Sorbonne Université

École doctorale Informatique, Télécommunications et Électronique (Paris)

Équipe LFI, LIP6

Interprétabilité locale post-hoc des modèles de classification "boîtes noires"

Par Thibault Laugel

Thèse de doctorat d'Informatique

Dirigée par Marie-Jeanne Lesot, Christophe Marsala et Marcin Detyniecki

Présentée et soutenue publiquement le 16 Mars 2020

Devant un jury composé de :

Fosca Giannotti	KDDLab, ISTI-CNR	Rapporteur
Jamal Atif	LAMSADE, Univ. Paris-Dauphine	Rapporteur
Chris Russell	Alan Turing Institute	Examinateur
Nicolas Maudet	LIP6, Sorbonne Université	Examinateur
Marie-Jeanne Lesot	LIP6, Sorbonne Université	Directeur de thèse
Christophe Marsala	LIP6, Sorbonne Université	Directeur de thèse
Marcin Detyniecki	AXA, Paris	Directeur de thèse

Local Post-hoc Interpretability for Black-box Classifiers

Résumé

Cette thèse porte sur le domaine de l'XAI (*eXplainable AI*), et plus particulièrement sur le paradigme de l'interprétabilité post-hoc locale, c'est-à-dire la génération d'explications pour une prédiction unique d'un classificateur entraîné. En particulier, nous étudions un contexte entièrement *agnostique*, c'est-à-dire que l'explication est générée sans utiliser aucune connaissance sur le classificateur (qui est alors traité comme une boîte noire), ni les données utilisées pour l'entraîner. Dans cette thèse, nous identifions plusieurs problèmes qui peuvent survenir dans ce contexte et qui peuvent être préjudiciables à l'interprétabilité. Nous nous proposons d'étudier chacune de ces problématiques et de proposer des critères et des approches nouvelles pour les détecter et les caractériser. Nous proposons de plus des méthodes de génération d'explications originales pour à ces problématiques. Les trois questions sur lesquelles nous nous concentrons sont : le risque de générer des explications qui sont hors-distribution ; le risque de générer des explications qui ne peuvent être associées à aucune instance de vérité de base ; enfin, le risque de générer des explications qui ne sont pas assez locales.

Afin de définir une explication locale, c'est-à-dire une explication permettant de comprendre une prédiction unique, nous proposons tout d'abord d'examiner le cadre des explications contrefactuelles. Nous proposons d'introduire une contrainte de parcimonie dans la fonction de coût qui en résulte. Nous proposons une procédure originale pour optimiser la fonction ainsi obtenue, appelée *Growing Spheres*, et nous montrons expérimentalement que cette approche permet d'obtenir des explications à la fois locales et faciles à comprendre, en accord avec les attentes de l'utilisateur. Nous étudions également la question des explications contrefactuelles hors-distribution, et montrons l'existence d'un risque auquel toutes les approches d'interprétabilité post-hoc sont vulnérables.

Puis, nous formulons un desideratum original pour les explications en termes de justification, qui peut être considéré comme un lien avec la connaissance de la vérité de terrain, afin qu'une explication ne soit pas basée sur des artefacts appris par le modèle de classification. Nous examinons ensuite le risque auquel sont confrontées les méthodes post-hoc, en particulier les explications contrefactuelles, en deux temps : nous proposons deux outils de diagnostic, d'abord pour mettre en évidence l'existence de ce risque, ensuite pour évaluer la vulnérabilité des approches contrefactuelles post-hoc. Nous montrons expérimentalement que ce risque existe et que les approches contrefactuelles y sont vulnérables. Nous étudions également le lien entre cette vulnérabilité et la localité des explications contrefactuelles.

Nous remettons ensuite en question cette notion de localité des explications en utilisant une seconde catégorie d'approches d'interprétabilité, reposant sur l'utilisation de modèles de substitution. Nous proposons de mesurer la fidélité

du modèle de substitution construit au classifieur "boîte noire" dans un voisinage de l'observation dont la prédiction est à expliquer. Le critère proposé, que nous appelons *Local Fidelity*, nous permet de définir la localité d'une explication comme étant la partie de la frontière de décision qui est approximée. En utilisant cette procédure d'évaluation, nous montrons que la façon dont les approches de substitution locales échantillonnent leurs instances d'entraînement a un impact important sur la localité de l'explication. C'est pourquoi nous proposons une nouvelle approche d'explication par substitution locale, qui utilise une procédure d'échantillonnage originale pour garantir des explications locales.

Les approches de substitution locales et les approches d'explication contrefactuelle reposant toutes deux sur la détection de la frontière de décision locale du classificateur, nous montrons qu'elles peuvent être mises en parallèle. A cette fin, nous introduisons la notion de généralisation d'une explication, étroitement liée à la fidélité locale d'un modèle de substitution linéaire, et l'utilisons pour suggérer que les approches de substitution locales sont une relaxation des approches d'explications contrefactuelles.

Abstract

This thesis focuses on the field of XAI (eXplainable AI), and more particularly local post-hoc interpretability paradigm, that is to say the generation of explanations for a single prediction of a trained classifier. In particular, we study a fully agnostic context, meaning that the explanation is generated without using any knowledge about the classifier (treated as a *black-box*) nor the data used to train it. In this thesis, we identify several issues that can arise in this context and that may be harmful for interpretability. We propose to study each of these issues and propose novel criteria and approaches to detect and characterize them, as well as original explanation methods to address them. The three issues we focus on are: the risk of generating explanations that are out-of distribution; the risk of generating explanations that cannot be associated to any ground-truth instance; finally, the risk of generating explanations that are not local enough.

To define a local explanation, i.e. an explanation allowing to understand a single prediction, we first propose to consider the framework of counterfactual explanations. We propose to introduce a sparsity constraint in the resulting cost function. We propose an original procedure to optimize it, called *Growing Spheres*, and show experimentally that this approach allows to obtain explanations that are both local and easy to understand, in accordance with a user's expectations. We also study the issue of out-of-distribution counterfactual explanations, and show the existence of a risk to which all explanation approaches are vulnerable.

Secondly, we formulate an original desideratum for explanations in terms of *justification*, which can be seen as a link with ground-truth knowledge, so that an explanation is not based on artifacts of the classifier. We then examine the risk faced by post-hoc methods, in particular counterfactual explanations, in two steps: we propose two diagnostic tools, first to highlight the existence of this risk, then to assess the vulnerability of post-hoc counterfactual approaches. We show experimentally that this risk exists and that counterfactual approaches are vulnerable to it. We also study the link between this vulnerability and the locality of the counterfactual explanations.

We then question this concept of explanation locality using a second category of interpretability approaches, called local surrogate models. We propose to measure the fidelity of the built surrogate model to the black-box classifier in a neighborhood of the observation whose prediction is to be explained. The resulting proposed criterion, that we call *Local Fidelity*, allows us to define the locality of an explanation as the part of the decision boundary that is being approximated. Using this evaluation procedure, we show that the way local surrogate approaches sample their training instances highly impacts the locality of the explanation. Therefore, we propose *Local Surrogate*, a new surrogate explanation approach using an original sampling procedure to ensure local explanations.

Since both local surrogate model approaches and counterfactual explanation approaches rely on the detection of the local decision boundary of the classifier, we show that they can be put in parallel. For this purpose, we introduce the notion of explanation generalization, closely related to the local fidelity of a linear surrogate model, and use it to suggest that local surrogate approaches are a relaxation of counterfactual explanation approaches.

Publications

The work conducted during the Ph.D program has led to the following publications:

Mentioned in this thesis

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Proc. of the Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'18)*, pages 100–111, 2018a

Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretablity. *ICML 2018 Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018b

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *Proc. of the 28th Int. Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 2801–2807, 2019c

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Unjustified classification regions and counterfactual explanations in machine learning. In *to appear in Proc. of the European Conf. on Machine Learning, ECML-PKDD'19*, 2019b

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Issues with post-hoc counterfactual explanations: a discussion. In *ICML 2019 Workshop on Human in the Loop Learning (HILL 2019)*, 2019a

Other works (joint collaborations)

The following works are not mentioned directly in this manuscrit, but have been conducted in parallel as collaborations on related topics:

Xavier Renard, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Detecting potential local adversarial examples for human-interpretable defense. In *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases, ECML-PKDD'18, Workshop on Adversarial Learning (Nemesis)*, 2018

Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Marcin Detyniecki, and Pascal Frossard. Imperceptible adversarial attacks on tabular data. In *NeurIPS 2019 Workshop on Robust AI for Financial Services*, 2019

Contents

1

Introduction

Over the recent years, Artificial Intelligence, and more specifically Machine Learning, has gained a phenomenal interest. Thanks to recent scientific and technological advances, collecting and processing data have become easier than ever. In addition, the performance of Machine Learning models has drastically improved over the last years, especially in tasks such as image or text classification with the development of deep neural networks. As a result, the applications of Machine Learning are now increasingly diverse and widespread. These applications concern multiple industries, among which for instance healthcare (e.g. tumor detection on radiography images), marketing (e.g. ad targetting), cybersecurity (e.g. spam detection in emails), transportation (e.g. traffic predictions), personal assistants in smartphones (e.g. voice and command recognition), etc.

Yet, such progress has been accompanied by an increase in the complexity of Machine Learning models. Today's best performing models (e.g. deep learning models or XGBoost) are highly opaque, to the point where they are often referred to as *black-boxes*. This term reflects that the model is viewed as a mysterious tool, whose behavior is not clear. This opacity can even be seen as dangerous, as attested by frequent stories of the unpredicted disastrous consequences of AI systems. An infamous example of such a disaster is the scandal of the COMPAS software in 2016[1]: an analysis by the non-profit organization ProPublica revealed that the COMPAS software, used by several jurisdictions in the US to predict the recidivism risk of convicts, was racially biased. The opacity of the algorithm considered in the software (a Machine Learning model) was making this bias difficult to assess, leading to judicial and ethical issues.

As an answer to this realization of the potential hazards created by AI systems, questions associated to these issues have gained interest in public discussions: the

[1] https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

general public is now more conscious about the potential negative impacts of AI on society. This is attested by the number of newspapers articles and books dealing with the dangers of AI: recent examples for instance include discussions on AI bias and discrimination[2], or on its negative impacts on society[3]. A major advance in this matter is the application of the *General Data Protection Regulation*[4] (GDPR), a regulation of the European Union. Beside other topics such as guaranteeing the respect of data privacy, the GDPR enforces the "right to an explanation" for citizens. This means that when being targetted with an algorithm (for instance when being recommended a product through ads, or when buying insurance online through an automated system), the organization (e.g. company or administration) responsible for the development of the algorithm is compelled to explain its decisions to the concerned citizen.

As will be discussed in more details in Chapter 2, in research, the fields of Interpretability, Fairness, Privacy and Robustness, sometimes all grouped under the terminology *Human AI*, *Trusted AI* or *Robust AI* are now more prominent than ever to meet these requirements. They are today among the hottest topics in AI research, as shown by the creation of numerous dedicated workshops and conferences[5]. Each of these sub-fields focuses on limiting the potential risks of undesirable behavior by AI systems, such as: unwanted bias, privacy attacks or lack of transparency. In particular, the field of *eXplainable AI (XAI)* directly addresses the problem of the opacity of AI systems, and therefore plays a central role with respect to these issues. XAI is further detailed below, as it is the focus of this thesis.

Explainable AI

The term XAI (standing for eXplainable AI) has been popularized by the DARPA (Defense Advanced Research Projects Agency) in a call for research proposals on AI explainability[6]. This term regroups multiple aspects that will be developed further. The idea they have in common is that they focus on providing explanations to a user for the decisions of AI systems. The considered context is thus one of a user aiming to perform a specific task with the help of an AI system: the system is making *decisions* that the user, depending on his/her final objective, can either follow, or use to achieve his/her task. For instance, a radiologist uses a visual recognition model to

[2] https://www.nytimes.com/2019/12/06/business/algorithm-bias-fix.html

[3] *Weapons of Math Destruction*, by Cathy O'Neil. Crown Books. 2016.

[4] https://gdpr-info.eu/

[5] To name a few, workshops include: WHI/HILL@ICML, XAI@IJCAI, FATML@KDD, several workshops at Neurips...; conferences include: AIES, FAT*, CHI, IEEE EuroS&P, ACM AsiaCCS...

[6] https://www.darpa.mil/program/explainable-artificial-intelligence

automatically detect the presence of a tumor in a radiography, in order to prescribe a treatment to the patient. In order to help the radiologist, the goal of XAI is to explain why the radiography is detected cancerous.

Multiple sub-fields of AI are concerned by XAI, such as autonomous agent behaviors (see e.g. Belahcène et al., 2015), recommender systems (see e.g. Heckel et al. (2017)), planning (see e.g. Zhang et al., 2015), or Machine Learning to name a few. The context of this thesis is Machine Learning Explainability. In particular, multiple works focus on the task of machine learning classification, which is considered in this thesis: the general goal is to generate explanations to give insights to the user about the reasons leading to predictions made by the model from the data. For instance in the case of the example of a radiologist given above, explanations would include identifying the tumor on the image using saliency maps, an interpretability method for image classification further detailed in Chapter 2.

Gaining insights from the data has been the focus of several related fields for a long time, such as Statistics for instance. In this field, the tasks of predicting and understanding (or describing) the effects of attributes over a target variable are traditionally separate (Shmueli, 2010). These two distinct tasks are thus associated to different objectives, the fulfillment of which is achieved using different models: for instance linear models for understanding feature effects, and Gaussian processes for prediction. Illustrating this distinction, a notion of *trade-off* between accuracy and understandability is often used (see e.g. Nisbet et al., 2009; Kuhn and Kjell, 2013): ensuring the best predictive performance is only possible with the use of complex models, therefore harder to understand. The same trade-off is also proposed in Fuzzy Machine Learning: this domain focuses on designing models to ensure that they are understandable to a human through the use of linguistic terms and fuzzy sets theory, although sometimes at the cost of predictive performance (see e.g. Yu and Xiao, 2009; Marsala, 2009).

Yet, in the light of the recent popularity of Machine Learning, sacrificing predictive performance is often not viewed as an acceptable option. Hence, the need to understand these predictions without degrading prediction accuracy has gained a renewed interest, leading to the development of the field called Machine Learning Interpretability.

Machine Learning Interpretability

One goal of Machine Learning interpretability considered in this thesis is to generate *explanations* to help a user understand a model's predictions (Doshi-Velez and Kim,

2017). However, because this concept is very general, no consensus over a formal definition or desideratum for explanations seems to exist. Numerous attempts to bring formal definitions and formalizations have been proposed (see e.g. Doshi-Velez and Kim, 2017; Lipton, 2017; Mueller et al., 2019; Weller, 2019). In parallel, numerous surveys (Guidotti et al., 2018; Biran and Cotton, 2019; Artelt and Hammer, 2019; Carvalho et al., 2019; Molnar, 2019) have proposed categorizations for interpretability approaches and presented an overview of multiple aspects related to interpretability to help organize these approaches. In particular, we discuss below two overlapping discussion axes which help framing the focus of this thesis.

Global and local explanations. One major distinction between interpretability approaches can be made between *global* and *local* approaches. Global approaches aim at explaining the behavior of a classifier in its entirety. On the other hand, local approaches, which are studied in this thesis, focus on explaining a single prediction made by the classifier. Numerous types of local explanations can be defined, depending on the considered context. For instance, counterfactual explanations (Wachter et al., 2018) aim at identifying the minimal perturbation to apply to an instance to alter its prediction. Another example is the case of local surrogate approaches (Ribeiro et al., 2016), which aim at approximating the local behavior of a trained classifier with a simple model. Both will be described in more details in Chapter 2.

Self-explaining models and post-hoc explanations. Another important distinction can be made between interpretability approaches that rely on building a *self-explaining model*, that is to say a classifier that generates its own explanations, such as a decision tree for instance; and approaches that focus on generating explanations for the predictions of a trained classifier. The latter, sometimes referred to as *post-hoc* approaches, thus make, by design, the generation of explanations independent from the prediction model and in particular its training step. This allows for more flexibility in terms of usage, as the classifier may thus be modified and retrained without modifying the explainer system for instance. Additionally, post-hoc approaches may use prior knowledge about the classifier or about existing data (training set or other). The absence of such knowledge is referred to as model and data agnosticity assumptions.

Research Questions

Local post-hoc interpretability constitutes the paradigm studied in this thesis because of its high relevance and numerous advantages. The considered context is thus the

generation of explanations for a single prediction of a trained classifier. In particular, a fully agnostic context is considered: no information is supposed to be available about the classifier, nor about any data. This paradigm has the upside of guaranteeing more flexibility for the user.

Despite these advantages, the constraining model- and data-agnosticity assumptions represent a source of potential issues: first, they raise the question of how relevant the generated explanations are with respect to the (inaccessible) training data. In particular, two aspects of the relation between explanations and training data are considered in this thesis: the risk of generating out-of-distribution explanations and the risk of generating unjustified explanations, that are more precisely defined in Chapters 3 and 4.

Furthermore, without any knowledge about the classifier nor any data, the sole task of defining the locality of an explanation is challenging. Although the independence between the generated explanation and any prior knowledge is intended, it raises questions regarding the relevance and usefulness of the explanations generated in the local post-hoc context.

In this thesis, these questions are formalized into desirable properties for explanations, and analyzed for two families of local post-hoc interpretability approaches, named counterfactual explainers and local surrogate models and described in Chapter 2. The study of these issues associated to these properties constitute the main proposition of this thesis.

Contributions

As mentioned above, the agnosticity assumptions considered represent a source of potential issues. The study of three of these issues is the focus of this thesis: the risk of out-of-distribution explanations, the risk of generating unjustified explanations, and the complexity of defining the locality of an explanation.

When no information is available about the classifier nor any data, generating a local explanation is challenging. Indeed, defining the mere concept of locality for an explanation is complex. An answer to this question is to focus on defining a local explanation as the minimal change to apply to the instance whose prediction is to be interpreted to alter its prediction. This is the goal of counterfactual explanation approaches, which aim at finding the minimal perturbation required to change the predicted class of the considered instance (see e.g. Wachter et al., 2018). We propose a new algorithm, called *Growing Spheres*, to generate counterfactual explanations in

a fully agnostic context, guaranteeing explanations that are both local and simple to understand. The latter objective is measured through the sparsity of the explanation. Although we illustrate experimentally that there is a trade-off between these two notions, we show that the proposed algorithm is successful in generating local explanations. However, the considered post-hoc paradigm raises issues that may hurt the interpretability of the generated explanations. Indeed, we show experimentally that there is a risk of generating counterfactual explanations that lie out of the distribution of the training data.

The second studied issue is the risk of having explanations that can not be related to any training instance. We propose a formal definition of this risk, that we call *risk of unjustification*, and a procedure, called *Local Risk Assessment*, to assess it. We show that besides depending on the considered classifier, the risk of unjustification is heavily linked to the notion of overfitting. An extension of this procedure, called *Vulnerability Evaluation*, is also proposed to highlight the vulnerability of existing counterfactual explanation approaches. Experiments across various classical benchmarks suggest that the considered approaches are vulnerable to this risk, although avoiding it may be possible at the cost of explanation locality.

This concept of explanation locality is then questioned. Using surrogate models, we propose another criterion, called *Local Fidelity*, to measure the locality of an explanation. Using this criterion, we experimentally show that the classical method LIME (Ribeiro et al., 2016) does not match this definition of locality. We identify the issue as being related to the sampling, one of the steps used by local surrogate approaches to ensure the locality of explanations. Therefore, we propose *Local Surrogate*, an approach using a new sampling procedure to solve this issue. Finally, we use the proposed *Local Fidelity* criterion to draw a link between local surrogates and counterfactual explanations.

Document Structure

The thesis is structured as follows. After a brief overview of the very vast domain of Machine Learning Interpretability, Chapter 2 presents some key elements of technical context that are relevant to this thesis and, in particular, the two main families of methods it focuses on: surrogate model approaches and counterfactual explanations.

Chapter 3 to 5 are then devoted to the study of the 3 issues presented in the previous section, as well as the algorithms and procedures proposed for each of them respectively: Chapter 3 tackles the risk of generating out-of-distribution explanations;

Chapter 4 addresses the risk of generating unjustified explanations; Chapter 5 is devoted to discussing the notion of explanation locality and how to define it using local surrogate models.

Finally, this manuscript ends by summarizing the contributions of this thesis and discussing the perspectives it opens.

2

Technical Context

As stated in Chapter 1, there is no consensus over the definition of Interpretability for Machine Learning in the literature. Even the mere term "interpretability" is not unanimous: other similar notions such as explainability, transparency or justification, to name a few, can often be found in the same contexts. Depending on the authors, these terms may or may not actually refer to the same notion and field. For instance, Lipton (2017) highlights that interpretability is the general field of understanding decisions of AI systems, while explainability refers to explanations generated for the predictions of a trained classifier. On the other hand, according to Carvalho et al. (2019) or Miller (2019), the latter definition refers to *post-hoc* interpretability, while interpretability and explainability essentially mean the same thing. The notions of transparency and justification are generally differentiated. The concept of prediction justification, introduced by Biran and Cotton (2019), refers to the process of giving insights about why a prediction is good, without actually explaining the decision process leading to the prediction. A common example of justification is a prediction confidence score (e.g. classification probability). Transparency can either refer to the more general field of understanding the behavior of machine learning models (Weller, 2019), or to some models whose inner workings can be inspected (Guidotti et al., 2018), for instance by looking at the algorithm (e.g. visualizing a decision tree).

These debates illustrate the plurality of aspects regrouped in the field, as well as its complexity. Under these notions, numerous approaches are being proposed, and older approaches regrouped under these new terminologies, as shown in recent surveys (Guidotti et al., 2018; Biran and Cotton, 2019; Artelt and Hammer, 2019; Carvalho et al., 2019; Molnar, 2019). A similar phenomenom can be observed beyond Machine Learning Interpretability, as the more global field of XAI faces the same issues. In our work, we use the general definitions proposed by Miller (2019) and

do not make any major difference between interpretability and explainability. On the other hand, the notions of transparency and justification are considered different, and left out of the scope of this thesis.

In view of the diversity of these existing notions, referencing all the related literature is impossible. However, we rely on existing global works such as surveys (Guidotti et al., 2018; Biran and Cotton, 2019; Artelt and Hammer, 2019; Carvalho et al., 2019; Molnar, 2019) and discussions (Lipton, 2017; Doshi-Velez and Kim, 2017; Miller, 2019) to present some key notions of interpretability, necessary to fully appreciate the content of this thesis. Besides, we focus on two families of interpretability approaches, especially interesting and studied in this thesis: surrogate model appraches and counterfactual explanation approaches.

This chapter is structured as follows: first, in Section 2.1, some key notions behind Machine Learning Interpretability are defined and discussed. Then, in Sections 2.2 and 2.3, the two families of interpretability approaches mentioned are presented in turn: surrogate model approaches first, and then counterfactual explanation approaches.

2.1 | Key Notions of Machine Learning Interpretability

In a general and imprecise definition, Machine Learning Interpretability can be seen as aiming to generate *explanations* to help understand a model's predictions (Doshi-Velez and Kim, 2017). The notion of explanation can for instance be defined referring to the field of cognitive sciences (see e.g. Hempel and Oppenheim, 1948; Lombrozo, 2006): it is generally defined as knowledge that help understand a concept, and appears to have a highly subjective component. As a consequence, to the best of our knowledge, no formal definition or desideratum for explanations seem to exist. This has lead to a lack of consensus over the notion of interpretability and its objectives in the machine learning literature, as attested by the numerous attempts to bring formal definitions and formalizations (see e.g. Doshi-Velez and Kim, 2017; Lipton, 2017; Mueller et al., 2019; Guidotti et al., 2018; Biran and Cotton, 2019; Artelt and Hammer, 2019; Weller, 2019; Carvalho et al., 2019; Molnar, 2019, among others).

In this section, we give an overview of the key concepts of Machine Learning Interpretability. First, in Section 2.1.1, we present in more details motivations for the need of understanding machine learning predictions. These motivations are important to understand how formal objectives of interpretability can be defined, which are

formulated in Section 2.1.2. Then, in Section 2.1.3, some major discussion axes from the literature are defined to help understanding the field of Machine Learning Interpretability. Finally, Section 2.1.4 is devoted to the task of evaluating interpretability approaches so as to dispose of tools to assess and compare explanations, a challenging issue of the field due to the subjectivity and lack of formal requirements.

2.1.1 | The Need for Interpretability

The motivations behind interpretability help understanding the absence of consensus in interpretability definitions. In this section, we motivate interpretability by presenting it as filling an incompleteness in the machine learning paradigm, in the light of the work of Doshi-Velez and Kim (2017).

As stated in Chapter 1, it is generally accepted that the need for Machine Learning Interpretability occurs in a context where a machine learning model is applied to help a user perform a specific decision-making task. Most of the time, the predictions returned by the model only are not sufficient to achieve this goal: there is a structural *incompleteness* in the problem formalization (Doshi-Velez and Kim, 2017; Miller, 2019). We propose to identify two reasons to explain this incompleteness. It can happen because: (i) the user does not trust the model, or (ii) the decision returned by the model does not fully match the final objective of the user. Each of these arguments is discussed below in turn.

Lack of trust in AI. When using a machine learning model, the user may be unsure whether the model is behaving as expected or not. This leads to a lack of trust in the model. It is especially crucial when some notion of safety is involved, or when the stakes impacted by the decision the user has to make are high. For instance, healthcare applications are especially concerned: incorrectly detecting the presence of a tumor in a radiography image may lead to disastrous consequences. This lack of trust is thus obviously partially caused by the fear of dealing with an incorrect prediction (Dietvorst et al., 2015). A common use case of Machine Learning Interpretability is therefore to focus on understanding prediction errors. Interpretability is then required in order to either make better use of the predictions or improve the model performance for instance (Breiman, 2002; Kabra et al., 2015; Lucic et al., 2019).

Other issues than misclassification may arise when training a model, and can also hurt the human-AI trust. A first example is the presence of unwanted biases, such as the case of the racial bias higlighted in the context of the COMPAS algorithm, used in the US judicial system to predict the risk of criminal recidive: studies (Larson et al.,

2016) have shown that the COMPAS algorithm was mainly predicting the recidive risk based on the skin color, raising legal and ethical questions. Additionally, the lack of robustness of the model, such as its vulnerability to malicious attacks (Biggio et al., 2013), may also hurt the human-AI trust. In these situations, interpretability can be used as a confirmation or refutation for the model behavior. For instance, the existence of biases in the data (Zeng et al., 2016) or learned by the model (Tan, 2018) can be proven using interpretable models and techniques. Another example is offered by Tao et al. (2018), who use Machine Learning Interpretability to detect malicious attacks on face recognition models. Besides an end in itself, interpretability methods can thus also be used to investigate various issues related to machine learning models, such as the case of machine learning fairness in the aforementioned example. This makes the role of Machine Learning Interpretability central, and its study crucial.

In such problematic situations, the estimated predictive performance alone (e.g. classification accuracy) is not enough to ensure trust in the model. This is even truer when the model is performing poorly in terms of predictive performance, in which case the urge for explanations is even stronger (Papenmeier et al., 2019). Interpretability is then required to help restore this trust so that the model may be useable (Miller, 2019).

A mismatch in objectives. There is often a gap between the decision returned by the model and the final task of the user (Doshi-Velez and Kim, 2017). In particular, the final objective of the user may be much more complex than the one encoded in the learning algorithm. In this context, explanations may be a complement that helps the user meeting his/her final objective. For instance, a fraud analyst may be using a model predicting which customer is the most likely to commit insurance fraud. However, the final objective of the analyst is not only to identify customers with a fraudulent behavior, but also to understand how they committed fraud, so as to take appropriate actions. In order to fulfill this task, he/she thus needs to understand *why* the customer is predicted to be conducting fraud. This goal can be met using Machine Learning Interpretability (see e.g. Collaris et al., 2018).

Be it to restore the trust of the user in the model or to compensate for the mismatch in objectives, understanding machine learning predictions is required to help the user perform his/her final task. Interpretability is thus defined with respect to the considered context: depending on the motivations of the user, interpretability can be brought in multiple ways. This leads to a large variety of formal definitions and goals of interpretability, without a consensus having been reached. These objectives

of interpretability approaches are the focus of the next section.

2.1.2 | The Two Sub-tasks of Interpretability Approaches

As explained in the previous Section 2.1.1, the objective of interpretability is dependent on the considered user and situation. For instance, the final objective of the user is important: saliency maps (a common form of explanation in the context of image classification, see e.g. Selvaraju et al., 2016) may be particularly appropriate in the context of specific tasks such as model debugging, where the final objective of the user is to build the best performing tumor detection model in radiographies for instance. However, because of their form, these approaches are less useful for tasks involving the necessity to provide understandable insights about how to change the prediction of the considered instance, such as face recognition models.

The domain knowledge of the user may also impact the way the explanations are built: it is expected that a domain expert (e.g. a physician in the context of healthcare) and a neophyte do not require the same information in order to understand and trust a prediction (Doshi-Velez and Kim, 2017; Weller, 2019).

Despite relying on a subjective and context-dependent notion, some common goals can be identified for most interpretability approaches. In order to categorize and better understand these approaches, we propose to define two distinct sub-tasks that are addressed when building an explanation. These are the following:

1. First, the information relevant to the predictions to explain is to be extracted from the model.

2. This raw information is then "translated" to be given to the user. This constitutes the final *explanation*.

These sub-tasks are inspired from the formalization of explanations for black-box models proposed by Guidotti et al. (2018). In this work, the generation of an explanation is defined as the construction of two functions: one to mimic the behavior of the black-box, and the second to use the information provided to generate the final explanations. However, in this work, the objectives associated to these two steps are not explicitly stated, and not studied separately. We therefore propose to explicitly define each sub-task, as they provide a reading grid ensuring a better understanding of interpretability approaches.

These two sub-tasks are respectively described in Sections 2.1.2.1 and 2.1.2.2. Then, in Section 2.1.2.3, we describe, using these sub-tasks, some existing interpretability approaches.

2.1.2.1 | Extracting Information from the Model

The goal of this first sub-task is to identify the rationale of the model behind the predictions. That is to say, the mechanisms activated by the model when making predictions. In the context of a decision tree for instance, this rationale would be a combination of the observations whose predictions are to be interpreted, and the relevant splits of the tree associated to these predictions. Similarly, a visualization of the neurons activated for a given prediction of a neural network would provide the user with this rationale.

However, depending on the context, identifying and extracting this information may not always be possible. This may happen for instance because the said rationale is too complex to be extracted, or because full knowledge of the classifier is not available (or no knowledge at all is available). For instance, in a situation where the explanations are generated without any information about the model being available (e.g. for confidentiality reasons), i.e. treating it as a *black-box*, retrieving the true rationale of the model is not possible. In such cases, the goal of Machine Learning Interpretability is then to identify *a* (instead of *the*) rationale behind the prediction. In other words, interpretability approaches aim at extracting a piece of information that helps the user understand the prediction: this piece of information may not be the actual complete reason behind the prediction, as stated by Miller (2019), Mittelstadt et al. (2019) or Rudin (2019).

In order to build these rationales, interpretability questions can be formulated to help specify which information is desired (Doshi-Velez et al., 2018). For instance, a question may be to identify the most impactful factor in a decision. This is the objective of feature importance explanations (e.g. Ribeiro et al., 2016; Lundberg and Lee, 2017) for instance. Another example is the case of counterfactual explanations (e.g. Wachter et al., 2018 and developed in Section 2.3, page 28), which aim at answering the particular question: *What changes need to be applied in order to alter the prediction?* These questions guide the generation of explanations by reducing the task of explaining a prediction to a single practical aspect, that can be translated into an objective function.

2.1.2.2 | Generating the Final Explanation.

The idea behind the second step of generating the explanation itself is to adapt the extracted information to the needs of the user. Depending on his/her knowledge or on the considered context, specific representations of this piece of information may be more appropriate than others. For instance in the context of a model predicting

whether a loan applicant is likely to default, counterfactual explanations are particularly relevant to provide insights to non-expert customers, as shown in Wachter et al. (2018).

A variety of forms of explanations. Besides impacting what information is required, the considered context thus also impacts the form the final explanations take. There is a big variety of existing forms of explanations. A non-exhaustive list of the most common forms of explanations is presented below:

- Feature importance vectors, such as the emblematic interpretability method LIME (Ribeiro et al., 2016), which will be particularly studied in Chapter 5. These explanations give values of how impactful each feature is for the considered predictions. Several definitions of feature importance can be considered, based on the needs of the user. For instance, the importance of a feature can be calculated using the decrease in accuracy triggered by the permutation of the feature's values (Breiman, 2001; Fisher et al., 2019). LIME uses the coefficients of a linear model as feature importances. Another example is to use the gradient of the model as a local feature importance vector to explain its predictions (Baehrens et al., 2010; Selvaraju et al., 2016). Additionally, some approaches use insights from game theory to compute feature importance coefficients, such as the famous Shapley values (Strumbelj et al., 2009; Lundberg and Lee, 2017). The ranking of these feature importance values is especially useful, as it gives a sense of which features have the most influence on the predictions. Similarly, a graphical representation of the feature importances may be provided (Ribeiro et al., 2016).

- Decision rules: such as the approaches proposed by Turner (2015); Zeng et al. (2016); Ribeiro et al. (2018); Guidotti et al. (2019a). These explanations give sufficient conditions that, when satisfied, lead to the considered predictions. These rules can be computed in various ways. For instance, LORE (Guidotti et al., 2019a) computes a rule-based explanation for a single prediction by extracting the path leading to the studied instance in a decision tree trained in its vicinity. Another example is MES, *Model Extraction System* (Turner, 2015), which selects the best rule-based explanation candidate with respect to their mutual information score with the model to interpret. Rule-based explanations have the upside of using a limited set of conditions to explain the predictions. This leads to them being more transparent than feature importance explanations for instance.

- Visualizations, such the approaches proposed in Friedman (2001); Krause et al. (2018); Ming et al. (2018). A well-known example of such an explanation is partial dependance plots (Friedman, 2001; Goldstein et al., 2015), which show the marginal effect of a feature over the model outcome. Also falling into this category are most approaches generating explanations for individual predictions in the context of image classification, such as Selvaraju et al. (2016).

- Particular instances used as comparison. This is the case for instance of prototype-based approaches (Kim et al., 2014), and counterfactual explanations (Martens and Provost, 2014; Lash et al., 2017a; Wachter et al., 2018), which are further discussed in Section 2.3, page 28 and are the particular focus of Chapters 3, page 41 and 4, page 69. Another type of approaches falling into this category are the ones that try to detect the training instances that are the most influential for a given prediction, generally by retraining the classifier (Kabra et al., 2015; Sharchilev et al., 2018).

- A classifier. In some cases, the original model performing the predictions can also generate its own explanations. This can be because the model itself is considered to be "simple" enough to be understood, as it is the case for low-complexity decision trees for instance, or sparse regression wich can be used to provide interpretable models in high-dimensional data (Alaya et al., 2019). Some classifiers are also designed to generate explanations when making a prediction, such as self-explaining neural networks (Alvarez Melis and Jaakkola, 2018). Another possibility is to use a *surrogate model*, that is to say an interpretable copy of a complex model as the explanation (Craven and Shavlik, 1996; Hara and Hayashi, 2016).

Most of these explanations are characterized using specific criteria. For instance rule-based explanations can be described by the number and size of the decision rules they provide. Similarly as the form of the explanation, these characteristics also depend on the considered context. Some of these criteria are further discussed in Section 2.1.4, page 19.

Knowing which user the explanation is destined to, as well as what the final task is, are thus obviously really important to define the right explanation.

2.1.2.3 | A First Reading Grid for Interpretability Approaches

Defining the two sub-tasks of explanation generation presented in the previous section allows to define a first reading grid, leading to a better readability of interpretability approaches. The two categories of interpretability approaches that are studied in this thesis, namely surrogate model and counterfactual explanation approaches, fit into this framework. Indeed, surrogate model approaches (which are the focus of Section 2.2, page 21) such as LIME (Ribeiro et al., 2016) generate explanations by training a surrogate model in order to extract information from the classifier (first sub-task). The final explanation is then generated using the linear coefficients of this surrogate model, presenting them in the form of a feature importance vector as well as using visualizations (second sub-task). Counterfactual explanation approaches (which are the focus of Section 2.3, page 28) such as the one by Wachter et al. (2018), first extract information from the model by identifying the closest touchpoint of its decision boundary to the observation whose prediction is to be interpreted (first sub-task). The final explanation is then provided to the user in the form of a list of actions needed to apply to change the studied prediction (second sub-task).

The distinction of these two sub-tasks also enables more relevant evaluation and comparison between approaches, as discussed in Section 2.1.4, page 19. In the next section, we propose more axes of discussion inspired from existing works (Lipton, 2017; Doshi-Velez and Kim, 2017; Guidotti et al., 2019a; Carvalho et al., 2019) that help categorizing interpretability approaches. Compared to the reading grid proposed in this section, which focused on technical aspects of interpretability, the following discussion axes are positioned at a higher level, as they discuss the objectives of interpretability approaches.

2.1.3 | Axes of Discussion

Several general categories can be defined for interpretability approaches, as proposed by existing surveys and discussions on the topic of interpretability (Lipton, 2017; Doshi-Velez and Kim, 2017; Guidotti et al., 2019a; Carvalho et al., 2019). These categories, some of them already mentioned in Chapter 1, page 4, generally overlap and can be positioned along axes of discussion, three of which are presented below. These discussion axes are studied in existing works (Lipton, 2017; Doshi-Velez and Kim, 2017; Guidotti et al., 2019a; Carvalho et al., 2019), and are especially relevant in the context of this thesis.

