

A General Framework for Personalising Post Hoc Explanations through User Knowledge Integration

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Marcin Detyniecki

▶ **To cite this version:**

Adulam Jeyasothy, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Marcin Detyniecki. A General Framework for Personalising Post Hoc Explanations through User Knowledge Integration. International Journal of Approximate Reasoning, 2023, 160, pp.108944. ⟨10.1016/j.ijar.2023.108944⟩. ⟨hal-04337026⟩

HAL Id: hal-04337026

https://hal.science/hal-04337026v1

Submitted on 12 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

A General Framework for Personalising Post Hoc Explanations through User Knowledge Integration

Adulam Jeyasothy[a], Thibault Laugel[b], Marie-Jeanne Lesot[a], Christophe Marsala[a], Marcin Detyniecki[a,b,c]

[a]*Sorbonne Université, CNRS, LIP6, Paris, F-75005, France*
[b]*AXA, Paris, France*
[c]*Polish Academy of Science, IBS PAN, Warsaw, Poland*

Abstract

The field of XAI aims at providing explanations about the behavior of AI methods to a user. In particular, local post-hoc interpretability approaches aim at generating explanations for a particular prediction of a trained machine learning model. It is generally recognized that such explanations should be adapted to each user: integrating user knowledge and taking into account the user specificity allows to provide personalized explanations and to improve the explanation understandability. Yet these elements appear to be rarely taken into account, and only in specific configurations. In this paper, we propose a general framework to allow this integration of user knowledge in post-hoc interpretability methods, relying on the addition of a compatibility term in the cost function. We instantiate the proposed formalization in two scenarios, varying in the explanation form they propose, in the case where the available user knowledge provides information about the data features. As a result, two new explainability methods are proposed, respectively named Knowledge Integration in Counterfactual Explanation (KICE) and Knowledge Integration in Surrogate Model (KISM). These methods are experimentally studied on several benchmark data sets to characterize the explanations they generate as compared to reference methods.

Keywords: eXplainable Artificial Intelligence, XAI, user knowledge, compatibility, counterfactual explanation, surrogate model explanations, local feature importance

1. Introduction

The democratization of Machine Learning (ML) applications has led to a raise in the awareness of the potential issues these systems may have. The eXplainable Artificial Intelligence (XAI) domain [1, 2, 3] focuses on answering questions a user may ask when faced with the result of such a ML model that constitutes a black box. Such questions can be expressed as "Why do I get this prediction?" or "What do I need to do to get the prediction I want?". Among the large variety of approaches proposed in this domain, post-hoc methods focus on explaining the predictions obtained from a given pretrained classifier. Different formats of explanations can be proposed, such as feature importance (e.g LIME [4] and SHAP [5]) or counterfactual examples [6] (e.g. Growing Spheres [7] and FACE [8]). Explanation methods also differ depending on the hypotheses they make regarding the information considered available: model agnostic and data agnostic approaches consider that no knowledge is available, neither on the ML model (e.g. on the form of the decision boundary), nor on data (e.g. on their distribution). This absence of knowledge has been shown to lead to numerous issues, such as their risk of generating meaningless explanations [9] or their lack of connection to the real-world actions required to change features [10]. As a result, they may then not be always understood by the user [11]. To tackle these issues, as detailed in Section 2, some methods propose to enrich the considered input and integrate the user in the loop by taking into account user knowledge [12, 13, 14], so the explanation is more understandable.

Focusing on the same issue, this paper takes over the general formalization we proposed in previous work [15]. It relies on the definition of a generic cost function that integrates the aforementioned user knowledge in the search for an explanation, independently of the explanation form and the knowledge representation. It consists in adding, to the traditional explanation-generation-optimization objective, a complementary *compatibility* term, taking as parameter the considered user knowledge. This general framework can be used to augment traditional post-hoc explanation methods of different types, e.g. counterfactual and surrogate model-based approaches, and make them compatible, or complementary depending on the considered context and objective, with knowledge of various forms.

The paper provides a discussion about the need for such a framework, as well as two instantiations of the framework, for different types of explanations, in a model and data agnostic paradigm. First, we focus on the case

of counterfactual explanations that favor actions along the features that are known by the user. Then, we consider the case of surrogate models to reach the same objective. To solve these two instantiations, two new explainability approaches are proposed, respectively named Knowledge Integration in Counterfactual Explanation (KICE) and Knowledge Integration in Surrogate Model (KISM), and their relevance is studied empirically. Finally, the paper proposes an exploratory discussion for the case of user knowledge expressed in a different, enriched, form: beyond the set of features the user indicates he/she understands, it considers user knowledge represented in the form of a rule-based system. It discusses the interest and challenges of using it to augment counterfactual explanations.

The paper is structured as follows: Section 2 details the background of XAI and existing dedicated approaches that take into account specific user knowledge forms for generating specific explanation forms. Section 3 describes the general framework we propose for integrating user knowledge into interpretability methods. Its instantiations to the case of feature based knowledge, for explanations in the form of counterfactual examples and surrogate models, are respectively presented in sections 4 and 5. Section 6 describes the experimental study conducted on several benchmarks to characterize the explanations generated by these approaches, as compared to reference methods. Perspectives for the case where the user knowledge is expressed as a rule-based system are discussed in Section 7. Finally Section 8 concludes the paper.

2. Background

This section provides background information to which the paper refers: it first describes some post-hoc explanation methods; then, it discusses different possible forms the user knowledge can take, before presenting some existing approaches that integrate knowledge in explanation generation.

2.1. Post-hoc Explanations

There exist numerous post-hoc explanation methods, i.e. methods that take as input a trained classifier considered as a black box and that can only request it to label new data points (see e.g. [16] for a survey). Many of them rely on generating explanations as a solution to an optimization problem, for a well-defined cost function. Among those, we are particularly interested in counterfactual and surrogate approaches.

The first ones [17, 18, 19] aim at answering the question: "What do I have to change to get the prediction I want?", generating so-called contrastive explanations that oppose the predicted output to the desired one [20]. They provide local explanations, i.e. explanations that depend on the instance the user asks a question about. The explanation is defined as the data point that is both predicted to belong to the desired class and as close as possible to the instance of interest: the difference between this generated data point and the reference instance indicates the required changes and as such constitutes the explanation. From the first propositions [6, 21], numerous variants have been proposed to take into account additional requirements, such as reducing the number of features to modify [7], improving the feasibility of the changes [8] or their realism [22] to name a few (see e.g. [17, 18, 19] for recent surveys). Most approaches consist in optimizing, under constraint, a cost function that expresses the desired characteristics of the counterfactual example. The basic form of this cost function is defined as the distance between the considered instance and the explanation, e.g. the Euclidean distance, possibly combined with the l_0 norm [7, 21] under the constraint its prediction equals the desired class.

On the other hand, surrogate-based explanations rather answer the question: "Why do I get this prediction?": they aim at capturing locally the behavior of the considered machine learning model, representing it by a local approximation performed by a simple interpretable model, such as decision trees, linear regressions or classification rules [4, 23, 24], that mimics its behavior. One popular surrogate model approach is LIME [4] that relies on a local linear approximation of the studied classifier: it optimizes the coefficients of a linear regression model and the number of non-zero coefficients to maximize the fidelity to the black box classifier and outputs the features with the highest absolute values of the coefficients. LIME bridges the gap to local feature importance explanation approaches that aim at extracting the locally most influential features.

2.2. Knowledge Representation

As discussed in the introduction, post-hoc explanations, as well as any explanation building method, can be enriched by the integration of available user knowledge, so as to improve their quality and intelligibility, in a human-in-the-loop paradigm. A preliminary question, before its integration in the explanation methods, concerns the formalism used to express this user knowledge. Many have been considered, with numerous variants in each case,

one can cite among others: class prototypes [25] (representative instances of classes that satisfy the characteristics necessary to belong to a class), distribution of data [8] (real data set in the studied context) or feature-based knowledge. In the latter case, detailed below as the case on which this paper focuses, the user provides personal information according to which all features are not equivalent: some of them may be more important, or related to others, which should be exploited to formulate the explanations.

