]. Indeed, in order to support decision challenging, it is necessary to provide interactions about justifications (why a given decision is the good one), and not only about explanations (why the ADS made or suggested this decision). This requires a form of argumentation currently beyond the scope of IBEX.

[12] Dataset, Optimizer, Model, Predictions, Evaluator and Goodness.

[13] They are called respectively "rule-extraction methods", "attribution methods" and "intrinsic methods".

References

[1] S. T. Mueller, R. R. Hoffman, W. Clancey, A. Emrey, Explanation in human-ai systems: A literature meta-review synopsis of key ideas and publications and bibliography for explainable ai 204.

[2] V. Arya, R. K. E. Bellamy, P.-Y. Chen, A. Dhurandhar, M. Hind, S. C. Hoffman, S. Houde, Q. V. Liao, R. Luss, A. Mojsilović, et al., One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques, arXiv:1909.03012 [cs, stat]ArXiv: 1909.03012 (Sep 2019).
URL http://arxiv.org/abs/1909.03012

[3] C. Henin, D. Le Métayer, A multi-layered approach for tailored black-box explanations, in: Proceedings of the ICPR'2020 Workshop Explainable Deep Learning- AI, Springer, 2021.

[4] C. Henin, D. Le Métayer, Towards a generic framework for black-box explanations of algorithmic decision systems (Extended Version), Inria Research Report 9276, https://hal.inria.fr/hal-02131174.

[5] M. T. Ribeiro, S. Singh, C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, San Francisco, California, USA, 2016, pp. 1135–1144. doi:10.1145/2939672.2939778.

[6] E. Štrumbelj, I. Kononenko, Explaining prediction models and individual predictions with feature contributions, Knowledge and Information Systems 41 (3) (2014) 647–665. doi:10.1007/s10115-013-0679-x.

[7] M. T. Ribeiro, S. Singh, C. Guestrin, Anchors: High-precision model-agnostic explanations, in: AAAI Conference on Artificial Intelligence, 2018.

[8] H. Lakkaraju, E. Kamar, R. Caruana, J. Leskovec, Interpretable & Explorable Approximations of Black Box Models, arXiv preprint arXiv:1707.01154 (2017).

[9] S. Wachter, B. Mittelstadt, C. Russell, Counterfactual explanations without opening the black box: Automated decisions and the GDPR, Harvard journal of law & technology 31 (2018) 841–887.

[10] C. M. Bishop, Pattern recognition and machine learning, corrected at 8th printing 2009 Edition, Information science and statistics, Springer, 2009.

[11] B. Kim, R. Khanna, O. O. Koyejo, Examples are not enough, learn to criticize! criticism for interpretability, in: Advances in Neural Information Processing Systems, 2016, pp. 2280–2288.

[12] S. M. Lundberg, S.-I. Lee, A Unified Approach to Interpreting Model Predictions, Curran Associates, Inc., 2017, p. 4765–4774.

[13] K. Sokol, P. Flach, Glass-box: Explaining ai decisions with counterfactual statements through conversation with a voice-enabled virtual assistant, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, 2018, pp. 5868–5870. doi:10.24963/ijcai.2018/865.
URL https://www.ijcai.org/proceedings/2018/865

[14] H. Nori, S. Jenkins, P. Koch, R. Caruana, Interpretml: A unified framework for machine learning interpretability, arXiv preprint arXiv:1909.09223 (2019).

[15] J. Klaise, A. Van Looveren, G. Vacanti, A. Coca, Alibi: Algorithms for monitoring and explaining machine learning models.
URL https://github.com/SeldonIO/alibi

[16] P. Biecek, Dalex: Explainers for complex predictive models in r, Journal of Machine Learning Research 19 (84) (2018) 1–5.
URL http://jmlr.org/papers/v19/18-416.html

[17] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Computing Surveys (CSUR) 51 (5) (2018) 93.

[18] A. Adadi, M. Berrada, Peeking inside the black-box: A survey on explainable artificial intelligence (XAI), IEEE Access 6 (2018) 52138–52160. doi:10.1109/ACCESS.2018.2870052.

[19] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai, arXiv:1910.10045 [cs]ArXiv: 1910.10045 (Dec 2019).
URL http://arxiv.org/abs/1910.10045

[20] E. Ventocilla, T. Helldin, M. Riveiro, J. Bae, N. Lavesson, Towards a taxonomy for interpretable and interactive machine learning, 2018. doi:10.13140/RG.2.2.14534.98886.

[21] G. Ras, M. van Gerven, P. Haselager, Explanation methods in deep learning: Users, values, concerns and challenges, CoRR abs/1803.07517 (2018). arXiv:1803.07517.
URL http://arxiv.org/abs/1803.07517

[22] A. Richardson, A. Rosenfeld, A survey of interpretability and explainability in human-agent systems 7.

[23] R. Tomsett, D. Braines, D. Harborne, A. D. Preece, S. Chakraborty, Interpretable to whom? a role-based model for analyzing interpretable machine learning systems, CoRR abs/1806.07552 (2018).

[24] Z. C. Lipton, The mythos of model interpretability, arXiv:1606.03490 [cs, stat]ArXiv: 1606.03490 (Jun 2016).
URL http://arxiv.org/abs/1606.03490

[25] A. Weller, Challenges for transparency, arXiv:1708.01870 [cs]ArXiv: 1708.01870 (Jul 2017).
URL http://arxiv.org/abs/1708.01870

[26] A. Dhurandhar, V. Iyengar, R. Luss, K. Shanmugam, A formal framework to characterize interpretability of procedures, arXiv:1707.03886 [cs]ArXiv: 1707.03886 (Jul 2017).
URL http://arxiv.org/abs/1707.03886

[27] P. Madumal, T. Miller, L. Sonenberg, F. Vetere, A grounded interaction protocol for explainable artificial intelligence, arXiv:1903.02409 [cs]ArXiv: 1903.02409 (Mar 2019).
URL http://arxiv.org/abs/1903.02409

[28] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, F. Giannotti, Local rule-based explanations of black box decision systems, arXiv:1805.10820 [cs]ArXiv: 1805.10820 (May 2018).
URL http://arxiv.org/abs/1805.10820

[29] C. Henin, D. Le Métayer, A framework to challenge and justify decisions based on machine learning algorithms, in: Proceedings of the 1st International Workshop on Deceptive AI (DeceptECAI), Springer, 2020.