The first one makes a distinction between approaches that propose to use classi-

fiers that generate their own explanations, and the ones that generate explanations for the predictions of a trained classifier. The second one aims at differentiating approaches depending on the assumptions they make on the classifier and existing data. Finally, the third proposed discussion axis separates the methods that aim at generating explanations for a single prediction and the ones that provide insights about the whole behavior of a model.

Self-explaining models and post-hoc explanations. A first natural distinction between interpretability approaches comes from the context in which the explanations are to be generated. Considering a specific classification task, a distinction is generally made between using a classifier that generates its own explanations (*self-explaining* model), and approaches that require the generation of an explanation to interpret the prediction of a trained classifier (*post-hoc* explanations).

In the first situation, explanations for the predictions can be directly extracted from the model. This is for instance the case of linear models, which rely on coefficients describing a simple, linear, relation between the target variable and a given attribute; similarly, the decision path of a decision tree can be visualized to understand the reasons leading to a specific prediction. However, these approaches are often limited by their predictive performance.

To circumvent this issue, post-hoc interpretability approaches propose to generate explanations for the output of a trained classifier in a step distinct from the prediction step, hence the name post-hoc. This can be conducted for instance by approximating the decision boundary of a complex classifier with a simple model to extract explanations, such as the approach proposed by Ribeiro et al. (2016). Such explanations have the upside of being flexible, as the trained classifier may be modified and retrained without changing the explainer system. Two categories of post-hoc approaches are the focus of this thesis: surrogate models and counterfactual explanations.

Agnosticity assumptions. A natural follow-up question in the post-hoc context is about *agnosticity* assumptions, that is to say defining what knowledge can be used to build these explanations. Indeed, some approaches may for instance rely on knowing which family of classifier was used (such as the approach proposed by Hara and Hayashi (2016), which focuses on tree-ensemble classifiers). Similarly, in the field of image classification, multiple interpretability approaches suppose that the classifier is a deep neural network (see e.g. Selvaraju et al., 2016) Others, however, suppose that no information is available: neither about the classifier, nor about any existing

data (training set or other instances, such as a dataset used to evaluate the model's predictive performance). This is the case for LORE (Guidotti et al., 2019a) for instance.

These assumptions impact the way these methods can be used. *Model-agnostic* approaches have the upside of being faster and more flexible to use since they do not depend on the classifier that has been trained. *Data-agnostic* approaches do not require any existing instances to be run. This can be interesting when privacy constraints make accessing data impossible for instance.

In this thesis we focus on post-hoc approaches, and analyze the impact of these model- and data- agnosticity assumptions over the quality of the generated explanations.

Local or global explanations. A third distinction can be made based on which predictions the explanations are generated for. One the one hand, *global* interpretability approaches aim at generating explanations to help the user gain knowledge about the whole model. In the aforementioned framework of Section 2.1.2, the information that is to be extracted thus concerns the general behavior of the model. For instance, the coefficients of a logistic regression give insights about the impact of each feature globally. Another example is provided by the method proposed by Kim et al. (2014), who use case-based reasoning to help the user understand the information learned by a classifier using prototypes.

On the contrary, *local* approaches aim at generating explanations for a specific prediction (Guidotti et al., 2018): they focus on a specific part of the rationale of the classifier. A famous example of local post-hoc approach is LIME (Ribeiro et al., 2016), which uses a local surrogate model to approximate the decision boundary of a classifier. In the context of LIME, the explanation is local because the surrogate model focuses on approximating a small portion of the decision boundary of the classifier. However, this definition of locality can be questioned, and is one of the focuses of this thesis. Therefore, local explanation approaches are further discussed in Sections 2.3, page 28 and 5.1.2, page 103.

2.1.4 | Evaluating Interpretability

One of the core issues of the field of Machine Learning Interpretability is the evaluation of interpretability approaches: the somehow vague and subjective objectives of interpretability, as well as the lack of consensus of the objectives of interpretability make the evaluation of interpretability approaches problematic, as illustrated by the discussions proposed by Doshi-Velez and Kim (2017); Lipton (2017). In this sec-

tion, we present an overview of the solutions that have been proposed in previous works, distinguishing between evaluation methods involving users in Section 2.1.4.1, quantitative criteria in Section 2.1.4.2 and methods proposing to audit interpretability approaches in Section 2.1.4.3. In each subsection, the difficulties raised by the considered evaluation method category are presented.

2.1.4.1 | User Experiments

As presented in Section 2.1, interpretability approaches aim to help a user understand predictions in order to help him/her perform better a given task. A first natural evaluation for interpretability therefore focuses on the measuring the efficiency of the user in performing the said task with the help of explanations. This is sometimes called *task-oriented* evaluation, or *application-grounded evaluation* (Doshi-Velez and Kim, 2017). However, conducting such experiments is complex and expensive since it supposes that the final task can be reproduced multiple times to ensure fair measurement.

Most interpretability approaches hence try to perform *human-grounded* evaluation: it consists in evaluating how helpful the explanations are to understand predictions of the model. For instance, Ribeiro et al. (2016) ask mechanical workers to evaluate the explanations generated for the predictions of a given model. However, this is of course a very subjective goal. As mentioned in Section 2.1.2, this objective may depend on parameters such as the user knowledge and the considered context (e.g. the considered classifier), and is therefore heavily prone to population bias.

2.1.4.2 | Quantitative Criteria

On the other hand, defining quantitative criteria to assess the quality of an explanation is a complex task due to the absence of consensus of the expected result and the subjective component. In the light of the two-task framework presented in Section 2.1.2, a good explanation has two objectives: (1) Capture the correct information and (2) Translate it to the user. Each of these objectives can be measured separately. Ensuring that the correct information is captured (objective 1) is hard to compare between approaches that do not represent the knowledge in the same form. For instance, comparing the information learned by a surrogate model to the one captured by a set of prototypes is complex.

As for the second task, i.e. translating this information, some criteria are commonly associated to the user understanding of the explanation. Generally, it is thus assumed that the explanation *complexity* is an important criterion that can often be

considered. This can be done through metrics such as sparsity (for feature importance explanations for instance), or the number of decision rules. In the case of counterfactual explanations, Guidotti et al. (2019a) aim at generating a counterfactual explanation involving as few rules as possible. The sparsity of the explanations generally measures how many attributes are involved in the explanation (e.g. number of non-null feature importance coefficients in Ribeiro et al., 2016), often measured with the l_0-norm.

These evaluation methods are also heavily dependent on the considered context and interpretability approach. Therefore, there is no consensus over which criteria to use.

2.1.4.3 | Diagnostic of Interpretability Approaches

Finally, some works focus on analyzing the issues raised by some interpretability approaches. These works generally conduct analyzes to highlight issues and limits of interpretability approaches. Although different from the two aforementioned categories of evaluation methods, these methods may overlap: the diagnostics may rely on analyzing user experiments and may lead to the proposition of numerical criteria.

For instance, Adebayo et al. (2018) study saliency maps, a common way of analyzing the predictions of a neural network in the context of image classification. They show that most of the time, the generated explanation is not model-dependent: this thus raises the question of the efficiency of the method in interpreting the model's predictions. Another example can be found in Alvarez Melis and Jaakkola (2018), who propose to measure the *robustness* of post-hoc interpretability approaches. They define robustness as the variation of the explanation with respect to the instance whose prediction is to be explained. The robustness of interpretability approaches is said to be linked to the trust the human gives to the model.

Most of the work conducted in this thesis falls into this category of evaluation method. In particular, this thesis focuses on the study of three issues that post-hoc interpretability methods face. The study of each of these issues leads to the proposition of numerical criteria to assess its importance.

2.2 | Surrogate Model Approaches

Using the key notions of Machine Learning Interpretability presented in the previous section, this section and the following one focus on two specific families of interpretability approaches: surrogate models and counterfactual explanations. These

categories of interpretability approaches both fall into the post-hoc paradigm (presented in Section 2.1.3) and are especially relevant in the context of this thesis. In particular in this section, surrogate model approaches are discussed.

Although mentioned in the whole thesis, surrogate model approaches are especially the focus of Chapter 5, page 101. In particular, the way surrogate models are designed to generate *local* explanations will be analyzed. Hence, we present in this section these approaches in light of this issue.

In Section 2.2.1, the general objectives and operating process of surrogate models is presented. Then, in Section 2.2.2, the most emblematic approaches are studied, with a particular focus on local interpretability approaches.

2.2.1 | General Objectives and Principle

This section first presents the general principle behind surrogate model approaches for interpretability. Then, the major steps considered to generate explanations with surrogate models are presented in Section 2.2.1.2.

2.2.1.1 | Principle

Surrogate model approaches are a type of post-hoc interpretability approach, hence designed to generate explanations for the predictions of a trained classifier. For this purpose, surrogate model approaches aim at fitting a *surrogate* model to imitate the behavior of the classifier while facilitating the extraction of explanations. Depending on the family of models chosen for the surrogate, these explanations may either be the model itself (for instance visualizing a decision tree), or other information extracted from the surrogate model, such as feature importance vectors, decision rules or gradient vectors to name a few. Often, the surrogate model is thus a *simpler* version of the original classifier.

We make a distinction between *global* and *local* surrogates. Global surrogates aim at replicating the behavior of the classifier in its entirety. On the other hand, local surrogate models are trained to focus on a specific part of the rationale of the trained classifier. The distinction between local and global surrogates may differ from the one between local and global interpretability. In particular, global surrogates can be used to generate local explanations (e.g. in the case of a global decision tree used to generate local explanations) and vice versa. More examples of such situations are given in Section 2.2.2, page 24.

The post-hoc nature of these interpretability methods raise natural questions about agnosticity (see Section 2.1.3). Indeed, prior knowledge of the original classifier or of

some existing data heavily impacts how the surrogate models are trained. For instance, Hara and Hayashi (2016) generate a surrogate model specifically designed for tree ensemble classifiers. On the other hand, Guidotti et al. (2019a) train a surrogate model in a fully agnostic context, relying on the generation of instances using a genetic algorithm.

Despite these distinctions, a common operating process can be identified for surrogate model approaches. This framework is discussed in the next section.

2.2.1.2 | Operating Process

Given a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ trained on a dataset $X \subseteq \mathcal{X}$, the goal is to generate explanations for a prediction $f(x)$, with $x \in \mathcal{X}$. For surrogate model approaches, the construction of this explanation requires the training of a surrogate model $h : \mathcal{X} \rightarrow \mathcal{Y}$ to mimic the behavior of f. In order to generate these explanations, we propose to identify a three-step architecture common to all surrogate model approaches. These three steps are presented in turn below:

1. **Sampling Step**

   One of the fundamental questions when it comes to surrogate model approaches concerns the data to be used to train the model h. Agnosticity hypotheses mentioned in Section 2.1.3 are central, as they define whether the data used to train the original classifier can be reused. However, even when ground-truth instances (be it training data or other labelled data) are available, new instances are often generated to make sure that sufficient information about the classifier is available (see for instance the approach proposed by Craven and Shavlik, 1996, described in Section 2.2.2.1, page 24). The question of how these data should be sampled in $\mathcal{X}$ is crucial to these procedures, as it heavily impacts the resulting surrogate model. This is especially important for *local* approaches, as studied in Chapter 5, page 101.

   Instances of $\mathcal{X}$ are thus either generated, or selected from X, in order to build X_h. These instances are then labelled using f. The obtained prediction vector $f(X_h)$ is used as a label for training the surrogate model. Using this predicted output allows to get insights about the decision boundary of f. In the context of classification, the output of the classifier f that is used can either be the predicted class or the continuous classification confidence scores (e.g. classification probabilities). An example of the latter case is the approach proposed by (Baehrens et al., 2010).

2. **Training Step**

   Once the training data has been defined, the surrogate model $h : \mathcal{X} \rightarrow \mathcal{Y}$ is trained on $(X_h, f(X_h))$. As mentioned earlier, the choice of the model h to use depends on several elements, such as: the desired form of the final explanations (e.g. decision rules vs. linear coefficients), the agnosticity assumptions considered (e.g. knowledge about f), and whether the surrogate should be local or global.

   For instance, the choice of the considered surrogate model may be particularly designed to approximate a specific type of classifier. This is for instance the case for the approach proposed by Hara and Hayashi (2016), specially designed for ensemble tree methods. Depending on the nature of the problem and the desired output, the cost function used to train the surrogate model may also take various forms. For instance, weights may be assigned to the training instances X_h in order to focus on a specific portion of the dataset, e.g. to make the surrogate model local, such as in the case of LIME (Ribeiro et al., 2016).

3. **Explanation extraction step.**

   The final explanations given to the user are extracted from h. Again, the form of the explanations depends on the nature of the surrogate model, as well as on the information desired by the user.

2.2.2 | Approaches

In this section, several surrogate model approaches are presented in light of the proposed three-step framework. In particular, we make a distinction between global surrogates, presented in Section 2.2.2.1, and local surrogates, presented in Section 2.2.2.2.

2.2.2.1 | Global Surrogates

This section presents three global surrogate approaches: the one proposed by Baehrens et al. (2010), that we call Parzen, TREPAN (Craven and Shavlik, 1996) and the one proposed by Hara and Hayashi (2016), that we call DT, for Decision Tree. Since these three approaches are representative examples of the framework presented in Section 2.2.1.2, we present them in light with the three identified steps to build a surrogate.

Sampling step. As mentioned in the previous section, global surrogates aim at giving insights about the inner workings of the whole classifier f. Therefore, the used

surrogate training data X_h is generally sampled *globally*, meaning to cover the whole input space of the classifier f. In some situations, the dataset X used to train f may be available. This is the case for DT and Parzen for instance. While the former sets $X_h = X$, the latter requires new instances to be sampled following the same distribution as X. However, this assumption implies that some areas of the feature space may not be well covered, leading to less accurate information learned by the surrogate.

To avoid such issues, TREPAN proposes to generate new instances in areas insufficiently covered by the training instances from X. In particular, since the associated surrogate is a decision tree, the proposed approach detects at each potential split the number of instances supporting the split. When the number of associated instances is below a certain value S_{min} (specified by the user), additional data is generated following the marginal distribution of each attribute of X, modeled with a kernel density estimation method.

Training step. A surrogate model h is then trained to approximate f over the whole input space and generate explanations. TREPAN relies on a form of decision tree. The specificity of the proposed approach is to consider *m-of-n* splits. The condition asociated to these particular splits is considered to be satisfied when at least m of the n specified conditions are satisfied. Contrary to classical decision trees such as the one learned using the CART algorithm (Breiman et al., 1984), these splits involve multiple attributes and allow TREPAN to learn more complex concepts. The final output is thus a decision tree that gives insights about the global behavior of f, despite no knowledge about f being available. Similarly, Parzen approximates the continuous probability output of an unknown classifier f using Parzen windows.

On the other hand, DT supposes that some knowledge about f is available. In particular, it focuses on generating explanations specifically for ensemble tree classifiers. By relying on the particular structure of this family of classifiers, DT aggregates the regions learned by the various decision trees involved in f and approximates them with a simple decision tree. In order to find the optimal values for the parameters of h to best replicate the behavior of f, an EM algorithm is used to minimize the Kullback-Leibler divergence between $f(X)$ and $h(X)$. In the end, the low number of regions defines a global surrogate explanation in the form of a decision tree.

Explanation extraction step. Both TREPAN and DT return a decision tree summarizing the global behavior of the model. However, another notable possibility is to use global surrogates to generate local explanations. This is the case for the Parzen approach. Using the gradient of the surrogate Parzen model, chosen to be differen-

tiable, explanations $\nabla h(x)$ can be given for an individual prediction $f(x)$. This type of approach thus allows the generation of local explanations using only one surrogate model, meaning requiring less training steps than a purely local approach. However, as discussed in Chapter 5, this is often at the cost of local explanation quality.

2.2.2.2 | Local Surrogates

As opposed to global surrogates, local surrogates focus on a specific part of the rationale of the classifier f to generate explanations for a single prediction. Each instance $x \in \mathcal{X}$ thus requires the training of a dedicated surrogate model $h_x : \mathcal{X} \to \mathcal{Y}$ on a dataset X_{h_x}. Considering the general framework presented in Section 2.2.1.2, the definition of the sampling step as well as the training step may heavily impact the obtained results. A discussion about how to define the desired level of locality for an explanation, and how to incorporate it in the proposed framework is further conducted in Section 5.1.2, page 103. This section presents two approaches that are studied in this thesis: LIME (Ribeiro et al., 2016) and LORE (Guidotti et al., 2019a). Both approaches are model-agnostic and make few assumptions about the data.

LIME (Ribeiro et al., 2016) LIME (*Local Interpretable Model-agnostic Explanations*) is the most emblematic local surrogate approach today. The idea behind LIME is to locally approximate the probability function of a classifier with a surrogate model h defined as a linear regression. Although extensions to image and text data are proposed by Ribeiro et al. (2016), we only focus on the version for tabular data.

- **Sampling step.** In LIME, the generation of the instances of X_{h_x} does not depend on x. However, a weight, the value of which depends on x, is associated to each instance.

  First, the instances of X_{h_x} are sampled following independent normal distributions for all features describing $\mathcal{X}$. The parameters of these distributions can be calculated using a possibly available dataset that may be given as input of the algorithm (for instance the training set X, but not necessarily). The idea behind this heuristic is to reproduce the range of the ground-truth data. However, using a normal distribution (instead of reproducing the distribution of the input dataset for instance) makes the generated explanation not directly dependent on the training data.This is somehow desirable since the goal is to interpret the decisions of the classifier f, not reproduce them using the same data.

The new dataset X_{h_x} is then labelled using f to return a continuous classification score (e.g. probability or confidence). Weights w_i are then calculated for each instance $x_i \in X_{h_x}$ based on their distance to x and using a RBF kernel: $w_i = e^{-||x-x_i||_2^2/\sigma^2}$. The value of the kernel width parameter σ is set by the user or by a heuristic proposed by the authors as: $\sigma = 0.75\sqrt{dim(\mathcal{X})}$. A study on the influence of this parameter is proposed in Section 5, page 101.

- **Training step.** These instances are used to train the surrogate model h_x, optimizing the following loss function:

$$\mathcal{L} = \sum_{x_i \in X_{h_x}} w_i (f(x_i) - h_x(x_i))^2 + \Omega(h_x)$$

with Ω a measure of *complexity* of a model, such as the number of non-zero coefficients in the case of the linear regression. This complexity measure allows the user to control the complexity of the explanation (number of variables involved in the explanation). The resulting loss function is optimized using a Lasso regression (Tibshirani, 1996).

- **Explanation extraction step.** To generate the final explanation, the linear regression coefficients are extracted. They are then given to the user through a visual interface provided with the LIME package by the authors[1].

As one of the most emblematic post-hoc interpretability approaches, LIME has been the focus of numerous extensions. Notable approaches include for instance Kernel SHAP (Lundberg and Lee, 2017), *SHapley Additive exPlanations*, which draws a link between LIME and game theory-based interpretability approaches (such as Strumbelj et al., 2009). In particular, Kernel SHAP proposes to specify weight values (in place of the RBF kernel) and model complexity to give theoretical guarantees to the generated explanations. These guarantees are associated to desirable properties for the explainer system, e.g. *local accuracy*, which states that the surrogate model and the black-box classifier should agree on the prediction of x.

LORE (Guidotti et al., 2019a) LORE (*LOcal Rule-based Explanations*) is a more recent model-agnostic local surrogate approach. The idea behind LORE is to use a decision tree to generate both a decision rule-based explanation and a counterfactual explanation. Thus, the approach is also discussed in Section 2.3, page 28, which focuses on counterfactual explanations. LORE is also the focus of Chapter 4, which is centered on

[1] https://github.com/marcotcr/lime

post-hoc counterfactual explanations. Initially designed for tabular data, extensions of LORE to non-tabular data have since been proposed (Guidotti et al., 2019b).

- **Sampling step.** First, instances X_{h_x} are generated using a genetic algorithm. The idea behind this proposition is to generate a relevant neighborhood well capturing the subtleties of the local decision boundary of f. In practice, instances $x_i \in X_{h_x}$ are generated by maximizing 2 fitness functions: one for the instances belonging to the same class as x, and one for the instances from the other class. These fitness functions, respectively denoted $fitness^x_=$ and $fitness^x_{\neq}$ are defined by:

$$fitness^x_=(x_i) = \mathbb{1}_{f(x)=f(x_i)} + (1 - d(x, x_i)) - \mathbb{1}_{x=x_i}$$
$$fitness^x_{\neq}(x_i) = \mathbb{1}_{f(x)\neq f(x_i)} + (1 - d(x, x_i)) - \mathbb{1}_{x=x_i}$$

  with d the normalized Euclidean distance. These fitness functions ensure that a generated instance x_i is evaluated to be more relevant if it is close but different from x. The authors show experimentally that choosing this distance, as well as this genetic algorithm, gives a better approximation of the classifier f by h_x. Besides, choosing to generate each class separately ensures that the generated training set X_{h_x} is balanced.

- **Training step.** The surrogate model, a decision tree, is then trained on this dataset. To build this tree, LORE uses a variant of the C4.5 algorithm.

- **Explanation extraction step.** Finally, two types of explanations are extracted from this tree by studying its structure. First, an explanation for $f(x)$ is generated by looking at the path taken to calculate $h_x(x)$. Thus, the final explanation takes the form of a list of decision rules. Additionally, a set of *counterfactual rules* is generated by looking at which conditions of the decision rule should be changed to alter the predicted class $f(x)$. This second approach is presented in more details in Section 2.3.2, page 35.

2.3 | Counterfactual Explanation Approaches

Counterfactual explanations are another type of local explanations that are the focus of Chapters 3 and 4. In these chapters, this family of approaches is studied to highlight potential issues of the post-hoc context. In this section, these approaches are

discussed. After describing their general principle in Section 2.3.1, several methods from the literature are presented in Section 2.3.2. Finally, some related works about adversarial examples are presented and discussed in Section 2.3.3 to underline their similarity and differences.

2.3.1 | Principle

This section presents the general principle behind counterfactual explanation approaches. First in Section 2.3.1.1, counterfactual explanations are defined and put in perspective with counterfactual reasoning, a cognitive sciences principle. Then, Section 2.3.1.2 is devoted to presenting some of their upsides. Finally, Section 2.3.1.3 presents a formal definition for counterfactual explanations.

2.3.1.1 | From Counterfactual Reasoning to Counterfactual Explanations in Machine Learning

Counterfactual reasoning. The term *counterfactual* originally comes from psychology (see e.g. Roese, 1997; Byrne, 2008), and Artificial Intelligence literatures (Lewis, 1973; Ginsberg, 1986). In these domains, counterfactual reasoning "is a concept that involves the creation of possible alternatives to life events that have already occurred (counterfactual world). This resoning revolves around answering the question *What if ...?* when thinking of how things could have turned out differently" (Wikipedia definition for Counterfactual Thinking[2]). In psychology, counterfactual reasoning is associated to multiple benefits, some of them mentioned in the next Section 2.3.1.2.

Loosely inspired from cognitive sciences, counterfactual reasoning in machine learning can be found in several tasks such as planning failures (Halpern and Pearl, 2005), reinforcement learning (Swaminathan and Joachims, 2015) or generative adversarial networks (Neal et al., 2018). Another highly studied topic is counterfactual fairness (Kusner et al., 2017), which uses counterfactual reasoning to assess the presence of bias. In this paradigm, a prediction of a model is considered to be biased if it differs in the counterfactual world.

Counterfactuals for Machine Learning Interpretability. Counterfactual explanations for machine learning predictions (also sometimes referred to as contrastive explanations) have been used for several years (Bottou et al., 2013). They have been recently the focus of attention since Wachter et al. (2018): this law research paper dis-

[2]https://en.wikipedia.org/wiki/Counterfactual_thinking

cusses the *right for an explanation* stated in the GDPR. More precisely, it proposes to use counterfactual explanations to explain individual predictions made by a black-box model. Since then, multiple works on counterfactual explanations have been proposed (see Artelt and Hammer (2019) for a recent survey on counterfactual explanations), some of them being presented further in this section. Several reasons explain this success, some of them detailed in the next section, after counterfactual explanations for Machine Learning Interpretability are defined here.

Counterfactual explanations for Machine Learning Interpretability can be seen as an adaptation of counterfactual reasoning to the context of Machine Learning Interpretability. The key idea is to analyze the predictions of a classifier by envisaging alternative situations that may modify them. Let us consider a trained classifier and the case where a single prediction, associated with an instance from the input space, is to be interpreted. In this context, the goal of counterfactual explanations is to consider an alternative version of this instance, that is to say another instance, and study the differences observed in their predictions. This difference between the original state and the counterfactual one constitutes a *change* that can be observed and measured. This change can also be interpreted as a list of *actions* that are required to alter the prediction. Hence, the objective of counterfactual explanations can be formulated as the following question:

What actions are required to alter this prediction?

Another, slightly different, formulation is:

How does applying this change impact the prediction?

Considering these questions, the following definition for a counterfactual explanation for a given prediction can thus be proposed:

Definition 1 (Counterfactual Explanation)**.** *In the context of post-hoc local interpretability, a* counterfactual explanation *is a list of* actions *to apply to an instance to alter its prediction.*

These *actions* are modifications in the feature values of the instance. The resulting modified instance is called *counterfactual example*.

Let us consider the context where a customer applies for a loan to a bank. A model is used to predict whether a customer, given the information he/she provides, is likely to default or not. Supposing that the model predicts the customer to default, a counterfactual explanation would take the following form:

The credit application of Customer x was rejected. In order to have it approved, Customer x would have needed: (i) to have a yearly salary ***increased by*** \$1000. *(ii) To smoke* 2 ***less*** *cigarettes per week.*

Obviously, multiple counterfactual explanations may be proposed for a same prediction. The goal of counterfactual explanation approaches is therefore to identify which actions would help best provide the user with insights about the classifier. Hence, counterfactual approaches for Machine Learning Interpretability focus on specifying what actions should be applied to ensure relevant explanations.

Although out of the scope of this thesis, counterfactual explanations can also be defined for the task of regression. Instead of relying on identifying a class change, these approaches generally rely on detecting a meaningful variation in the predicted value (see for instance Bottou et al., 2013; Lucic et al., 2019).

2.3.1.2 | Motivations

Counterfactual explanations have multiple upsides, both in terms of usability and scientific justification. This section lists some of them, as a motivation for their importance in this thesis. First, we present upsides related to cognitive sciences. Then, we present more practical arguments.

Upsides of counterfactual reasoning. To some extent, several upsides of counterfactual explanations can be seen as inherited from counterfactual reasoning. In the cognitive sciences literature, such counterfactuals, created when envisaging an alternative reality, have been shown to be a very natural way of thinking, emerging in the child's mind since the early age (O'Connor et al., 2014). In addition, creating these alternatives has been shown to help in learning from experience, modulating emotional state and contributing to decision-making and social functioning (McCrae, 1987; Roese, 1997). These counterfactual thoughts are shown to be a spontaneous, even systematic ("irresistible") process in case something bad happens (Goldinger et al., 2003).

Transposed to the context of Machine Learning Interpretability, these benefits are also highly valuable. Indeed, as stated in Section 2.1.1, page 11, Machine Learning Interpretability is generally considered to assist in decision-making. Thus, proposing an explanation following a natural reasoning process is a relevant paradigm. Furthermore, machine learning explanations have been shown to be more important in cases where the predictions of the model are wrong (Doshi-Velez and Kim, 2017), as

discussed in Section 2.1.1, page 11. As mentioned earlier, in such a situation, counterfactual reasoning has been showed to be a systematic thought process for the user. In this sense, counterfactuals seem to be a natural solution to generate explanations.

Explaining by comparing. Another upside of counterfactual explanations comes from their reliance on specific instances (counterfactual examples) to explain individual predictions. Indeed, this can be related to the task of learning through examples in teaching sciences (see e.g. Watson and Shipman, 2008). Explaining through particular instances has been shown to facilitate the learning process of a user, especially in cases where the concept to be learned is complex (see e.g. Decyk, 1994; Watson and Shipman, 2008; Mvududu and Kanyongo, 2011). For instance, a child learning to identify pictures of animals could be given an explanation in the form of: *"Had this animal had longer ears, it would be a hare instead of a rabbit."* Similarly, Watson and Shipman (2008) show through experiments that generated examples help students "see" abstract concepts that they had trouble understanding with more formal explanations.

In the context of Machine Learning Interpretability, this is of course especially relevant in situations where the classifier decision to explain is very complex and other types of interpretability approaches may fail to provide meaningful explanations.

A more practical explanation Because they can be associated to explicit actions required to change the prediction of an instance (see Section 2.3.1.1), counterfactual explanations provide a sense of *tangibility*: by giving exact instructions on how to act on the model, the generated explanations are directly understandable and actionable (Wachter et al., 2018). Because they explicitly state which actions impact the predictions, counterfactual explanations are particularly appropriate in the context of the *right for an explanation* of the GDPR. For instance, a customer of a bank getting his loan application denied needs to understand, besides the reasons leading to this rejection, what he would need to change in order to have it accepted. This can be opposed to other explanations, in particular the ones using feature importance vectors for instance (such as LIME, see 24). Arguably, these forms of explanation are harder to use and understand, especially for a non-expert user (Molnar, 2019).

2.3.1.3 | Formal Principle

In this section, we propose a definition for counterfactual explanations inspired from counterfactual reasoning. As in the previous sections, let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a classi-

fier, and $x \in \mathcal{X}$ the instance whose prediction $f(x)$ is to be interpreted. In Section 2.3.1.1, counterfactual explanations have been defined as answers to the question: *What actions are required to alter this prediction?* This is equivalent to constructing an instance $e \in \mathcal{X}$, such that:

$$f(e) \neq f(x) \tag{2.1}$$

The difference vector $e - x$ is thus the change required to alter the prediction.

The notion of *required* change to alter the prediction implies a notion minimal effort. Although multiple counterfactual explanations may be proposed, they share the principle that e that is associated to the *smallest* change $e - x$. This leads to defining a cost function $c : \mathcal{X} \to \mathbb{R}$, associated to any change $e - x$. Finding a counterfactual explanation can then written as the minimization problem:

$$\begin{aligned} e^* = \underset{e \in \mathcal{X}}{\arg\min} \quad & c(e) \\ \text{subject to} \quad & f(e) \neq f(x) \end{aligned} \tag{2.2}$$

This is an inverse classification problem. Therefore, some inverse classification approaches can be considered to be counterfactual approaches (Barbella et al., 2009; Martens and Provost, 2014; Lash et al., 2017b). Some of these approaches are described in more details later in this section.

Three elements control the counterfactual problem of Equation 2.2: the cost function c, the explored space in which the solution is searched, and the considered optimization method. These elements, often defined through the considered context and user needs (task, agnosticity assumptions, etc.), are discussed in turn below.

Defining the cost function. Choosing how to define the cost function c is obviously crucial. For this purpose, l_2 and l_1 distances are common choices: Lash et al. (2017b) consider the l_2 norm, while Wachter et al. (2018) use the l_1 norm. However, other metrics can be considered: for instance, LORE (Guidotti et al., 2019a) defines c as the l_0 distance. The advantage of l_0 is to introduce naturally a sparsity contraint on the explanation.

Other possibilities include having unequal costs for changes in different features. Considering the previous credit application example, the aforementioned Customer x might have an easier time reducing his/her number of smoked cigarettes than changing his/her salary. This could be translated as weights in the cost function. Similarly, one could envisage asymetric costs for actions, that is to say different costs for increasing or decreasing a given continuous variable. Lash et al. (2017a) propose a

double extension of the inverse classification problem, which considers both of these possibilities.

Defining the exploration space. The exploration space is the space in which the solution to the problem of Equation 2.2 is searched: depending on the considered context, this space may actually be different (smaller) from (than) $\mathcal{X}$.

This situation commonly occurs when the goal is to use the explanation to impact the prediction of a real life observation: some of the attributes describing $\mathcal{X}$ may not be not directly actionable. In this case, a first solution is to exclude the corresponding features from the space in which the solution of Equation 2.2 is searched. For instance, an explanation given in the context of the aforementioned credit application example may be useless in terms of actionability if Customer x is asked to change his/her age to become younger.

Going further, Lash et al. (2017a) propose to split the features describing $\mathcal{X}$ into three sets: the ones on which the user can have a direct impact $\mathcal{F}_d$, the ones that can be indirectly impacted $\mathcal{F}_i$ and the ones that cannot be impacted at all $\mathcal{F}_u$. Besides excluding the latter ones from the search space, the authors propose to infer $\mathcal{F}_i$ from $\mathcal{F}_d$ using a predictive model, allowing to restrict the feature space to $\mathcal{F}_d$ only. An example of such situation is for instance in the case of an online marketing model predicting whether a customer is going to buy a product. A feature describing the customer is the number of ads he/she has seen. The company cannot impact this value directly. It can, however, impact the number of ads it sends to the customer, which in turn impacts the number of seen ads.

Another example of restricting the exploration space can be found in the approach proposed by Martens and Provost (2014), who use inverse classification to explain document classification. In particular, for a given document x, they identify words it contains which, when removed, change its prediction. In this situation, the exploration space is thus made of the words contained in x.

Solving the minimization problem. Solving the problem defined by Equation 2.2 depends on the considered paradigm. In particular, agnosticity assumptions heavily impact how this problem can be solved. For instance Barbella et al. (2009) propose to generate counterfactual explanations specifically for the predictions of a SVM classifier by identifying meaningful support vectors. Another example is the works of Ustun et al. (2019) and Russell (2019), who propose efficient search algorithms in the case where the classifier is linear. In these works, the counterfactual problem is solved using specific knowledge of the classifier.

However, when no information is available about the classifier, solving the problem of Equation 2.2 is complex. Model-agnostic approaches thus generally rely on finding an approximate solutions by sampling numerous instances in an area around x in order to detect a class change (see e.g. Lash et al., 2017a). This topic is examined in detail in Chapter 3.

2.3.2 | Two Counterfactual Explanation Approaches

This section describes in turn two counterfactual approaches, namely HCLS proposed by Lash et al. (2017a), and the already presented LORE (Guidotti et al., 2019a), that are relevant for the rest of the thesis. In particular, these approaches are the focus of Chapter 4, page 69.

HCLS (Lash et al., 2017a) As mentioned earlier, finding a counterfactual explanation is equivalent to solving a specific in inverse classification problem. Lash et al. (2017b) consider a framework for model-agnostic budget-constrained inverse classification. The idea behind this problem is to solve an inverse classification problem with a notion of maximum budget, capping the allowed cost $c(e)$. The proposed problem is therefore slightly different from the one written in Equation2.2:

$$\begin{aligned} e^* = \arg\max_{e \in \mathcal{X}} \quad & p(e) \\ \text{subject to} \quad & c(e) \leq B \end{aligned} \tag{2.3}$$

where $p : \mathcal{X} \rightarrow [0, 1]$ is the probability or confidence function returned by the black-box classifier to belong to a specific class, and $B \in \mathbb{R}^+$ a hyperparameter defining the budget, that is to say the maximum cost allowed to change x into e.

To solve this problem, the authors propose several heuristic-based algorithms. As the obtained results do not differ much across these models, we have chosen to focus on one of them, called HCLS (*Hill-Climbing + Local Search*). The principle HCLS relies on is to iteratively perform a local search to identify which direction is the most promising to increase the probability of belonging to the targeted class. The local search is performed by iteratively applying Gaussian perturbations to the instance, and identifying the perturbation that induces the biggest variation in classification probability p. Additionally, at each step, the instance is projected to the space:

$$\Delta_B = \{z \in \mathcal{X} \mid c(z) \leq B\}$$

The procedure is repeated a given number of iterations. In the end, HCLS returns a local maximum for p, as well as the corresponding instance e^*. It is important to note

that at the end of the run of the procedure, e^* may not necessarily verify $f(e^*) \neq f(x)$. This can be caused by two reasons. First, as implied by Equation 2.3, the value of p within Δ_B may not be high enough to guarantee a class change. Furthermore, the iterative aspect of the HCLS implies that it is vulnerable to local maxima. If such a situation arises, no matter how high the number of iterations is, no counterfactual solution may be found. While not problematic in the context of inverse classification, this represents a limit of the use of HCLS as a counterfactual explanation approach.

LORE (Guidotti et al., 2019a) As mentioned in Section 2.2.2.2, page 26, LORE generates a model-agnostic counterfactual explanation using a surrogate model. As a reminder, given a trained classifier f and an instance $x \in \mathcal{X}$ whose prediction is to be interpreted, LORE uses a genetic algorithm in order to generate a dataset in a local region centered on x. A decision tree classifier is then trained on this dataset, and a counterfactual explanation is built by exploring this tree. The idea is to use the tree structure to identify the minimal number of attribute values of the considered instance that need to be modified in order to change $f(x)$. For each leaf $\mathcal{Q}$ of the tree that leads to a predicted label $l \neq f(x)$, the number of modifications that need to be applied to x in order to have $x \in \mathcal{Q}$ is calculated. The paths that lead to the leaves that require the least amount of changes to x are proposed as counterfactual explanations. Moreover, in case of equal number of required changes, multiple counterfactual rules may be proposed.

The final explication is thus a list of counterfactual *rules*, instead of counterfactual instances. In Chapter 4, page 69, a method is proposed to transform such counterfactual rules into counterfactual instances.