A first type of such feature-based user knowledge provides information on the individual features: the user can for instance indicate the so-called actionable features [14], i.e. the features that can be modified, as opposed to the ones that must remain unchanged. This information may be in particular crucial in the case of explanation expressed as counterfactual examples, as a way to express preferences regarding the features that need to be changed to be predicted in a different class. For instance in a credit application scenario, a user may indicate that the budget is not actionable, whereas the duration of the credit reimbursement is. At a more refined level, knowledge may also indicate that some modifications can only apply in a given direction: common knowledge for instance indicates that age cannot be decreased [21]. The user can also give the importance of each feature. The knowledge is then a feature importance vector. This type of knowledge gives an idea of the impact of the features on the prediction according to the user. This information can be used in the case of feature importance explanations [26].

A second type of feature knowledge provides information on their interactions. It may be expressed through their covariation [12, 27] (e.g. "An increase in education level leads to an increase in age") or, more generally and more formally, through a causal graph that expresses functional dependencies between the feature values [12, 13].

It can be underlined that knowledge can have different sources and it can be useful to distinguish between the notions of prior, expert or user knowledge: all three can be represented in the same formalism, but may have different impacts on the explanation. Prior knowledge can be considered as common knowledge, normally shared between various users and that needs to be taken into account generally. On the other hand, expert and user knowledge are more specific and are the ones that should lead to personalized explanations. The difference between these two types is that all users are not experts and that their own knowledge may be sometimes unreliable or imposing strong constraints that may endanger the existence of explanations satisfying them all. For instance a user may have too few actionable features

for counterfactual examples to exist. This case of user knowledge is however the one this paper focuses on.

2.3. Integration of Knowledge

Beside the question of the form the available knowledge can take, the question of its exploitation and integration in the explanation must be considered. Several approaches have been proposed, to the best of our knowledge they are dedicated to specific combinations of the knowledge type and the explanation form they consider.

In the case where knowledge indicates actions which are possible or not and explanations take the form of counterfactual examples, Ustun et al. [14] propose to generate actionable explanations. The set of actions is defined as a set of features where an action is feasible, i.e when values can be modified. This allows to avoid proposing inapplicable explanation such as "In order to get the credit, you need to decrease your age". To the classic choice of a distance-based cost function (see Section 3.2.1), Ustun et al. add a density-based criterion which is the Cumulative Distribution Function. One of the limits of this approach is that it can only applied in the case where the classifier or regressor to explain is a linear models. The KICE method we propose in this paper (see Section 4) relies on the same form of explanation and user knowledge as Ustun et al. but it considers a model agnostic context.

For image data, Zhao et al. [26] consider the integration of feature weight knowledge in feature importance explanations: in this case the considered knowledge and explanation types are identical, the final explanation is a compromise between the two feature score vectors.

Considering the case of counterfactual explanations, Mahajan et al. [12] focus on the case where knowledge is expressed as causal interactions between features. They propose to integrate the latter in the definition of a distance measure that quantifies the extent to which a candidate explanation satisfies the causal relationships. Unlike Ustun et al. [14], they do not exclude candidates that do not satisfy these constraints, but penalize them, in a more flexible approach. This method makes it possible to avoid unrealistic explanations of the form "In order to get the credit, you need to increase your budget and to decrease your salary".

Frye et al. [13] are also interested in the integration of causal relationships, in the case of explanations in the form of local feature importance vectors. They focus on the asymmetry brought by the causal link orientation and

propose to exploit it by deriving weights to define personalized variants of the cost function.

These four approaches integrate different types of user knowledge into different forms of explanation. They consider the available knowledge as an additional constraint in the explanation generation. This constraint is represented in different forms: reduction of the search space, addition of penalty in the cost function or weight in the cost function. In the following section, we propose a general formalization for such methods integrating knowledge.

3. Proposed General Framework

This section describes the general framework for building explanations that takes into account user knowledge, independently of the type of explanation and the type of knowledge. After describing the notations used throughout the paper, the proposed model is described and then its generality is illustrated showing how it can be instantiated to represent a reference existing approach.

3.1. Notations

The considered objective is to explain the prediction of a machine learning model denoted $f : \mathcal{X} \longrightarrow \mathcal{Y}$ where $\mathcal{X}$ and $\mathcal{Y}$ respectively denote the input and the output spaces. We consider a domain whose elements are described by d numerical features, i.e. $\mathcal{X} \subseteq \mathbb{R}^d$, equipped with the Euclidean distance. The case of categorical features, not explicitly studied in this paper, requires to modify this distance. Focusing on the common case of explanation for classification tasks, we consider $\mathcal{Y}$ to be a discrete set, e.g. $\{0, 1\}$ for binary classification or $\{c_1, \ldots, c_k\}$ for k classes. In the considered model and data agnostic paradigm, no further information is available about f nor about $\mathcal{X}$ except for the user knowledge. The latter is denoted E in its most general form, it will be instantiated in the following sections, Sections 4, 5 and 7.

$\mathcal{E}$ denotes the set of candidate explanations, independently of the their type: it may contain counterfactual examples or surrogate models. $e \in \mathcal{E}$ denotes a candidate explanation.

In the considered local explanation case, the explanation is required for a reference instance $x \in \mathcal{X}$ whose prediction $f(x)$ needs to be explained. In the following x may be used as subscript of the elements depending on it.

3.2. Proposed Optimization Problem

Following the principle of most post-hoc explanation methods, briefly reminded in Section 2.1, the generic framework we proposed is formulated as an optimization problem of an enriched cost function: the latter adds, to the classical penalty term that assesses the quality of a candidate explanation with respect to the considered instance x and classifier f, a term that depends on the user knowledge E. It is formally defined as

$$\underset{e \in \mathcal{E}}{\operatorname{argmin}}\, agg(penalty_x(e, f), incompatibility_x(e, E)) \tag{1}$$

where $penalty_x$, $incompatibility_x$ and agg are three functions described in turn in the following. As discussed below, the exact expression of these three functions depends on the studied context: the motivations of the user, the type of explanation to be generated or the considered type of knowledge.

3.2.1. Penalty Function

As recalled in Section 2.1, most existing approaches to generate explanations minimize a cost function denoted here $penalty_x$: it takes as argument a candidate explanation e and the classifier to be explained f, and it may depend on the studied instance x in the case of local explainers. The lower the penalty, the better the candidate explanation e.

In the case of counterfactual examples, penalty can be defined as the distance between the candidate and the considered instance, $d(x, e)$, to which an arbitrary high value Z is added if the class predicted for e is the same[1] as the one for x, i.e. $f(e) = f(x)$: $penalty_x(e, f) = d(x, e) + \mathbb{1}_{f(e)=f(x)} \times Z$. As a consequence, only candidates with a predicted class different from the one of x are considered. This is equivalent to restricting the set of explanations $\mathcal{E}$ to a local definition $\mathcal{E}_x = \{e \in \mathcal{E} / f(e) \neq f(x)\}$. Regarding the distance, usual choices include the Euclidean distance [21], or the l_0 norm to incorporate a sparsity constraint [7] to name a few.

In the case of surrogate explainers, penalty can be defined as the local fidelity of the surrogate e to the classifier f in the neighborhood of the studied instance x.

[1] In the case where the user makes explicit the class he/she is interested in, denoted $c^* \in \mathcal{Y}$ with $c^* \neq f(x)$, the additional term penalises candidates such that $f(e) \neq c^*$ and not only $f(e) = f(x)$. In the following we keep to the simple case where c^* is only required to differ from $f(x)$.

3.2.2. Incompatibility Function

In order to integrate the user knowledge, we propose to add a function that measures the extent to which the candidate explanation is incompatible with this knowledge, so as to favor the explanations that are in agreement with the latter.

Indeed, in the case of non-expert users, the aim is to increase the intelligibility of the explanation, relying on a common language. For instance in a credit application scenario, an explanation for a non-expert customer asking how to get his/her credit application accepted needs to focus on common knowledge actions such as increasing his/her salary. On the other hand, in the case where the explanation is generated for a domain expert, the incompatibility can represent the level of fidelity of the explanation with respect to the expert's comprehension of the domain. The lower the incompatibility, the more fidel the explanation.

Detailed instantiations are discussed in the next sections, we illustrate here the principle with a basic case. For a given candidate explanation e, we denote A_e the set of features is involves, independently of its form. For counterfactual example, it can for example contain the modified features, for local feature importance, it can be defined as the set of features with non-zero scores. Similarly, we denote A_E the features involved in the user knowledge, independently of its form. An incompatibility function then aims at measuring the discrepancy between A_e and A_E. Decreasing incompatibility can then mean minimizing the number of features the explanation takes into account that are not part of the user knowledge:

$$incompatibility_x(e, E) = Card(A_e \setminus A_E) \tag{2}$$

Note on a Different Semantics for Knowledge Integration. The above discussion considers that the explanation must be compatible with the user knowledge, which appears as a relevant choice that may contribute to increase the user's confidence in the machine learning models and his/her willingness to use it. However, we wish to discuss here an alternative, and actually contradictory, semantics for knowledge integration, that makes sense in a different scenario, for a different type of users with different objectives.

Indeed, in some cases, a user may be interested in an explanation that is actually orthogonal to his/her knowledge: he/she may wish to get a complementary explanation, likely to provide him/her new information he/she did not dispose of. In other words, his/her aim in such a case is to acquire

new, enriching, pieces of knowledge that should not be redundant with what he/she already knows. The desired explanation should then be not redundant with the knowledge he/she expresses, which is at cross-purposes of the knowledge integration principle discussed above.

In such a case, considering the example and notations given above, expressing that the explanation needs to be complementary to the knowledge, i.e. minimizing the redundancy, may for instance advocate for a function such as

$$incompatibility_x(e, E) = Card(A_e \cap A_E)$$

3.2.3. Aggregation Function

The aggregation function combines the penalty and incompatibility functions. There exists an abundant literature on aggregation operators (see e.g. [28, 29]) which can be divided into several categories depending on their behavior and semantics. For illustration purposes, we focus on three of them, called conjunctive, disjunctive or compromise operators. A conjunctive operator, for example the minimum function, returns a high value only if all criteria to be aggregated, here both penalty *and* incompatibility, are high. A disjunctive operator, such as the maximum function, requires only *at least one* value to be high. Trade-off operators such as the weighted average allow for the compensation of low values by high values. Other categories include operators offering a so-called reinforcement behavior, according to which if all criteria take high values, they result in an even higher value.

The choice of the aggregation function can be made according to the user preferences, instead of necessarily considering the same function for all of them: one way to personalize the explanation to a user is to let him/her choose the aggregation he/she wants to perform.

3.3. Illustration of the Proposed General Framework

In order to illustrate the generality of the proposed framework, we show in this section how it can be instantiated to express the state-of-the-art approach proposed by Ustun et al. [14], highlighting the definition of the three involved functions.

In Ustun et al. [14] approach, the user knowledge E is local and depends on the instance of interest: it is denoted $A(x)$ and defined as the set of allowed modifications, that can be applied to instance x. Its integration makes it possible to generate an actionable counterfactual explanation. The

latter is defined as the solution to the following optimization problem [14]:

$$\eta^* = \underset{\eta \in A(x)}{\operatorname{argmin}} \, cost_fct(\eta, x) \text{ with } f(x+\eta) \neq f(x)$$

$cost_fct$ is defined as the distance between x and $x+\eta$. The generated explanation η^* expresses the move between the instance of interest x and the closest instance in the opposite class $x+\eta^*$.

We propose to rewrite this optimization problem as:

$$e^* = \underset{e \in \mathcal{E}_x}{\operatorname{argmin}} \, cost_fct(e-x, x) + \mathbb{1}_{|x-e| \notin A(x)} \times Z$$

where $\mathcal{E}_x = \{x' \in \mathcal{X} \mid f(x') \neq f(x)\}$ and Z is an arbitrarily large real number. Here e^* is the closest instance in the opposite class, and it enables to retrieve η^* as $\eta^* = e^* - x$.

This expression, equivalent to the previous one, makes it possible to identify the *penalty*, *incompatibility* and *agg* functions, respectively defined as, for an arbitrary high value Z'

$$\begin{aligned} penalty_x(e, f) &= cost_fct(e-x, x) + \mathbb{1}_{f(e)=f(x)} \times Z' \\ incompatibility &= \mathbb{1}_{|x-e| \notin A(x)} \times Z \\ agg(u, v) &= u + v \end{aligned}$$

Indeed, the penalty function equals $cost_fct$ defined in [14] with the introduction of a arbitrary high value Z' for candidates that are predicted to be in the same class as x as discussed in Section 3.2.1. The latter expresses the restriction of the search space $\mathcal{E}$ to $\mathcal{E}_x$. The incompatibility function represents the presence or absence of the modified feature in the user knowledge; it takes only two values 0 or Z. An incompatible counterfactual explanation thus has a very high cost function, resulting in only compatible counterfactuals being considered. Finally, aggregation is performed by a sum. However, as the incompatibility function is binary, only compatible counterfactuals are considered, so the resulting counterfactual is both compatible and of good quality. In this setting, the aggregation function has a conjunctive behavior.

4. Knowledge Integration in Counterfactual Explanation (KICE)

This section introduces a first instantiation of the general framework described in the previous section, leading to a new explanation method taking

into account user knowledge in the case of counterfactual example explanations. It first describes the characteristics and motivations for the considered type of user knowledge, relevant both for counterfactual explanations and surrogate models considered in Section 5. It then describes in turn the proposed cost function and the algorithm optimizing it.

4.1. Characteristics of the Knowledge Type and Motivations

The knowledge we propose to consider takes the simple form of a set of features, $E = \{X_i, i = 1 \dots m\}$ with $m \leq d$ where d is the total number of features: within the framework discussed in Section 2.2, it corresponds to a case of individual feature-based user knowledge. It can be seen as similar to the case of actionable features, but its meaning differs: the information provided by the user does not provide indication about his/her ability to modify the associated values, but refers to a more basic level. As discussed in Section 3.2.2, two interpretations can be distinguished depending on whether the user is an expert or not, as illustrated in the following.

We consider the following fictitious example regarding a classification of vegetables: let us consider a sample predicted to be a carrot. We are interested in two candidate explanations e_1 and e_2 which explain why the model predicts that the sample is a carrot. The first explanation e_1 is: "The color of the sample is orange and the taste is sweet". The second explanation e_2 is: "The sample has a high rate of provitamine A and a high rate of saccharose".

We consider two users: a non-expert and an expert of the domain who express the same knowledge: $E = \{color, flavor\}$. Even if the knowledge is the same, it represents two different semantics. For the non-expert user, this knowledge represents the characteristics that he knows about and that he agrees to find in an explanation in order to better understand it. A non-expert user wants an explanation in his language so that it is understandable. He then prefers explanation e_1 because it contains only features presented in his knowledge. For the expert user, this knowledge represents the features whose impacts on the prediction are already known by him. He then wants additional information and an explanation that is not redundant with his knowledge. Thus, he wants a complementary explanation that does not contain the elements he already knows. He then prefers explanation e_2 because it has features that he has not expressed. In the following, we focus on the case of non-expert users who expect explanations in their language.

4.2. Proposed Cost Function: Instantiation of the Framework

This section describes the instantiation of the general cost function given in Eq. (1) for the case of explanation in the form of counterfactual examples $\mathcal{E}_x = \{x' \in \mathcal{X} \mid f(x') \neq f(x)\}$, user knowledge as a set of features and an explanation expressed in the user language. It describes in turn the penalty and incompatibility functions and the aggregation operator.

Penalty Function. For the penalty function, we propose to use the classical counterfactual function which is the Euclidean square distance:

$$penalty_x(e, f) = \| x - e \|^2 \tag{3}$$

with the restriction of the search space $\mathcal{E}$ to $\mathcal{E}_x$ to guarantee the prediction associated with the counterfactual example differs from the one of the reference instance.

Incompatibility Function. As specified above, the objective is to propose a counterfactual explanation in agreement with the user knowledge, meaning that ideally, the counterfactual modifications must be only performed according to features appearing in E. However, for some instances, focusing only on a subset features raises the risk of never being able to meet the decision boundary of f, leading to no counterfactual explanation being generated.