By minimizing the number of modifications to apply to x, LORE maximizes the sparsity of the explanation vector. However, it is important to note that this counterfactual explanation is searched within the space covered by the generated local neighborhood. Therefore, the generated counterfactual maximizes the sparsity of the explanation within a local region (estimated with a decision tree classifier). As a result, writing the problem addressed by LORE in the form of Equation 2.2 is difficult. A discussion on this topic is proposed in Chapter 3. Additionally, the counterfactual explanations generated are analyzed in Chapter 4.

2.3.3 | Related Works: Adversarial Examples

In this section, we give a brief overview of the field of adversarial learning, and show how close yet distinct adversarial and counterfactual examples are.

Adversarial examples. Adversarial learning is a subdomain of machine learning that focuses on its security aspects. In particular, the notion of adversarial attack (or evasion attack), introduced by Biggio et al. (2013), aims at *fooling* a trained classifier f. This means maliciously generating an instance $\tilde{x}$ such that $f(\tilde{x})$ is *wrong*, thus exploiting the gap between what has been learned by the model and the reality it tries to approximate. This notion of *fooling* is not formally defined. Yet, this is commonly not seen as an issue since most works in the field of adversarial learning use deep neural networks for image (Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016) or text (Biggio and Roli, 2018) classification: in these situations, a human can generally recognize the true class of an instance, and thus assert whether the model is wrong.

Numerous works focus on generating attacks to fool the classifier, or studying how to protect classifiers from these attacks and study their robustness (see e.g. Papernot et al., 2016; Pinot et al., 2019).

Similarities with counterfactual examples. In order to generate adversarial examples, most approaches rely on existing ground-truth instances and on the assumption that very slightly modifying this instance should, in fact, not alter its prediction. This is of course especially true in the context of text or image classification, where the input dimensionality is so high that minor perturbations do not change the overall ground-truth label of the instance. Given a ground-truth instance x, the goal of these approaches is thus to find a perturbation $\epsilon \in \mathcal{X}$ as small as possible, such that $\tilde{x} = x + \epsilon$ satisfies $f(\tilde{x}) \neq f(x)$. Of course, this formulation is the same problem as the counterfactual problem written in Equation 2.2. Because they are generated through the same inverse classification problem, a confusion thus exists between counterfactual explanations and adversarial examples. From a formal point of view, these two concepts are identical.

Differences. No proper definition has been proposed to make the distinction between these two concepts. The general consensus is therefore that because counterfactual examples are an interpretability topic and adversarial examples a security topic, they differ in their objective. While most adversarial attacks focus on generating an *imperceptible* perturbation, a similar objective would be useless in the context of interpretability: not being able to "see" (and *a fortiori* understand) the counterfactual explanation vector $e - x$ would make the explanation useless for the user. Counterfactual explanations are therefore relying on identifying a small yet perceptible perturbation, while adversarial attacks focus on imperceptible perturbations.

This difference in objective is usually translated into different optimization problems and cost functions. While solely minimizing a l_2 program is desirable in the context of adversarial examples, it is not in the context of interpretability. As presented in Section 2.3.1.3, counterfactual explanation approaches thus often include some notion of sparsity to make them more useful.

Adversarial examples and counterfactual examples are thus similar in formalization. However, their different objectives make them easily distinguishable.

2.4 | Conclusion

After having sketched some key elements of interpretability, this chapter presented two families of interpretability approaches: local surrogates and counterfactual explanations. Understanding the numerous possible objectives and characteristics helps to understand the diversity of interpretability approaches.

The works presented in this thesis fall into the local post-hoc paradigm. The considered context is therefore the generation of explanations for a single prediction of a trained classifier. Additionally, a fully agnostic context is considered: no knowledge is available about either the classifier, nor any data (as presented in Section 2.1.3). This paradigm corresponds to realistic constraints for interpretability use cases, for instance when confidentiality constraints make any access to knowledge about the classifier or existing data impossible.

However, in this paradigm, several issues can arise. This thesis proposes tools to identify and study three issues:

- In Chapter 3, we analyze the difficulty to generate local explanations in a fully agnostic context. This study is conducted by focusing on counterfactual explanations, presented in Section 2.3, page 28. Moreover, we study a second issue and propose an analysis to show that, in the considered post-hoc paradigm, there is a risk of generating explanations that lie out of the distribution of ground-truth data.

- The link between explanations and ground-truth data is further studied in Chapter 4. In this chapter, in light with the diagnostic methods for interpretability approaches discussed in Section 2.1.4, page 19, we propose an assessment of the risk of generating *unjustified* explanations, a desirable property that we define for counterfactual explanations. In a second analysis, we also explore the

link between justification and the notion of explanation locality, presented in Section 2.1.3, page 17.

- The concept or explanation locality is further studied in Chapter 5. Focusing on surrogate model approaches, presented in Section 2.2, page 21, we discuss the complexity of defining the locality of an explanation in a fully agnostic context. We also draw a parallel between counterfactual explanations and surrogate model approaches.

2.5 | Notations

As most of the work of this thesis uses identical assumptions, we introduce in this section several notations to avoid redefining them in each chapter.

We consider a binary classifier f mapping the input space $\mathcal{X}$ of dimension $dim(\mathcal{X})$ to an output space $\mathcal{Y} = \{0, 1\}$. When specified, the output of f may be a confidence score, or probability. In all the thesis, no knowledge about f whatsoever is available: it is considered a *black-box* classifier. The classifier f is trained on a dataset X of instances of $\mathcal{X}$. Unless specified (such as in Chapter 4), no information about X is available either. However, we suppose that $\mathcal{X}$ is described only by numerical features.

In the context of local interpretability, we focus on generating explanations for a single prediction. Let $x \in \mathcal{X}$ be the observation whose prediction $f(x) \in \mathcal{Y}$ is to be interpreted. This observation is not necessarily a training instance: $x \notin X$.

Other notations, specific to the context of each chapter, are introduced further.

3

Generating Post-hoc Counterfactuals and the Risk of Out-of-distribution Explanations

In this chapter, we discuss the complexity of generating post-hoc local explanations that are easily understandable in a fully agnostic context. Although rarely presented with this perspective, counterfactual explanations are naturally well suited to answer this problem. Indeed, existing counterfactual approaches generally rely on the cognitive upsides provided by this form of explanation, discussed in Section 2.3.1.2, page 31. Among these, the counterfactual explanation provided is thus supposed to be easily understandable by the user. Yet, this objective of understandability is rarely formulated as such by the existing approaches, a fortiori optimized. In this chapter, we propose to measure this explanation understandability with the explanation sparsity, and propose to directly integrate it as a constraint to generate counterfactual examples.

After discussing the desired behavior of a post-hoc counterfactual explanation, we propose a formalization to guide the generation of explanations. We show that ensuring an explanation that is both local and easy to understand is a complex problem that is not directly addressed in the literature. We propose to answer the formulated objective problem with the *Growing Spheres* algorithm (GS). After showing the efficiency of this approach, we use it to address a second issue associated to the considered post-hoc paradigm in this setting and tackled in this thesis: the generation of explanations that lie out of the distribution of ground-truth data.

In Section 3.1, we discuss the desired behavior of of counterfactual explainers and propose an idea to generate explanations that are both local and simple to understand. This idea is formalized in Section 3.2, and implemented with the *Growing*

Spheres algorithm. This proposition is then validated experimentally in Section 3.3. Finally, Section 3.4 is devoted to studying the risk of having out-of-distribution explanations.

Parts of the work presented in this chapter are the subject of the papers *Inverse Classification for Comparison-based Interpretability in Machine Learning*, published at the IPMU 2018 conference (Laugel et al., 2018a); and *Issues with post-hoc counterfactual explanations: a discussion*, published at ICML 2019's Human in the Loop Learning Workshop (Laugel et al., 2019a).

3.1 | Motivations

The post-hoc paradigm, described in Section 2.1.3, page 17, raises questions about how to define local explanations. Although particularly suited to answer this problem, counterfactual explanations are generally not presented as such. In this section, we propose to give motivations for counterfactual explanations in light of this problem of generating a local explanation in the post-hoc context.

First, in Section 3.1.1, we explain why counterfactuals are a good answer to the issue of generating a local explanation. Then, in Section 3.1.2 we discuss the desired behavior of a post-hoc counterfactual explanation. Finally, additional assumptions that contribute to defining the studied problem are described in Section 3.1.3.

3.1.1 | From Locality to Counterfactuals

As presented in Section 2.1.3, page 17, *local* interpretability approaches aim at generating explanations to help a user understand the reasons for a single prediction by a classifier by focusing on identifying some local behavior of the classifier (Rueping, 2006; Guidotti et al., 2018; Carvalho et al., 2019). Following the discussion of Section 2.1.2, page 13, the local explanation should give insights about the "parts" of the model that are activated specifically when making the prediction. These parts can be easily identified in the context of specific classifiers: one can mention as examples the list of activated rules of a decision tree (Guidotti et al., 2019a) or the visualization of the neuron activations of a deep neural network (Yosinki et al., 2015; Selvaraju et al., 2016).

However, identifying and extracting such a local rationale is not possible in the post-hoc context, as it requires knowledge and access to the inner workings of the model to be explained. Furthermore, this local behavior can be hard to define for other models such as SVM (e.g. with a complex unknown kernel) or ensemble meth-

ods for instance: the former may use non-linear kernels which make the definition of the locality difficult, while the latter aggregate the decisions of multiple weak classifiers, making the notion of local behavior complex to define for the ensemble model. Therefore, we propose to reduce the generation of a local explanation to identifying the local decision boundary of the classifier. In this context, the local decision boundary is defined as the *closest* part of the decision boundary to the observation whose prediction is to be interpreted.

For that reason, counterfactual explanations (presented in Section 2.3, page 28) appear as a good answer to the task of generating a local explanation. Indeed, they try to identify the closest touchpoint of the decision boundary of the classifier, and therefore generate the most local explanation possible. Nevertheless, as discussed in Section 2.3.1.3, page 32, generating a counterfactual explanation relies first and foremost on defining a cost function that is relevant for the considered problem and user. This problem is tacked in the next section, in which the desired behavior of the post-hoc counterfactual explainer (hence the proposed cost function) is discussed, and used to define our objective function.

3.1.2 | Discussion about the Desired Behavior

Two objectives: locality and sparsity. As explained in Section 2.3.1.3, page 32, given a black-box classifier and an observation whose prediction is to be interpreted, a counterfactual explanation is associated to a specific data point that is predicted to belong to the other class. The final explanation given to the user is then expressed in the form of the change vector between the observation and the identified data point (see Definition 1, page 30). Following the principle of locality discussed in the previous section, the explaining data point must be as close as possible to the observation whose prediction is to be interpreted. However, defining the associated cost function in the post-hoc context is challenging. Due to the considered agnosticity assumptions, no information about the potential metric functions used by the classifier is available to help define the associated cost function of the explanation vector. Moreover, the classifier might have used some specific distance function, sometimes even defined in another feature space, to make predictions. In this context, using the Euclidean distance to define this closeness has been shown to be a reasonable assumption (see e.g. Strumbelj et al., 2009; Lash et al., 2017b). We thus propose to identify the l_2-closest touchpoint of the decision boundary.

However, recalling the second objective of explanation methods defined in Section 2.1.2, page 13, in order to be usable, the explanation should furthermore be

easily understandable by the user. In this context, focusing solely on the Euclidean distance may lead to explanations that are difficult to read. This is especially true in the case of high dimensional data for instance: finding a counterfactual example by looking for the closest in terms of Euclidean distance would lead to an explanation involving a high number of attributes. As evoked in Section 2.1.4, page 19, including the sparsity of the explanation in the formulated objective seems therefore important.

Combining these two criteria. This raises the question of the combination of these two criteria (Euclidean distance and sparsity of the explanation) should be performed. Aggregation operators have been well-studied (see e.g. Detyniecki, 2000) and are often considered in the context of multi-criteria optimization. However, the main aggregation operators cannot be considered in the studied context. Indeed, one of the specificities of the desired behavior of the explainer system is that the two considered criteria (Euclidean distance and sparsity) are not comparable. As a matter of fact, the generated explanation should ideally be a local explanation that is also sparse. Therefore, a compromise between these two criteria (e.g. using a weighted mean) does not seem desirable either. Indeed, the simplicity of the explanation (measured by the counterfactual sparsity) should be obtained in addition to their locality (measured by the Euclidean distance), not at its cost. Comparing these two criteria, and a fortiori compensating a lack of locality with more sparsity is thus undesirable. Therefore, aggregators such as disjunctive, conjunctive and variable behavior operators cannot be used.

Existing post-hoc counterfactual approaches (namely the ones described in Section 2.3.2, page 35) do not address this problem. One the one hand, HCLS (Lash et al., 2017b) does not take into account sparsity at all. On the other hand, LORE (Guidotti et al., 2019a) proposes to identify the most sparse counterfactual explanation in the leaves of a decision tree generated in a local neighborhood. As a result, a certain level of locality is guaranteed, delimited by the generated neighborhood. However, sparsity is still obtained at the cost of locality within this neighborhood.

To circumvent this issue, we propose to split the problem into two parts that we solve successively: first, we focus solely on minimizing the Euclidean distance, and then we try to make the produced explanation as sparse as possible. This approach of the problem allows us to get the "best of both worlds", and not having to compromise locality for sparsity. This idea is further detailed and formalized in Section 3.2.1.

3.1.3 | Additional Assumptions

In addition to the considered agnosticity constraints, we make additional assumptions, described in turn below.

No categorical attribute. In this chapter, like in the rest of this thesis, we focus on numerical attributes. Indeed, the proposed approach relies on computing distances between observations, which can be difficult in the context of unordered categorical data. For the sake of simplicity, we thus choose to exclude these attributes from the study.

Output of the classifier. Furthermore, we focus on the case where the classifier is assumed to return only a label, and not a continuous classification confidence score (e.g. probability). This is intended to make the proposed approach more general. Finally, we focus on binary classification only. The extension of the proposed approach to multiclass classification is not particularly challenging; this topic is discussed in Section 3.2.4, page 56.

3.2 | Proposed Problem Formalization and the *Growing Spheres* Algorithm

As stated in the previous section, generating a counterfactual explanation that is both local and sparse is challenging in the post-hoc paradigm. Existing approaches thus generally rely on optimizing one of these notions at the cost of the other. In this section, we propose a formalization of the objective of the explainer and use the discussions of Section 3.1 to propose a post-hoc counterfactual explanation approach, called *Growing Spheres*.

First, in Section 3.2.1, a formalization of the counterfactual problem we want to solve is proposed. Section 3.2.2 describes the *Growing Spheres* algorithm. Then, a discussion about the main hyperparameters that *Growing Spheres* is relying on is proposed in Section 3.2.3. Finally in Section 3.2.4, we tackle the extension of the proposed approach to the context of multiclass classification.

3.2.1 | Problem Formalization

We use the same notations as the ones introduced in Section 2.5, page 39, and suppose that no information is available about the considered classifier. The goal of the

proposed counterfactual explanation approach is to explain a prediction $f(x)$ through another observation $e \in \mathcal{X}$, belonging to another class, i.e. such that $f(e) \neq f(x)$.

Following the discussion of Section 3.1, we propose to use the counterfactual formulation written in Equation 2.2, page 33, and therefore define the function $c : \mathcal{X} \to \mathbb{R}^+$ such that $c(e)$ is the cost of moving from observation x to the counterfactual example e. Using this notation, it is recalled that the problem we focus on can be written as:

$$\begin{aligned} &\underset{e \in \mathcal{X}}{\text{minimize}} \quad c(e) \\ &\text{subject to} \quad f(e) \neq f(x). \end{aligned} \tag{3.1}$$

As explained in Section 2.3.1, page 29, the final form of explanation is the difference vector $e - x$.

In order to obtain an explanation that is both local and sparse, we proposed in Section 3.1.2 to decompose the problem into two parts that are to be solved successively. As a result, the problem we propose to formalize in this section is an extension of the counterfactual problem written in Equation 3.1. This section formalizes these successive objectives: first, the problem of minimizing the Euclidean distance is presented in Section 3.2.1.1. Then, the problem of making the produced explanation as sparse as possible is tackled in Section 3.2.1.2. Finally, in Section 3.2.1.3 a discussion is proposed about the feasibility of the problem and the questions it raises.

3.2.1.1 | Solving the l_2 Problem First

In Section 2.3.1.3, page 32, we explained that three parameters need to be defined to tackle the counterfactual problem: the explored space, the cost function considered, and the optimization method. However, defining these elements is complex in the considered post-hoc context. They are discussed in turn in the following paragraphs.

Exploration space. An important parameter of the counterfactual problem is to define the space in which the solution is to be searched. Had the training set X been available, it could have been used to help finding the solution to Equation 3.1 by reducing the size of the explored space. Yet, due to the post-hoc assumption made, no data or knowledge is available to reduce the scope of this problem. Therefore, the solution needs in this case to be searched within the entire feature space $\mathcal{X}$.

Cost function. In the previous section, we explained that the desirable behavior of the explainer system is to build explanations minimizing both the Euclidean distance

and the sparsity of the explanation vector. The sparsity of the explanation vector $e - x$ can be measured with the l_0 norm, defined for a given vector $z = [z_i]_{i=1...dim(\mathcal{X})} \in \mathcal{X}$ by:

$$||z||_0 = \sum_{i \leq dim(\mathcal{X})} z_i^0 = \sum_{i \leq dim(\mathcal{X})} \mathbb{1}_{z_i=0}$$

A first idea thus revolves on integrating both criteria in the cost function c. However, as discussed in Section 3.1.2, defining the cost function c as a compromise between the Euclidean distance and sparsity, such as $c(x, e) = ||x - e||_2 + \gamma||x - e||_0$, with $\gamma \in \mathbb{R}^+$ for instance, is not desirable. To circumvent this issue, we propose to split the problem into two parts and first focus solely on the l_2 component, which leads to the problem:

$$\begin{aligned} e^* = \underset{e \in \mathcal{X}}{\arg\min} \quad & ||x - e||_2 \\ \text{subject to} \quad & f(e) \neq f(x). \end{aligned} \tag{3.2}$$

The solution $e^* \in \mathcal{X}$ of this problem is the l_2 closest touchpoint of the decision boundary of f, evoked in Section 3.1.

Optimization method. In practice, because we have no knowledge about the classifier, solving *analytically* the counterfactual problem of Equation 3.2 is impossible. Hence, like most post-hoc approaches (such as for instance the approaches described in Section 2.2.2.2, page 26, also fully agnostic), the program is to be solved using Monte-Carlo estimation. Through the sampling and labelling of the generated instances with f, an approximation of the solution of this problem can be found. Solving this problem therefore necessarily induces an approximation error (in addition to potential errors induced by numerical instability for instance). The existence of this error leads to an approximate solution $\tilde{e} \in \mathcal{X}$ for Equation 3.2. This solution is by definition sub-optimal, that is to say such that:

$$||x - \tilde{e}||_2 > ||x - e^*||_2$$

Such a sub-optimal solution means that $\tilde{e}$ is located further away from x than the theoretical solution e^*, i.e. further away "behind" the decision boundary of f. Figure 3.1 illustrates this principle. Considering a trained black-box classifier f whose decision boundary is represented by the black line, the prediction $f(x)$ of an instance $x \in \mathcal{X}$ is to be interpreted. The instance e^* represents the theoretical solution of Equation 3.2, i.e. the closest l_2-touchpoint of the decision boundary. On the other hand, $\tilde{e}$ represents a sub-optimal solution approximating e^*.

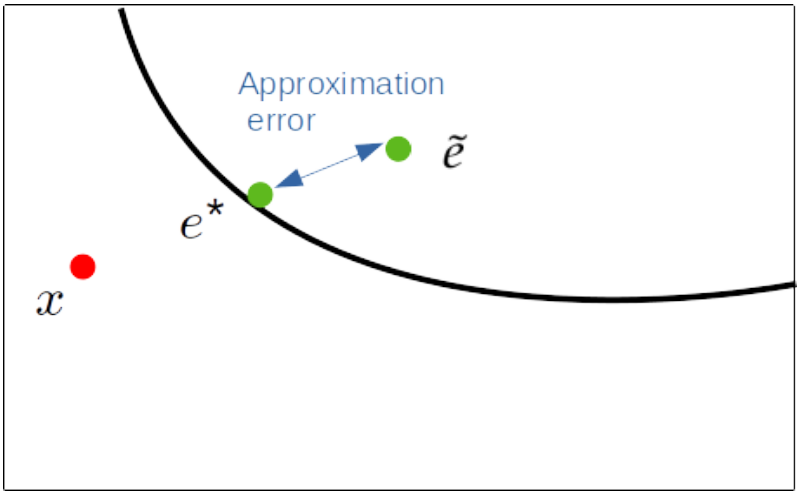


Figure 3.1: Illustration of the principle behind the definition of e^* and $\tilde{e}$, respectively the theoretical and approximate solutions of Equation 3.2.

This approximation error is obviously not desirable. However, the existence of the induced distance between the found solution and the decision boundary can be used to maximize the sparsity of the explanation.

3.2.1.2 | Sparsity through Projections

The obtained explanation vector $\tilde{e} - x$ is supposedly not sparse and therefore complex to understand. The objective of this step is to:

- make it sparser, i.e. minimize $||x - \tilde{e}||_0$;
- without sacrificing locality, i.e. the l_2 distance;
- while still ensuring $f(\tilde{e}) \neq f(x)$, the counterfactual condition.

To do so, we consider compositions of orthogonal projections of the found solution $\tilde{e}$ on the hyperplanes $\mathcal{H}_i$ defined by the coordinates of x: $\mathcal{H}_i = \{z \in \mathcal{X} \text{ s.t. } z_i = x_i\}$, for $i \in \{1, ..., dim(\mathcal{X})\}$. Orthogonally projecting $\tilde{e}$ on $\mathcal{H}_i$ means indeed setting $\tilde{e}_i = x_i$ while leaving the other coordinates unchanged, thus reducing $||\tilde{e} - x||_0$ by 1. Consequently, the Euclidean distance is also reduced, since setting $\tilde{e}_i = x_i$ for a given i also implies reducing $||\tilde{e} - x||_2$.

Maximizing the sparsity of the explanation through these projections is thus equivalent to setting as many attribute values of $\tilde{e}$ as possible to the ones of x, without changing the predicted class. In other words, identifying the largest set of indices $\mathcal{I}$

such that:

$$f(\text{proj}_{\mathcal{I}}(\tilde{e})) \neq f(x)$$

where:

$$\text{proj}_{\mathcal{I}}(\tilde{e}) = z \text{ such that } \begin{cases} \forall i \notin \mathcal{I}, z_i = \tilde{e}_i \\ \forall i \in \mathcal{I}, z_i = x_i \end{cases}$$

Additionally, we introduce the following notation:

$$\begin{aligned} P_{\tilde{e}} &= \{\text{proj}_{\mathcal{I}}(\tilde{e}), \quad \mathcal{I} \subseteq \{1, ..., dim(\mathcal{X})\}\} \\ &= \{z \in \mathcal{X} \quad s.t. \quad \forall i \leq dim(\mathcal{X}), z_i \in \{\tilde{e}_i, x_i\}\} \end{aligned}$$

$P_{\tilde{e}}$ is the set of all possible projections of $\tilde{e}$ on the hyperplanes $\mathcal{H}_i$ as well as their combinations. The second expression allows to easily determine that the cardinal of $P_{\tilde{e}}$ is $2^{dim(\mathcal{X})}$.

We note that by definition we have: $x \in P_{\tilde{e}}$ since performing the maximum number $dim(\mathcal{X})$ of projections of $\tilde{e}$ on the hyperplanes defined by the coordinates of x leads to x: $\text{proj}_{\mathcal{H}_1}(...(\text{proj}_{\mathcal{H}_{dim(\mathcal{X})}}(\tilde{e})...) = x$. However, x obviously does not constitute a desirable solution to the considered problem.

Using the notations introduced, we have the following equivalence:

$$\begin{array}{lcl} \min\limits_{e \in \mathcal{P}_{\tilde{e}}} ||x - e||_0 & \iff & \max\limits_{\mathcal{I} \subseteq \{1, ...dim(\mathcal{X})\}} |\mathcal{I}| \\ \text{s.t. } f(x) \neq f(e) & & \text{s.t.} \quad f(\text{proj}_{\mathcal{I}}(\tilde{e})) \neq f(x) \end{array}$$

Let $\mathcal{I}^*$ be a set of $\{1, ..., dim(\mathcal{X})\}$ associated one of the solutions of this problem. Our proposition revolves around making the counterfactual explanation vector as sparse as possible. Instead of returning $\tilde{e}$, the solution of the l_2-minimization problem, we thus propose to return $e_f = \text{proj}_{\mathcal{I}^*}(\tilde{e})$. By definition, this solution satisfies $||e_f - x||_0 = |\mathcal{I}^*|$, meaning that the counterfactual explanation vector is potentially sparse (if $|\mathcal{I}^*| < dim(\mathcal{X})$).

3.2.1.3 | Feasibility of the Problem

Orthogonally projecting $\tilde{e}$ without changing the predicted class, as proposed in the previous section, is only feasible because we focus on $\tilde{e}$ instead of e^*. This section explains why.

Because the presented problem is solved sequentially, the solution found of the first optimization program impacts how the second one is defined. Indeed, the solution found for the l_2 minimization program $\tilde{e}$ obviously impacts the definition of $P_{\tilde{e}}$. Since $P_{\tilde{e}}$ is the subset in which the solution of the l_0 minimization problem is searched, its definition obviously impacts the final solution e_f.

Therefore, it must be underlined that this approach is only possible because of the approximation error, i.e. because $\tilde{e} \neq e^*$. Indeed, we can formulate the following theorem:

Proposition 1. *Let e^* be the analytical solution of Equation 3.2 and the associated l_0 minimization problem:*

$$\begin{aligned} \min_{e \in P_{e^*}} \quad & ||x - e||_0 \\ s.t. \quad & f(e) \neq f(x). \end{aligned}$$

This problem has a unique solution e^*.

Proof. Let e^* be the theoretical solution of the l_2 minimization program of Equation 3.2. Let us suppose that there exists a solution to the l_0 minimization problem different from e^*, i.e. that there exists an instance $z \in P_{e^*}$ satisfying $f(z) \neq f(x)$ and such that $z \neq e^*$.

$$\begin{aligned} ||x - z||_2^2 &= \sum_{i=1}^{dim(\mathcal{X})} (z_i - x_i)^2 \\ &= \sum_{i \notin \mathcal{I}} (e_i^* - x_i)^2 + 0 \\ &< \sum_{i=1}^{dim(\mathcal{X})} (e_i^* - x_i)^2 \end{aligned} \tag{3.3}$$

Hence, $||z - x||_2 < ||e^* - x||_2$, which contradicts the fact that e^* is the solution of Equation 3.2. Thus, we have $P_{e^*} = \{e^*\}$, leading to a unique solution e^* for the l_0 minimization problem. □

There is thus no possibility of making the vector $e^* - x$ sparse without changing the predicted class. However, when considering $\tilde{e}$, there is greater chance (that is to say, strictly greater than zero when considering e^*) of having a potential solution in the set $P_{\tilde{e}}$ distinct from $\tilde{e}$. An illustration of this idea in a simple 2-dimensional setting is shown in Figure 3.2. The same situation as for Figure 3.1 is represented. Additionally, we note that it is not possible to project the analytical solution e^* of the l_2 counterfactual problem in order to guarantee a sparse explanation. However, because

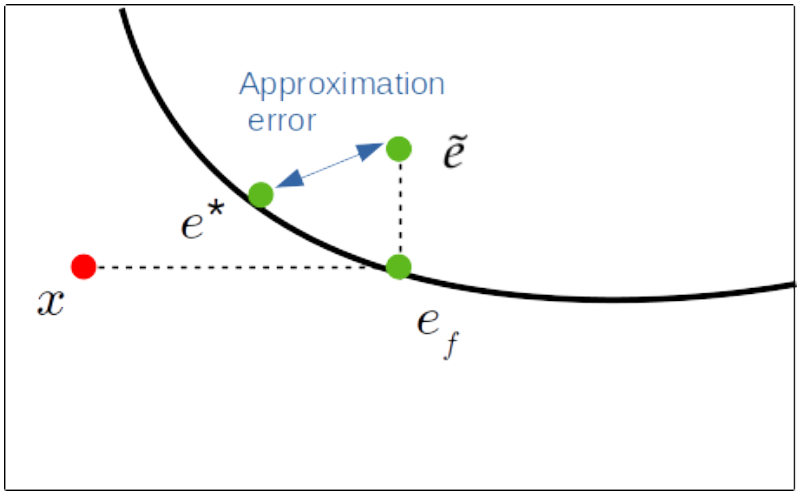


Figure 3.2: Illustration on the discussion on approximation errors for the case of a binary classifier. Because $\tilde{e}$ is returned instead of e^*, a solution e_f to the l_0-minimization problem can be found.

it is located further away, it is possible to project the solution found numerically $\tilde{e}$, leading to the final counterfactual e_f.

Obviously, being able to accurately solve the l_2 minimization problem would require to design a new heuristic to favor explanation sparsity afterwards. Another possibility would be to extend the proposed formalization by explicitly searching for a sub-optimal l_2 solution, that is to say located "behind" the decision boundary of f. This would favor the feasibility of the orthogonal projections. However, it would also be equivalent to forcing the formulation of an explicit trade-off between sparsity and locality, which, as stated before, is not desirable.

Therefore, the final objective we propose to focus on in this chapter can be written as the following:

$$\begin{aligned} e_f = \underset{e \in P_{\tilde{e}}}{\arg\min} \quad & ||x - e||_0 \\ \text{subject to} \quad & f(e) \neq f(x) \\ \text{with} \quad & \tilde{e} \approx e^* = \underset{z \in \mathcal{X}}{\arg\min} \{||x - z||_2 \quad \text{s.t.} \quad f(z) \neq f(x)\} \end{aligned} \tag{3.4}$$

The solution e_f of this problem ensures that the counterfactual explanation vector $e_f - x$ is both local and sparse. In Section 3.2.2, a heuristic is proposed to minimize this objective.

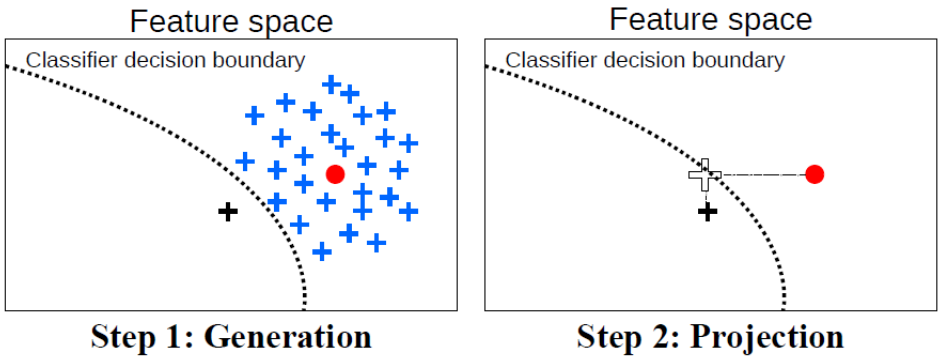


Figure 3.3: Illustration of *Growing Spheres*: The black dashed line represents the unknown classifier decision boundary, while the red circle represents the observation x whose prediction is to be interpreted. The plus signs are the generated instances (blue (resp. black) for instances classified similarly (resp. differently). Here, the only instance represented that is predicted to belong to another class than x is $\tilde{e}$. The white plus is the final counterfactual example e_f used to generate explanations.

3.2.2 | The *Growing Spheres* Algorithm

In this section, in order to solve sequentially the two presented minimization programs we propose the *Growing Spheres* algorithm. This procedure is a two-step heuristic approach that returns e_f, the final solution of the problem given in Equation 3.4. These two steps, namely the generation step, which solves the l_2 minimization problem (presented in Equation 3.2), and the projection step, which solves the l_0 minimization problem, are described in turn in the new two sections and are illustrated in Figure 3.3.

3.2.2.1 | Generation Step

The generation step aims at solving the program given in Equation 3.2. As a reminder, the considered context is fully agnostic. Therefore, the minimization of the considered objective, detailed in Algorithm 2, is conducted without relying on any existing data. Thus, for the considered observation x, there is no information about the direction in which the closest classifier boundary might be. A greedy approach to find the closest counterfactual example relies on exploring the input space $\mathcal{X}$ by generating instances in all possible directions, further and further until the decision boundary of the classifier is crossed. More precisely, the algorithm generates observations in the feature space in l_2-spherical layers around x until an instance predicted to belong to a different class than $f(x)$ is found. Formally, given two positive numbers a_0 and a_1,

Algorithm 1 Hyperspherical Layer Generation (HLG)

Require: x, the center of the hyperspherical layer
Require: a_0 and a_1 the bounds delimiting $\mathcal{SL}(x, a_0, a_1)$
Require: n, number of desired points
Output: $Z = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{SL}(x, a_0, a_1))$
1: $Y = \{y_i\}_{i \leq n} \sim \mathcal{N}(0, 1)$
2: $Y \leftarrow \frac{Y}{||Y||_2}$
3: $U = \{u_i\}_{i \leq n} \sim \mathcal{U}([0, 1])$
4: $R \leftarrow a_0 + a_1 U^{1/dim(\mathcal{X})}$
5: $Z \leftarrow R^T Y + x$
6: **return** Z

we define the (a_0, a_1)-spherical layer $\mathcal{SL}$ around x as:

$$\mathcal{SL}(x, a_0, a_1) = \{z \in \mathcal{X} \ : \ a_0 \leq ||x - z||_2 \leq a_1\}$$

To generate observations following a uniform distribution over these subspaces, we propose a generalization of the algorithm proposed in Muller (1959). This algorithm aims at efficiently generating instances distributed uniformly over the surface of the unit sphere by normalizing $dim(\mathcal{X})$ normal distributions. In its adaptation, provided for information in Algorithm 1 and called Hyperspherical Layer Generation (HLG), the radial distance of these observations is rescaled (line 3 to 5 of Algorithm 1): as a result, we obtain observations that are uniformly distributed over $\mathcal{SL}(x, a_0, a_1)$. Using this algorithm allows for much better computational performance than a naive method such as generating instances uniformly in a hypercube and rejecting the instances that are not in $\mathcal{SL}$.

As shown in Algorithm 2, the HLG algorithm is then used iteratively, for increasing values of the parameters a_0 and a_1 to generate instances in increasing hyperspherical layers, as detailed in the iteration step described below. The width of these layers is set to be constant, and controlled by a hyperparameter $\eta = a_1 - a_0$ set by the user. The algorithm also relies on a second hyperparameter n, which is the number of instances generated in each hyperspherical layer.

Initialization. The initial step of the generation algorithm consists in generating uniformly n observations in the l_2-ball of radius η and center x, with n and η hyperparameters of the algorithm: this ball also corresponds to $\mathcal{SL}(x, 0, \eta)$ (line 1 of Algorithm 2). It is important to note that in a context where information about some existing instances would be available (i.e. in a non-fully agnostic situation, unlike the context of this chapter), the η parameter could be easily set using the distance between x

Algorithm 2 Growing Spheres Generation

Require: $f : \mathcal{X} \to \{-1; 1\}$ a binary classifier
Require: $x \in \mathcal{X}$ an observation to be interpreted
Require: Hyperparameters: η, w, n
Output: $\tilde{e} \approx e^* = \underset{e \in \mathcal{X} \text{ s.t. } f(e) \neq f(x)}{\arg\min} ||x - e||_2$

1: Generate $Z = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{SL}(x, 0, \eta))$ using HGL
2: **while** $\exists\, e \in Z$ s.t. $f(e) \neq f(x)$ **do**
3: $\quad \eta \leftarrow \eta / 2$
4: $\quad$ Generate $Z = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{SL}(x, 0, \eta))$ using HGL
5: **end while**
6: Set $a_0 = \eta$, $a_1 = \eta + w$
7: **while** $\nexists\, e \in Z$ s.t. $f(e) \neq f(x)$ **do**
8: $\quad$ Generate $Z = \{z_i\}_{i \leq n} \sim \mathcal{U}(\mathcal{SL}(x, a_0, a_1))$ using HGL
9: $\quad a_0 \leftarrow a_1$
10: $\quad a_1 \leftarrow a_0 + w$
11: **end while**
12: $\tilde{e} = \underset{e \in Z \text{ s.t. } f(e) \neq f(x)}{\arg\min} ||x - e||_2$
13: **return** $\tilde{e}$

and the closest of these instances z which would happen to satisfy $f(x) \neq f(z)$. Indeed, this distance would provide an upper bound for the searched solution e^*, since the closest touchpoint searched would necessarily be closer than this instance z (by definition of e^*).

These instances are then labelled using f. Two situations can arise: if at least one of the generated instances z satisfies $f(z) \neq f(x)$, a local search is performed to make sure that the algorithm did not miss the closest decision boundary. This is done by updating the value of the initial radius: $\eta \leftarrow \eta$ / 2 and repeating the initial step until no instance satisfying $f(z) \neq f(x)$ is found in the initial ball $SL(x, 0, \eta)$ (lines 2 to 5 of Algorithm 2) anymore. However, if no instance from the other class is found within $SL(x, 0, \eta)$, this means that the explored space is too narrow to include some section of the decision boundary of f (this supposes, of course, that n is not set to an absurdly low value). In such case, the explored area is expanded, as described in the next paragraph.