Therefore, we propose to relax this constraint by penalizing the modifications according to the features from $\overline{E}$, i.e. features that are not present in the knowledge E. This allows, when the boundary cannot be found along the sole features of E, to make sure that a solution can still be found. Thus, we propose the incompatibility function that computes the Euclidean square distance only using absent features:

$$incompatibility_x(e, E) = \| x - e \|^2_{\overline{E}} = \sum_{i \notin E} (x_i - e_i)^2 \tag{4}$$

Minimizing this incompatibility avoids generating counterfactual explanations that greatly modify the unknown features.

Aggregation Function. In order to combine the penalty function with the incompatibility function, we propose to use a compromise operator, more precisely a weighted average:

$$agg(u, v) = u + \lambda v \tag{5}$$

where $\lambda \in \mathbb{R}^+$ is a user-set hyper-parameter. The higher λ, the more intransigent the user is with respect to to his/her knowledge, i.e. the less open he/she is to features he/she may not understand. A high value of λ implies that the counterfactual example may be located further away from the reference instance, meaning the user may need to perform more modifications. On the other hand, these modifications only apply to features he/she understands the meaning of.

Global Function. We then obtain the following optimization problem:

Denoting $\mathcal{E}_x = \{x' \in \mathcal{X} \mid f(x') \neq f(x)\}$ and $\lambda \in \mathbb{R}^+$,

$$\begin{aligned} e^* &= \underset{e \in \mathcal{E}_x}{\operatorname{argmin}}\, cost_{x,E}(e) \\ \text{with } cost_{x,E}(e) &= \| x - e \|^2 + \lambda \| x - e \|_E^2 \end{aligned} \tag{6}$$

4.3. Description of algorithm KICE

This section describes the algorithm we propose to solve the optimisation problem defined in Equation (6) named KICE for Knowledge Integration in Counterfactual Explanation. Its general principle is presented below before a more detailed description of its generation step.

Principle. KICE exploits the principle of iterative generation of instances proposed by the Growing Spheres algorithm [7]: given f the considered classifier, x the instance for which an explanation is requested, E the set of user known features and λ the weight of the incompatibility, KICE generates instances around x in increasingly larger spaces until it finds one predicted by f with a different class. The integration of E in the incompatibility term in the cost function calls for a non uniform generation: the space is distorted, with features being associated to different weights.

Indeed, for any ν, the equation $cost_{x,E}(e) = \nu$ with *cost* presented in Equation (6) defines an ellipse with center x and radius $\sqrt{\frac{\nu}{1+\lambda}}$ with respect to features in $\overline{E}$ and $\sqrt{\nu}$ with respect to features in E. The spaces to be considered are thus not spherical layers, but ellipsoidal ones.

This principle is illustrated on Figure 1 that shows a toy 2D-data set, the classifier prediction with the colored regions and an instance of interest x. The considered user knowledge is defined as $E = \{X_2\}$. As a consequence, it is less costly to modify feature X_2 than feature X_1. Tentative counterfactual examples are thus iteratively generated in the "vertical" ellipses until reaching an instance predicted in the other class e^*.

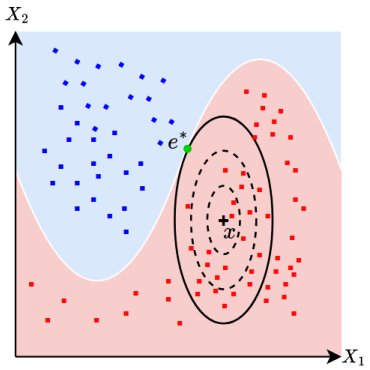


Figure 1: Illustration of KICE ellipsoidal layers generation step for a toy 2D data set: f is indirectly defined by the colored regions, user knowledge E is $\{X_2\}$, x denotes the instance of interest, the ellipses the considered increasingly large spaces around x, e^* the finally generated counterfactual.

Uniform Generation in Ellipsoidal Layers. In order to generate candidates uniformly in the ellipsoidal layers, KICE relies on a modified version of the HLG algorithm [30] presented in Algorithm 1. The latter generates instances uniformly in the spherical layer $SL(x, a_0, a_1)$ defined as the set of points at a distance greater than a_0 and less than a_1 from x. Modification of the algorithm HLG we propose relies on the random variable $U = \{u_i\} \sim \mathcal{U}([0,1])$ used to generate points according to $a_0 + a_1 U^{\frac{1}{dim(\mathcal{X})}}$.

To modify HLG we propose to distinguish between features in E and in $\overline{E}$: for the former, the same procedure as HLG is applied, considering $a_0 + a_1 u_i$ for $i \in E$. For the latter, the generation is weighted by $1/\sqrt{1+\lambda}$, i.e. considers $\frac{a_0}{\sqrt{1+\lambda}} + \frac{a_1}{\sqrt{1+\lambda}} u_i$ for $i \in \overline{E}$.

KICE overview. Globally, the KICE algorithm covers the space by generating instances iteratively: in an initialisation step, n instances are generated in the ellipse of center x and radius $\sqrt{\frac{\nu_0}{1+\lambda}}$ with respect to the features in $\overline{E}$ and $\sqrt{\nu_0}$ with respect to the features in E, where n and ν_0 are hyperparameters. If none of these instances belong to the opposite class, KICE generates instances in the ellipsoidal layer between ν_0 and $\nu_0 + \epsilon$ where $\epsilon > 0$ is a third hyperparameter. Then, at each iteration if none of generated instances belong to the opposite class, new instances are generated in a new layer with new base radius $\nu_0 \leftarrow \nu_0 + \epsilon$.

Algorithm 1 Ellipsoïdal Layer Generation

Require: x, the center of the ellipsoïdal layer
Require: E, the prior knowledge
Require: a_0 and a_1 the bounds of the layer
Require: n, number of desired points
Ensure: $Z = \{z_i\}_{i \leq n}$

```
Y = {y_i}_{i≤n} ~ N(0, 1)
Y ← Y / ||Y||_2
U ← {u_i}_{i≤n} ~ U([0, 1])
R ← a_0 + a_1 U^{1/dim(X)}
W ← R^T Y + x
R' ← a_0/√(1+λ) + a_1/√(1+λ) U^{1/dim(X)}
W' ← R'^T Y + x
for i ← 1 to n do
    if i ∈ E then
        Z_i ← W_i
    else
        Z_i ← W'_i
    end if
end for
return Z
```

5. Knowledge Integration in Surrogate Models (KISM)

This section presents a second instantiation of the general framework described in Section 3: it considers the same type of user knowledge as KICE, as discussed in Section 4.1, but describes its integration to explanations in the form of linear surrogate models. This section first describes the proposed cost function and then presents the algorithm proposed to optimize it.

5.1. Proposed Cost Function: Instantiation of the Framework

This section discusses in turn the three components of the general cost function, penalty, incompatibility and aggregation function, using the same notations as in Section 4. $\mathcal{E}$ denotes the set of candidate surrogate models, for instance linear regression models in the case of LIME [4]. In this case, $\mathcal{E} = \{g | \forall z \in \mathcal{X}, g(z) = \sum_{i=1}^{d} w_i z_i + w_0\}$ and the final explanation is a feature importance vector obtained from the w^* associated to the chosen $e^* \in \mathcal{E}$. To

simplify the notations, we do not note the link between the model g and his coefficients w, i.e. we use a simplified notation w_i instead of w_i^g. The set $\mathcal{E}$ may also be a set of decision trees which are then considered as transparent models and thus directly define an interpretable explanation. For any $e \in \mathcal{E}$, and for any data point $z \in \mathcal{X}$, $e(z)$ then denotes the prediction performed by the surrogate model e for z. For such surrogate approaches the black box classifier f to be explained is usually considered to output continuous probability scores that are later thresholded to get a prediction result.

Penalty Function. We propose to define the penalty using the classical surrogate cost function: it assesses the fidelity of e which respect to f, measured by their l_2 difference on a neighborhood $\mathcal{Z}_x$ around the instance of interest x. The definition of the neighborhood $\mathcal{Z}_x$ is a crucial and non trivial step of the processus, as a.g discussed by Laugel et al. [31]. Formally,

$$penalty_x(e, f) = \sum_{z \in \mathcal{Z}_x} (f(z) - e(z))^2 + \Omega(e) \tag{7}$$

The first term captures the local fidelity of the surrogate model to f and the second term $\Omega(e)$ captures the complexity of the candidate explanation e. The latter depends on the type of surrogate classifiers. In the classical case of linear regression models, the complexity can be the l_0, l_1 of l_2 norms of the regression coefficients, depending on whether a simple, lasso or ridge regression is performed. In the following, we consider the case of the l_2 norm with weight α: $\Omega(e) = \alpha \sum_{i=1}^{d} w_i^2$.

Incompatibility Function. In order to value the agreement with the user knowledge, we consider that the surrogate-model explanation can be associated with weights on the features, denoted w_i: this is obviously the case for linear regression surrogates; when decision trees are considered, the depth of the occurrence of the feature allows to derive such weights.

We then interpret the user knowledge E as providing the most important features and wish to favor, when possible, the coefficients associated to these variables. For this purpose, we thus propose to penalize the coefficients associated to features that do not belong to E.

To match our choice to define the complexity Ω using a l_2 penalty, we propose to define the incompatibility function as:

$$incompatibility_x(e, E) = \sum_{i \in \overline{E}} w_i^2 \tag{8}$$

with w_i the weights associated with the features.

Aggregation Function. As in Section 4.2, we propose to formulate the new explanation objective as an explicit trade-off between the quality and the compatibility of the explanation, through a weighted average.

Global Function. As a consequence, the proposed optimization problem is:

$$e^* = \underset{e \in \mathcal{E}}{\operatorname{argmin}}\, cost_{x,E}(e) \tag{9}$$

$$\text{with } cost_{x,E}(e) = \underset{e \in \mathcal{E}}{\operatorname{argmin}} \sum_{z \in \mathcal{Z}_x} (f(z) - e(z))^2 + \alpha \sum_{i=1}^{d} w_i^2 + \lambda \sum_{i \in \overline{E}} w_i^2$$

with α and $\lambda \in \mathbb{R}^+$.

5.2. Algorithm Description

We propose to solve the optimization problem defined in Eq. 9 in the case of linear surrogate models, i.e. when $\mathcal{E} = \{g | \forall z \in \mathcal{X}, g(z) = \sum_{i=1}^{d} w_i z_i + w_0\}$. The solution of a typical ridge regression problem to minimize can be expressed as:

$$W = (X^t X + \alpha I)^{-1} X^t y$$

where X denotes the matrix representing the neighborhood data, W the vector of w_i, y the labels associated to X and I the identity matrix.

Now the cost function defined in Eq. 9 belongs to this framework, with an additional λ to weight the features outside the user knowledge. Thus, denoting $I_{\overline{E}}$ an identity matrix containing 1 only on the rows associated to unknown features, the solution to the considered optimization problem can be rewritten

$$(X^t X + \alpha I + \lambda I_{\overline{E}})^{-1} X^t y$$

where X denotes the data points generated in the neighborhood $\mathcal{Z}_x$ and y the vector of their associated prediction by the classifier f to be explained.

This solution makes explicit the integration of knowledge within the method. The integration of a new term does not change the complexity and the procedure to apply the algorithm.

6. Experiments

This section presents the experiments performed to study the explanations generated by the proposed methods KICE and KISM and to assess their relevance.

Datasets	λ	ϵ	n	ν_0
Half-moons	4	0.01	200	0.1
Boston	3	0.02	1000	0.2
Breast Cancer	6	0.3	2000	5

Table 1: Values of the hyperparameters λ, ϵ, n and ν_0 of KICE chosen for the three datasets.

6.1. Experimental Protocol

Experiments are conducted on three classical benchmark tabular data sets: Half-Moons, Boston and Breast Cancer where the first one is a dataset generated with the package sklearn and the two last ones are UCI datasets. As a pre-treatment, the data are normalized and in the case of the Boston data set, the regression value is transformed into a binary class by discretization: the price is "expensive" if it is greater than $\$21,000$, and "cheap" otherwise. The data sets are split into train and test data subsets (80%-20%).

Within the considered post-hoc explanation framework, the classifier choice does not matter, we apply SVM with a Gaussian kernel, that achieves a high accuracy on the three data sets: 0.99, 0.98 and 0.93 respectively for Half-Moons, Boston and Breast cancer.

Explanations e^* are then generated for each instance x of the test data set using different methods. First, python implementation of KICE and KISM, using the numpy, sklearn and lime libraries, are run, using the λ, ϵ, n and ν_0 parameters values shown in the Table 1. ϵ, n and ν_0 are chosen according to the dimension of the datasets and the number of features considered for E: the more features the dataset contains, the higher the value of the parameters. λ is chosen arbitrarily, in order to obtain interesting results, to have explanations that differ from the competitor ones, for the competitors described below.

The first competitor, generating the explanation denoted e_{ref}, corresponds to the case where no knowledge is taken into account: it solves the reference optimization problem that only minimizes the penalty function. This competitor corresponds to Growing Spheres [7] in the case of counterfactual explanations, and LIME [4] in the case of surrogate models. This corresponds to an extreme case of aggregation of Eq. (6) where the incompatibility term is ignored, i.e. setting $\lambda = 0$.

In the case of counterfactual explanation, a second competitor is proposed

by imposing to comply strictly with the knowledge under the constraint it is predicted in the opposite class. Its associated cost function thus equals the incompatibility function, a naive way of integrating knowledge in the explanation. In this case, we denote e_{user} the counterfactual instance that solves the associated problem:

$$e_{user} = \underset{e \in \mathcal{E}}{\operatorname{argmin}} \parallel x - e \parallel_E^2$$

This corresponds to the other extreme case of aggregation of Equation (6), where the penalty term is ignored, i.e. λ arbitrarily large. We do not consider Ustun et al. as competitor although it shares common points with the proposed KICE because it is restricted to linear classifiers and cannot be applied to the SVM with Gaussian kernel considered here).

Finally, regarding user knowledge E, we consider that the user disposes of less features than the classifier. In order to build somehow realistic knowledge, we train a decision tree with low depth on the train data (more precisely the maximal depth is set to half the number of data features), the set of features E then contains the ones that occur in this tree.

6.2. KICE Experimental Study

This section presents the experiments conducted to evaluate the KICE algorithm: the purpose of these experiments is to show that the proposed method finds the expected counterfactual instance, i.e. it allows to achieve a trade-off between the quality and the compatibility of the explanation.

6.2.1. Illustrative Examples

First, we illustrate the behavior of the methods with the 2D Half-Moons data set, denoting X_0 and X_1 the two dimensions. Figure 2 shows the counterfactual examples obtained for three different instances x. The other points represent the training examples, the blue and red regions represent the predicted classes. The considered knowledge system contains a single feature, it is represented by the horizontal line: $E = \{X_1\}$.

As expected, we can observe that: (i) the counterfactual instance e_{ref} is the closest point belonging to the opposite class; (ii) e_{user} is further away than e_{ref} and only modifies feature X_1 which is the knowledge feature; (iii) e^* is a compromise between e_{ref} and e_{user}. It requires less modifications according to X_0 than e_{ref}, hence it is more compatible. It is also closer to the studied instance than e_{user}.

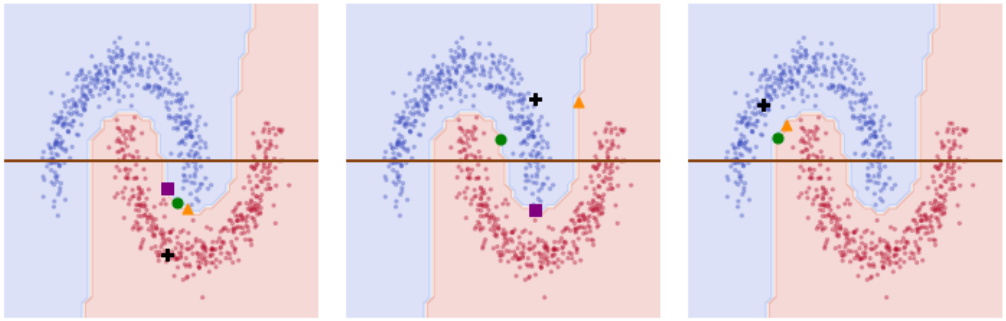


Figure 2: Examples of resulting e_{ref}, e_{user} and e^* for three instances x (+: x, ▲: e_{ref}, ■: e_{user}, ●: e^*)

In the graph, on the right, we notice that there is no e_{user}. In this case, there is no counterfactual instance modifying only feature X_1. The proposed KICE method is then useful because it allows to obtain a more compatible counterfactual than e_{ref}.