Iterations. To detect the closest touchpoint of the decision border, the explored area is widened by updating a_0 and a_1: a_0 is set to η (the initial radius value), and a_1 to $a_0 + w$, with w a hyperparameter describing the width of the spherical layers. New instances are thus generated over $\mathcal{SL}(x, a_0, a_1)$ and labelled using f. This process is re-

Algorithm 3 Growing Spheres Projection

Require: $f : \mathcal{X} \to \{-1; 1\}$ a binary classifier
Require: $x \in \mathcal{X}$ the observation to be interpreted
Require: $\tilde{e} \in \mathcal{X}$ such that $f(\tilde{e}) \neq f(x)$, the solution of Algorithm 2
1: Set $e' = \tilde{e}$
2: Set $e_f = \tilde{e}$
3: Set $\mathcal{I} = \emptyset$
4: **while** $f(e') \neq f(x)$ **do**
5: $\quad e_f = e'$
6: $\quad i = \underset{j \in [1:dim(\mathcal{X})],\, e'_j \neq x_j, j \notin \mathcal{I}}{\arg\min} |e'_j - x_j|$
7: $\quad$ Update $e'_i = x_i$
8: $\quad \mathcal{I} \leftarrow \mathcal{I} \bigcup \{i\}$
9: **end while**
10: **return** e_f

peated by setting a_0 to the former value taken by a_1 until the first instance $f(z) \neq f(x)$ is met. If several of these instances are found, the l_2 closest one is returned. These steps are detailed in lines 6 to 11 of Algorithm 2 and illustrated in a 2-dimensional setting in Figure 3.3, page 52: the decision boundary of the black-box classifier (black dashed line) is met after instances have been generated in hyperspherical layers (blue crosses).

In the end, the Generation algorithm returns $\tilde{e}$, an approximation of the solution e^* of the l_2 minimization program of Equation 3.2 (represented by the black cross in Figure 3.3). Once this is done, we focus on making the associated explanation as easy to understand as possible, through the projection step described in the next section.

3.2.2.2 | Projection Step

In this second step, we focus on making the difference vector $\tilde{e} - x$ as sparse as possible to minimize the objective of Equation 3.4. To do so, as discussed in Section 3.2.1.2, we aim at performing the highest possible number of projections of $\tilde{e}$ on the hyperplanes $\{\mathcal{H}_i\}_{i \leq dim(\mathcal{X})}$ to reduce the number of features used when moving from x to $\tilde{e}$. Since finding the exact minimum by trying out all the possibilities is expensive (we showed in Section 3.2.1.2 that: $card(P_{\tilde{e}}) = 2^{dim(\mathcal{X})}$ possibilities), we consider a heuristic approach based on the idea that the coordinates of the vector $\tilde{e} - x$ with the smallest absolute values might be less relevant locally regarding the classifier decision boundary and should thus be the first ones to be ignored. The proposed algorithm thus tries to align as many coordinates of $\tilde{e}$ with x as possible, as

long as the predicted class does not change. It stops when no projection of $\tilde{e}$ can be performed anymore. The proposed Projection algorithm is detailed in Algorithm 3.

The final explanation provided to interpret the observation x and its associated prediction is the vector $x - e_f$, with e_f the final solution identified by the algorithms. This step is illustrated in the left image of Figure 3.3, page 52: e_f (represented with the big white cross) is obtained through projections of $\tilde{e}$ (black cross).

3.2.3 | *Growing Spheres* Hyperparameters: n, η and w

The *Growing Spheres* algorithm relies on three hyperparameters that highly impact the performance of the algorithm and its outputs. While the hyperparameter η (radius of the initial hypersphere) mostly influences the computation time of the algorithm rather than its results, this is not the case for the hyperparameters n and w. Indeed, at each iteration of the Generation step, n instances are drawn following a uniform distribution over the hyperspherical layers $\mathcal{SL}$, defined by its width w. Yet, the approximation error considered to define $\tilde{e}$ with respect to the analytical solution e^* directly depends on the density of the instances drawn in the space, i.e. the volume of $\mathcal{SL}$ divided by the number of generated instances n. However, because this value is less transparent for the user, we choose to consider w and n separately instead of defining the density as a hyperparameter. In practice, we choose an arbitrary value for w (that needs to be small for the considered problem) and calibrate *Growing Spheres* using n only. An obvious consequence of this choice is that the density of the generated instances decreases exponentially at each iteration, meaning that the average approximation error increases with the number of iteration. However, this assumption seems more reasonable than requiring either n to increase exponentially or w to decrease exponentially for convergence reasons.

Nevertheless, considering the highly stochastic nature of the *Growing Spheres* algorithm, although these parameters influence the produced results, their exact value is not crucial.

3.2.4 | Adapting the Algorithm to Multiclass Classification

This chapter, and in particular the proposed *Growing Spheres* algorithm, focuses on binary classification. However, adapting the considered problem to the context of multiclass classification is not particularly challenging. When there are more than two classes, two possibilities can arise: counterfactual explanations can either be *tar-*

geted or *untargeted*. Considering a multiclass classifier $f : \mathcal{X} \to \mathcal{Y}$, the following modifications ensue:

Untargeted counterfactuals. Untargeted counterfactuals aim at generating the closest instance classified differently. The predicted class itself of the final counterfactual example is thus not important. Applying this to *Growing Spheres* is thus easy, since the algorithm does not need to be modified: the closest instance $e_f \in \mathcal{X}$ satisfying $f(e_f) \neq f(x)$ is still thus to be returned.

Targeted counterfactuals. Targeted counterfactuals aim at generating the closest instance predicted to belong to a specific class $l \in \mathcal{Y}$. The stopping criterion of the generation step thus becomes $f(\tilde{e}) = l$. Similarly, the condition in the projection step becomes $f(e') = l$ (instead of line 4 of Algorithm 3) to ensure that $f(\text{proj}_H(\tilde{e})) = l$.

3.3 | Experimental Validation

In this section, experiments are performed to show the efficiency of the *Growing Spheres* algorithm. After describing the considered datasets and classifiers in Section 3.3.1, illustrative examples are shown to give intuitions about both the behavior and the outputs of the *Growing Spheres* algorithm in Section 3.3.2. Finally, in Section 3.3.3, quantitative results about the sparsity of the generated explanations are given, and a study on the impact of the hyperparameters is conducted.

3.3.1 | Experimental Protocol

This section describes the datasets and protocol considered in the experiments, which aim at validating the efficiency of the *Growing Spheres* approach. Figures summarizing this information can be found in Table 3.1.

Datasets. The datasets considered for these experiments incude 4 datasets: Boston Housing (Harrison and Rubinfeld, 1978), Propublica Recidivism (Larson et al., 2016), News Popularity (Fernandes et al., 2015) and German Credit from the UCI repository (Dua and Graff, 2017). These datasets have been chosen because of their easily understandable features and are commonly featured in the interpretability (and fairness) literature. Both the Boston Housing and News popularity datasets are originally regression tasks that are transformed into classification problems. This is conducted

Dataset	**# of instances**	$dim(\mathcal{X})$	**Classifier**	Accuracy	η	w	n
Half-moons	2000	2	SVM	1.00	0.1	0.01	200
Boston	506	13	SVM	0.85	0.2	0.02	2000
Credit	1000	20	RF	0.74	0.5	0.1	5000
Recidivism	10000	18	RF	0.5	0.68	0.1	10000
News	4954	58	RF	0.66	0.5	0.1	10000
MNIST	70000	784	SVM	0.97	1.5	0.1	10000

Table 3.1: Characteristics of the datasets, classifiers and *Growing Spheres* parameter values considered in the experiments.

by using the median value of the targeted variable as a threshold to define two balanced classes. Unordered categorical data are dropped out, as they remain out of the scope of the study. Additionally, the remaining numerical attributes are standardized (same mean and standard deviation). In addition, the MNIST dataset (LeCun et al., 1998) is considered for the discussion in Section 3.4, page 64.

In order to give intuitions about the proposed approach, a 2-dimensional toy dataset (half-moons) is also considered. This dataset is relevant to the whole thesis, as it also appears in Chapters 4 and 5 for the same reasons. The half moons dataset is generated using a scikit-learn (Pedregosa et al., 2011) procedure[1]. Instances are generated in 2 dimensions, distributed in interlaced moon crescents. Each crescent (half-moon) is associated to a label. Beside the number of instances to generate, a *noise* parameter is available to control how separated the two crescents are. In this thesis, the value of this parameter is set according to the purpose of the illustrative experiment. For instance, in Section 3.3.2, its value is set to 0.05, leading to well-separated classes (see Figure 3.4, page 60). On the other hand, in Section 5.2.1, page 106, this parameter is set to 0.4 to create a more complex decision boundary for f (see Figure 5.1, page 106).

Protocol. For each considered dataset, a train-test split of the data is performed with a $90\% - 10\%$ proportion, and a binary classifier is trained. As the considered context is post-hoc, the choice of the classifier does not matter. The considered algorithms are nevertheless specified in Table 3.1. The classifiers used were selected for their predictive accuracy on the test set and their parameters chosen through cross-validation. More details about the obtained preprocessed datasets and classifiers can be found in Table 3.1.

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

The test set is then used to run the experiments: for each instance x it contains, the GS algorithm is run to generate a counterfactual e_f. As discussed earlier, the values of the GS hyperparameters may impact the results. The values used are chosen to ensure a reasonable computation time and are given in Table 3.1.

Quantitative criteria. In order to assess the quality of the generated counterfactual explanations, we look at the two criteria mentioned in Section 3.1.2: the Euclidean norm $||e_f - x||_2$ and the sparsity of the explanation vector $||e_f - x||_0$. In order to give insights about how the method behaves on a whole dataset, average values of these two criteria over each test set are also calculated.

3.3.2 | Illustrative Results

This section illustrates the behavior of *Growing Spheres* (GS) and its results. First, the efficiency of the proposed approach in generating local counterfactual explanations is illustrated in a 2-dimensional setting in Section 3.3.2.1. Then, an example of output returned by *Growing Spheres* for the News dataset is presented and discussed in Section 3.3.2.2.

3.3.2.1 | Locality of the Generated Explanations

This first illustrative experiment aims at showing that *Growing Spheres* does generate local explanations. In particular, we verify that the generated counterfactual examples lie on the decision boundary of the black-box classifier. Figure 3.4 shows the results obtained with GS on the half-moons dataset. A Support Vector Machine classifier is run on 90% of the generated instances, and its decision regions are represented by the red and blue areas. The white area represents between the blue and red regions thus represents the decision boundary of f. The accuracy of the classifier is 100%. For each considered instance x of the test set (only one of the classes is shown in the figure, as 32 red instances), counterfactual explanations e_f are generated using *Growing Spheres* with parameters $\eta = 0.1, w = 0.01, n = 200$. The obtained counterfactual examples are represented by the green instances.

We observe that these counterfactual examples lie on the decision boundary of the classifier, proving that in this example, the method achieves generating local explanations. However, it should be noted that in this 2-dimensional setting, only 2 out of the 32 explanations can be made sparse by projecting $\tilde{e}$.

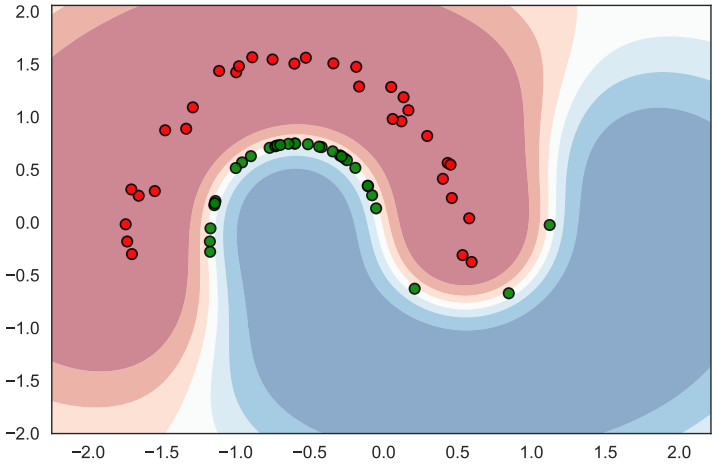


Figure 3.4: Scatterplot of one class of the half-moons dataset (red instances) and of the corresponding counterfactual explanations generated using *Growing Spheres* (green instances). The red and blue areas represent the decision regions of the black-box classifier.

3.3.2.2 | Example of *Growing Spheres* Output

In this section, we provide two output examples highlighting the usefulness of the explanations generated using *Growing Spheres*.

The News Popularity dataset (Fernandes et al., 2015) contains articles from the news website Mashable. The 58 features encode information about the format and content of the articles, such as the number of words in the title, a measure of the content subjectivity or the popularity of the used keywords, etc. The binary classification task aims at predicting whether an article is popular or not, where popularity is defined as having been shared more than 1400 times.

We consider two observations from the test set: Article A1, entitled *Apple's App Store Passes 40 Billion Downloads*[2], that is predicted to be not popular by the considered classifier, and Article A2, entitled *Twitter Reactions to Zimmerman Verdict Run Hot and Cold*[3], predicted to be popular. The explanation vectors given for each article by *Growing Spheres* are shown in Table 3.2.

For article A1, its associated prediction can be explained by two characteristics: to be predicted as popular, the maximum number of shares associated to the keywords of the article should be increased by 4569 in average (this feature is called *Avg. keyword (max. shares)* in Table 3.2); moreover, the number of shares of the least popular article among the ones referenced would need to be increased by 788 (*Min. shares of*

[2]https://mashable.com/2013/01/07/apple-40-billion-app-downloads/
[3]https://mashable.com/2013/07/14/twitter-george-zimmerman/

Article/class	Feature	Move
A1 Not Popular	Avg. keyword (max. shares)	+4569
	Min. shares of referenced articles in Mashable	+788
A2 Popular	Average shares of referenced articles in Mashable	−502
	Average polarity (negative words)	+0.0014

Table 3.2: Output example of *Growing Spheres*

referenced articles in Mashable in Table 3.2). This means that according to the classifier, using more popular keywords and citing more popular articles would lead to more popularity.

As for article A2, the average number of shares of the Mashable articles cited would need to decrease by 502. Additionally, the average polarity score of the negative words used would need to increase by 0.0014. In other words, article A2 would be predicted to be not popular by the considered classifier if it was written in a less neutral tone and if the references cited were less popular themselves.

These examples illustrate the usefulness of counterfactual explanations and how they can be leveraged. Using *Growing Spheres*, it is possible to precisely understand what changes need to be applied to alter the model prediction, without any actual understanding or knowledge about the model whatsoever.

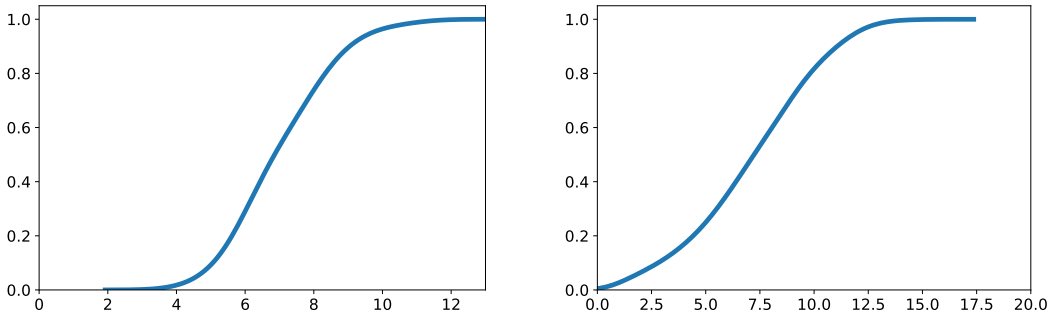
3.3.3 | Quantitative Results

In this Section, we conduct experiments over several datasets to quantify and discuss the efficiency of the *Growing Spheres* procedure. First, the sparsity of the explanations is studied in Section 3.3.3.1. Then, the feasability of the proposed counterfactual program, discussed in Section 3.2.1, page 45, is studied in Section 3.3.3.2. In particular, a trade-off between locality, sparsity and computation time is highlighted.

3.3.3.1 | Measuring the Sparsity of Explanations

Using the experimental protocol presented in the previous section, we use *Growing Spheres* to generate explanations and measure their sparsity. Figure 3.5 shows the (smoothed) cumulative distribution of the l_0 norm of the final explanation vector $x - e_f$ for all test instances, for 4 datasets. The X-axis is defined so that it is limited by the total number of attributes of the dataset (as reported in Table 3.1, page 58).

We observe that for instance the maximum value taken by the curve on the Recidivism dataset (bottom left) is reached when the X-axis value is 7, meaning that each



(a) Boston Housing

(b) German Credit

(c) Recidivism

(d) Online News

Figure 3.5: Cumulative distribution curves of the l_0 norm of the final explanations obtained for four datasets. X-axis: number of attributes; Y-axis: percentage of associated instances. Reading: "on the Boston Housing Dataset 40% of the observations of the test set have explanations that use 6 features or less."

observation in the test set only needs to change at most 7 out of the 18 attribute values available to cross the decision boundary of the classifier. It is important to note that this does not mean that only 7 attributes are required to explain all the observations, as each explanation may use different features. Beside the considered classifier and dataset, the steepness of the observed curves is influenced by the values chosen for the hyperparameters of *Growing Spheres*, since they may impact the sparsity of the explanations.

Nevertheless, this shows that the proposed approach does achieve sparsity in order to make explanations more understandable.

3.3.3.2 | Study of the Feasability of the Proposed Program

We now propose to assess the claimed feasibility (see Section 3.2.1, page 45) of the proposed counterfactual program. As a reminder, generating a sparse explanation in

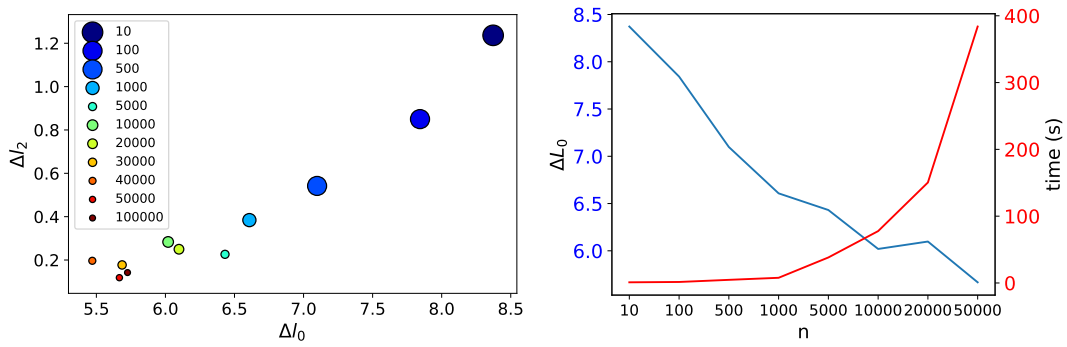


Figure 3.6: Left: scatter plot of several runs of the *Growing Spheres* procedure on the Boston dataset and for several values of n, on the plane defined by Δl_0 and Δl_2. Right: Δl_0 and computation time as functions of n, for several runs of the *Growing Spheres* procedure on the Boston dataset.

the proposed context is only made possible because of approximation errors induced in the l_2 minimization program. In Section 3.2.3, page 56, we discussed how this approximation error is intrinsically linked to the number n of generated instances in the Generation Step of the *Growing Spheres* algorithm. Therefore, in this experiment, we study the extent to which increasing the value of this hyperparameter impacts sparsity of the generated solution.

For this purpose, we introduce the l_0-gain defined by:

$$\Delta l_0 = ||x - \tilde{e}||_0 - ||x - e_f||_0$$

This score evaluates how many projections of $\tilde{e}$ could be performed in the projection step. Since the generation step almost never returns a sparse solution, $||x - \tilde{e}||_0$ is almost always equal to $dim(\mathcal{X})$. In particular, we are interested in the average value of Δl_0 over multiple instances.

Similarly, we also define the l_2 gain Δl_2:

$$\Delta l_2 = ||x - \tilde{e}||_2 - ||x - e_f||_2$$

The left image of Figure 3.6 shows a scatter plot of the average values over the test set of the Boston Housing dataset of the l_0 (X-axis) and l_2 gains (Y-axis) for several values of n (colors) while the other hyperparameters are left untouched. The size of the circles represent the standard deviation of the observed values.

A first observation is that higher values of n seem associated to some extent to smaller values of Δl_0. This can be explained by the fact that increasing n leads to a better approximation of e^* by $\tilde{e}$, which means that $\tilde{e}$ is located closer to the decision

boundary of f, leading to a lesser chance of finding projections that are predicted to belong to the same class.

The same observation can be made for Δl_2, which is to be expected since with our assumptions, reducing the l_0 norm induces an automatic reduction of the l_2 norm. However, the average l_2 and l_0 gains seem to be stagnating after reaching a certain value (here starting from $n = 10000$). This can be interpreted as the fact that at some point, the resulting approximation errors cannot easily be suppressed by increasing the value of n.

Furthermore, increasing the value of n is also obviously associated to an increase in algorithmic complexity. The right graph of Figure 3.6 illustrates this idea, and displays the same average l_0-gain depending on the value chosen for n, in addition to the running time of the procedure. Increasing the value of n is thus costly, leading to little gain in terms of l_2 and l_0.

These experiments show that the proposed approach manages to generate useful explanations that are both close and sparse. Although the results have been shown only for the Boston Housing Dataset, similar observations have been made on the other considered datasets.

3.4 | Discussion: Out-of-Distribution Counterfactuals

The previous experiments show the efficiency of the proposed procedure for the generation of post-hoc counterfactual explanations. However, as mentioned in Chapter 2, page 9, the post-hoc paradigm may also create potential issues that may hurt the generation of relevant explanations. In this section, we propose a discussion about one of these limitations of the generation of counterfactual explanations in a fully agnostic context: the risk of generating explanations that lie out of the ditribution of the ground-truth data. The discussion around this issue constitutes one of the contributions of this thesis.

As discussed in Section 2.1.3, page 17, generating an explanation method with agnosticity assumptions guarantees a larger freedom in terms of usage: the same approach can be used whatever the classifier or data is, and updating the model does not in theory require any modification on the side of the explainer. However, this strength can also be a weakness since making no assumption means having no knowledge and therefore potentially creating useless explanations. In particular, when building the explanation through queries to the black-box in the post-hoc context, there is no pos-

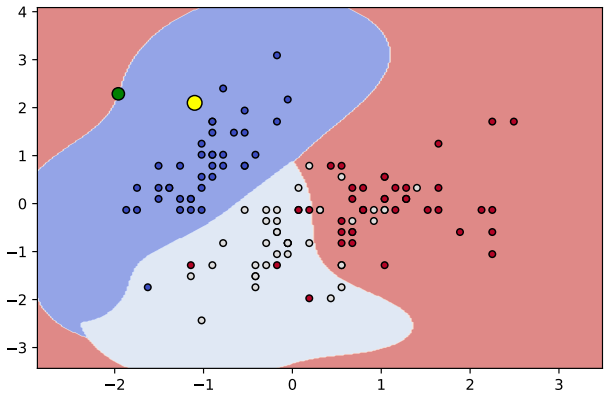


Figure 3.7: Training instances of a 2-dimensional iris dataset (blue, light blue and red instances), used to train a SVM classifier (blue, light blue and red areas represent its decision regions). The output of *Growing Spheres* is used on an instance x (yellow instance) is representerd by the green instance.

sibility of making a distinction between a prediction that is made because of some ground-truth knowledge learned by the classifier and one that would purely rely on the generalization capabilities of the classifier.

The risk of generating out-of-distribution counterfactuals. This general issue is in particular the focus of Chapter 4, in which we study the connectedness of counterfactual examples to ground-truth data. In this section, we focus on another issue: the risk of generating instances that lie out of the distribution of the ground-truth data used to train the classifier. This risk exists due to the agnosticity assumptions made, and raises the problem of generating potentially useless explanations.

Generating counterfactuals that lie out of the distribution of the ground-truth data can frequently happen even in low dimension, when the classifier overfits the training data for instance. Figure 3.7 shows the training data of a 2-dimensional version of the iris dataset (3 classes), on which a SVM classifier with a RBF kernel and parameters $C = 1.0$ and $\gamma = 100$ is trained. The resulting classifier obviously overfits the training data, as shown by the shape of the decision boundary in upper left corner. When trying to generate an explanation for an instance that is located on the edge of the ground-truth distribution (yellow instance), the counterfactual built by *Growing Spheres* lies in an empty region (green instance), leading to an explanation of little interest.

A problem of feature space definition. Another way of seeing this problem is to look at feature representations. In particular, there can be cases where the feature

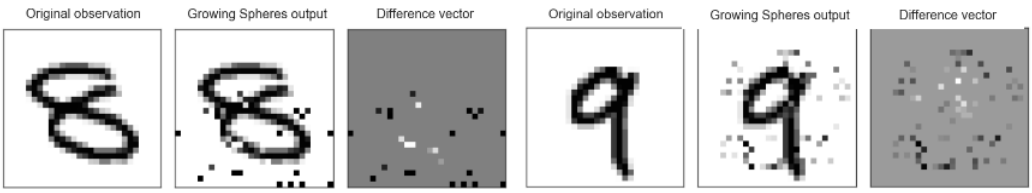


Figure 3.8: *Growing Spheres* output examples. From left to right: example of the original instance x, counterfactual explanation found e_f, explanation vector $e_f - x$. First for an 8, then for a 9.

space relevant to a user in the context of a specific classification task is not the one given as input to the classifier. Such situations can arise for instance when a user does not actually understand the impact of an input feature, or when the features that makes sense to him/her are too complex to be directly extracted (e.g. positivity/negativity of a text) and given as input. Another example is image classification, as it is well known that users do not use the pixel representation that is fed to the models to identify the label of an image.

To illustrate this phenomenom, we apply the *Growing Spheres* algorithm to the MNIST dataset (LeCun et al., 1998) that contains images of handwritten digits, encoded in the form of images of 28 by 28 pixels. For the sake of simplification, we train a binary classifier to identify the digits 8 and 9. The classifier used (a SVM classifier, arbitrarily chosen) and the *Growing Spheres* algorithm are used to generate explanations (see Table 3.1, page 58 for the parameters values). Just like the instances of the dataset, the counterfactual examples e_f generated with *Growing Spheres* as well as the explanation vectors $e_f - x$ are also described as pixels. Therefore, they can also be visualized as images. Figure 3.8 shows two examples correctly classified as 8 and 9, as well as their resulting explanations (counterfactual example e_f and difference vector $e_f - x$). While a naive transformation to turn a digit 8 into a digit 9 would be to focus on the bottom-left part of the digit to "close" or "open" the bottom loop, the outputs of *Growing Spheres* do not concur. In fact, the first thing we observe is that the counterfactual explanations e_f found by the algorithm in both cases are not even proper digits at all but rather noisy versions of the original image x. In fact, this makes them more likened to adversarial examples (presented in Section 2.3.3, page 36), although the perturbation is here visible. This can be explained by the fact that pixels involved in the required modifications to change the predicted class are located all around the picture, rather than around the digit. Besides, they display more variations of gray than actual digits where colors are more contrasted, making these modifications irrelevant for a human in the considered context.

Although seemingly indesirable, these observations are consistent with the principle of the proposed approach: the goal of *Growing Spheres*, and more generally post-hoc approaches, is to explain the classifier decision, not the reality it is approximating. In this case, the fact that the classifier apparently considers these pixels to be influential for the classification of these digits could be an evidence of the learned boundary inaccuracy compared to the real world.

Such explanations raise questions however regarding their actionability and tangibility, which can be desirable in some contexts. This is even more problematic in the case of counterfactuals since these characteristics are supposed to be some of the strong arguments for their use (as explained in Sections 2.1.1, page 11 and 2.3.1.2, page 31). Defining the locality of explanations in the post-hoc context may thus lead to potential issues in terms of interpretability.

3.5 | Conclusion

This chapter proposes to answer the problem of generating a local explanation in the post-hoc context by solving a counterfactual problem. Besides model- and data-agnosticity, additional constraints are imposed such as not having access to some classification confidence score. To solve this problem, an approach is proposed that takes advantage of approximation errors to circumvent the necessity of formulating a trade-off between l_2 and l_0 norm. This proposition is implemented in *Growing Spheres*, whose efficiency is shown through illustrative and quantitative results.

Several improvements and extensions could be envisaged. The results obtained with *Growing Spheres* depend on the values chosen for the hyperparameters η, w and n. The impact of increasing n, studied in Section 4.2.4, suggests that its role remains directly associated to the one of w. In our experiments, the values chosen for these hyperparameters are selected among an arbitrarily chosen range of values, and by taking into account the presented tradeoff between approximation accuracy and computing performance. Defining a better way to select these hyperparameters would lead to more clarity for the proposed approach as well as transparency of the results.

A more crucial extension for the proposed approach should focus on the formalization of its objective. By relying on approximation errors to generate a sparse solution, the question of what would be done in the case where the approximation of the solution of the l_2 minimization program is better remains. Sparsity of explana-

tions would then be impossible to bring, and a different heuristic would need to be envisaged.

4

The Risk of Unjustified Explanations

In this chapter, we study a second issue of local post-hoc explanations: the risk of generating explanations that cannot be directly associated to any ground-truth knowledge. Despite concerning local post-hoc explainers in general, this problem is studied from the perspective of counterfactual explanations. In this chapter, we discuss that a desirable property for counterfactual explanations is that they can be *justified*, which we propose to define as being connected through a continuous path to a ground-truth instance from the same class. However, we show that in the post-hoc context, explainers do not have the capability to avoid unjustified classification regions that may be created by the classifier. We therefore propose a procedure to assess the risk of having such undesirable counterfactual examples disturb the generation of counterfactual explanations. Additionally, we design a second procedure to exhibit that when facing this risk, state-of-the-art post-hoc counterfactual approaches may generate explanations that are unjustified. Despite sharing the same name, the notion of *justification* we propose is different from the one proposed by Biran and Cotton (2019). In the context of their work, a justification for a prediction is an insight given to a user illustrating why a certain prediction can be trusted (e.g. classification confidence score): it is not used to characterize an explanation.

In Section 4.1, we motivate and define the notion of explanation justification, central to this chapter. In Section 4.2, a procedure called *Local Risk Assessment* (LRA) is proposed to assess the risk of generating unjustified counterfactual explanations. This risk is exhibited in Section 4.3, and its link with the notion of classifier overfitting is analyzed. Finally, a second procedure, called *Vulnerability Evaluation* (VE) is proposed in Section 4.4 to assess the vulnerability of post-hoc counterfactual approaches when confronted to the risk of unjustification.

Most of the work presented in this chapter was the subject of two papers: *The*

Dangers of Post-hoc Interpretability: Unjustified Counterfactual Explanations, published at the IJCAI 2019 conference (Laugel et al., 2019c), and *Unjustified Classification Regions and Counterfactual Explanations in Machine Learning*, published at the ECML-PKDD 2019 (Laugel et al., 2019b) conference.

4.1 | Ground-truth Justification

This section presents motivations and formalization for the notion of ground-truth justification, central to this chapter. This notion is a desirable property for explanations, defined upstream of the assumptions made to generate them. In particular, this definition is not placed in the post-hoc context usually considered in the thesis. The restriction to the post-hoc framework and to the constraints it imposes is discussed in the following sections, and is detailed in the motivations below.

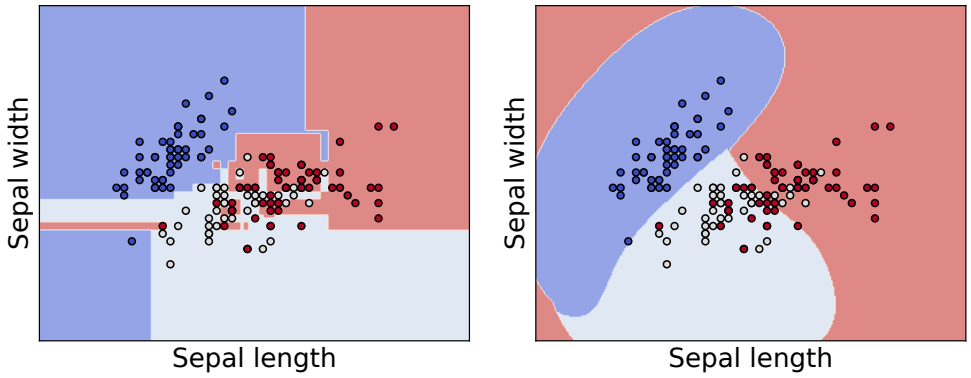
First, we discuss the goal of the proposed notion in Section 4.1.1. Then, its definition is proposed in Section 4.1.2, and its implementation addressed in Section 4.1.3.

4.1.1 | Ground-truth Based Decision vs. Artifact

In the previous chapter, in Section 3.4, page 64, the risk of generating counterfactual explanations that lie out of the distribution of the training data is tackled. The existence of this risk is attributed to the considered post-hoc context and its agnosticity assumptions. In this chapter, another issue, also raised by the post-hoc paradigm, is studied. This issue, called *unjustification risk*, also relies on an assessment of the link between a post-hoc counterfactual explanation and ground-truth data. However, instead of assessing whether explanations lie in the distribution of the ground-truth data distribution, we focus on identifying explanations that rely on *questionable decisions* made by the black-box classifier.

Two types of classifier decisions. We call *artifacts* these questionable decisions, as opposed to decisions that can be directly associated to some ground-truth knowledge. In the context of this chapter, ground-truth knowledge is represented by training data. Such artifacts can be created in particular because of a lack of robustness of the model, or because it is forced to make a prediction for an observation in an area that is not covered by the training set.

Figure 4.1 shows an illustration of how easily such situations can arise in a trivial problem. A 2D version of the iris dataset is used to train two classifiers, a random forest and a SVM classifier with RBF kernel. The right image illustrates the scenario



(a) RF with 3 estimators

(b) SVM with RBF kernel

Figure 4.1: Two classifiers have been trained on 80% of a 2D version of the iris dataset and have the same accuracy over the test set, 0.78. Left picture: because of its low robustness, the random forest classifier makes questionable generalizations (e.g. small red square in the dark blue region). Right picture: the support vector machine classifier makes questionable decisions in regions far away from the training data (red area in the top left corner).

previously considered in Section 3.4, page 64. The decision regions of these classifiers are represented by the red, light blue and dark blue regions (three classes). In both cases, some regions can be found where the classifier makes questionable improvisations, i.e. areas which it has no information about (no training data). In the left image, this occurs because the complexity of the classifier is not adapted to the problem, leading to insufficient generalization capabilities: this is for instance the case of the small isolated red square in the dark blue region. In the right image, because the classifier overfits the training data, questionable decisions appear in empty regions: this is for instance the case of the red regions at the top right and bottom right corners of the image. In this situation, the problem we propose to tackle in this chapter can therefore overlap with the one of explanations that rely on out-of-distributions examples, studied in Section 3.4, page 64. Nevertheless, the left image shows that it is not restricted to studying this generation of out-of-distribution explanations.

Artifacts are harmful for interpretability. As presented in Section 2.3.1.3, page 32 and in Chapter 3, counterfactual explanations rely on specific instances to explain the predictions of a trained classifier. Assessing whether the prediction associated to the counterfactual example generated is an artifact or not is therefore crucial.

The notion of ground-truth justification we propose to define in this chapter thus aims at making a distinction between an explanation that has been generated because of some previous knowledge (training data) and one that would be a consequence of

an artifact of the classifier. In the considered post-hoc paradigm, explainer systems do not have access to the training data; making this distinction is thus impossible. By relying on generating instances and labelling them with the black-box classifier, the risk for post-hoc explainers would thus be to build explanations that are helping the user understand a prediction using these artifacts instead of actual learned knowledge. This would thus lead to less useful explanations. In the case of counterfactual explanations, this is of course even more problematic since the generated instances are directly provided to the user as the final explanation. However, all post-hoc interpretability approaches are concerned.

It should be noted that predictions that cannot be directly associated to any ground-truth knowledge may not be harmful in the context of pure supervised learning: a desirable property of a classifier remains its ability to generalize to new observations. However, in the context of interpretability, this is less desirable as it may create new risks at the decision level. To illustrate this case, let us consider the example of a physician using a diagnostic tool and an explanation system to help him guide his drug prescriptions. An explanation built on predictions that are not based on existing medical cases would then be conceptually useless, if not very dangerous. Moreover, despite the fact that any other local post-hoc explanation method is also confronted to this issue, it is even more relevant for counterfactuals since it harms their main upside: their tangibility (see Section 2.3.1.2, page 31). This distinction between artifacts and ground-truth based decisions can be used to defined an important desideratum for counterfactual explanations. This desideratum, called *justification*, is defined in the next section.

Link with adversarial examples. In Section 2.3.3, page 36, we discussed the similarities between the notions of counterfactual examples and adversarial examples. A natural question in this regard is whether the proposed desideratum of justification aims to differentiate these two concepts. This is not the case: in Section 2.3.3, page 36, we stated that although similar, these notions are differentiated by their purpose. Adversarial examples are thus not generated in the context of interpretability. Yet, the proposed notion of justification is specifically designed for this purpose: therefore, it is important to highlight that its goal is not to detect adversarial examples. A discussion in more details about applying this notion to adversarial examples is proposed in Chapter 6, page 125, and a preliminary experiment exploring this direction is proposed in Appendix A, page 133.

4.1.2 | Proposed Definitions

We propose to define the notion of justification as a relation between an explanation and some existing knowledge (ground-truth data used to train the black-box model), more precisely with the topological notion of path, used when defining the path connectedness of a set. In order to be more easily understood and employed by a user, we argue that the counterfactual instance should be continuously connected to an observation from the training dataset belonging to the same class, i.e. without crossing the decision boundary. This statement holds even for counterfactual explanations generated in a post-hoc context. Although being generated in the post-hoc paradigm, without any information about the training data, ensuring that the counterfactual explanations are linked to ground-truth knowledge is important to guarantee relevant explanations.