6.2.2. Evaluation of the KICE Method

In this section, we further compare KICE with its competitors and study quantitatively the gain in terms of the proposed cost function.

We apply the three methods described in the previous section on the test set of the three data sets. Among the test set, for some instances no counterfactual example is found by the method that modifies only the features in E (e_{user}). This happens to the half-moons and breast cancer data sets, for respectively 20% and 11% of the instances. For the other instances, Table 2 shows the mean and standard deviation values of the penalty, incompatibility and cost functions for the three approaches.

We observe that, as expected, the proposed counterfactual examples have a penalty greater than that of e_{ref} but lower than that of e_{user}. Moreover, the incompatibility function is much smaller than that of e_{ref}. Thus, the modifications on the unknown features decrease, which increases the understanding of the explanation. Finally, e^* cost function is the lowest. We can notice that the standard deviations are high, which is due the fact that the instances are at different distances from the boundary.

Table 3 shows the execution times to obtain the three counterfactuals. As expected, for all three data sets, we notice that the time associated with e^* is higher than that of e_{ref}. Since only one feature is taken into account,

		$penalty_x(e)$	$incompatibility_x(e, E)$	$cost_{x,E}(e)$
Half moons	e_{ref}	**0.32 ± 0.21**	0.14 ± 0.13	0.86 ± 0.56
	e_{user}	1.48 ± 1.3	**0.0 ± 0.0**	1.48 ± 1.3
	e^*	0.42 ± 0.29	0.08 ± 0.11	**0.73 ± 0.52**
Boston	e_{ref}	**1.48 ± 1.75**	0.7 ± 1.03	3.57 ± 4.72
	e_{user}	2.26 ± 2.71	**0.0 ± 0.0**	2.26 ± 2.71
	e^*	1.72 ± 2.09	0.13 ± 0.19	**2.12 ± 2.54**
Breast cancer	e_{ref}	**8.82 ± 9.22**	7.27 ± 8.25	52.41 ± 58.63
	e_{user}	22.42 ± 24.87	**0.0 ± 0.0**	22.42 ± 24.87
	e^*	10.74 ± 9.85	1.25 ± 1.33	**18.24 ± 16.83**

Table 2: Quantitative evaluation of the three considered approaches on the three datasets for metrics: *penalty* defined in Eq. (3), *incompatibility* defined in Eq. (4) and *cost* defined in Eq. (6). Average and standard deviation are computed on all instances x of the test set for which all three approaches provide an explanation.

	e_{ref}	e_{user}	e^*
Half-moons	0.06 ± 0.04	0.25 ± 0.11	0.15 ± 0.10
Boston	0.17 ± 0.23	0.22 ± 0.19	0.30 ± 0.47
Breast Cancer	0.69 ± 0.65	0.19 ± 0.09	1.61 ± 1.57

Table 3: Execution times of the three considered approaches to obtain e_{ref}, e_{user} and e^* for the three datasets.

the e_{user} counterfactuals are further away, and require more iterations to be identified, increasing the execution time. For the two datasets Boston and Breast Cancer, the time associated with e^* is higher than the other two. The higher the dimension of the dataset, the longer the execution time. As in the evaluation of the metrics, we also note that the standard deviations are high as the studied instances are located in different areas of the data space.

To verify that KICE minimizes the cost function as opposed to the other two on all instances, Figure 3 shows the value of the cost function obtained by e^* (that is defined as minimizing it) as compared to the value it takes for e_{ref} (left) and e_{user} (right), for each of the test instances of the Half-moons dataset. The figures show that, as expected, all points are above the line $y = x$.

On the right hand graph, the points are more scattered but they remain above the line. We notice that the generated counterfactual instances are closer to the cost function of e_{ref} than to e_{user}. The user knowledge contains

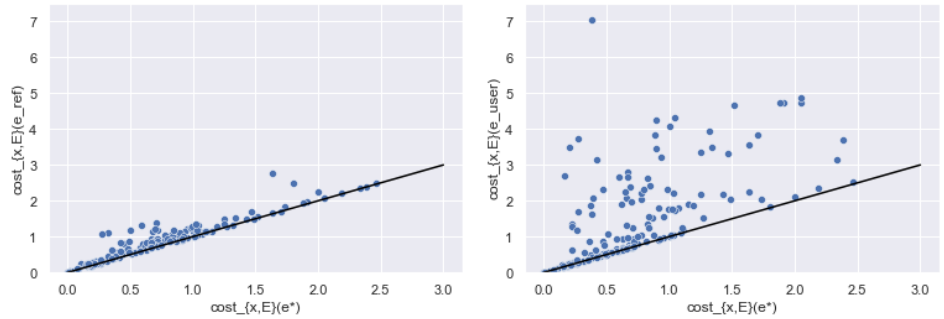


Figure 3: Cost functions *cost* as defined in Eq. (6) of e_{ref}, e_{user} and e^* for the 80% of the Half-moons test set for which all three counterfactual instances are defined. (left) Cost functions associated to e_{ref} and e^*, (right) cost functions associated to e_{user} and e^*.

a single feature, it is difficult to get a close counterfactual instance only by modifying this feature. It is then necessary to move further away from the studied instance to get 100% compatible one. It is also possible to increase λ so that the points are less scattered.

6.3. KISM Experimental Study

This section describes the experiments conducted with the KISM method by presenting illustrative examples and by showing that the algorithm achieves its goal, i.e. it associates a more important weight to the features included in the user knowledge. For the experiments, we arbitrarily set their weight in the ridge regression to 1.

In order to make the numerical evaluation easier, through the study of the rank associated to the features present in the user knowledge, we restrict the latter to contain a single feature: for Half Moons, except for the illustrative examples given below, $E = \{X_1\}$, for Boston, $E = \{X_7\}$ and for Breast Cancer, $E = \{X_3\}$. The value of λ is set to 500, the LIME parameters are set to their default values.

Explanations e^* are compared to one competitor that corresponds to LIME. This competitor solves the reference optimization problem that only minimizes the penalty function, the explanation is denoted e_{ref}.

6.3.1. Illustrative Examples

First, we illustrate the behavior of the methods using the same 2D Half-Moons data set as before: for the instance x represented with a +, Figure 4 shows the decision boundaries (corresponding to a 0.5 probability) of the

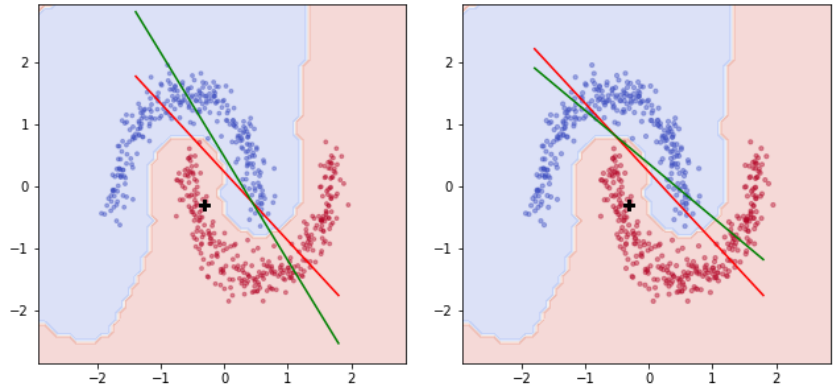


Figure 4: Surrogate models for KISM e^* and reference e_{ref} for the same instance x and different user knowledge: left $E = \{X_0\}$, right $E = \{X_1\}$ (+: x, —: e_{ref}, —: e^*)

	e_{ref}	e^*	Ranking gain
Half-Moons	1.63 ± 0.48	1.55 ± 0.5	0.09 ± 0.28
Boston	6.08 ± 2.89	4.49 ± 2.74	1.58 ± 1.35
Breast Cancer	13.18 ± 5.19	8.51 ± 4.86	4.67 ± 2.46

Table 4: Average rank, with standard deviation, of the user defined feature, for the two considered methods and the three data sets, together with the ranking gain.

two surrogate models obtained by KISM, denoted e^* and shown in green, and the reference e_{ref} shown in red. On the left graph, the considered user knowledge is $E = \{X_0\}$ and on the right graph, it corresponds to the other feature $E = \{X_1\}$.