This relies on the idea that the training data represents, in some way, the user's prior, trustworthy, knowledge. Having a counterfactual example being continuously linked to a correctly predicted training instance thus means that it is possible for the user to generalize the trusted behavior of the model to the counterfactual example. This thus leads to counterfactual explanations that can be more easily trusted, and therefore used. Obviously, the assumption that the training data is trustworthy is debatable, as situations may arise where the labels of the training data are wrongly assigned for instance. In this work, we make the assumptions that the training data is correctly labelled.

It should be noted that the idea of *justification* is not to identify the instances that are *responsible* for a prediction, like some other explainability approaches are trying to. For instance, Kabra et al. (2015) identify the importance of a training instance over a prediction by retraining the model with different labels and measuring the variation in prediction. In our case, the goal is rather to identify the training instances that are correctly being predicted to belong to the same class as the instance whose prediction is to be interpreted, for similar reasons.

We introduce the following definitions

Definition 2 (Justification)**.** *Given a classifier $f : \mathcal{X} \to \mathcal{Y}$ trained on a dataset X, a counterfactual example $e \in \mathcal{X}$ is* justified *by an instance $a \in X$ correctly predicted if $f(e) = f(a)$ and if there exists a continuous path h between e and a such that no decision boundary of f is met.*

Formally, e is justified *by $a \in X$ if: $f(a)$ is a correct prediction and: $\exists\ \gamma : [0,1] \to \mathcal{X}$ such that:*

(i) γ *is continuous*

(ii) $\gamma(0) = a$

(iii) $\gamma(1) = e$

(iv) $\forall t \in [0,1], f(\gamma(t)) = f(e)$.

To adapt this continuous notion to the constraint of discrete data instances, we replace the connectedness constraint with ϵ-*chainability*, with $\epsilon \in \mathbb{R}^+$: an ϵ-*chain* between e and a is a finite sequence e_0, e_1, ... $e_N \in \mathcal{X}$ such that $e_0 = e$, $e_N = a$ and $\forall i < N$, $d(e_i, e_{i+1}) < \epsilon$, with d a distance function. It is important to note that the instances $\{e_i\}_i$ are simply instances from the feature space $\mathcal{X}$, and are not sampled from the training set X.

This leads to the following definition for ϵ-*justification*:

Definition 3 (ϵ-justification)**.** *Given a trained classifier $f : \mathcal{X} \to \mathcal{Y}$ trained on a dataset X, a counterfactual example $e \in \mathcal{X}$ is* ϵ-justified *by an instance $a \in X$ correctly predicted if $f(e) = f(a)$ and if there exists an ϵ-chain $\{e_i\}_{i \leq N} \in \mathcal{X}$ between e and a such that $\forall i \leq N, f(e_i) = f(e)$.*

This definition is equivalent to approximating the function γ used in Definition 2 by a sequence $(e_i)_i$: when ϵ decreases towards 0, this definition becomes a weak version of Definition 2.

Figure 4.2 illustrates both the idea behind the notion of justification and its approximation, ϵ-justification. The left picture illustrates an instance x whose prediction by a binary classifier is to be interpreted, as well as two potential counterfactual explanations, CF_1 and CF_2. CF_2 can be connected to a ground-truth instance $a \in X$ through a continuous path γ without crossing the decision boundary of f and is therefore justified. On the contrary, CF_1 is not, since it lies in a classification region that does not contain any ground-truth instance from the same class (green "pocket"). In the right picture, the same idea is represented for ϵ-justification: an ϵ-path links CF_2 to a. It is important to note that the proposed notion of justification is based on connectedness, not convexity: as illustrated in these images, the considered paths and ϵ-chain do not necessarily form a straight line segment.

Consequently, we call a *justified counterfactual example* (denoted JCF) a counterfactual example that satisfies Definition 3. A counterfactual example that does not satisfy Definition 3 is called *unjustified counterfactual example*, denoted UCF.

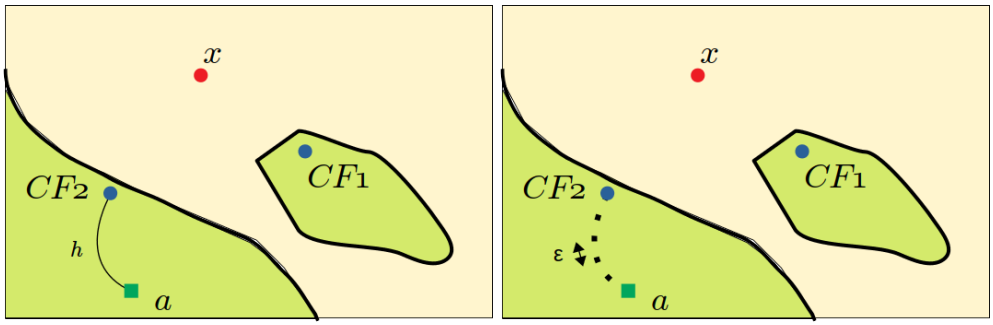


Figure 4.2: Left picture: illustration of the connectedness notion. The decision boundary learned by a classifier (illustrated by the yellow and green regions) has created two green regions. CF_1 and CF_2 are two candidate counterfactual explanations for x. CF_1 can be connected to the training instance a by a continuous path that does not cross the decision boundary of f, while CF_1 cannot. Right picture: same idea with a discretized path to the training instance, illustrating the notion of ϵ-justification.

Definition 3 depends on a hyperparameter, ϵ: setting its value allows to build an ϵ-graph of instances classified similarly. In such a graph, each instance is a node, and two nodes are linked if the distance between them is lower than ϵ. An ϵ-chain is thus a path in a ϵ-graph. The idea of using such a graph to approximate connectedness is also similar to the one found in some manifold learning approaches (e.g. Isomap Tenenbaum et al., 2000), where local neighborhoods help to approximate connectedness and can thus be used to reduce the dimension of the data. This idea is also used in some clustering methods, such as DBSCAN (Ester et al., 1996). We further develop this idea in the next section.

4.1.3 | Implementation

Implementing ϵ-justification using DBSCAN. It is possible to draw a link between connectedness and density-based clustering. In particular, the well-known algorithm DBSCAN (Ester et al., 1996) uses the distance between observations to evaluate the density of the data and derive clusters of dense observations. In the DBSCAN algorithm, two parameters $\epsilon \in \mathbb{R}^+$ and $minPts \in \mathbb{N}$ control the resulting clusters. The clusters are built from the *core samples* and *non-core samples*. Core samples are instances that have at least $minPts$ neighbors in their neighborhood, defined as a hyperball of radius ϵ. When an instance does not satisfy this condition, it is called a non-core sample if it neighbors a core sample, and an outlier otherwise.

In the specific case where $minPts = 2$, DBSCAN becomes a single-linkage clustering method with a constraint on ϵ: the instances grouped together are the ones that

have at least one neighbor closer than ϵ.

Thus, having two instances being connected by an ϵ-chain is equivalent to having them both belong to the same DBSCAN cluster when setting the parameters $minPts = 2$ and same ϵ. Drawing this parallel allows us to take advantage of the highly optimized implementations of the DBSCAN algorithm (e.g. using the *scikit-learn* package[1]) to efficiently assess whether a counterfactual example is justified, as described in the next section.

Restriction to numerical data. The previous definitions implicitly consider a numerical representation of the instances, for which a continuous distance can be applied. For unordered categorical data, connectedness cannot be properly defined. Therefore, the notion of justification cannot be directly applied to these domains. Hence, similarly to the previous chapter (as described in Section 3.1.3, page 45), we restrict the analysis of this chapter to numerical data.

4.2 | LRA: an Algorithm to Detect Unjustified Classification Regions

Using the definitions proposed in the the previous section, two procedures are proposed to analyze the risk of unjustification. First, in this section, we propose a test to assess the existence of this risk by detecting the presence around an observation of unjustified classification regions. Given a trained classifier and an instance whose prediction is to be interpreted, a procedure, called *Local Risk Assessment* (LRA), is thus proposed to assess the risk of having such regions in a neighborhood around the instance. Several experiments are then conducted with this procedure, and presented in Section 4.3, page 86. Then, in Section 4.4, page 93, a second procedure is proposed, called *Vulnerability Evaluation* (VE): the instances identified to be at risk by the LRA procedure are considered to assess the vulnerability of counterfactual explanation approaches to the risk of unjustification.

The LRA procedure, which uses the definition of justification and the parallel drawn with the DBSCAN algorithm in Section 4.1.2, is described in Section 4.2.1. In Section 4.2.2, the two criteria used to assess the risk of unjustification are presented. Illustrative results to show the efficiency of the proposed procedure are then presented in Section 4.2.3. Finally, a discussion about the hyperparameters of the LRA procedure, the values of which highly impact the results, is conducted in Section 4.2.4.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

4.2.1 | Local Risk Assessment Procedure: LRA

Given an instance $x \in \mathcal{X}$ whose prediction is to be interpreted, the classifier f and the training data X, the aim of the LRA procedure proposed in this section is to assess the risk of generating unjustified counterfactual examples in a local neighborhood of x. As mentioned before, it is a diagnostic tool for post-hoc approaches that relies on assessing their connectedness with ground-truth data. To do this, we propose a generative approach that aims at identifying the regions of this neighborhood that are ϵ-connected to a training instance of X, i.e. regions that satisfy Definition 3. In the rest of the chapter, given a subset of instances A of $\mathcal{X}$, we note, for a prediction class $l \in \mathcal{Y}$:

$$A^l = \{z \in A \mid f(z) = l\}$$

$$A^{\neq l} = \{z \in A \mid f(z) \neq l\}$$

The LRA procedure, commented below, is detailed in Algorithm 4, page 81 and illustrated in a two-dimensional setting in Figure 4.3: the red dot represents x, the observation whose prediction is to be interpreted, while the three green squares (a_0, a_1 and a_2) are correctly classified instances from the training set X. The decision boundary of f, the considered binary classifier, is represented by the black lines.

For clarity purposes, the procedure is split into three sequential steps discussed in turn in the 3 subsections below: first, the studied area is defined in the Definition step. Then, an Initial Risk Assessment step is performed in this area. Finally, if needed, the procedure is repeated in the iteration step.

4.2.1.1 | Definition Step

We first define the studied local neighborhood of x, i.e. the region of $\mathcal{X}$ around x whose risk we are trying to assess: it is the ball with center x and whose radius equals the distance between x and its closest neighbor from X correctly predicted to belong to another class: $\mathcal{B}(x, d(x, a_0))$, with:

$$a_0 = \arg\min\{d(x, z) \mid z \in X^{\neq f(x)} \text{ s.t. } f(z) \text{ is correct}\}$$

The distance $d(x, a_0)$ represents an upper bound for the minimal distance between x and the decision boundary of f. It is hence a reasonable distance to define the local region of $\mathcal{X}$ in which we look for counterfactual examples: as it is ϵ-connected to itself, a_0 represents a "close" justified counterfactual explanation.

The boundary of the defined local area $\mathcal{B}(x, d(x, a_0))$ is illustrated in the left picture of Figure 4.3 as the blue dashed circle.

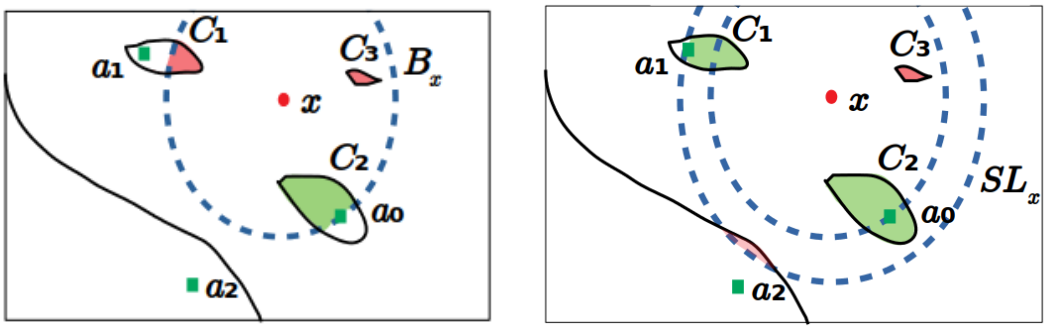


Figure 4.3: Illustration of the Local Risk Assessment procedure in the context of binary classification. Left: Definition and Initial Assessment steps; right: Iteration step.

4.2.1.2 | Initial Assessment Step

A high number n of instances $B_x = \{x_i\}_{i \leq n}$ are then sampled in $\mathcal{B}(x, d(x, a_0))$ following a uniform distribution, and labelled using f. Setting d as the Euclidean distance allows us to use the HLG algorithm (introduced in the context of *Growing Spheres* in Chapter 3 and described in Algorithm 1, page 53) to generate these instances. The HLG algorithm is thus used, with inputs: x, 0 and $d(x, a_0)$.

Among these sampled instances, the ones that are predicted to belong to the same class as a_0, i.e. $B_x^{f(a_0)}$, are kept. Indeeed, they are candidate couterfactuals for x, as close points of a different class. Recalling Definition 3, the goal of the LRA procedure is to identify the instances of $B_x^{f(a_0)}$ that are ϵ-justified, i.e. connected to an instance of X through an ϵ-chain. The process to set the value of ϵ is detailed in Section 4.2.4, page 83. As mentioned earlier, an easy way to implement this is to use the clustering algorithm DBSCAN on $B_x^{f(a_0)} \cup \{a_0\}$ with parameter values ϵ and $minPts = 2$. We note $\{C_t\}_t$ the resulting clusters and outliers. Because B_x consists in numerous instances generated uniformly, keeping only the instances of $B_x^{f(a_0)}$ means that this uniformity is broken and DBSCAN can thus easily identify the instances that belong to a same classification region. Each cluster thus corresponds to a connected classification region, separated one from another by regions of low density. This also allows us to directly detect which of these instances are justified: according to Definition 3 and as described in Section 4.1.3, the instances that belong to the same cluster as a_0 are ϵ-justified and thus labelled as JCF.

This Initial Assessment step is written in lines 2 to 8 of Algorithm 4, and illustrated in the left image of Figure 4.3: instances are generated in B_x (represented as the blue dashed circle; the generated instances are not shown in the illustration), the ones belonging to $B_x^{f(a_0)}$ are represented by the colored areas and assigned to clusters C_1,

C_2 and C_3. At this step, cluster C_1 and C_3 are detected as unjustified, and therefore represented by a red area. On the other hand, cluster C_2 is detected as justified, since $a_0 \in C_2$, it is therefore represented by a green area.

4.2.1.3 | Iteration Step

At this step, there is no certainty that the other instances of $B^{f(a_0)}$ (i.e. belonging to other clusters than the one that contains a_0) are unjustified: they can either simply be connected to other instances from $X^{f(a_0)}$ than a_0, or using an ϵ-path that cannot be fully included in the explored area $\mathcal{B}(x, d(x, a_0))$. This situation is illustrated in Figure 4.3: the instances belonging to cluster C_1 are actually justified by a_1, but are not detected as such in the Intial Assessment step since they are not connected within B_x.

To address this issue, we define a_1 as the second closest instance of $X^{f(a_0)}$ to x that is correctly predicted, and broaden the exploration region to the hyperspherical layer between a_0 and a_1, which is defined the same way as in Section 3.2.2, page 52:

$$\mathcal{SL}_1 = \{z \in \mathcal{X} \text{ s.t. } d(a_0, x_0) \leq d(z, x_0) \leq d(a_1, x_0)\}.$$

SL_1 is then a set of instances generated uniformly in $\mathcal{SL}_1$. The instances from $SL_1^{f(a_0)} \cup \{a_1\}$ are clustered using DBSCAN with the same parameter values. Using the same criterion, i.e. whether their minimum distance to an existing instance is less than ϵ, some of these new clusters can be merged to the ones detected at the previous step. As a result, some of the clusters C_t defined at the previous step grow (i.e. they are updated by some of the newly generated instances of SL_1), others remain identical (meaning that the classification region associated to the cluster was fully enclosed in $\mathcal{B}(x, d(x, a_0))$), and others are created at this step (i.e. formed by instances from SL_1 that cannot be attached to any existing cluster).

Depending on which cluster the ground-truth instance a_1 belongs to, new instances from the initial explored area can be labelled as JCF. In case an initial cluster of non-connected instances is being not updated by new instances, this means that the cluster was associated to an unjustified region that was fully enclosed in exploration step. This cluster is therefore labelled as UCF. This process is illustrated in the right image of Figure 4.3: cluster C_1 can now be connected to a_1 through the instances generated in SL_1 and the associated instances are therefore labelled as JCF (green region), while cluster C_3 has not been updated at this step and therefore remains labelled as an unjustified region (red region).

This step is repeated by generating hyperspherical layers defined by all instances from $X^{f(a_0)}$ until all the initial clusters C_t can either be justified or are not being updated by any new instances anymore (this is for instance the case of cluster C_3 of

Figure 4.3). This step is illustrated in the "while" loop of Algorithm 4, lines 9 to 19. It is also performed using the HLG algorithm when considering the Euclidean distance. If some non-connected clusters are still being updated when all instances from the training set have been explored (e.g. red region in the top left corner of the left picture of Figure 4.1), they are labelled as unjustified.

In the end, the LRA procedure returns n_J (respectively n_U) the number of JCF (respectively UCF) originally generated in $\mathcal{B}(x, d(x, a_0))$. If $n_U > 0$, there exists a risk of generating unjustified counterfactual examples for the considered data point x whose prediction is to be explained.

4.2.1.4 | Complexity of the Procedure

The procedure is quite costly. Indeed, it relies on generating numerous instances and clustering them using the DBSCAN algorithm. The complexity of the DBSCAN algorithm is at worst $O(n^2)$. As a result, the complexity of the LRA procedure is quadratic.

In addition, the DBSCAN algorithm needs to be run several times, one at each iteration step of the procedure. The number of steps required before the procedure stops obviously depends on the considered instance x. The worst case scenario is that some of the generated instances remain non-connected (i.e. still belong to $\mathcal{C}_{NC}$ after all the iteration steps), as it would be in a situation such as the one represented in the right image of Figure 4.1, page 71. In such a situation, because the steps correspond to expanding the studied area based on the instances from $X^{f(a_0)}$, the number of required steps would therefore be $p = |X^{f(a_0)}|$. The number of iterations thus linearly scales with the number of training instances. Therefore, the complexity of the LRA procedure is quite high. Additionally, the parameters of the procedure also impact the computation time. A discussion on this topic is proposed in Section 4.2.4, page 83.

Nevertheless, two things are important to note regarding this complexity. First, in practice, the procedure hardly ever performs p steps. In the experiments conducted in Section 4.3, page 86, no instance required p steps for the procedure to end. In fact, most of them only needed less than 10 steps, with the value of $|X^{f(a_0)}|$ being up to several thousands. Second, it is important to keep in mind that the proposed procedure is meant to be used as a *diagnostic tool*, as illustrated by the experiments conducted in Section 4.3.1, page 86. It is not a new interpretability approach, it rather aims at exposing the existence of an issue, as some of the approaches mentioned in Section 2.1.4, page 19. Therefore, the complexity (and computational time) of the

Algorithm 4 Local Risk Assessment procedure: LRA

Require: x, f, X, η

Sort the correctly predicted instances from $X^{\neq f(x)} = \{a_0, a_1, ...\}$ in increasing order of their distance to x
$B_x = \{x_i\}_{i \leq n} \sim \text{Uniform}(\mathcal{B}(x, a_0))$, using $HLG(x, 0, d(x, a_0))$
$B_x^{f(a_0)} = \{x_i \in B_x \ : \ f(x_i) = f(a_0)\} \cup \{a_0\}$
Set ϵ according to Equation 4.1, page 84
$\{C_t\}_t \leftarrow \text{DBSCAN}(B_x^{f(a_0)}, \epsilon, \text{minPts} = 2)$
$\mathcal{C}_J = C_{t_0}$ s.t. $a_0 \in C_{t_0}$; $n_J = |\mathcal{C}_J|$
$\mathcal{C}_{NC} = \bigcup_{t \neq t_0} C_t$; $n_{NC} = |\mathcal{C}_{NC}|$
$\mathcal{C}_U = \{\}$; $n_U = 0$
$k = 0$
while $n_{NC} > 0$ **do**
    $k = k + 1$
    $SL_k = \{x_i\}_i \sim \text{Uniform}(\mathcal{SL}_k)$ using $HLG(x, d(x, a_{k-1}), d(x, a_k))$
    $SL_k^{f(a_k)} = \{x_i \in SL_k \ : \ f(x_i) = f(a_k)\}$
    $\{C'_t\}_t \leftarrow \text{DBSCAN}(SL_k^{f(a_k)} \cup \{a_k\}, \epsilon, \text{minPts} = 2)$
    Update $\mathcal{C}_J$ and $\mathcal{C}_{NC}$ with $\{C'_t\}_t$
    Update $\mathcal{C}_U$ with clusters from $\mathcal{C}_{NC}$ that are not growing anymore
    Update n_J, n_U and n_{NC}
end while
return n_J, n_U

procedure is not crucial.

4.2.2 | Quantifying the Risk of Unjustification

Using the result of the LRA procedure, we propose to evaluate the risk of generating UCF when explaining the prediction for x with 2 criteria:

$$S_x = \mathbb{1}_{n_U > 0} \qquad \text{and} \qquad R_x = \frac{n_U}{n_U + n_J}.$$

S_x, which is the crucial criterion of the study, labels the studied instance x as being vulnerable to the risk if its neighborhood contains UCF candidates, i.e. if $n_U > 0$. The risk itself, measured by R_x, describes the likelihood of having an unjustified counterfactual example in the studied area when looking for counterfactual examples. Since the counterfactual examples described by the numbers n_u and n_J are generated following a uniform distribution, the R_x score can also be understood as a Monte-Carlo estimation of the relative size of the detected unjustified classification region in the

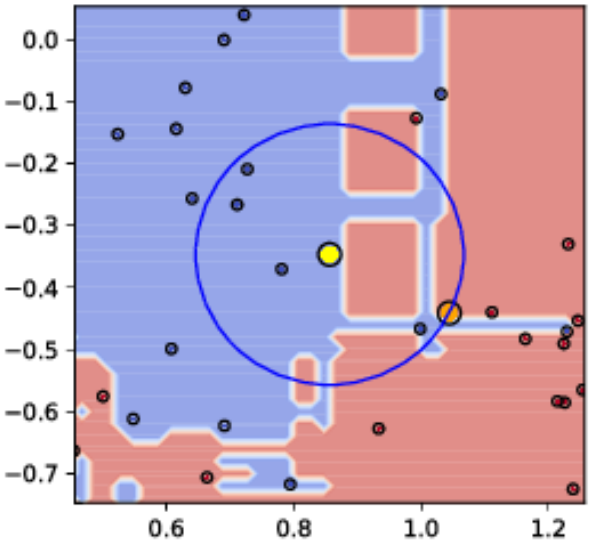


Figure 4.4: Illustrative result of the Local Risk Assessment procedure (left: $S_x = 1$) for an instance of the half-moons dataset, commented in Section 4.2.3.

explored area. On the other hand, the criterion S_x labels the mere existence of the unjustified region.

These scores are defined for a specific instance x whose prediction is to be interpreted. We are also interested in their average values $\bar{S}$ and $\bar{R}$ over multiple instances. Additionally, in practice, since the calculation of these scores relies on a random generation component, we compute them several times and look at the average values (10 runs of the procedure) for each instance x.

4.2.3 | Illustrative Results

For the purpose of giving insights about the proposed LRA procedure, a toy dataset (half-moons dataset, described in Section 3.3.1, page 57) is used. A classifier, deliberately chosen for its low complexity (random forest classifier with only 3 trees), is trained on 70% of the data. The classifier achieves 98% accuracy on the testing set.

Figure 4.4 illustrates the LRA procedure for a specific instance x (yellow point), exploring the neighborhood $\mathcal{B}(x, d(x, a_0))$ (blue circle) delimited by its closest neighbor from opposite class a_0 (orange instance). The red and blue dots are the instances used to train the classifier, and the red and blue areas represent its decision function: x is predicted to belong to the blue class and a_0 to the red one. Within $\mathcal{B}(x, d(x, a_0))$, a red square area is detected as a group of unjustified counterfactual examples since there is no red instance in this pocket. As a result, $S_x = 1$ (and $R_x = 0.33$).

This simple example illustrates the fact that the risk of generating unjustified counterfactual examples does exist for the half-moon dataset processed by a small random forest classifier. Full results of the LRA procedure run over multiple instances of several datasets are described and commented in Section 4.3, page 86.

4.2.4 | LRA Parameters: n and ϵ

The values of the two hyperparameters, n and ϵ, are obviously crucial since they define the notion of ϵ-justification and density of the generated instances. This can be related to what is discussed in Section 3.2.3, page 56, with the hyperparameters n and w of *Growing Spheres* impacting the results through the density of the sampling. Indeed, choosing inadequate values for n and ϵ may lead to having some unconnected regions not detected as such, and thus incorrect estimations of the explanation justification. First, we discuss how these parameters impact the precision and computational time of the procedure and propose a method to set the value of ϵ. Then, we discuss how the value of the hyperparameter n should be set.

Trade-off between precision and computational time. Broadly speaking, the higher the value of n and the smaller the value of ϵ are, the better the approximation of the topology of the local neighborhood is: on the one hand, if the value of n is too low, small regions of unjustified counterfactual examples might be missed. On the other hand, if the value of ϵ is too high, an ϵ-chain is not a good approximation of a connected path between two instances. This remark would advocate for high n and small ϵ values. However, on the other hand the higher n and the smaller ϵ are, the higher the computational time is: there is an obvious a trade-off between the algorithm precision and its computational time of the algorithm. Indeed, the complexity of the algorithm has been shown to be at worst quadratic in n, with at worst p steps, with $p = |X^{f(a_0)}|$ (see Section 4.2.1.4, page 80).

As mentioned earlier, these two values are linked: n defines, for a given $x \in X$, the density of the sampling in the the LRA procedure, hence it determines the average pairwise distance between the generated observations. Now this average distance is compared to ϵ in DBSCAN. In addition to the local topology of the decision boundary of the classifier, identifying an adequate value for ϵ therefore depends on n as well.

In practice, because the instances B_x are generated in the initial assessment step before running DBSCAN, a reasonable practical choice is to set the value of ϵ as a function of the n instances generated in B_x. Concretely, we propose to update $B_x^{f(a_0)}$

with the instance a_0 (see line 3 of Algorithm 4), and to set the value of ϵ to the maximum value of the distances of the obtained set $B_x^{f(a_0)}$ to their closest neighbors:

$$\epsilon = \max_{x_i \in B_x^{f(a_0)}} \min_{x_j \in B_x^{f(a_0)} \setminus \{x_i\}} d(x_i, x_j) \tag{4.1}$$

We choose to use this value for ϵ in the LRA algorithm (see line 4 of Algorithm 4, page 81).

Using this value, the training instance a_0 is guaranteed to belong to an actual cluster (i.e. not to be detected as an outlier). Indeed, according to Equation 4.1, the value of ϵ is then, by construction, greater than the distance between a_0 and the furthest instance of $B_x^{f(a_0)}$. Therefore, a_0 belongs by design to a DBSCAN cluster with parameters $minPts = 2$ and ϵ. This is a desirable property of the approach: it is expected that since a_0 is correctly predicted, it should be possible to generate a close neighbor classified similarly (in the same classification region). However, this requires the whole LRA procedure to be run several times to mitigate the risks that, because of the random generation of instances, ϵ does not take an absurd value due to a particular generation scenario. For instance, the case where a single instance x_{i^*} would be generated far away from the others would result in having all of the generated instances being detected as JCF: ϵ would then be taking an absurdly high value as for all $j \neq i^*$, $d(x_{i^*}, x_j)$ would be high.

Using this definition implies that the value of ϵ directly depends on the value of n (which impacts the density of the generated instances and therefore ϵ), and the choice of LRA hyperparameters thus becomes that of setting the value of n alone.

Setting the value of n. The role of parameter n is to make sure that the explored area is "saturated" enough and that no subtlety of the model's decision border, as well as potential unjustified counterfactual examples, are left undetected. In order to have the best performance, n should thus have the highest value as possible. However, this also increases dramatically the running time of the algorithm, that depends quadratically on n. Thus, for each observation x we assume there exists a threshold n_x above which the complexity of the decision boundary of f would be fully "captured". When this value is met, increasing n above n_x has very little impact over the found clusters and therefore on the risk score R_x. However, the same issues as with the number of instances generated in *Growing Spheres* of Chapter 3 are encountered: beside depending on the local complexity of the classifier's decision boundary, the n_x value required to saturate the local space increases exponentially with the dimensionality of the problem. Furthermore, as the radius of generation increases during the itera-

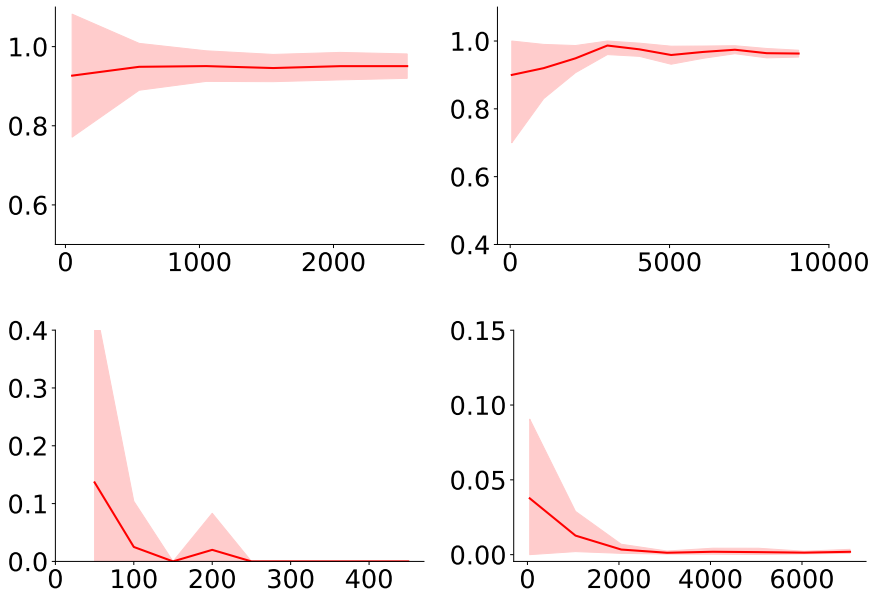


Figure 4.5: Average R_x score for four instances of the half-moons dataset as a function of n for 10 runs of the LRA procedure. The lightly colored area represent the standard deviations obtained for 10 runs of the procedure. The instances in the top row satisfy $S_x = 1$, while the ones in the bottom row satisfy $S_x = 0$. After n reaches a certain value, R_x hardly changes anymore.

tion steps, the number of instances should also increase to guarantee constant space saturation across various steps. Instead, to avoid convergence or memory issues, we choose to set a high initial value of n at the first step and keep it constant.

In this context, we are interested in identifying the value n_x that properly captures the complexity of the local decision boundary of the classifier without generating an unrequired amount of instances. Following the assumption made about the existence of a threshold guaranteeing that the complexity is correctly captured, we look at the value taken by R_x for several instances x and several values of n, so as to detect the threshold above which generating more instances does not change the output of the LRA procedure. Figure 4.5 illustrates the resulting values for R_x for four instances of the half-moons dataset. A random forest classifier is used as the black-box classifier, and the LRA procedure is run 10 times for each considered instance. More precisely, among these 4 instances, two instances such that $S_x = 1$ (top row), and two such that $S_x = 0$ (bottom row). As expected, we observe that for small values of n, there is a high variability in the obtained values for R_x. In all four cases, the graphs show that the R_x score reaches a plateau after a certain value n_x. Using this observation, the LRA procedure can be tested with various values of n to ensure a reasonable value is

chosen for the results of the other experiments presented in the next section.

Requiring several runs of the procedure obviously burdens all the more the use of the LRA procedure, on top of its own high complexity. However, as mentioned in Section 4.2.1.4, since the proposed procedure is a single-use diagnostic tool and not an interpretability method, the complexity and running time of the algorithm is not crucial.

4.3 | Experimental Assessment of the Local Risk of Generating Unjustified Counterfactuals

Using the LRA procedure and the proposed metrics, we propose to highlight and characterize the *risk* of having unjustified counterfactual regions harm interpretability independently of any specified explainer. These proposed experiments lie therefore out of the usually considered post-hoc context, as the training data X is supposed to be accessible. Two experiments are conducted, whose experimental protocols are detailed in Section 4.3.1. First in Section 4.3.2, we apply the LRA procedure to several datasets to assess the considered risk, and analyze the behavior of various classifiers. Then in Section 4.3.3, we study the risk of unjustification in the light of the notion of model overfitting, and analyze how these two concepts are related.

4.3.1 | Experimental Protocol

This section describes the datasets, classifiers and protocol considered in the experiments.

Datasets. The considered datasets include 2 low-dimensional datasets (half-moons and wine Dua and Graff, 2017) as well as 4 real datasets: Boston Housing (Harrison and Rubinfeld, 1978), German Credit (Dua and Graff, 2017), Online News Popularity (Fernandes et al., 2015) and Propublica Recidivism (Larson et al., 2016). As mentioned in Chapter 3, these structured datasets present the advantage of naturally understandable features and are commonly used in the interpretability (and fairness) literature. All datasets contain less than 70 numerical attributes. After keeping only the numerical attributes, the data is rescaled.

Classifiers. Several binary classifiers are trained on each dataset: a random forest classifier (RF), a support vector machine classifier with Gaussian kernel (SVM), an

Dataset	RF	SVM	XGB	NB	KNN	1-NN
Moons	0.98 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.95 ± 0.01	0.99 ± 0.00	0.95 ± 0.01
Wine	0.98 ± 0.01	0.99 ± 0.00	0.99 ± 0.00	0.95 ± 0.01	0.99 ± 0.00	0.95 ± 0.01
Boston	0.96 ± 0.02	0.97 ± 0.04	0.97 ± 0.03	0.87 ± 0.08	0.93 ± 0.06	0.85 ± 0.04
Credit	0.75 ± 0.05	0.64 ± 0.04	0.70 ± 0.08	0.66 ± 0.04	0.66 ± 0.02	0.55 ± 0.05
News	0.68 ± 0.02	0.68 ± 0.01	0.70 ± 0.02	0.65 ± 0.02	0.65 ± 0.02	0.55 ± 0.01
Recidivism	0.81 ± 0.01	0.82 ± 0.01	0.84 ± 0.01	0.78 ± 0.02	0.81 ± 0.02	0.68 ± 0.02

Table 4.1: AUC scores obtained on the test sets for a random forest (RF), support vector machine classifier (SVM), XGBoost (XGB), naive Bayes classifier (NB) k-nearest neighbors (KNN) and nearest neighbor (1-NN). Their respective parameters are optimized in a 5-fold cross validation.

XGboost classifier (XGB), a Naive Bayes classifier (NB) and a k-nearest-neighbors classifier (k-NN), as well as the extreme case of 1-NN. Unless specified, the associated hyperparameters are chosen using a 5-fold cross validation to optimize accuracy.

The *Area Under Curve* (AUC) score values obtained on the test set (10% of the data) with these classifiers are shown in Table 4.1. These values are given for the sake of completeness, but are not relevant to the study, except when mentioned otherwise. Several variations of the same classifier are also considered for the second experiment (see Section 4.3.3), changing the values of their associated hyperparameters, one at a time: the maximum depth allowed for each tree of the random forest algorithm and the kernel width γ of the Gaussian kernel of the support vector machine classifier.

Protocol. For each considered dataset, a train-test split of the data is performed with 90%-10% proportion to train and evaluate the accuracy of the classifiers. The LRA procedure is then applied to each instance of the considered test sets, and the scores $\bar{S}$ and $\bar{R}$ are calculated and analyzed for each dataset and classifier.

4.3.2 | Result Analysis

The goal of this first experiment is to assess the existence of the risk of unjustification for the considered datasets and classifiers. Additionally, we study the importance of the choice of the classifier on the creation of unjustified classification regions.

Table 4.2, page 88 shows the proportion $\bar{S}$ of the studied instances that have unjustified classification regions in their neighborhood (i.e. x such that LRA returns $S_x = 1$). Every classifier shown appears to be generating such unjustified regions: in some cases, as much as 93% of the tested instances are concerned (XGB classifier trained on the German Credit dataset). While these figures seem high, it is important

Dataset	RF	SVM	XGB	NB	KNN
Half-moons	0.37	0.00	0.05	0.00	0.02
Wine	0.21	0.08	0.15	0.08	0.15
Boston	0.63	0.29	0.62	0.44	0.25
Credit	0.93	0.76	0.93	0.27	0.92
News	0.85	0.72	0.86	0.57	0.68
Recidivism	0.81	0.50	0.61	0.36	0.73

Table 4.2: Proportion of instances being at risk of generating a UCF ($\bar{S}$ score) over the test sets for 6 datasets.

to keep in mind that it does not mean that the classifiers have created many unjustified classification regions: in this case, all these instances (93% of the test set) may be all exposed to the same unconnected region(s).

However, it can be observed that the extent to which each classifier is vulnerable varies greatly. For instance, among the considered classifiers, the random forest and XGBoost algorithms seem to be more exposed than the other classifiers (average $\bar{S}$ value across dataset respectively 0.63 and 0.54, vs. 0.39 for the SVM classifier for instance), despite presenting good predictive performance. An assumption to explain this observation is that because they aggregate the decisions of diverse weak classifiers, ensemble methods are more prone to generating unjustified classification regions.