In both graphs, the classic surrogate model e_{ref} shown in red, that does not take into account any user knowledge, provides the same result. On the left graph, we observe the weight according the feature X_0 is higher for e^* (green line) than for e_{ref} (red line). On the right graph, we observe the weight according to feature X_1 is higher for e^* (green line) than for e_{ref} (red line). The proposed KISM method is thus useful, as it effectively allows for an explanation to give more importance to known features than to unknown features.

6.3.2. Evaluation of the method

This section discusses the evaluation of the proposed method according to different criteria. First, for each obtained explanation, we rank the features

		$penalty_x(e)$	$incompatibility_x(e, E)$	$cost_{x,E}(e)$
Half-moons	e_{ref}	**274.11 ± 166.31**	0.05 ± 0.03	298.93 ± 190.32
	e^*	279.18 ± 171.00	**0.03 ± 0.03**	**291.91 ± 183.23**
Boston	e_{ref}	**213.19 ± 169.60**	0.08 ± 0.11	245.22 ± 186.16
	e^*	204.97 ± 171.79	**0.04 ± 0.05**	**233.82 ± 180.03**
Breast cancer	e_{ref}	**10.00 ± 12.24**	0.04 ± 0.08	31.94 ± 48.67
	e^*	14.47 ± 18.56	**0.02 ± 0.04**	**25.65 ± 37.55**

Table 5: Results with the two considered approaches on the three data sets for metrics: *penalty* defined in Eq. (7), *incompatibility* defined in Eq. (8) and *cost* defined in Eq. (10)

of the explanation in increasing order of absolute value of weights. Then, we study the position in this ranking of the user's feature, denoted E with a notation abuse below. Table 4 gives, in the first column, the average ranking (with standard deviation) over all instances of the test set in the explanations obtained without taking into account the knowledge (e_{ref}, provided by LIME). In the second column, the same average ranking is shown for the proposed KISM approach. The third column gives the number of positions gained between the two methods. We observe that the studied feature moves upward in the ranking in average (between 0.09 for half-moons and 4.67 for the breast cancer data set) when it is taken into account by the method. This effect is more visible for the data sets with more features. Indeed, in the 2D Half-Moons case, only two rank values are possible, which makes it harder to modify the values. Altogether, this shows that the proposed KISM method helps increasing the importance given to the user feature.

Second, we study the two methods according to the three criteria presented in Section 3: penalty, incompatibility and cost function. To do so, we apply LIME and the proposed method on all test instances of the different data sets. As expected e_{ref} minimizes the penalty function and e^* the incompatibility function. As compared to e_{ref}, the proposed explanation e^* does have a higher penalty, a lower incompatibility and altogether a lower cost function, meeting the desired behavior.

The proposed approach is based on the same procedure as LIME and only changes the type of considered model. Thus, the complexity of the proposed approach is the same as that of LIME, the time needed to obtain e_{ref} and e^* is the same.

7. Discussion on Knowledge Expressed as Rules

This section proposes to discuss the relevance and challenge of considering a richer type of user knowledge, beyond the case of a set of features, in the form of logical rules, and integrating it into counterfactual explanations. We name rKICE this rule Knowledge Integration in Counferfactual Explanation principle.

7.1. Characteristic of the Knowledge Type and Motivations

User knowledge expressed as a rule based system is defined as a set of logical implications that allow the user to make his/her own prediction for any instance. Formally, for any $x \in \mathcal{X}$, a rule is defined as

$$\bigwedge_{i=1}^{d} x_i \in [v_{inf}^i, v_{sup}^i] \Longrightarrow class = y$$

with $v_{inf}^i \in \mathbb{R}$ and $v_{sup}^i \in \mathbb{R}$ the lower and upper bounds of the interval over the i-th feature, possibly with infinite values. Such a rule can also be written

$$\bigwedge_{i \in A_R} x_i \in [v_{inf}^i, v_{sup}^i] \Longrightarrow class = y$$

where A_R denotes the set of features whose associated interval is not $\mathbb{R}$. A_R then corresponds to the user knowledge considered in the previous section.

As compared to the knowledge type considered in the previous sections, defined as set of features, rule-based knowledge is much richer, as it provides two additional pieces of information: each involved feature is associated with an interval and the associated prediction of a class.

We consider that for any instance of interest x, a single rule of the user knowledge E is triggered. This rule could be denoted $E(x)$, we use the notation and keep E, not making explicit this dependency. This hypothesis is not restrictive as several rules can be combined into a single one using the appropriate logical operators.

It seems reasonable to make the assumption that the user requests an explanation for instances for which the prediction of the classifier differs from the rule's decision.

7.2. Proposed Cost Function for rKICE

As in Sections 4 and 5, this section discusses the definition of the three components of the generic framework, the penalty, incompatibility and aggregation functions, for the integration of rule-based knowledge in counterfactual explanations.

Penalty Function. We propose to evaluate a candidate counterfactual example using the classical counterfactual penalty, as in Eq. (3):

$$penalty_x(e) = \| x - e \|^2$$

with the restriction of $\mathcal{E}$ to $\mathcal{E}_x$ to guarantee the prediction associated with the counterfactual example differs from the one of the reference instance.

Incompatibility Function. In order to integrate the user knowledge in the explanation, we propose to impose that the counterfactual example belongs to the same region as defined by the user rule set, as the instance of interest: in other words the generated example needs to satisfy the premise of the rule triggered by the instance. Indeed, this premise describes a region where the user possesses some knowledge and makes sens to him/her.

Hence we propose to penalize counterfactual candidates whose values for the features involved in the rule premise do not satisfy the associated rule constraints. Formally the incompatibility function is thus defined as:

$$\sum_{i \in A_E} (x_i - e_i)^2 \times \mathbb{1}_{e_i \notin [v^i_{inf}, v^i_{sup}]}$$

with A_E the set of features present in the premise of the rule E.

Aggregation Function. As in the previous proposed instantiations of the generic framework, we propose to aggregate the two terms of penalty and incompatibility using a trade-off operator, the weighted average.

Global Function. The derived optimization problem for rKICE can thus be rewritten

$$\begin{aligned} e^* &= \underset{e \in \mathcal{E}}{\operatorname{argmin}}\, cost_{x,E}(e) \qquad (10) \\ \text{with } cost_{x,E}(e) &= \underset{e \in \mathcal{E}_x}{\operatorname{argmin}} \| x - e \|^2 + \lambda \sum_{i \in A_E} (x_i - e_i)^2 \times \mathbb{1}_{e_i \notin [v^i_{inf}, v^i_{sup}]} \end{aligned}$$

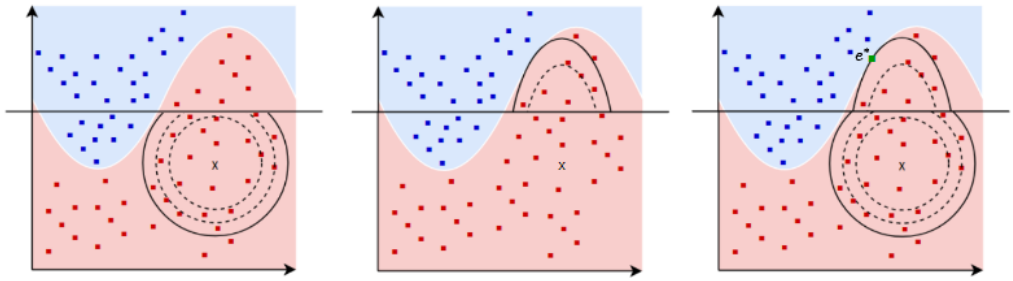


Figure 5: Right: rKICE generation mode, as a combination of the Growing Sphere (left) and KICE (middle) modes. The user rule is "if $X_1 < t$, then class=blue".

7.3. Comparison of KICE and rKICE principles

This section analyzes the rKICE optimization problem defined in Eq. (10) and compares it to the KICE optimization problem defined in Eq. (6). The first difference between these two problems is that rKICE incompatibility function depends on the features present in the user knowledge, whereas, on the contrary, the KICE one depends on the features absent from the user defined set of features. The principle of KICE may be combined to that of rKICE, through a more complex incompatibility function additionally penalizing the use of features absent from the premise of the triggered rule. However, the KICE incompatibility term is motivated by the notion of features known to the user, i.e. that he/she may understand. Following these lines it could be relevant to define the rule induced set of features as the union of the sets of features involved in all the rules. This variety of possibilities, that are all instantiations of the proposed generic framework, shows that the definition of an incompatibility term for rich knowledge as expressed by a rule set is a complex task that requires a refined study, out of the scope of this paper.

Focusing on the basic definitions given in Eq. (10) and Eq. (6) for rKICE and KICE respectively, one can distinguish between two cases for a given candidate explanation e: in the case where it satisfies the triggered rule premise, the incompatibility term equals zero and the cost function reduces to the classical counterfactual example case. Otherwise, let us denote A'_E the set of features involved in the triggered rule E whose interval constraint is not satisfied: the cost function is then of the form of the KICE one, for a user feature set defined as $\overline{A'_E}$. Figure 5 illustrates the combination of these two cases in a 2D toy data set, in the case where the user triggered rule is

"if $X_1 < t$, then class=blue", where t is a threshold value as represented on the graph (this rule is chosen to be a not appropriate one with respect to the classification task, for illustrative purposes). For the given instance x predicted as pink for the considered black box, the candidate counterfactual examples are generated along spherical layers in the region that does satisfy the rule premise, i.e. below the line and along ellipsoidal layers, in a KICE mode, above the line. The resulting combined generation form for rKICE is shown on the right graph.

8. Conclusion

In this paper, we propose a general framework defining an optimization problem to integrate user knowledge in post-hoc model-agnostic explanations. Using this framework, we propose two methods integrating knowledge as a set of features, interpreted as the features the user understands and can thus be exploited to build the explanation: KICE for the case of counterfactual explanations and KISM for the case of linear surrogate model explanations. KICE generates counterfactual explanations in the knowledge language by minimizing modifications according to unknown features, Similarly, KISM generates a linear model approximating the classifier to be explained in the knowledge language by minimizing the weights of unknown features. The relevance of these approaches are illustrated and evaluated numerically on several benchmark data sets. The paper also proposes a discussion extending the principles to the case of richer user knowledge in the form of a set of logical rules, showing the challenges of defining the incompatibility function.

The opened directions for future works are numerous and first include experiments with real users, to assess their satisfaction of the offered enriched explanations. A crucial question is also that of collecting their knowledge to be integrated in the generated explanations, which probably calls for the design of dedicated and user-friendly interfaces. It seems that an interactive process, e.g. generating first explanations and allowing the users to react and ask questions regarding some involved features, might be a direction to explore, in line with the desirable interactive property of explanations [20]. Ongoing works aim at developing the exploitation of user knowledge expressed as rule-based systems, among others the impact of rule fidelity with respect to the classifier to be explained. Other works aim at exploring and proposing

instantiations of the proposed generic framework for other explanation and knowledge types.

References

[1] N. Burkart, M. F. Huber, A Survey on the Explainability of Supervised Machine Learning, Journal of Artificial Intelligence Research 70 (2021) 245–317.

[2] P. Linardatos, V. Papastefanopoulos, S. Kotsiantis, Explainable AI: A Review of Machine Learning Interpretability Methods, Entropy 23 (1) (2021).

[3] C. Molnar, Interpretable machine learning, A Guide for Making Black Box Models Explainable, `https://christophm.github.io/interpretable-ml-book/`, 2022.

[4] M. T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?" Explaining the predictions of any classifier, in: Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining, 2016, pp. 1135–1144.

[5] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Proc. of the 31st Int. Conf. on Neural Information Processing Systems, 2017, pp. 4768–4777.

[6] 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.

[7] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, Comparison-based Inverse Classification for Interpretability in Machine Learning, in: Proc. of Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2018, pp. 100–111.

[8] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, P. Flach, FACE: Feasible and Actionable Counterfactual Explanations, in: Proc. of the AAAI/ACM Conf. on AI, Ethics, and Society, 2020.

[9] T. Laugel, M.-J. Lesot, C. Marsala, X. Renard, M. Detyniecki, The dangers of post-hoc interpretability: Unjustified counterfactual explanations, in: International Joint Conference on Artificial Intelligence, 2019.

[10] S. Barocas, A. D. Selbst, M. Raghavan, The hidden assumptions behind counterfactual explanations and principal reasons, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, 2020, pp. 80–89.

[11] C. Rudin, Stop Explaining Black Box Machine Learning Models for High Stakes decisions and Use Interpretable Models Instead, Nature Machine Intelligence 1 (2019) 206–215.

[12] D. Mahajan, C. Tan, A. Sharma, Preserving Causal Constraints in Counterfactual Explanations for Machine Learning Classifiers, NeurIPS workshop (2019).

[13] C. Frye, C. Rowat, I. Feige, Asymmetric Shapley values: incorporating causal knowledge into model-agnostic explainability, in: Proc. of Advances in Neural Information Processing Systems, Vol. 33, 2020.

[14] B. Ustun, A. Spangher, Y. Liu, Actionable Recourse in Linear Classification, in: Proc. of the Conf. on Fairness, Accountability, and Transparency, Association for Computing Machinery, 2019, p. 10–19.

[15] A. Jeyasothy, T. Laugel, M.-J. Lesot, C. Marsala, M. Detyniecki, Integrating prior knowledge in post-hoc explanations, in: Proc. of the Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2022.

[16] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A Survey of Methods for Explaining Black Box Models, ACM Comput. Surv. 51 (5) (2018).

[17] S. Verma, J. Dickerson, K. Hines, Counterfactual explanations for machine learning: A review (10 2020). arXiv:2010.10596.

[18] I. Stepin, J. M. Alonso, A. Catala, M. Pereira-Fariña, A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence, IEEE Access 9 (2021) 11974–12001.

[19] R. Guidotti, Counterfactual explanations and how to find them: literature review and benchmarking, Data Mining and Knowledge Discovery (2022) 1–55.

[20] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1–38.

[21] M. T. Lash, Q. Lin, N. Street, J. G. Robinson, J. Ohlmann, Generalized Inverse Classification, in: Proc. of the SIAM Int. Conf. on Data Mining, 2017, p. 162–170.

[22] R. K. Mothilal, A. Sharma, C. Tan, Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations, in: Proc. of the Conf. on Fairness, Accountability, and Transparency, ACM, 2020.

[23] T. Peltola, Local Interpretable Model-agnostic Explanations of Bayesian Predictive Models via Kullback-Leibler Projections, in: Proc. of the 2nd Workshop on Explainable Artificial Intelligence (XAI 2018) at IJCAI/ECAI 2018, 2018.

[24] K. Sokol, A. Hepburn, R. Santos-Rodriguez, P. Flach, bLIMEy: Surrogate Prediction Explanations Beyond LIME, in: Proc. of the HCML@NeurIPS, 2019.

[25] A. Van Looveren, J. Klaise, Interpretable Counterfactual Explanations Guided by Prototypes, in: Proc. of European Conf. on Machine Learning, 2021.

[26] X. Zhao, W. Huang, X. Huang, V. Robu, D. Flynn, Baylime: Bayesian local interpretable model-agnostic explanations, in: Proc. of the 37th Conf. on Uncertainty in Artificial Intelligence, Vol. 161 of Proceedings of Machine Learning Research, PMLR, 2021, pp. 887–896.

[27] M. Drescher, A. H. Perera, C. J. Johnson, L. J. Buse, C. A. Drew, M. A. Burgman, Toward rigorous use of expert knowledge in ecological research, Ecosphere 4 (7) (2013).

[28] T. Calvo, G. Mayor, R. Mesiar (Eds.), Aggregation Operators: New Trends and Applications, Vol. 97, Springer, 2002.

[29] M. Grabisch, J. Marichal, R. Mesiar, E. Pap, Aggregation Functions, no. 127 in Encyclopedia of Mathematics and its Applications, Cambridge Univ. Press, 2009.

[30] M. E. Muller, A Note on a Method for Generating Points Uniformly on N-Dimensional Spheres, Commun. ACM 2 (4) (1959) 19–20.

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