The learning algorithm, and therefore the associated complexity of the learned decision boundary, thus heavily influences the creation of classification regions. Hence, a link between justification and predictive accuracy can be expected, as illustrated by comparing the justification scores $\bar{S}$ with the predictive accuracy scores of the classifiers shown in Table 4.1: simple classifiers, with worse predictive accuracy, seem to be performing better in terms of justification. An example is the Naive Bayes classifier: while this classifier seems to be the more robust to the studied problem (average value of $\bar{S}$ across all datasets takes the relatively low value of 0.29), it can be noted that it is also the classifier that performs the worst in terms of prediction (beside 1-NN). Additionally, it can be noted that other models such as logistic regression, decision tree or nearest neighbor classifier (not appearing in the table) have, by construction, no UCF ($\bar{S} = 0.0$): a logistic regression creates only two connected classification regions, the prediction associated to the leaf of a decision tree is based on the presence of ground-truth instances and the predictions of a 1-NN classifier are by construction connected to their closest neighbor from the training data.

These results are further confirmed by the values of $\bar{R}$ shown in Table 4.3: $\bar{R}$ pro-

Dataset	RF	SVM	XGB	NB	KNN
Half-moons	0.07 (0.17)	0.00 (0.00)	0.01 (0.02)	0.00 (0.00)	0.00 (0.00)
Wine	0.01 (0.02)	0.02 (0.07)	0.00 (0.01)	0.01 (0.02)	0.01 (0.01)
Boston	0.16 (0.25)	0.06 (0.13)	0.14 (0.24)	0.07 (0.14)	0.03 (0.05)
Credit	0.44 (0.37)	0.10 (0.14)	0.45 (0.37)	0.06 (0.17)	0.31 (0.27)
News	0.35 (0.28)	0.18 (0.28)	0.33 (0.30)	0.12 (0.24)	0.37 (0.38)
Recidivism	0.26 (0.30)	0.14 (0.21)	0.21 (0.28)	0.08 (0.20)	0.20 (0.30)

Table 4.3: Average risk of generating an UCF ($\bar{R}$ score) and standard deviations for 6 datasets.

vides additional information to $\bar{S}$, though an indication of the relative size of these unjustified classification regions. For instance, despite having similar values for $\bar{S}$ on the German Credit dataset, RF and XGboost have higher $\bar{R}$ values than KNN, indicating that the formed unconnected regions are wider in average.

An important observation to make is that the $\bar{R}$ score varies greatly across the instances of a given dataset (high standard deviation), as well as across the datasets. This can be explained by the complexity of both the used datasets and classifiers. Supposedly, an instance located far away from the decision boundary of the classifier has a greater chance to generate unjustified counterfactual examples than an instance located closer, since the neighborhood explored by the Local Risk Assessment procedure is wider. However, for the same reason, the $\bar{R}$ value returned for an instance x that is confronted to a given unjustified region would be higher if x is located closer to the decision boundary of f than if it is located further. More generally, these observations on the variability of the justification scores depend on several factors, such as characteristics of the data (e.g. the considered dimensionality and density) and labels (e.g. number of classes, classes separability). These have not been studied here and constitute interesting prospective works; they are somehow illustrated by the variability of $\bar{R}$ between datasets. An interpretation for these results is that more complex datasets (e.g. less separable classes, higher dimensionality...) may lead to classifiers learning more complex decision boundaries, at the risk of favoring overfitting and therefore the creation of unjustified classification regions. This phenomenon is further studied in the second experiment, described in the following section.

4.3.3 | Link Between Justification and Overfitting

To further study the relation between the creation of unjustified regions and the learning algorithm of the classifier, we analyze the influence of overfitting over the considered quality criteria $\bar{R}$ and $\bar{S}$. Obviously, the creation of unconnected regions raises

questions about the generalization capacities of the considered classifiers. For this purpose, we conduct in this section a second experiment where we attempt to control the overfitting of the classifier and analyze the impact on the local risk scores.

First, we detail the experimental protocol to attempt to control overfitting. Then, in order to give more insights about the experiment and its results, we discuss an illustrative example on a toy dataset. Finally, some quantitative results are presented.

4.3.3.1 | Controlling Overfitting

Overfitting is controlled by changing the values of the hyperparameters of two classifiers:

- The maximum depth allowed for a tree, written *max_depth* for the random forest classifier. The random forest algorithm (Breiman, 2001) aggregates the decisions of multiple decision trees, each trained on a subset of instances and using a subset of features. Limiting the maximum depth allowed for each tree limits the overfitting capacity of each tree (Breiman et al., 1984). While random forests are not supposed to overfit due to the feature sampling, the assumption is that setting the value of the maximum tree depth while keeping the others parameters constant impacts the overfitting of the whole random forest.
- The kernel width of the RBF kernel, written γ, for the SVM classifier (Boser et al., 1992). Given two instances $(x, x') \in \mathcal{X}^2$, the kernel is defined as:

$$K(x, x') = e^{-\gamma ||x - x'||_2^2}$$

The γ parameter therefore controls the influence of each single training instance over training: increasing γ reduces the radius of the area of influence of the support vectors, leading to more overfitting (Han and Jiang, 2014). On the contrary, decreasing γ reduces the effect of the support vectors, leading to a global behavior that is more similar to the one of a linear model, at the risk of underfitting.

4.3.3.2 | Illustrative Example

To give an intuition about how limiting overfitting through these parameters impacts the creation of unjustified regions, we apply the LRA procedure on the two-dimensional half-moons dataset to a random forest classifier with only 3 trees and change the value of *max_depth*. Figure 4.6 shows a zoom on an area of the decision boundary (represented by the separation between the colored areas; the green

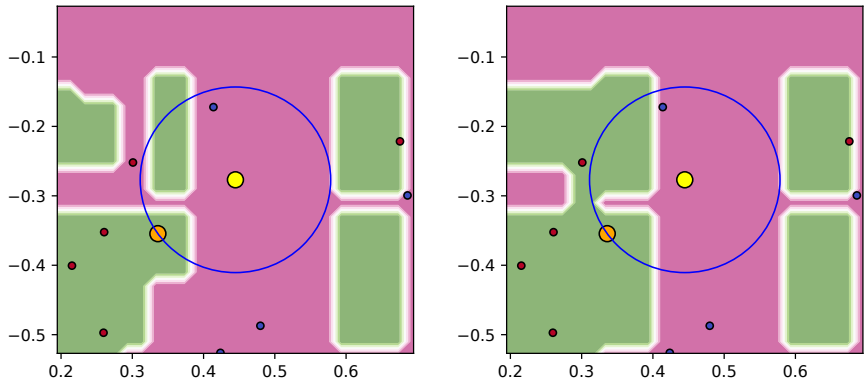


Figure 4.6: Illustration of the LRA procedure applied to an instance of the half-moons dataset. Left: RF with no constraint on the maximum depth allowed. Right: maximum depth set to 10.

and purple dots represent the training instances), as well as the result of LRA for a specific instance x (yellow instance). In the left figure, the considered classifier has no limitation on the depth of the trees it can use (and therefore reaches a maximum depth of 14), whereas in the right one this parameter is set to 10. As explained earlier, LRA explores the local neighborhood of x (blue circle), delimited by its closest neighbor from the training set correctly classified a_0 (orange instance). In the left figure, within this neighborhood, a green rectangular region is detected as an unjustified region (top left from x): there is no green instance in this region, hence $S_x = 1$ (and $R_x = 0.13$). However, in the right picture, this region is connected to green instances: $S_x = 0$ (hence also $R_x = 0$).

In this example, reducing the value of the maximum depth allowed for the trees of the random forest classifier, overfitting was reduced. As a result, two classification regions appearing as separate in the left image are merged in the right one, leading to the suppression of the unjustified region.

4.3.3.3 | Quantitative Results and Discussion

Quantitative results of this phenomenon are shown in Figure 4.7, which illustrates the evolution of the $\bar{R}$ score for the two mentioned classifiers on the Boston dataset. On the left picture, a random forest classifier is trained for several values of max_depth (the other parameters are kept constant). Setting max_depth = "None" means that no constraint is imposed on the maximum depth of the trees. On the right picture, a SVM classifier is trained for several values of the RBF kernel width γ. As expected, the more overfitting is allowed (i.e. when the maximum tree depth of RF and the γ

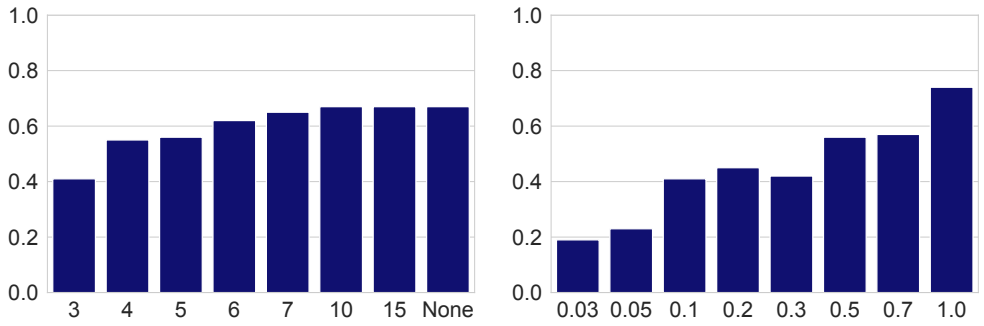


Figure 4.7: $\bar{R}$ scores for RF (left) and SVM (right) classifiers on the Boston dataset for different hyperparameter values (respectively the maximum depth of the trees and the width of the Gaussian kernel). "None" (X-axis of the left figure) means that no maximum tree depth restriction is set.

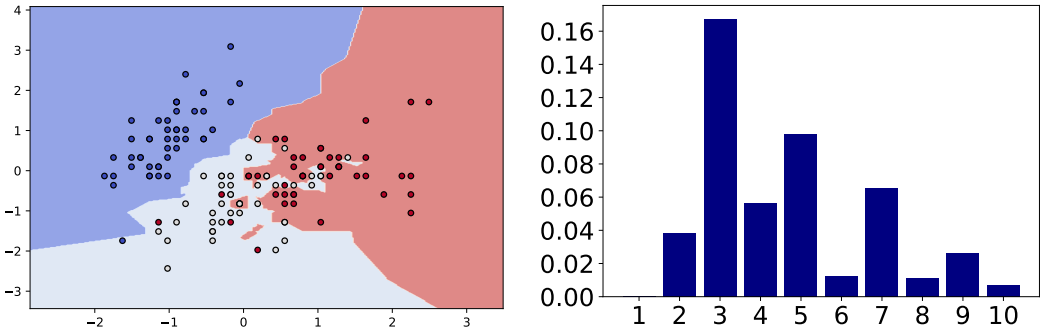


Figure 4.8: Left: unjustified regions created by a k-NN classifier ($k = 3$) on 2-dimension version of the iris dataset (the represented instances are the training set). Right: $\bar{R}$ scores for a k-NN classifier trained on 70% of the half-moons dataset for several values of k.

parameter of the RBF kernel of SVM increase), the more prone to generate unjustified regions these two classifiers seem.

However, it is important to note that the notion of overfitting is not sufficient to fully explain the creation of unjustified regions, as controlling it does not allow to ensure that no unjustified region is created. Thus, there is not a clear trade-off between overfitting and justification. Additionally, as shown in the first experiment, this behavior is obviously dependent on the considered classifier. In the case of the k-NN classifier for instance, the opposite behavior can be observed between justification and overfitting: a nearest neighbor classifier ($k = 1$), as mentioned earlier, does not create any unjustified classification region despite heavily overfitting. However, attempting to reduce this overfitting by increasing the value of k leads to the possibility

of creating unconnected regions, as shown in the left picture of Figure 4.8. Using a k-NN classifier with $k = 3$ on a 2-D version of the iris dataset (predictive accuracy: 0.98) leads to the creation of some unconnected classification regions. In the right image, the values obtained for the $\bar{R}$ scores on the half-moons dataset with a k-NN classifier for various values of k are shown: we observe that increasing the value of k does not lead to a clear decreasing tendency in $\bar{R}$.

The creation of unjustified regions thus depends heavily on the classifier beyond its overfitting tendency, and seems therefore hard to control.

This can be harmful in the context of post-hoc interpretability, as no knowledge about the classifier is available. In the next section, a study is proposed to assess the vulnerability of post-hoc counterfactuals approaches from the state-of-the-art.

Summary of the results. The experiments performed in this section show that there is a risk of generating UCF for several datasets and classifiers. Additionally, several characteristics of the classifiers, including among others their overfitting tendency, seem to impact the risk of unjustification. These results raise the question of the vulnerability of counterfactual explanation approaches when being confronted to this risk. In the next section, a new procedure, called *Vulnerability Evaluation* (VE), is proposed to assess this vulnerability.

4.4 | VE: An Algorithm to Assess the Vulnerability of Post-hoc Counterfactual Approaches

Once the risk of generating UCF has been established, we analyze how troublesome it is for existing counterfactual approaches. This section presents a second procedure, called *Vulnerability Evaluation*, which aims at assessing how a post-hoc interpretability method behaves in the presence of UCF. As for LRA, the goal of this section is to propose a diagnostic for post-hoc explainers. Therefore, the considered context is not strictly post-hoc, as training instances are assumed to be accessible.

The VE procedure is described in Section 4.4.1. We then apply the proposed procedure to three post-hoc counterfactual approaches, namely LORE-I, an adaptation of LORE (Guidotti et al., 2019a) that we propose to define counterfactual examples beyond counterfactual rules, HCLS (Lash et al., 2017a), as well as a variant of *Growing Spheres*, proposed in Chapter 3, on several datasets. The experimental protocol is presented in Section 4.4.2, while illustrative and quantitative results are shown in Section 4.4.3.

4.4.1 | Vulnerability Evaluation Procedure: VE

The goal of the VE procedure is to assess the risk for counterfactual explanation methods to generate UCF in risky regions. Given an instance $x \in X$, we use the LRA procedure to assess the risk R_x and focus on the instances where this risk is "significant", e.g. by imposing $R_x \geq 0.25$. Using the counterfactual method to be evaluated, a counterfactual explanation $E(x) \in \mathcal{X}$ is generated.

To check whether $E(x)$ is justified or not, a procedure similar to LRA is used called *Vulnerability Evaluation* (VE), described in Algorithm 5, page 95: instances $B_{E(x)}$ are generated uniformly in a local region defined as the hyperball with center $E(x)$ and radius $d(E(x), b_0)$, where b_0 denotes the closest training instance to $E(x)$ that is correctly predicted to belong to the same class:

$$b_0 = \arg\min\{d(E(x), z) \mid z \in X^{f(E(x))} \text{ s.t. } f(z) \text{ is accurately predicted}\}$$

These instances are labelled with f, and the DBSCAN algorithm is also used on the ones that are predicted to belong to the same class as $E(x)$ and b_0. If $E(x)$ is assigned to the same cluster as the closest instance to b_0, then there exists an ϵ-chain linking $E(x)$ and b_0, meaning that $E(x)$ is a JCF according to Definition 3.

If not, similarly as previously, the explored area is expanded to the hyperspherical layer defined by the distance to b_1, the second closest instance from $X^{f(E(x)}$ that is correctly predicted. Just like for the LRA procedure, this step is repeated by widening the studied area as many times as necessary: if no instance from $X^{f(E(x))}$ can be connected, then $E(x)$ is labelled as being unjustified. In the end, the VE procedure returns 1 if $E(x)$ is justified, 0 otherwise. The VE procedure is highly similar in construction to the LRA procedure being applied to $E(x)$. The major difference is that it focuses on instances from the same class as $E(x)$, instead of trying to identify counterfactual regions.

An illustration of the procedure in a 2-dimensional binary classification setting is shown in Figure 4.9, page 95 for two counterfactual explanations CF_1 and CF_2 (blue dots), generated for the observation x (red dot). In the left picture, two clusters (hatched areas) are identified by DBSCAN in the explored area (blue dashed circle): CF_1 and a, the closest training instance, do not belong to the same cluster, defining CF_1 as unjustified. In the right picture, CF_2 belongs to the same cluster as a and is therefore defined as justified.

Like for the LRA procedure, the output of VE depends on the parameters n and ϵ. Since VE is to be run after LRA to focus on the instances that satisfy a certain local risk (i.e. $R_x \geq 0.25$), the same parameter values can be used.

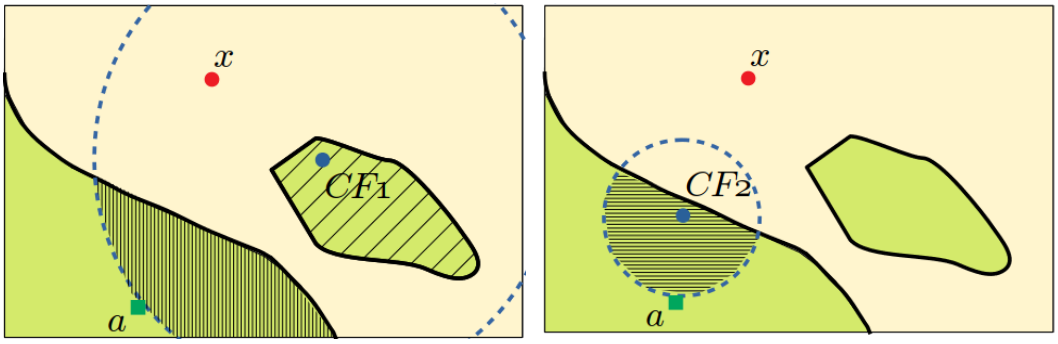


Figure 4.9: Illustration of the VE procedure for two counterfactual explanation candidates. Left: CF_1, which is not justified. Right: CF_2, justified

Algorithm 5 Vulnerability evaluation: VE procedure

Require: $E(x)$, f, X

Sort correctly predicted instances from $X^{f(E(x))} = \{b_0, b_1, ...\}$ in increasing order of their distance to $E(x)$

$B_{E(x)} = \{x_i\}_{i \leq n} \sim \text{Uniform}(\mathcal{B}(E(x), b_0))$

$B_{E(x)}^{f(b_0)} = \{x_i \in B_x \ : \ f(x_i) = f(b_0)\} \cup \{b_0\}$

Set ϵ according to Equation 4.1, page 84

$\{C_t\}_t \leftarrow \text{DBSCAN}(B_x^{f(b_0)}, \epsilon, \text{minPts} = 2)$

$\mathcal{C}_J = C_{t_0}$ s.t. $b_0 \in C_{t_0}$

$C_{E(x)}$ s.t. $E(x) \in C_{E(x)}$

$k = 0$

while $C_{E(x)} \notin \mathcal{C}_J$ **do**

  $k = k + 1$

  $SL_k = \{x_i\}_i \sim \text{Uniform}(\mathcal{SL}_k)$

  $SL_k^{f(a_k)} = \{x_i \in SL_k \ : \ f(x_i) = f(b_k)\}$

  $\{C'_t\}_t \leftarrow \text{DBSCAN}(SL_k^{f(b_k)} \cup \{b_k\}, \epsilon, \text{minPts} = 2)$

  Update $\mathcal{C}_J$ with $\{C'_t\}_t$

  Update $C_{E(x)}$

end while

$J_{E(x)} = 1$ if $C_{E(x)} \in \mathcal{C}_J$, 0 otherwise

return $J_{E(x)}$

4.4.2 | Experimental Protocol

In this section, we describe how the experiments are conducted: first, the quality criteria defined to assess the risk of generating an unjustified counterfactual example are defined in Section 4.4.2.1. Then, the counterfactual explanation approaches used in the evaluation are presented in Section 4.4.2.2. Finally, the experimental protocol

itself is described in Section 4.4.2.3.

4.4.2.1 | Quality Criteria

Given an instance x the goal is to check whether a counterfactual example $E(x)$ to explain $f(x)$ is justified or not. For this purpose, we simply define the justification score $J_{E(x)}$ as a binary score that equals 1 if $E(x)$ is justified, 0 otherwise. Again, we measure the average value $\bar{J}$ of $J_{E(x)}$ over multiple instances x and multiple runs to mitigate the impact of the random generation.

In addition to $J_{E(x)}$, the distances between each considered instance x and its generated counterfactual explanations are calculated: $d = ||E(x) - x||_2$. As a reminder, this criterion was presented as a measure of the locality of the explanation in Chapter 3. Similarly, we look at the average value $\bar{d}$ across various x.

4.4.2.2 | Counterfactual Approaches

Three post-hoc counterfactual approaches, listed below, are used for the experiments. As these approaches have been described in Section 2.3.2, page 35, a simple reminder about the key points of their functioning is given here. The first approach is HCLS (Lash et al., 2017a). The second one is a variant of LORE (Guidotti et al., 2019a), called LORE-I. The goal of LORE-I is to adapt LORE to the context of the study. In particular, LORE-I proposes to use the output of LORE to return a single instance instead of counterfactual rules, so their justification can be assessed. Additionally, the generation part of the *Growing Spheres* algorithm is used; we call it GS-G.

HCLS (Lash et al., 2017a): HCLS is a hill-climbing method using the classification confidence score returned by the black-box classifier to maximize the probability of belonging to a specified class and then to return $E(x)$. The authorized move $E(x) - x$ is limited using a budget cost associated to the move vector and a maximum budget constraint B. In the experiments, we define the budget cost as the Euclidean distance: $c(E(x)) = ||x - E(x)||_2$ and we set B to the distance to the closest ground-truth neighbor predicted to belong to another class: $B = \min_{a \in X^{\neq f(x)}} ||x - a||_2$. Setting B to this value ensures that there is enough maximum budget to reach a justified classification region, i.e. to reach $a \in X$, hence removing a potential bias in the study.

LORE-I LORE-I is a variant of the interpretability method LORE (Guidotti et al., 2019a) (LOcal Rule-based Explanations). As a reminder, LORE samples instances locally around x using a genetic algorithm, and trains a decision tree on these instances

labelled with the black-box classifier. The counterfactual explanation is built by looking for the l_0-closest decision boundary in this decision tree. The counterfactual explanation is a list of rules that leads to a counterfactual region (tree leaf) rather than to a specific instance.

To make it adapted to the VE procedure (which requires a specific counterfactual instance as input), we propose LORE-I. The goal of this new procedure is to select an instance $E(x)$ of the counterfactual region provided by LORE. This instance is identified by applying the counterfactual changes returned by LORE to the instance x, and checking that the resulting instance does satisfy $f(E(x)) \neq f(x)$. In case this condition is not satisfied, an instance is picked randomly in the leaf corresponding to the counterfactual explanation returned by LORE until it satisfies $f(E(x)) \neq f(x)$.

GS-G We use the generation step of the *Growing Spheres* approach, the proposition of Chapter 3, page 41. Since the goal of the experiment is to study the counterfactual regions detected, the projection step presented in Section 3.2.2, page 52 is not used as it would create the possibility of generating a counterfactual explanation $E(x) = e_f$ in a different classification region from $\tilde{e}$, thus making the results harder to read. Therefore, we focus on the first part of the algorithm and set $E(x) = \tilde{e}$. Since the same datasets as for the experiments of Chapter 3 are considered, we use the same values for the *Growing Spheres* parameters, n, η and w, as the ones specified in Table 3.1, page 58.

4.4.2.3 | Protocol

A random forest classifier is trained over the considered datasets. For each instance x of the associated test sets, the LRA procedure is run to compute the local risk R_x. For all the instances that face a significant justification risk (i.e. such that $R_x \geq 0.25$, where 0.25 is a threshold arbitrarily set), each considered counterfactual explanation approach is applied to generate three explanations. For each of them, the VE procedure is then applied to calculate the associated justification scores and distances, and return $\bar{J}$ and $\bar{d}$.

4.4.3 | Experimental Results

Illustration. The experimental protocol is first applied to a toy dataset to illustrate how the considered counterfactual explanation approaches behave. The same setup as the illustrative results for the LRA procedure, shown in Section 4.2.3, is considered.

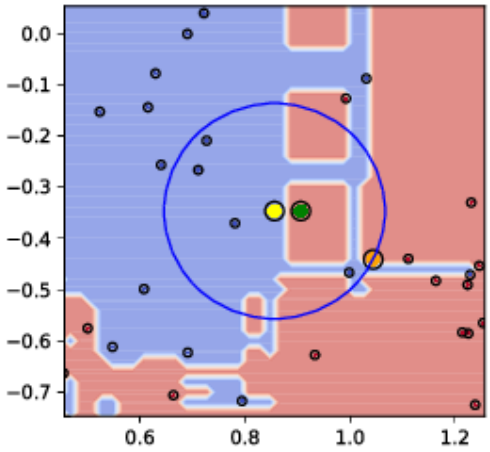


Figure 4.10: Illustrative result of the Vulnerability Evaluation procedure procedure for an instance of the half-moons dataset. The counterfactual explanation generated using HCLS (Lash et al., 2017a) is unjustified ($J_{E(x)} = 0$)

Since the instance x then considered is shown to be facing a consequent unjustification risk (the LRA procedure returns a score $R_x \geq 0.25$), we use it to test the VE algorithm.

A counterfactual example is generated using HCLS with the parameters presented is Section 4.4.2.2. The output $E(x)$ is represented in Figure 4.10 by the green instance. In this situation, $E(x)$ lies in the red square that was identified as being an unjustified classification region. The VE procedure, that indeed fails to connect $E(x)$ to any ground-truth instance from the same class, thus labels the explanation generated with HCLS as an unjustified counterfactual example ($J_{E(x)} = 0$).

This example illustrates the fact that, when there is a consequent risk R_x, post-hoc approaches can be vulnerable to it.

Quantitative Results The results of the VE procedure applied to all the considered datasets are shown in Table 4.4. As expected, when confronted to situations with a local risk of generating unjustified counterfactual examples ($R_x \geq 0.25$), the considered approaches fail to generate JCF: the proportion of generated JCF can fall as low as 30% (GS on the News Popularity dataset). This confirms the assumption that post-hoc counterfactual approaches are, by construction, vulnerable to the studied issue.

As in the first experiment, a major variability in $\bar{J}$ can be observed across the datasets. This is assumed to be caused by the variation in complexity of the decision boundaries of the classifier, not accessible in the post-hoc context.

Differences can also be noticed in the results obtained by the various counterfac-

Dataset	HCLS		GS		LORE-I	
	$\bar{J}$	$\bar{d}$	$\bar{J}$	$\bar{d}$	$\bar{J}$	$\bar{d}$
Half-moons	0.83	0.45 (0.27)	0.67	0.48 (0.26)	0.83	1.19 (0.18)
Boston	0.86	1.99 (0.88)	0.84	0.84 (1.03)	1.0	1.58 (0.98)
Credit	0.65	1.78 (0.94)	0.59	0.82 (0.71)	1.0	1.57 (1.11)
News	0.46	1.81 (0.75)	0.30	1.68 (0.99)	0.77	1.74 (0.83)
Recidivism	0.91	0.89 (1.08)	0.70	0.70 (1.09)	0.98	1.23 (0.90)

Table 4.4: Proportion of generated counterfactuals that are justified ($\bar{J}$) for vulnerable instances ($R_x \geq 0.25$), and average and standard deviation values of their distance to x.

tual approaches: HCLS and LORE-I seem to achieve better performance than GS in terms of justification across all datsets (average $\bar{J}$ across datasets respectively equals 0.74 and 0.91 for HCLS and LORE-I, against only 0.62 for GS). However, we observe that the average distance $\bar{d}$ is also higher (respectively 1.38 and 1.46 for HCLS and LORE-I, against 0.90 for GS). This can be explained by the fact that GS directly minimizes a l_2 distance (which is also the distance considered to explore the space in the LRA and VE procedures), while LORE-I minimizes a l_0 distance in a local neighborhood. By looking for counterfactuals in the direct vicinity of x, the GS algorithm thus tends to 'fall' in unjustified regions more easily than the other approaches, whereas looking further away from the decision boundary probably enables LORE-I to favor explanations located closer to ground-truth instances, assumed to be therefore more frequently justified. The Euclidean distance was presented as a measure for explanation locality for counterfactuals (see Section 3.1.1, page 42). The results shown thus tend to highlight a trade-off betwen explanation justification and locality.

Another observation is that despite achieving better performance than GS , HCLS still comes short in terms of justification. As explained in Section 2.3.2, page 35, HCLS directly tries to optimize the classification confidence to generate a counterfactual explanation. One could have expected the confidence score to be related to justification, and therefore HCLS to avoid unjustified regions. Yet, the results suggest otherwise: some unconnected regions may thus have high classification confidence. This raises the question of the relevance of classification confidence as a way to detect unjustified classification regions and guarantee good explanations. Further research is needed to answer this question, as discussed in Chapter 6.

Thus, when confronted to unconnected classification regions, post-hoc counterfactual approaches are indeed vulnerable to the risk of generating unjustified explanations. Because of the location of these unconnected regions in the feature space, the obtained results suggest that approaches that best avoid unconnected regions are the

ones that generate the least local counterfactual explanations. The creation of problematic unconnected regions is a consequence of the training of the classifier. However, it is the post-hoc paradigm that makes the interpretability methods vulnerable.

4.5 | Conclusion

This chapter proposes to study the problem of having post-hoc explanations that cannot be directly associated to any ground-truth instance by defining the *justification* property. Two procedures are proposed, LRA and VE, in order to assess the risk of creating unjustified classification regions as well as generating unjustified counterfactual explanations with existing counterfactual approaches. We suggest that the risk of creating unjustified counterfactual regions is related to the overfitting of the classifier, and that controlling overfitting could thus help reducing this risk.

In addition, we show that when the risk exists, post-hoc counterfactual approaches are vulnerable to it. Experimental results suggest that this vulnerability might be related to the locality of the explanations, as approaches that generate less local explanations are less susceptible of generating unjustified explanations. In the next chapter, we propose to define the locality of the explanation differently by focusing on another type of interpretability methods: local surrogate models.

Several improvements and extensions to the procedures LRA and VE proposed in this chapter can be envisaged. Similarly as in Chapter 3, the LRA and VE procedures rely on the generation of numerous instances, controlled by hyperparameters (here ϵ and n) that impact the obtained results and act as a control in a tradeoff between the precision of the measurement and computational performance.

Another limit is the scalability of the LRA approach to high dimensional data. While the issue of unjustified classification regions may supposedly be even more problematic when the dimension increases, the presented study may need some adaptation as the DBSCAN algorithm may face issues in high dimension. Furthermore, higher dimensional data is also problematic because of the complexity of the proposed definitions and procedures. Ensuring a better scalability of the procedure would make it easier to use it in real-world contexts. Furthermore, it would also make it more practical to use it as a quality criterion to assess the validity of a candidate counterfactual explanation for instance. A possibility would then be to using the justification of generated counterfactual explanations measured with the VE procedure as an objective for counterfactual approaches that can be directly optimized.

5

Defining Explanation Locality for Post-hoc Surrogate Models

In this chapter, a third potential issue for post-hoc local explanations is tackled: defining the locality of the explanations. As presented in Section 3.1.1, page 42, literature provides a somewhat vague definition for a local explanation, and no consensus seems to exist. In Chapter 3, we formulated the problem of the generation of a local explanation as an inverse classification problem, that we proposed to answer with counterfactual explanations. However, the studies conducted in Chapters 3 and 4 suggest that this notion of locality is associated, in the post-hoc context, to issues that may hurt the generation of explanations. Therefore, in this chapter, we propose a new formulation for the problem of generating a local explanation. Similarly as in Chapter 3, this problem is studied through the analysis of the local decision border of the classifier. However, instead of using counterfactual explanations, we propose to extend the study to another type of explainer systems: local surrogate models.

For these methods, we discuss how explanation locality can be taken into account and propose an evaluation criterion measuring the fidelity of the built surrogate model to the black-box classifier in a neighborhood of the observation whose prediction is to be explained. Using this evaluation procedure, we show that LIME, the most emblematic existing local surrogate approach, does not really ensure locality. To circumvent this issue, we propose LS, a new local surrogate approach that achieves better performance in terms of locality. Finally, we discuss the limits of the proposed study by introducing the notion of explanation generalization, and use it to draw a link between local surrogates and counterfactual explanations.

This chapter is structured as follows: in Section 5.1, we give motivations for the use of surrogate model approaches, and analyze how locality can be ensured for the

generated explanations. In Section 5.2, we highlight the existence of locality issues with the LIME approach, and propose a criterion called *Local Fidelity* to assess the extent of this problem. Section 5.3 is then devoted to the proposition of LS, a new local surrogate approach that generates more local explanations. Experiments are conducted to prove this point in the same section. Finally, in Section 5.4, a discussion about locality and the link between local surrogates and counterfactual explanations is conducted.

Part of the work presented in this chapter was the subject of the paper *Defining Locality for Surrogates in Post-hoc Interpretability*, presented at the Workshop on Human-Interpretable Machine Learning (WHI) at ICML 2018, as well as the short paper *Issues with Post-hoc Counterfactual Explanations: a Discussion*, presented at the Human in the Loop Learning workshop (HILL) at ICML 2019.

5.1 | Locality for Local Surrogate Models

The context of this chapter is similar to the rest of the thesis: we focus on approaches that try to generate local explanations in a post-hoc context, i.e. to explain the predictions of a trained black-box classifier. The post-hoc paradigm supposes that no information about the classifier nor any data is available.

In this section, we discuss existing local surrogate model approaches, which are the focus of this chapter. Contrarily to the presentation of these approaches conducted in Section 2.2, page 21, we focus here on the mechanisms of surrogate model approaches to incorporate locality to their explanations.

First, we motivate the need to study locality in Section 5.1.1. In Section 5.1.2, we then analyze how the locality of the explanations can be ensured when training a surrogate model. We then focus on the particular case of LIME, the emblematic local surrogate approach proposed by Ribeiro et al. (2016), in Section 5.1.3.

5.1.1 | Motivations for Studying Explanation Locality

As stated in Chapter 2, generating a post-hoc local explanation does not have a clear definition. Often, it is reduced to the study of the local decision boundary of the trained classifier, in the vicinity of the instance whose prediction is to be interpreted. A first type of explanation approach that follows this idea has been studied in Chapter 3: counterfactual explanations. By solving an inverse classification problem, counterfactual explanations address the question of locality by focusing on the closest touchpoint of the decision boundary. The locality of the explanation $E(x)$ was thus

measured by the Euclidean distance between the instance whose prediction is to be interpreted and the counterfactual example: $||x - E(x)||_2$.

However, in Chapters 3 and 4, experiments show that this definition of locality is associated, in the post-hoc paradigm, to issues that are potentially harmful for interpretability: the risk of generating out-of-distribution counterfactuals (see Section 3.4, page 64), and the risk of generating unjustified explanations (studied in Chapter 4, page 69). Moreover, the importance of the latter risk of unjustification has been suggested experimentally to be related to the locality of the explanation itself, i.e. its distance to the counterfactual example (see Section 4.4.3, page 97 for the results and discussion). This raises the question of the relevance of such a definition. In this chapter, we propose to question this point of view by focusing on another type of interpretability approach: local surrogate models.

Presented in Section 2.2, surrogate model approaches are post-hoc explainer systems that focus on training an interpretable model (e.g. linear regression or decision tree with low complexity) to imitate the decisions of the black-box classifier. Explanations are then extracted from this surrogate model, in the form of feature importance vectors for linear regressions for instance. In particular, the idea behind *local* surrogates is to focus on a specific part of the rationale of the black-box classifier to generate explanations for a single prediction. Studying how to define this portion of the rationale that the explanation should focus on is referred to as defining the locality of the explanation. It is the focus of this chapter.

5.1.2 | Integrating Locality in Surrogate Models

In Section 2.2, page 21, a general three-step framework is proposed for the generation of explanations using a surrogate model approach; this section gives a short reminder of this framework and analyzes how it can be used to integrate locality. In this section and in the rest of the chapter, the considered notations are the same as the ones presented in Section 2.5, page 39 and used in the previous chapters.

The three main steps that can be identified for surrogate model approaches are:

1. **Sampling step:** First, because the training set X is unavailable, a surrogate training set X_h is built by generating instances in $\mathcal{X}$. They are labelled using f, leading to the prediction vector $f(X_h)$.

2. **Surrogate training:** A surrogate model $h : \mathcal{X} \to \mathcal{Y}$ is trained on $(X_h, f(X_h))$. Optionally, weights can be assigned to the training instances.

3. **Explanation extraction:** The final explanations given to the user are extracted from h. The form of the explanations depends on the nature of the surrogate model, as well as on the information desired by the user.

In Section 2.2.2, page 24, the distinction between *global* and *local* surrogates was presented. In this chapter, we focus on *local* surrogates, which aim at generating explanations for a single prediction. Although some global surrogate models can be used to generate local explanations (such as the approach proposed by Baehrens et al., 2010, discussed in Section 2.2.2.1, page 24), the locality of their explanations is generally defined by design and does not require further study. Therefore, these approaches lie out of the scope of this chapter.

For local surrogate models, however, defining this explanation locality is required. For this purpose, local surrogate approaches adapt the first two of the aforementioned steps in order to generate a specific surrogate model for each prediction $f(x)$ to explain. Each prediction that needs to be interpreted thus requires the training of a specific local surrogate model: the notations h and X_h above are therefore replaced by h_x and X_{h_x} .

In particular, this chapter focuses on studying what sampling X_{h_x} should be performed in order to ensure local explanations. As mentioned in Section 2.2.1.1, page 22, defining an adequate sampling strategy is crucial for surrogate models, as it directly impacts the explanations. In the next section, the LIME (Ribeiro et al., 2016) approach is studied in light of this discussion.

5.1.3 | The Case of LIME

LIME (*Local Interpretable Model-agnostic Explanations*), proposed by Ribeiro et al. (2016), has already been presented in Section 2.2.2.2, page 26. This section gives a reminder of its general procedure with a focus on the way it addresses the problem of locality. LIME constitutes the most emblematic local surrogate approach, hence our focus on this approach.

To generate its explanations, it considers a post-hoc context but makes the assumption that the black-box classifier f returns a continuous classification score, instead of only a predicted class. We propose to identify the following instanciation of the three-step surrogate framework presented in the previous section for LIME:

1. **Sampling step:** X_{h_x} is generated by drawing instances following a normal distribution with the same mean and standard deviation as the data given as input (e.g. the original training set X if available), independently of the location of x.

For the labels $f(X_{h_x})$, LIME uses continuous classification confidence scores returned by f.

2. **Surrogate training:** A linear regression is trained to approximate the continuous classification scores $f(X_{h_x})$. In order to make the approximation local, each instance of X_{h_x} is associated to a weight calculated using a kernel function (RBF kernel by default): instances closer to x are assigned a higher importance weight during the training. A regularization parameter is included in the loss function, optimized with a Lasso regression algorithm (Tibshirani, 1996) to train the surrogate model. This allows to make the regression (and thus the final explanation) sparse, hence more understandable.

3. **Explanation Extraction:** Finally, human-interpretable explanations for the prediction $f(x)$ are generated by extracting the regression coefficients of the trained surrogate h_x.

The locality of the explanation generated by LIME is therefore a consequence of the use of the RBF kernel. This kernel is controlled by its width σ, which is a parameter the user can set. Increasing σ makes the kernel function more discriminant, therefore putting more emphasis on the instances located closer from x. In order to help setting the value of this parameter, Ribeiro et al. (2016) propose the heuristic definition $\sigma = 0.75\sqrt{dim(\mathcal{X})}$.

However, despite the use of this kernel, the way the surrogate dataset X_{h_x} is generated raises questions about the locality of the associated surrogate model. Indeed, using a normal distribution over the whole feature space means that the sampled instances X_{h_x} are essentially the same for every x. Thus the sampling step does not contribute in making the explanation specific to $f(x)$. In the next section, we investigate the explanations learned by LIME and highlight some issues resulting from this sampling approach.

5.2 | Measuring Locality: the *Local Fidelity* Criterion

The global sampling strategy considered in LIME raises questions about the efficiency of the approach in correctly capturing the local nuances of the decision boundary of f. In this section, we propose to visualize these issues in a simple scenario. The resulting observations help us formulate an intuitive desideratum for the behavior of local surrogates, which leads us to propose an evaluation criterion for local surrogate

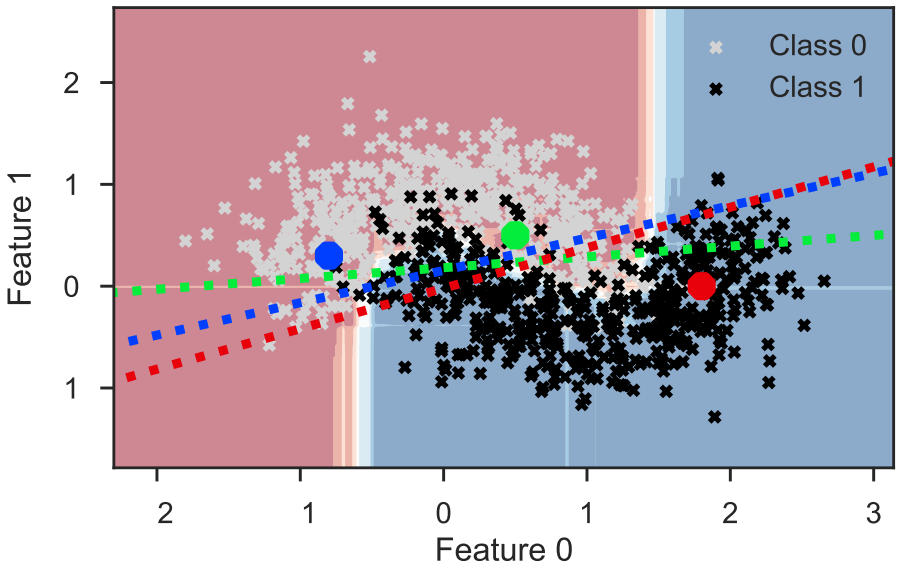


Figure 5.1: Local linear approximations (dashed colored lines) provided by LIME for three predictions (big color points) on the half-moons dataset. The black-box decision regions are represented by the blue and red areas.

models. We call this criterion *Local Fidelity* and show that it efficiently captures the local behavior of surrogate approaches.

In Section 5.2.1, we use a toy dataset to visualize the local decision boundaries learned by LIME. In Section 5.2.2, we propose the definition of the *Local Fidelity* score. Finally, we assess the efficiency of the proposed criterion in the previously used toy dataset, and discuss the issues of LIME in Section 5.2.3.

5.2.1 | Illutrative Example

We propose a method to visualize the explanations learned by LIME and the locality issues that may arise in a 2-dimensional context. A black-box classifier (here a Random Forest algorithm) is trained on 70% of the instances of the half-moons dataset (already described in previous chapters, see Section 3.3.1, page 57). The parameters of the classifier are optimized for accuracy using cross-validation. The final accuracy measured over the test set is 0.93. The decision regions of f are represented by the red and blue areas in Figure 5.1, while the white and black crosses represent the training instances X. We apply LIME on three instances of the test set, chosen for their location in the feature space (blue, green and red instances of Figure 5.1).

For each instance x, the surrogate model h_x is a linear regression. Therefore, no actual decision boundary is available to be visualized. To circumvent this issue, we draw the line corresponding to the hyperplane defined by $h_x(a) = 0.5$, for $a \in \mathcal{X}$. This allows for easy visualization of the linear coefficients of h_x, which correspond to both the *direction* to the learned decision boundary and the final explanation provided by LIME. This is important to keep in mind as the specific threshold $h_x(a) = 0.5$ is chosen arbitrarily and does not particularly represent a class change intended by the authors: the final explanation of LIME is the direction to this decision boundary, not the boundary itself. These hyperplanes are represented by the blue, green and red dashed lines in Figure 5.1 and can be associated to the final explanations returned by the LIME procedure for each instance: the slopes of the dashed lines can be expressed with the values of the regression coefficients, which constitute the final explanation given to the user. In this situation, the green dashed line being more horizontal than the other ones means that the coefficient associated to Feature 0 is relatively smaller for the green instance than for the other two. LIME is used as provided by the library developed by its authors[1] with default parameters, after slight modifications to return the material needed to plot the LIME decision boundary.

The first observation we make is that the decision boundaries learned by LIME do not really match the direction of the local decision boundaries of the black-box classifier. Indeed, for the three considered instances, much more "vertical" (aligned with Feature 1) borders could be expected, especially for the blue and red instances. For the green instance, looking at the shape of the closest decision boundary of f, a negative slope could have been expected for the decision boundary learned by LIME rather than a positive one. Another observation is that the slopes of the decision boundaries learned by LIME for these 3 instances, scattered across the dataset, are quite similar, despite their respective local black-box decision boundaries being seemingly different.

These observations show that the decision boundaries learned by LIME tend to approximate the global shape of the black-box decision boundary rather than the local ones. As a result, this leads to local feature influences being mitigated in favor of global feature influence, meaning the explanation is not local enough. In the following section, we use these observations to propose a criterion to measure the locality of the learned explanation and assess the importance of this issue.

[1] https://github.com/marcotcr/lime

5.2.2 | Measuring Locality for Surrogates: Local Fidelity

Since surrogate model approaches try to imitate the black-box classifier as much as possible, a comparison between the predictions of f and the surrogate model is usually conducted. The associated metrics is referred to as *fidelity*. Several definitions for this criterion exist (see for instance Craven and Shavlik, 1996; Hara and Hayashi, 2016; Ribeiro et al., 2016). However, they rely on comparing the predictions of the classifier f and the surrogate model globally. Therefore, we propose a new definition to incorporate the notion of locality. This new criterion is defined in Section 5.2.2.1, and its two parameters are discussed in turn in Sections 5.2.2.2 and 5.2.2.3.

5.2.2.1 | Proposed Criterion

In order to analyze the *local* behavior of the black-box classifier, an intuitive proposition thus revolves around measuring the fidelity of the surrogate model in a local neighborhood. For a given instance x and trained surrogate h_x, we therefore propose the *Local Fidelity score* (LF), defined as the fidelity of h_x to f within a neighborhood $\mathcal{V}_x$ around x:

$$LF(x, h_x, \mathcal{V}_x) = Acc_{\mathcal{V}_x}(f, h_x) \tag{5.1}$$

where Acc is a metrics to evaluate how similar the predictions of f and h_x are, calculated over instances of $\mathcal{V}_x$. The higher this value is, the better the surrogate model is at replicating the local behavior of the black-box classifier.

Choosing the criterion Acc, that is to say defining how to evaluate the fidelity of the surrogate model, as well as delimiting the right neighborhood $\mathcal{V}_x$ is obviously crucial. These two elements are discussed in the following subsections.

5.2.2.2 | Choosing the Fidelity Criterion Acc

As mentioned earlier, in the case of LIME, h_x is trained on the classification probabilities of f. A proposition by Ribeiro et al. (2016) is to measure the quality of this approximation using the R-squared score (coefficient of determination), weighted with the RBF kernel used in the training. However, this regression score does not seem particularly adapted to the context of post-hoc interpretability. Indeed, the R-squared coefficient automatically and spuriously increases with the number of considered attributes (see for instance, among many references: Berk, 2004). This can be harmful in the context of interpretability, since the number of chosen attributes also impacts the complexity of the explanation.

Furthermore, in the context of this work, we are interested in evaluating the final explanation provided by LIME, rather than measuring the quality of the linear estimation of the values of the classification probabilities. Yet, since h_x is linear, this final explanation is given by the coefficients of the regression, i.e. the learned *direction* of the decision boundary of f. In this context, using classification metrics that evaluate the direction of the class change seems to be more relevant than the R-squared coefficient. We propose to use as a measure of accuracy the *Area Under Curve* ($Acc = AUC$) to compare the predictions of h_x and f.

It should be noted that using other fidelity metrics may be more relevant for other surrogate model approaches. For instance, in the case where h_x is a decision tree such as by Hara and Hayashi (2016) and Guidotti et al. (2019a), the final explanation is not the direction to the decision boundary but rather a decision rule. In this context, assessing that h_x correctly replicates the *class* predicted by f is important. Therefore, using the classification accuracy may be more relevant than the AUC score.

5.2.2.3 | Delimiting the Neighborhood $\mathcal{V}_x$

Defining a local region. Delimiting the right neighborhood $\mathcal{V}_x$ is obviously crucial to properly measure the quality of the learned approximation. Since $\mathcal{V}_x$ is to be used to evaluate h_x, it needs to be defined independently. We propose to define the neighborhood of x as $\mathcal{V}_x = \mathcal{B}(x, r_{fid})$, the l_2-hypersphere centered on x and of radius r_{fid}, discussed below. One of the upsides of using this intuitive definition is that the neighborhood $\mathcal{V}_x$ then depends on a single parameter, this radius r_{fid} of the fidelity hypersphere, making it easy to use. Through this parameter, the size of the region in which the fidelity of the surrogate is evaluated is controlled. The parameter r_{fid} thus acts as a proxy for the degree of locality considered to evaluate the approximation of the behavior of the black-box.

The value of r_{fid} needs to be set appropriately for each instance x. Choosing a too low value creates the risk of defining a neighborhood $\mathcal{V}_x$ that only contains instances x_i that satisfy $f(x_i) = f(x)$, meaning that no part of the decision boundary of f is included in $\mathcal{V}_x$. This is of course not desirable since understanding the local behavior of f does require to study its decision boundary. A suitable minimum value for r_{fid} thus equals to the distance between x and the l_2-closest touchpoint of the decision boundary of f, with the latter being however unobservable in the post-hoc context. On the contrary, setting too high a value r_{fid} would mean looking at the behavior of the surrogate model at a global scale rather than a local one. In practice, interesting insights can be gained by looking at how the LF score evolves for various

values of r_{fid}. Indeed, this may give intuitions about the area in which the behavior of the black-box is being properly (or not) replicated.

For the sake of clarity, the value of the parameter r_{fid} will be expressed in the rest of the thesis as a percentage of the maximum distance between the instances of the dataset (unavailable in the post-hoc context) and x, whose prediction is being interpreted.

Generating instances in $\mathcal{V}_x$. Having defined $\mathcal{V}_x$, we propose to define a set of n instances, where n is a fixed parameter, $\{x_i\}_{i=1,\dots,n} \subseteq \mathcal{V}_x$, drawn following a uniform distribution $\mathcal{U}$ over $\mathcal{V}_x$. The evaluation of the fidelity thus depends on this parameter n. We call V_x the resulting set of instances, defined as:

$$V_x = \{x_i\}_{i=1,\dots,n} \sim \mathcal{U}_{\mathcal{B}(x,r_{fid})}$$

Ideally, n should be as high as possible to make sure that the neighborhood $\mathcal{V}_x$ is correctly covered. In practice, we set this value to an arbitrarily high number (e.g. 10000).

Final definition. In light of these discussions, we thus consider the following final definition for LF:

$$LF(x, h_x, r_{fid}) = AUC_{V_x}(f, h_x) \tag{5.2}$$

In order to get insights about the average quality of a local explanation approach over a whole dataset, we are also interested in calculating the average LF score over the instances of the test set. We note $\overline{LF}$ this value.

5.2.3 | Illustrative Result

We use the proposed criterion to evaluate the locality of the explanations generated with LIME on the half-moons dataset. Figure 5.2 first shows the value of the LF score obtained by the explanations generated for the 3 instances shown in Figure 5.1, page 106, as a function of r_{fid}. The number of instances n generated in $\mathcal{V}_x$ is arbitrarily set to 1000. Each curve matches in color its associated instance.

We observe that at a local scale, that is to say for low values of radius r_{fid}, the obtained Local Fidelity score is significantly worse than at a global scale, for higher values of r_{fid}. This is in agreement with the previous discussion (see Section 5.2.1, page 106): the approximation learned by the local surrogate of LIME is influenced by global features. This leads to a decrease in the local fidelity of h_x at the local scale,

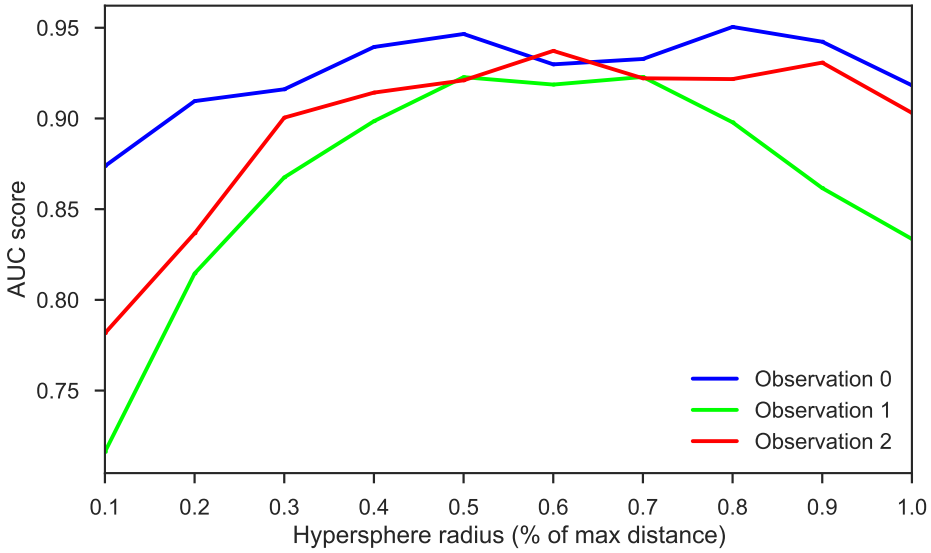


Figure 5.2: Local Fidelity for the explanations provided by LIME for 3 instances (shown in Figure 5.1, page 106) of the half-moons dataset, for increasing values of r_{fid}.

but to an increase of LF for higher values of r_{fid}. This seems especially true for the green and red instances: their expected local decision boundaries appear to be the most different from the one learned by LIME (see Figure 5.1). This illustrates the relevance of the LF criterion to assess the locality of the explanations. Moreover, our assumption that the explanations generated using LIME are quite global is confirmed.

This is further observed by conducting the same evaluation for all the instances of the half-moons dataset: Figure 5.3 shows a heatmap where each point of the test set is colorized depending of the LF score of LIME for $r_{fid} = 0.3$ (value chosen by studying the LF curves obtained for several instances). We can observe that LIME has more difficulty approximating areas where the local decision boundary of the black-box classifier f differs from the decision boundary approximating the whole dataset. This further suggests that when features with a local influence differ from the features that have a global influence, LIME seems to perform poorly: it tends to generate global explanations.

These illustrative results thus highlight the existence of locality issues for the surrogate model LIME. Our hypothesis for the rationale behind this observed behavior is that, for a local surrogate to fit properly a local decision boundary of the black box, the instances used to learn the surrogate X_{h_x} should also be generated *locally*. Assigning weights to the sampled instances X_{h_x} with a kernel function of the distance

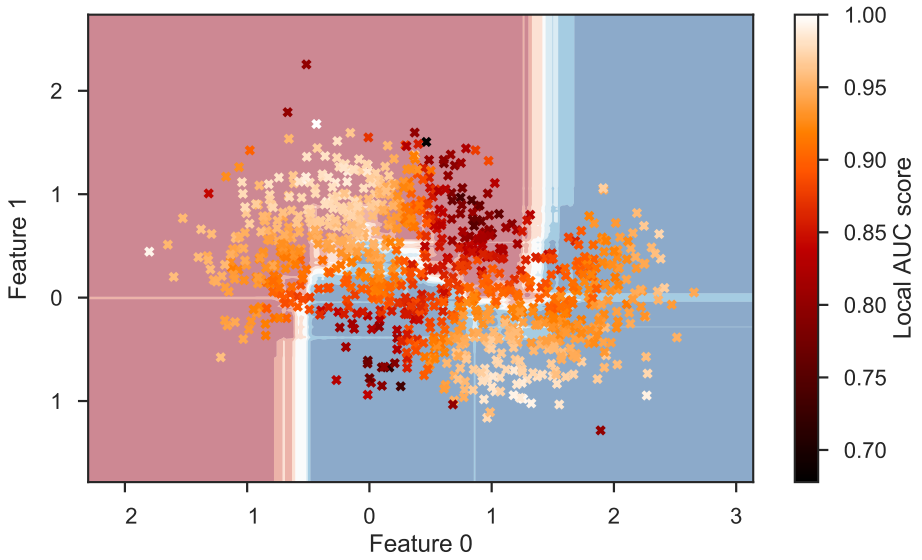


Figure 5.3: Visualization of the Local Fidelity score ($r_{fid} = 0.3$) of LIME for each instance of the half-moons test set.

to the observation x to explain helps LIME to focus on the local decision boundary. However, because these instances are sampled over the whole dataset, this local focus is not enough. This is illustrated by the LF score values obtained by LIME for low values of r_{fid}. Therefore, LIME fails to properly generate local explanations. In the next section, we propose a new sampling approach to confirm this assumption and correct this bias.

5.3 | A New Local Surrogate Approach: the LS Algorithm

In order to overcome the locality issues highlighted in the previous section, we propose a new local surrogate approach, called LS (*Local Surrogate*). LS relies on a new sampling step in the local surrogate training workflow presented in Section 5.1.2, page 103. In particular, we propose to center the sampling directly on the local decision boundary of f to ensure that the approximation correctly captures the local behavior. In Section 5.3.1, the LS approach is detailed. Experiments are then conducted in Section 5.3.2 to show that LS does lead to more local explanations.

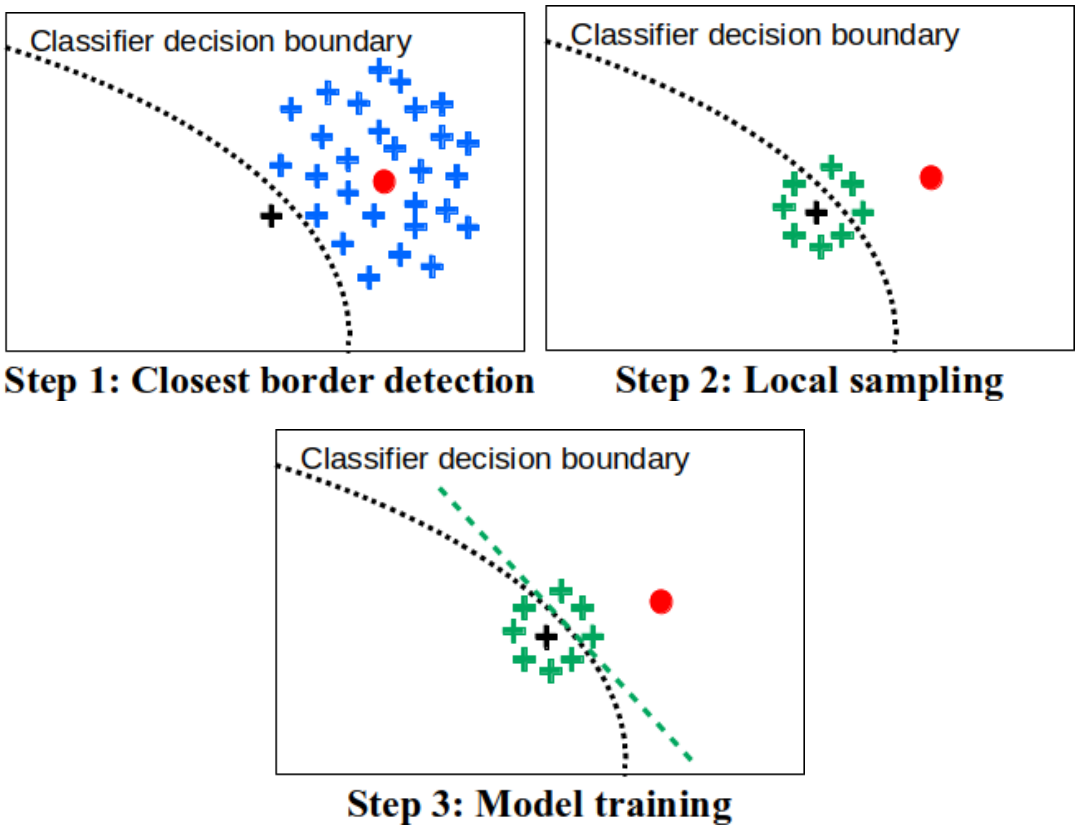


Figure 5.4: Illustrations of the LS algorithm applied to an instance represented by the red dot. Top left: the *Growing Spheres* algorithm (blue crosses) is used to detect the closest touchpoint of the decision boundary of f, x_{border} (black cross). Top right: N instances are sampled uniformly around x_{border} and labelled with f (green crosses). Bottom: the surrogate model h_x is trained on these instances.

5.3.1 | The Local Surrogate Procedure (LS)

This section first describes the whole proposed LS procedure, and then discusses its parameters in Section 5.3.1.2.

5.3.1.1 | Proposed Procedure

The main idea behind LS relies on the assumption that, in order to approximate a local decision boundary, the data X_{h_x} used for the training of the surrogate model should be sampled precisely around the decision boundary itself. This may seem surprising, since the criterion to be maximized, the Local Fidelity score, is calculated in an area $\mathcal{V}_x$ centered around the instance x: an intuitively reasonable proposition would rely on sampling X_{h_x} in $\mathcal{V}_x$ to make the approximation as efficient as possible. However,

the final objective of h_x remains to approximate the classification decision boundary of f. Hence, focusing the sampling around this boundary rather than around x is important.

Given an individual prediction to explain x and a black-box classifier f, our proposition for the sampling stage is the following. First, the closest decision boundary of f is detected by looking for the closest instance x_{border} satisfying $f(x_{border}) \neq f(x)$. This, of course, can be achieved by generating a counterfactual explanation for x in the sense of Equation 3.2, page 47. Since no information about f nor any data is supposed to be available, we propose to identify x_{border} using the generation part of *Growing Spheres* (see Algorithm 2, page 54), introduced in Chapter 3.

Once x_{border} is found, N training instances are sampled uniformly in the vicinity of x_{border}. This neighborhood is defined as the hypersphere of radius r_x (discussed in the next section) centered on x_{border}:

$$X_{h_x} \sim \mathcal{U}_{\mathcal{B}(x_{border}, r_x)}$$

This sampling allows LS to perform an approximation of the local decision boundary of f.

Finally, the surrogate h_x itself is trained on X_{h_x}. The method is detailed in Algorithm 6 and illustrated in Figure 5.4.

5.3.1.2 | LS Parameters

As discussed in the previous chapters, using adequate hyperparameters for *Growing Spheres* (the number of instances generated at each step n, the initial radius η and the radius step w) is required to accurately detect the closest decision boundary of f. However, it should be noted that in the context of this chapter, the approximation errors of the *Growing Spheres* algorithm studied in Section 3.2.1, page 45, are not as problematic. Indeed, the found solution x_{border} is only used to center the sampling: finding an optimal solution of the counterfactual problem is not required to perform a relevant sampling. In the experiments, we set the values of these hyperparameters depending on the dimensionality of the problem to ensure reasonable computational time.

The LS procedure relies on two other parameters: r_x, which defines the radius of the sampling area, and N, the number of generated instances. These two parameters obviously impact the efficiency of the procedure and the quality of the approximation. The value of N can be set to an arbitrarily high value. Setting the value of r_x is more complex, as its impact on the LF score depends on the considered instance x. In

Algorithm 6 *Local Surrogate* algorithm.

Input: $x \in \mathcal{X}$, $f : \mathcal{X} \to \mathcal{Y}$, r_x, N
$x_{border} \leftarrow GrowingSpheresGeneration(f, x)$ using Algorithm 2, page 54
$X_{h_x} \leftarrow$ Draw uniformly N instances in $\mathcal{B}(x_{border}, r_x)$
$Y_{h_x} \leftarrow f(X_{h_x})$
Train h_x on (X_{h_x}, Y_{h_x})
Return: h_x

practice, we propose to set r_x by performing a grid search to maximize the LF score for a given value of r_{fid}.

The choice of the surrogate h_x (e.g. linear model vs. decision tree) also impacts the overall fidelity. For instance, if the local decision border of f presents some major non-linearities, using a decision tree algorithm for h_x and increasing the number of training instance N will increase the overall fidelity of the surrogate. This link is further discussed in Section 5.4.

However, unless specified, the surrogate model h_x considered in the rest of this chapter is a linear regression model trained on classification confidence scores, in order to ensure fair comparison with LIME and show the efficiency of this new sampling procedure.

5.3.2 | Experiments

Designed to ensure that the generated explanations are more local, the new sampling procedure of the LS approach is confronted to LIME in this section. In particular, we first analyze using illustrative results how LS performs in terms of the LF score proposed in Section 5.2.2, page 108. After describing the experimental protocol in Section 5.3.2.1, illustrative and quantitative results are given, respectively in Section 5.3.2.2 and Section 5.3.2.3.

5.3.2.1 | Experimental Protocol

Datasets In addition to the 2-dimensional dataset half-moons, the datasets considered for this experiment are the following: Breast Cancer dataset, German Credit dataset, Online News Popularity dataset and Tennis Major Tournament Match Statistics. All these datasets are openly available on the UCI repository (Dua and Graff, 2017). Like in the previous chapters, the unordered categorical features are dropped and the data is rescaled.

Competitors The proposed algorithm LS is compared to LIME. LIME depends on a single parameter σ, the width of the RBF kernel that can be set by the user to enforce locality in a more or less aggressive fashion. The default kernel width value proposed by the authors is $\sigma = 0.75\sqrt{dim(\mathcal{X})}$.

However, we have illustrated in Section 5.2.1, page 106 that this default value is not satisfying to ensure local explanations. In this context, a natural competitor for the proposed LS approach is to try to find a value of σ that maximizes the local fidelity. It is expected that reducing the value of σ is desirable. In this experiment, we perform a grid search on σ to find its optimal value for the corresponding $\mathcal{V}_x$. We call the resulting approach LIME-K.

Protocol Each dataset is split into a training and test sets ($70\% - 30\%$). For each dataset, a random forest classifier with 200 trees and default parameters from the scikit-learn package is trained on the training set. As always, it is important to note that the considered local surrogates approaches are model-agnostic, and therefore the choice of the classifier does not matter. For each instance of the test set, LIME, LIME-K as well as the proposed approach LS are applied to generate local explanations, the local fidelity of which is measured using the LF score ($r_{h_x} = 0.3$, chosen as an arbitrarily low value). The average and standard deviation values of the obtained LF scores across the considered test sets are calculated.

5.3.2.2 | Illustrative Results

In order to give insights about the efficiency of the proposed procedure, we compare the explanations generated for an instance of the half-moons dataset, in the aforementioned setup. Figure 5.5 shows the decision boundaries of the different competitors for a randomly picked instance of the dataset, represented by the green dot (the instance is different from the ones considered in Section 5.2.1). The decision boundary of LIME (default kernel width $\sigma = 0.75\sqrt{dim(\mathcal{X})} \approx 1.06$ in this case) is shown in green. The decision boundary of LIME-K, whose optimized kernel width parameter is set to $\sigma \approx 0.5$ here, is shown in blue. Finally, the decision boundary of LS (trained with $r_x = 0.3$) is shown in red. The closest touchpoint of the local decision boundary x_{border}, found with the *Growing Spheres* algorithm, is represented by the red instance.

Similarly to what was observed in Section 5.2.1, page 106, the decision boundary learned by LIME (green dashed line) is almost horizontal, the same way as the one of a global model would be. Even though reducing the kernel width helps making the learned decision boundary more local for LIME-K (blue dashed line), we observe

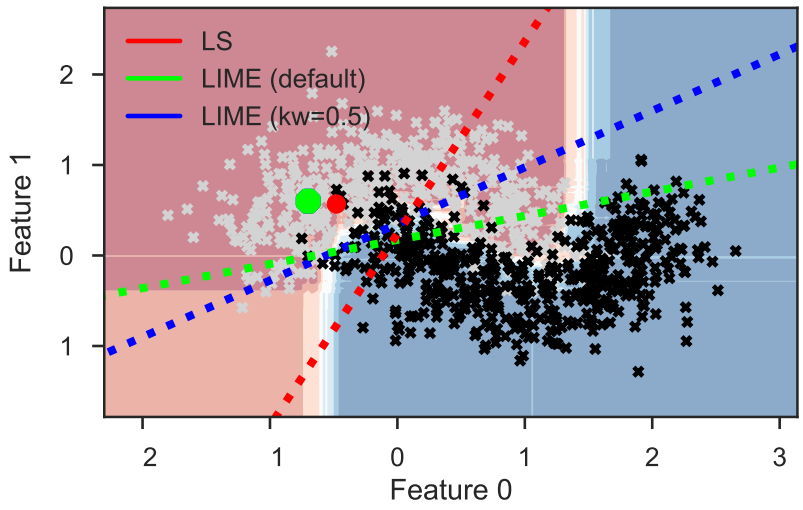


Figure 5.5: Example of the linear approximations performed by LIME (with default kernel width), LIME-K (with reduced kernel width) and the proposed Local Surrogate for a randomly picked instance from the half-moons dataset (green dot x). The red dot corresponds to the closest instance from the other class x_{border} found with the *GrowingSpheres* algorithm.

that it seems to remain not enough to approximate properly the local border of the black-box classifier f. In comparison, LS (red dashed line) seems to approximate a much more local border direction: its slope is much more vertical, which matches the expected behavior for this instance since Feature 1 seems relatively less important locally. Because its sampling is centered on the decision boundary, LS provides a more local, therefore satisfying, explanation.

These results can be further observed in Figure 5.6, which shows the LF scores obtained for these three competitors for the same instance x as a function of r_{fid}. The first observation is that for most values of r_{fid}, the LF scores for LS and LIME-K are quite similar. On the other hand, LIME achieves lower overall LF scores. However, for small values of r_{fid} (between 0.1 and 0.2), the proposed approach LS achieves higher Local Fidelity than both LIME and LIME-K. This tends to confirm our initial assumption that even if weighting does help ensuring some degree of locality to the generated explanation, the global sampling performed by LIME and LIME-K tends to mitigate the local feature effects in favor of the global ones. As a result, local explanations are only guaranteed by LS.

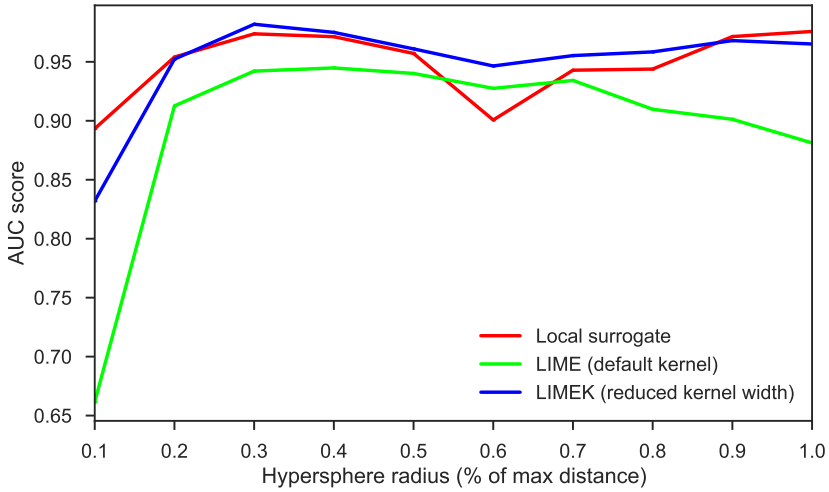


Figure 5.6: Local Fidelity scores for the instance of the half-moons dataset shown as the green dot on Figure 5.5, for several values of r_{fid}.

5.3.2.3 | Quantitative Results

The average Local Fidelity score $\overline{LF}$ is then calculated across all test instances for the datasets listed in Section 5.3.2.1. As explained in the protocol, the value of σ for LIME-K is chosen through grid-search to maximize Local Fidelity. The value of r_x (sampling radius for LS) is set up arbitrarily depending on the dimension of the problem, similarly to the value of r_{fid}. The results are shown in Table 5.1.

The average Local Fidelity of the proposed Local Surrogate approach is markedly higher than the one obtained with LIME (between $+0.08$ and $+0.18$ across all datasets) and LIME-K (between $+0.01$ and $+0.15$ across all datasets). The gain in Local Fidelity heavily depends on the considered dataset, and in some cases is not visible. For instance, the $\overline{LF}$ scores for LS and LIME-K are equivalent for the half-moons dataset. However, this tends to show that in general, despite an optimized kernel width, LIME fails to approximate properly the black-box classifier locally.

The higher standard deviation values obtained with LIME and LIME-K (compared to LS) also confirm our observations of Section 5.2.1, page 106. Indeed, the global and local decision boundaries of f may, for some instances, match. In such case, the explanation learned by LIME, which matches the global shape of the decision boundary of f, is accurate and results in a high LF score. However, when the local and global decision boundaries of f are not similar, the LF score decreases significantly. This variation thus results in a high variability of the LF score. This confirms

Dataset	LIME	LIME-K	LS
half-moons	0.89 (0.07)	0.96 (0.06)	**0.97 (0.03)**
cancer	0.86 (0.07)	0.87 (0.07)	**0.96 (0.02)**
credit	0.67 (0.21)	0.70 (0.18)	**0.85 (0.12)**
news	0.64 (0.10)	0.67 (0.10)	**0.79 (0.07)**
tennis	0.85 (0.12)	0.83 (0.13)	**0.98 (0.02)**

Table 5.1: Average and standard deviation of the Local Fidelity scores ($r_{fid} = 0.05$) for LIME, LIME-K and *LS* (our proposition) across all test instances for 5 datasets.

the observations previously made with Figure 5.3, page 112 for LIME: because of the local decision boundaries of the instances, the LF score is high for the instances represented in white in the heatmap. It is low for the instances represented in red, leading to a high standard deviation value: 0.07 for the half-moons dataset, up to 0.21 for the German credit dataset.

Thus, LIME generally fails to generate local explanations consistently over the whole dataset. On the other hand, LS achieves better Local Fidelity across all datasets with lower standard deviation, thus providing more accurate local explanations for the predictions made by f.

5.4 | Discussion: Local Surrogates and Counterfactuals

In Chapter 3, we proposed to answer the problem of generating a local explanation in the post-hoc framework using counterfactuals. This definition was then questioned in this chapter, where we proposed to focus on local surrogate model approaches and proposed a new criterion, Local Fidelity, to measure the explanation locality. In this section, we propose to put these two definitions in parallel and make a connection between local surrogate approaches and counterfactual explanations. This connection is discussed using two perspectives. First, in Section 5.4.1, we propose to define counterfactual explanations as the most local surrogate model possible. Then, in Section 5.4.2, we propose a second connection between these two families of interpretability approaches using a new concept: explanation generalization.

5.4.1 | The Most Local Surrogate

Both the proposed criterion, the LF score, and the proposed local surrogate approach, LS, rely on hyperparameters: regarding LF, r_{fid}, the radius of the studied local area $\mathcal{V}_x$,

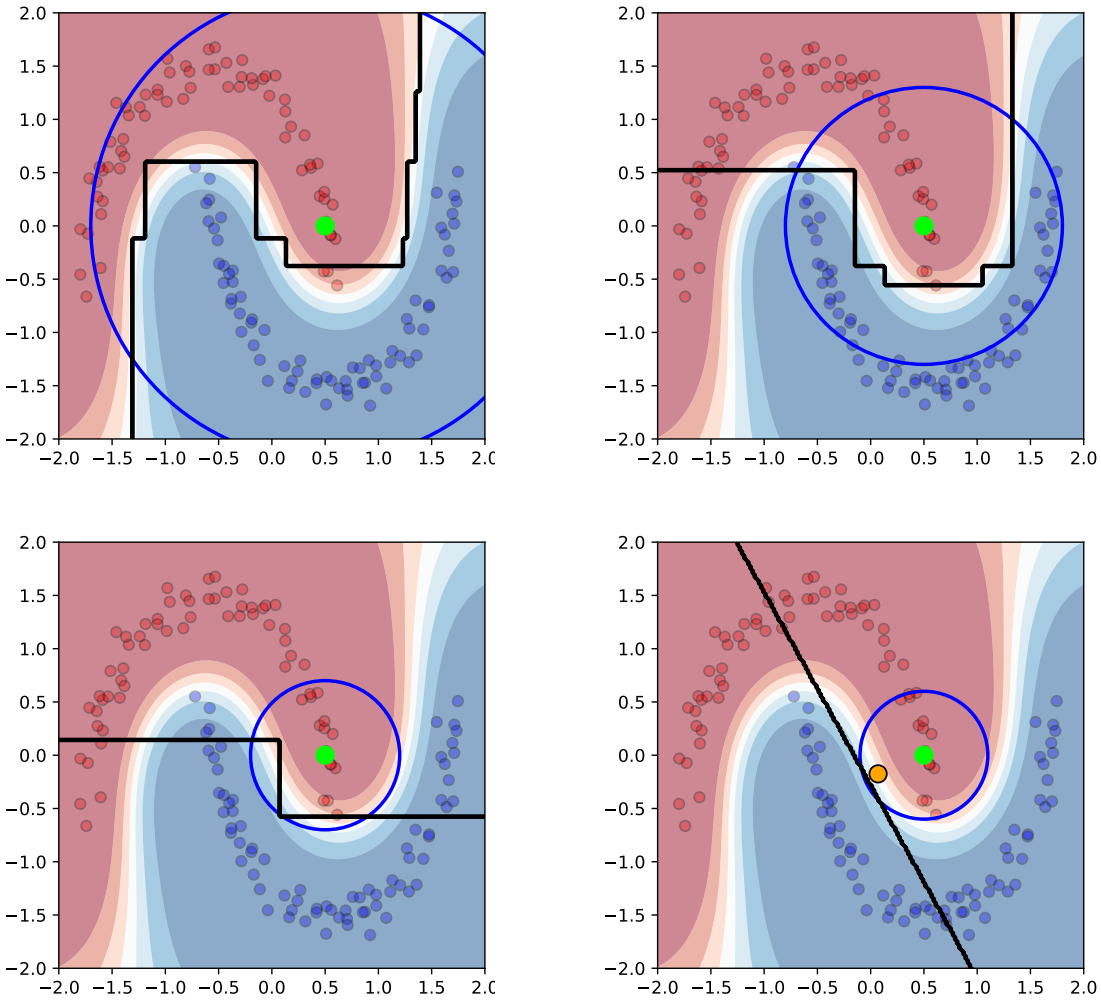


Figure 5.7: Local decision boundaries learned by LS with a surrogate h_x (model and parameters) chosen to ensure a LF score higher than 0.95, for several values of r_{fid}: from left to right, top to bottom: 0.94, 0.51, 0.30, 0.26 (relative value to the maximum distance).

controls the area of the feature space that the local explainer focuses on. For LS, r_x controls the sampling width. Section 5.3.1.2, page 114 studies the impact of r_x on the local fidelity. In this section, we consider this impact in interaction with r_{fid}.

Impact of the choice of the surrogate model. The parameter r_{fid} controls the part of the decision boundary of f that is to be included in the studied area $\mathcal{V}_x$, hence that the surrogate model focuses on. Therefore, in order to ensure the highest Local Fidelity score possible, the value taken by r_x should be defined such that this portion of the decision boundary is also covered by the sampling area. However, since the

goal is ultimately to replicate this decision border, the choice of the surrogate h_x naturally also comes into play. For a given instance x and radius r_{fid}, the shape of the section of the decision boundary that is included in $\mathcal{V}_x$ impacts the complexity of the surrogate model that is required to correctly approximate it. In the case of a complex local decision border (i.e. with numerous non-linearities and variations), defining h_x as a linear model would lead to low local fidelity. Reciprocally, imposing a linear model for h_x would mean that only a really narrow neighborhood $\mathcal{V}_x$ can be correctly approximated.

Illustrative experiment with the half-moons dataset. A visualization of this idea is shown in Figure 5.7 for a given instance x of the half-moons dataset, on which a black-box classifier f (here a SVM classifier with RBF kernel) is trained. Each image corresponds to a different value of the radius r_{fid}. From left to right, top to bottom, these values are: $0.94, 0.51, 0.30, 0.26$. These values are chosen so as to properly illustrate the discussion by delimiting decision boundaries of various shapes. For each associated neighborhood considered $\mathcal{V}_x$, the LS approach is used: a surrogate model h_x is chosen to *correctly* approximate f locally. We say that a local approximation is correct if it satisfies the condition: $LF(x, h_x, r_{fid}) \geq 0.95$. The decision boundary of h_x is represented in each image by the black line. In left and top images, a linear model is not complex enough to satisfy $LF(x, h_x, r_{fid}) \geq 0.95$ in the considered neighborhood. We thus change the surrogate model to a decision tree. Beside the Local Fidelity, we measure the complexity of the associated explanation by counting the number of nodes of each tree. In each case, we perform a grid search to find the least complex model satisfying $LF(x, h_x, r_{fid}) \geq 0.95$. The resulting complexity of the surrogate models h_x are, from left to right, top to bottom: $31, 19$, and 7. In the last image, the studied neighborhood $\mathcal{V}_x$ allows for a correct approximation of the decision boundary using only a linear model. Thus, this suggests that the narrower $\mathcal{V}_x$ is, the simpler the surrogate needs to be in order to correctly approximate the local decision boundary of f. Further experiments are required to assess the existence of a direct correlation.

Link between local surrogates and counterfactuals. Pursuing this idea, it is possible to draw a connection between local surrogate models and counterfactual explanations. As mentioned in Section 5.2.2, page 108, the value of r_{fid} can be reduced up to $||x - x_{border}||_2$. Satisfying this condition means that the only part of the decision boundary of f included in $\mathcal{V}_x$ is in a close vicinity of x_{border}. In such a case, the final explanation associated to a linear surrogate model h_x with high LF score

would be pointing to this small portion of the decision boundary. Hence, this explanation would be similar to the vector $x_{border} - x$. This vector is also the counterfactual explanation vector returned by *Growing Spheres* in this situation. This idea can be visualized in the bottom right image of Figure 5.7. The orange instance is a counterfactual example e_f returned by *Growing Spheres*. In this case, the direction to the local decision boundary learned by the surrogate model is similar to the vector $e_f - x$. One of the reasons behind this link is the fact that the distance used to defined the neighborhood $\mathcal{V}_x$ is the same as the one considered in the minimization problem of the counterfactual explanation approach. Of course, this is especially true for LS since it also uses the *Growing Spheres* algorithm to center its sampling. Nevertheless, further research may help highlighting this link for local surrogate approaches in general, as long as they satisfy the local fidelity criterion. In this context, counterfactual explanations can be seen as "the most local" version of local surrogate explanations.

5.4.2 | Explanation Generalization

We propose another discussion for the link between counterfactual explanations and surrogate models by introducing the notion of explanation generalization.

Given an instance x and its local neighborhood $\mathcal{V}_x$, let h_x be a trained surrogate model such that h_x correctly approximates f over $\mathcal{V}_x$. In the context of this chapter, the explanation associated to the considered linear surrogate model h_x is a direction pointing to the closest decision boundary. In this case, having h_x that correctly approximates f over $\mathcal{V}_x$ means that the same direction can be used for other neighboring instances in $\mathcal{V}_x$ to alter their predictions.

Yet, as mentioned earlier, this direction can be associated to a counterfactual explanation indicating the required change to alter the prediction of x. Therefore, this correct linear approximation implies that the same counterfactual direction can be used for other instances in $\mathcal{V}_x$. The area defined by high local fidelity surrogate models can thus be seen as an area in which the same counterfactual explanation can be used to explain the predictions. This allows us to draw another link between local surrogates and counterfactual explanations.

Pursuing this idea, having an explanation generated for a given instance x being also valid for other instances essentially means that the knowledge conveyed by the explanation can be *generalized* to other instances. This idea of generalization is close, albeit different, to the notion of explanation *stability* (or robustness), introduced by Alvarez Melis and Jaakkola (2018): in their work, the similarity of the explanations generated for similar predictions is studied. This is therefore different from the con-

cept of explanation generalization that we propose to discuss, which focuses on the validity of an explanation generated for a certain prediction to similar instances.The concept of knowledge generalization, deeply studied in cognitive sciences and neuro sciences (see for instance Didierjean, 2003), implies that a user is able to adapt some knowledge to a similar yet different situation. Yet, the concept of generalization of explanations has been hardly addressed in the Machine Learning Interpretability literature. Its importance and relation to key concepts of learning (see for instance Didierjean, 2003; Lombrozo, 2006) make it seemingly important to reach the final objective of machine learning interpretability: having the user truly *understand* the model, leading to a higher level of AI-human trust and therefore efficiency.

5.5 | Conclusion

This chapter proposes to go back to studying the notion of explanation locality. In particular, local surrogate model approaches are considered, which try to approximate the local behavior of the black-box classifier. We propose the *Local Fidelity* criterion to assess the quality of this approximation, and propose the LS approach to optimize it. To ensure local explanations, this approach is based on a new sampling method centered around the local decision boundary of the black-box classifier. This allows us to draw a connection between local surrogates and counterfactual explanations.

Several improvements and extensions to the proposed criterion and procedure can be envisaged. First, easing the setup of the hyperparameters is required. Both the proposed quality criterion, LF, and the proposed interpretability approach, LS, indeed rely on several parameters (mainly r_{fid} and r_x). Yet, the way they interact one with another, as well as how to set their values, needs to be studied in more details.

Furthermore, the introduction of the notion of explanation generalization raises interesting prospects. In particular, an upside for the use of counterfactuals often mentioned in this thesis is their actionability: analyzing how the notion of generalization interacts with this actionability promises also an interesting area of work. Moreover, while this concept of generalization has been evoked only in the case of counterfactuals, extending this concept to other interpretability approaches, such as methods returning explanations based on feature importances, could be useful.

6

Conclusion and Perspectives

6.1 | Summary of the Contributions

We consider in this thesis the local post-hoc interpretability paradigm, that is to say the generation of explanations for a single prediction of a trained classifier. In particular, we study a fully agnostic context, meaning that the explanation is generated without using any knowledge about the classifier nor the data used to train it.

We identify three issues that can arise in this context and that may be harmful for interpretability. We propose to study each of these issues and propose novel criteria and approaches to characterize them, as well as two original explanation methods to address them, respectively in the counterfactual and surrogate frameworks. The issues we focus on are: the risk of generating explanations that are out-of distribution; the risk of generating explanations that cannot be associated to any ground-truth instance; finally, the risk of generating explanations that are not local enough.

Adopting a slightly different point of view, the contributions made in this thesis can be organized into two topics: the ones that focus on the notion of explanation locality in the post-hoc context, and the ones that aim at studying the relevance of counterfactual explanations under agnosticity assumptions. We use this new point of view to summarize our contributions below. Additionally, we present how these contributions can be used to draw conclusions in the more general context of the field of post-hoc interpretability.

Defining Locality in the Post-hoc Context

Defining the locality of an explanation in a context where no information about the classifier nor any data is available is complex. In Chapter 3, we first propose to define

a local explanation as an explanation built using the classifier's decision boundary. This provides an additional justification for using counterfactual explanations, which rely on identifying the minimum perturbation required to alter a given prediction. After raising the question of how to make local explanations using this definition easy to understand, we propose a novel approach based on projections of the solution of the Euclidean problem to tackle a sparsity objective. Our proposal, the *Growing Spheres* algorithm, thus proposes post-hoc explanations that are both local and sparse.

A second category of approaches we consider in Chapter 5 is local surrogate models, which aim to approximate the local decision border of the black-box classifier with a simple model, from which a final explanation is extracted. For surrogate model approaches, we propose to measure the fidelity of the built surrogate model to the black-box classifier in a neighborhood of the observation whose prediction is to be explained. The resulting proposed criterion, that we call *Local Fidelity*, thus allows us to define the locality of an explanation as the section of the decision boundary that is being approximated. Using this evaluation procedure, we show that the way local surrogate approaches sample their training instances highly impacts the locality of the explanation. Therefore, we propose *Local Surrogate*, an approach using a new sampling procedure to ensure local explanations.

Since both of the proposed interpretability methods rely on the detection of the local decision boundary of the classifier, we show that they can be put in parallel. For this purpose, we introduce the notion of explanation generalization, closely related to the local fidelity of a linear surrogate model, and use it to suggest that local surrogate approaches can be interpreted as counterfactual explanation approaches.

Issues Raised by Data-agnosticity Assumptions

Beside the definition of the locality of an explanation, we also focus on issues associated to the the data-agnosticity assumption considered in the post-hoc context. Without any information being available about any data whatsoever, we show that generating a relevant post-hoc explanation is complex. For this purpose, we focus on counterfactual approaches in Chapters 3 and 4, and conduct two main studies.

First, we analyze the risk of generating explanations that lie out of the distribution of the training data of the black-box classifier. Through the example of *Growing Spheres*, we thus show that by relying on the greedy generation of numerous instances, post-hoc explanation approaches are highly vulnerable to this issue, especially when there is a mismatch between the features that are used by the user and the ones describing the dataset, such as in the case of image classification.

Secondly, we study the risk of generating explanations that cannot be directly associated to any ground-truth knowledge. Specifically, we discuss that a desirable property for counterfactual explanations is that they can be *justified*, which we propose to define as being connected through a continuous path to a training instance from the same class. In addition, we show that under the data-agnosticity assumption, explainers do not have the capability to avoid unjustified classification regions that may be created by the classifier. We therefore propose a diagnostic approach, *Local Risk Assessment*, to assess the risk of generating unjustified counterfactual explanations. Experimental results suggest that characteristics of the classifier such as the considered algorithm and its overfitting tendency impact this risk. Additionally, a second approach, *Vulnerability Evaluation*, is proposed to exhibit that when facing this risk, post-hoc counterfactual approaches may indeed generate unjustified explanations. Experimental results suggest that the vulnerability of post-hoc counterfactual approaches is related to the locality of the explanations, as approaches that generate less local explanations are less susceptible of generating unjustified explanations.

The Dangers of Post-hoc Interpretability

In Chapter 2, we presented how the post-hoc interpretability paradigm, and more precisely the considered data- and model-agnosticity assumptions, can be viewed as a strength. Indeed, interpretability approaches built under these assumptions can be used in a variety of contexts, leading to more flexibility for the user. However, all the issues studied in this thesis are also a direct consequence of these assumptions. Thus, this suggests a paradox raised by the local post-hoc interpretability context.

6.2 | Future Works

The contributions of this thesis open several promising directions for further works. Beyond some perspectives announced in the chapters' conclusions, these include prospective works on the proposed interpretability approaches and criteria, as well as more generally on post-hoc interpretability.

Four main research directions are identified, developed in the following sections. First, we discuss the questions opened by our study on locality. Then, we discuss how the different issues studied in this thesis can be used to propose new evaluation methods for interpretability approaches. Next, we propose prospective studies to extend the work conducted in thesis to other machine learning problems, such as the detec-

tion of adversarial examples. Finally, we identify perspectives opening the discussion on the consequences of the contributions of this thesis for post-hoc interpretability.

Deepening the Study on Explanation Locality

The contributions of this thesis open the way for other studies on the locality of explanations, to characterize further this notion and possibly adapt it to ensure better explanations. The introduction of the notion of local fidelity represents an interesting contribution in this regard, and its reliance on a hyperparameter r_{fid} offers a criterion to quantify this dimension and allow for user adaptation, i.e. a possible personalization in line with the subjective component of interpretability. However, it also raises questions regarding its practical utility. Indeed, identifying a desirable level of locality seems challenging, especially for non-expert users. Further works focusing on adapting this notion to make it rely on a user-friendlier parameter may therefore be beneficial.

Moreover, it opens questions about the mere fact that the locality of explanation should be a user-defined parameter. An assumption could be made that for a given prediction, only a single level of locality should be acceptable. For instance, explaining some predictions would thus rely on generating global explanations, for some predictions, explanations should be expressed at a global level. On the other hand, for other predictions, the global feature influences may be deemed useless compared to local ones. This hypothetically *right* level of explanation locality may depend on several factors: beside obviously depending on the instance whose prediction is to be interpreted and the classifier, taking into account other parameters such as expert knowledge might also be required. To explore this idea, conducting more theoretical studies, as well as user studies, about explanation locality would thus be highly beneficial.

Defining Criteria for Explanations

Another promising direction of work consists in extending the studies of the proposed criteria. The issues studied in this thesis have led to the proposition of assessment criteria: the justification scores R_x and J_x in Chapter 4, and the locality score LF in Chapter 5. Because these criteria define desirable properties for interpretability approaches, a natural follow-up idea would be to use these criteria as quality metrics or in the generation of explanations. These criteria could thus be implemented to evaluate explanations or to improve classifiers.

Evaluating Explanations

A first idea is to use these criteria to evaluate the quality of a generated explanation, and therefore design approaches that directly optimize these criteria. For instance, this would mean evaluate counterfactual explanation approaches depending on how justified or how in-distribution (see Section 3.4) the counterfactual examples are. However, several issues make this idea difficult.

First of all, the issues of out-of-distribution and unjustified counterfactuals are defined with respect to the training data. Yet, the considered approaches generate explanations in a post-hoc context. Therefore, using these criteria as defined as an optimizable objective to generate counterfactual explanations is obviously impossible by design.

Instead of integrating them in the optimization objectives, these criteria could thus be used as quality metrics for explanations, in the light of the contributions made in this thesis. However, this purpose would also benefit from further studies. One of the difficulties raised by the proposed criterion of justification lies in the way it is calculated. Indeed, the proposed scores R_x, S_x and J_x rely on the *Local Risk Assessment* and *Vulnerability Evaluation* approaches to be calculated. Yet, as discussed in Section 4.2.4, page 83, these procedures are complex. And, they rely on the stochastic generation of instances, meaning that the criteria may differ in values for several identical runs. To circumvent this issue, we proposed to run the algorithms several times, increasing the cost of the calculation of these criteria even more. Therefore, working on the robustness as well as the reduction of the computational cost of these procedures promises relevant results.

Improving Classifiers

The ideas presented below lie beyond the context of the post-hoc generation of local explanations, as they provide recommendations for the training of classifiers. Indeed, a possibility is to define new classification algorithms facilitating the satisfiability of the properties studied in this thesis. For instance, an interesting desideratum for a classifier would be that it does not create any unjustified region. Although presented as an issue encountered in the post-hoc paradigm, the risk of unjustification is initially caused by the classifier creating unconnected regions. Moreover, the study presented in Chapter 4 suggests that the risk of unjustification is related to the choice of the classifier and to its overfitting tendency. Using this knowledge to ensure that no unjustified region is created without sacrificing too much accuracy is therefore a promising perspective.

Similarly, the study in Chapter 5 suggests that the choice of the local surrogate model heavily impacts its local fidelity to the black-box classifier. This raises the question of the possibility of building classifiers with decision boundaries that facilitate this approximation by a given local surrogate. For instance, in the context of linear surrogates such as LIME and LS, the procedure proposed in Chapter 5, an idea is to make the decision boundaries as linear as possible in some regions without any major loss of predictive accuracy. Recently, the work of Alvarez Melis and Jaakkola (2018) goes in this direction: it proposes to add a Lipschitz constraint in the loss function of a neural network to make the local decision boundaries as linear as possible. The resulting classifier is said to be self-explainable, as it makes the generation of local explanations easier.

Counterfactuals and Adversarial Examples

Beside evaluating explanations, questions can be raised regarding how the proposed notions of justification and out-of-distribution could be used to characterize a classifier's predictions in general, counterfactual examples: beyond the context of explanation characterization, it may be interesting to analyze the relation between justification and classification accuracy. Indeed, the classification confidence scores provided classifiers have often been proven to be misleading: this is attested for instance by the existence of adversarial examples, which are very confidently misclassified by the model. Moreover, the results obtained with HCLS highlight that some classifications regions can be both unjustified and classified with a high confidence by the model. In this context, one can wonder if justification could be used to gain information about how trustworthy a given prediction is. A potential use of the justification notion and the VE procedure could thus be to build a post-hoc uncertainty criterion, in the light of Jiang et al. (2018) for instance and incidentally coinciding with the *justification* notion proposed by Biran and Cotton (2019).

Using the J_x score alone (asserting whether or not a prediction is justified) may however not be enough, as suggested by preliminary experiments conducted in this thesis, the results of which are shown in Appendix A, page 133. In this experiment, we assessed the ϵ-connectedness of adversarial examples generated for the MNIST dataset. The results are that all of the adversarial examples generated are, in fact, justified. This also confirms the conclusion of existing papers such as Fawzi et al. (2018), asserting the connectedness of adversarial examples. This suggests that adversarial examples satisfy the justification property, and therefore further studies are necessary to define a relevant confidence criterion.

More generally, this raises questions about how the fields of counterfactual examples and adversarial examples can benefit from each other. As mentioned in Section 2.3.3, the distinction between these two notions can be expressed in terms of the perceptibility of the perturbation. However, the fact that these two subfields generally focus on different types of data make it difficult to study: adversarial examples have been first and foremost defined in the contexts of image and text classification, whereas most of the work on counterfactual explanations focuses on tabular data. Therefore, further studying the distinction between counterfactuals and adversarial examples would require proposing definitions for each concept that do not rely on the nature of the data. Some works recently start exploring these directions: Hendricks et al. (2018) have proposed counterfactual explanations for image classification, and Kulynych et al. (2018) propose to define adversarial attacks for discrete data. However, none of these works have proposed a general definition for these concepts.

Post-hoc Interpretability: a Shift of Paradigm

The work conducted in this thesis highlighs several issues raised by the post-hoc paradigm. In particular, model- and data- agnosticity assumptions indeed create the risk of generating explanations that may not be useful to the users. Although studied here in the context of counterfactual explanations and local surrogates, these issues concern all explanation methods using similar assumptions. A potential solution to this problem would obviously be to release the strong agnosticity assumptions imposed and explore the possibility to dispose of some prior knowledge to limit the perturbations created by the algorithm to ensure that the generated counterfactual explanation lies in a plausible domain. In a context such as images, learning specific representations that are relevant for the user may be helpful. On the MNIST dataset for instance, this would mean using what actually constitutes a digit and using this knowledge to generate counterfactual explanations in the manifold of digits. As a consequence, the question of the cost of the relaxation of these assumptions compared to the gain in explanation quality needs to be studied.

Beyond, this raises question about new use cases for post-hoc interpretability approaches. Indeed, the contributions of this thesis suggest that post-hoc methods may encounter unexpected issues hurting interpretability. In this context, this suggests that these methods might be more useful for specific tasks such as model debugging for instance, where the goal is to detect the vulnerabilities of the classifier. Namely, the presence of unnatural non-linearities can be detected using the *Local Fidelity* criterion, and the presence of unjustified regions with the LRA procedure. Beside relax-

ing the considered agnosticity assumptions, another research direction is therefore to identify a new framework for the application of post-hoc interpretability methods.

A

Justification of Adversarial Examples on MNIST

In Chapter 4, page 69, the notion of justification was proposed as a desirable criterion for counterfactual explanations. Due to the similarity of the concepts of adversarial and counterfactual examples, discussed in Section 2.3.3, page 36, the question of whether adversarial examples are justified or not is raised; this experiment is a perliminary study to analyze the justification of adversarial examples.

Using the MNIST dataset (introduced by LeCun et al. (1998) and described in Table 3.1, page 58), a convolutional neural network f is trained on 90% of the data (with default architecture). The resulting accuracy over the test set is 0.99.

Generating adversarial examples. In Figure A.1, 5 randomly chosen examples of handwritten digits, correctly classified, are represented (top row). Adversarial examples are then generated using the FGSM evasion attack method (Goodfellow et al., 2015), which uses the sign vector of the gradient calculated on an instance x to generate an adversarial example $\tilde{x}$, looking similar but classified differently. FGSM relies on a parameter to determine the maximum allowed distance $||x - \tilde{x}||_2$ and which we set to 0.05. In the middle row of Figure A.1, the obtained adversarial examples are shown. Each adversarial example $\tilde{x}$ represented is classified differently from x: $f(\tilde{x}) \neq f(x)$.

Justification of adversarial examples. We then focus on analyzing the justification of these adversarial examples, using the proposed Definition 3, page 74. Considering an adversarial example $\tilde{x}$, the goal of this experiment is therefore to identify whether there exists a training instance $a \in X$ such that $\tilde{x}$ and a are ϵ-connected. This falls into the application frame of the VE procedure, described in Section 4.4.1, page 94.

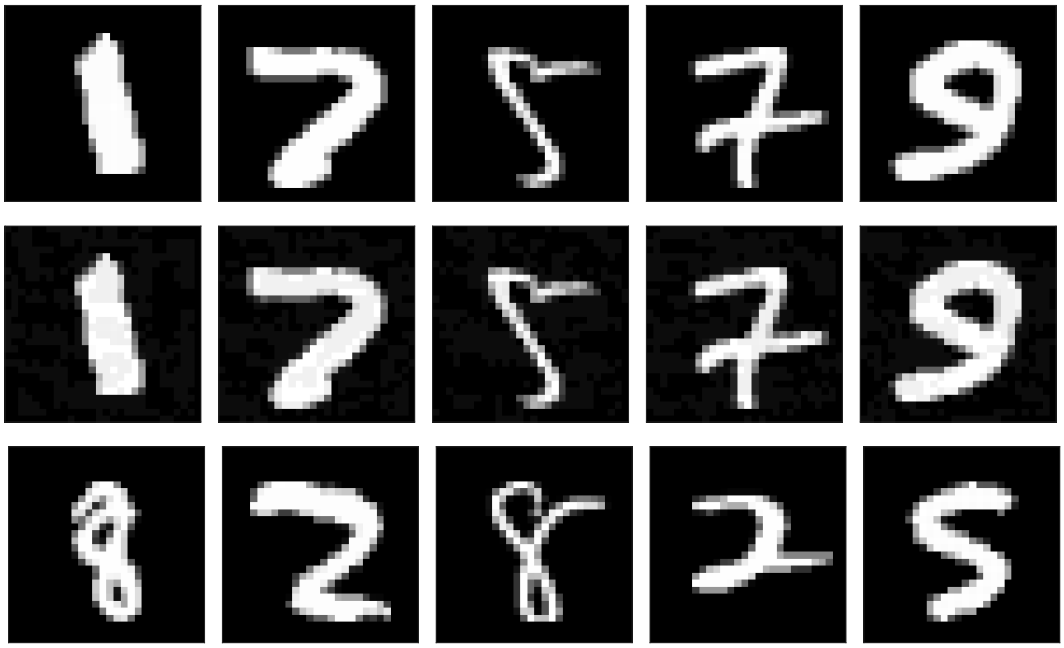


Figure A.1: Example of a normal (top) and adversarial (middle row) MNIST instances. In the bottom row, the closest ground-truth neighbor from the generated adversarial example predicted to belong to the same class is shown.

However, as mentioned in Sections 4.2.1.4, page 80 and 4.5, page 100, the scaling of the VE procedure makes it difficult to use in a high dimensional context such as the MNIST dataset.

Therefore, we propose therefore a simple experiment, consisting in assessing whether each adversarial example is connected to its closest ground-truth neighbor from the same class. For each adversarial example $\tilde{x}$, we thus propose in turn to:

- Detect its l_2-closest neighbor $a_0 \in X$ correctly predicted and satisfying: $f(a_0) = f(\tilde{x})$.
- Generate 10000 instances z_i uniformly distributed on the segment between a_0 and $\tilde{x}$. The number of generated instances is arbitrarily chosen to be high.
- Calculate the predicted class $f(z_i)$ for each of these instances z_i.
- Measure the proportion P of instances assigned to the same class as $\tilde{x}$:

$$P = \frac{|\{z_i \ : \ f(z_i) = f(a_0)\}|}{10000}$$

If $P = 100\%$, this means that all of the generated instances z_i are predicted to belong to the same class, hence that $\tilde{x}$ can be assimilated to being ϵ-connected to a_0.

This assessment is run over 1000 adversarial examples (each generated for a different test instance).

Results. Among the generated adversarial examples, the totality of them was satisfying $P = 100\%$. The last row of Figure A.1 shows these ground-truth neighbors a_0, to which the adversarial examples are connected.

Although imperceptibly different from x, each adversarial example is thus predicted to belong to a different class and to be justified to a ground-truth instance, according to a variant of the connectedness definition of justification: there exists a training instance a_0 with $f(a_0) = f(\tilde{x})$ such that all points on the line between a_0 and $\tilde{x}$ are in the same class, i.e. belong to the same classification region without crossing the decision boundary of f.

The fact that this trivial assessment detected the totality of the generated adversarial examples as being justified suggests that the notion of justification is not sufficient to define a relevant counterfactual explanation. Indeed, as discussed in Section 2.3.3, page 36, due to the imperceptibility of the perturbation induced, adversarial examples do not represent satisfying counterfactual explanations. Moreover, it confirms that the proposed notion cannot be used as a mean to detect adversarial examples. However, this experiment suggests a promising research direction to the question of the adaptation the concept of adversarial examples to the context of tabular data. In particular, it raises interesting questions regarding the link between the notion of imperceptibility, which adversarial examples rely on, and justification. Further works are necessary in that regard, including testing the proposed assessment for other adversarial attack approaches and datasets.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems 31*, pages 9505–9515. 2018.

Mokhtar Z. Alaya, Simon Bussy, Stéphane Gaïffas, and Agathe Guilloux. Binarsity: a penalization for one-hot encoded features in linear supervised learning. *Journal of Machine Learning Research*, 20(118):1–34, 2019.

David Alvarez Melis and Tommi Jaakkola. Towards robust interpretability with self-explaining neural networks. *Advances in Neural Information Processing Systems 31*, pages 7786–7795, 2018.

Andre Artelt and Barbara Hammer. On the computation of counterfactual explanations - a survey. *preprint arXiv:1908.00735*, 2019.

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

Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Marcin Detyniecki, and Pascal Frossard. Imperceptible adversarial attacks on tabular data. In *NeurIPS 2019 Workshop on Robust AI for Financial Services*, 2019.

David Barbella, Sami Benzaid, Janara M. Christensen, Bret Jackson, Victor X. Qin, and David Musicant. Understanding support vector machine classifications via a recommender system-like approach. In *Proc. of the 2009 Int. Conf. on Data Mining*, pages 305–311, 2009.

Khaled Belahcène, Christophe Labreuche, Nicolas Maudet, Vincent Mousseau, and Wassila Ouerdane. Explaining robust additive utility models by sequences of preference swaps. *Theory and Decision*, 82:151–183, 2015.

Richard Berk. *Regression Analysis: A Constructive Critique*. Thousand Oaks, 2004.

Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Machine Learning and Knowledge Discovery in Databases*, pages 387–402, 2013.

Or Biran and Courtenay Cotton. Explanation and justification in machine learning: A survey. *IJCAI 2017 Workshop on Explainable Artificial Intelligence (XAI)*, 2019.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, pages 144–152, 1992.

Léon Bottou, Jonas Peters, Joaquin Quiñonero Candela, Denis X. Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14:3207–3260, 2013.

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

Leo Breiman. Looking inside the black box. *Wald Lecture II, UC Berkeley*, 2002.

Leo Breiman, Jerome Friedman, Charles Stone, and Richard Olshen. *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

Ruth Byrne. The rational imagination: how people create alternatives to reality. *Behavioral and Brain Sciences*, 5(30), 2008.

Diogo V. Carvalho, Eduardo M. Pereira, and Jaime S. Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

Dennis Collaris, Leo Vink, and Jack van Wijk. Instance-level explanations for fraud detection. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018.

Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained neural networks. *Advances in Neural Information Processing Systems*, 8:24–30, 1996.

Betsy N. Decyk. Using Examples to Teaching Concepts. In *Changing College Classrooms: New teaching and learning strategies for an inscreasingly complex world*, pages 39–63. 1994.

Marcin Detyniecki. *Mathematical Aggregation Operators and their Application to Video Querying*. PhD thesis, Université Pierre et Marie Curie (UPMC), Paris, France, November 2000.

André Didierjean. Is case-based reasoning a source of knowledge generalisation? *European Journal of Cognitive Psychology*, 15(3):435–453, 2003.

Berkeley Dietvorst, Joseph Simmons, and Cade Massey. Algorithm aversion: People erroneously avoid algorithms after seeing them err. *Journal of Experimental Psychology*, 2015.

Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *preprint arXiv:1702.08608*, 2017.

Finale Doshi-Velez, Mason Kortz, Ryan Budish, Chris Bavitz, Sam Gershman, David O'Brien, Stuart Shieber, James Waldo, David Weinberger, and Alexandra Wood. Accountability of AI under the law: The role of explanation. *Privacy Law Scholars Conference*, 2018.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD'96)*, pages 226–231, 1996.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'18)*, 2018.

Kelwin. Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. *Proc. of the 17th EPIA 2015 - Portuguese Conference on Artificial Inteligence*, 2015.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

Matthew L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30(1):35–79, 1986.

Stephen Goldinger, Heather M. Kleider, Tamiko Azuma, and Denise R. Beike. "Blaming the victim" under memory load. *Psychological Science*, 14(1):81–85, 2003.

Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A survey of methods for explaining black box models. *ACM Comput. Surv.*, 51(5), 2018.

Riccardo Guidotti, Anna Monreale, Fosca Giannotti, Dino Pedreschi, Salvatore Ruggieri, and Franco Turini. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, 34(6):14–23, 2019a.

Riccardo Guidotti, Anna Monreale, Stan Matwin, and Dino Pedreschi. Black-box explanation by learning image exemplars in the latent feature space. In *to appear in Machine Learning and Knowledge Discovery in Databases*, 2019b.

Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach. part I: Causes. *British Journal for the Philosophy of Science*, 56(4):843–887, 2005.

Henry Han and Xiaoqian Jiang. Overcome support vector machine diagnosis overfitting. *Cancer Informatics*, 13 (supplementary 1):145–158, 2014.

Satoshi Hara and Kohei Hayashi. Making tree ensembles interpretable. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2016)*, 2016.

David Harrison and Daniel Rubinfeld. Hedonic prices and the demand for clean air. *Environment Economics and Management*, 5:81–102, 1978.

Reinhard Heckel, Michail Vlachos, Thomas Parnell, and Celestine Duenner. Scalable and interpretable product recommendations via overlapping co-clustering. In *Proc. of the IEEE 33rd Int. Conf. on Data Engineering (ICDE'17)*, pages 1033–1044, 2017.

Carl G. Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of Science 15*, 2, 1948.

Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. Generating counterfactual explanations with natural language. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems 31*, pages 5541–5552, 2018.

Mayank Kabra, Alice Robie, and Kristin Branson. Understanding classifier errors by examining influential neighbors. *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'15)*, pages 3917–3925, 2015.

Been Kim, Cynthia Rudin, and Julie A Shah. The Bayesian Case Model: A generative approach for case-based reasoning and prototype classification. In *Advances in Neural Information Processing Systems*, pages 1952–1960, 2014.

Josua Krause, Adam Perer, and Enrico Bertini. Using visual analytics to interpret predictive machine learning models. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018.

Max Kuhn and Jonhson Kjell. *Applied predictive modeling*. Springer, 2013.

Bogdan Kulynych, Jamie Hayes, Nikita Samarin, and Carmela Troncoso. Evading classifiers in discrete domains with provable optimality guarantees. *preprint arXiv:1810.10939*, 2018.

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

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the compas recidivism algorithm. *ProPublica*, 2016.

Michael Lash, Qihang Lin, Nick Street, Jennifer Robinson, and Jeffrey Ohlmann. Generalized inverse classification. In *Proc. of the 2017 SIAM Int. Conf. on Data Mining*, pages 162–170, 2017a.

Michael T. Lash, Qihang Lin, W. Nick Street, and Jennifer G. Robinson. A budget-constrained inverse classification framework for smooth classifiers. In *Data Mining Workshops (ICDMW), IEEE Conf. on Data Mining*, 2017b.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Comparison-based inverse classification for interpretability in machine learning. In *Proc. of the Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'18)*, pages 100–111, 2018a.

Thibault Laugel, Xavier Renard, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Defining locality for surrogates in post-hoc interpretablity. *ICML 2018 Workshop on Human Interpretability in Machine Learning (WHI 2018)*, 2018b.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Issues with post-hoc counterfactual explanations: a discussion. In *ICML 2019 Workshop on Human in the Loop Learning (HILL 2019)*, 2019a.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. Unjustified classification regions and counterfactual explanations in machine learning. In *to appear in Proc. of the European Conf. on Machine Learning, ECML-PKDD'19*, 2019b.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. The dangers of post-hoc interpretability: Unjustified counterfactual explanations. In *Proc. of the 28th Int. Joint Conference on Artificial Intelligence (IJCAI'19)*, pages 2801–2807, 2019c.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791.

David K. Lewis. *Counterfactuals*. Blackwell, 1973.

Zachary C. Lipton. The mythos of model interpretability. *ICML Workshop on Human Interpretability in Machine Learning (WHI 2017)*, 2017.

Tania Lombrozo. The structure and function of explanations. *Trends in Cognitive Sciences 10*, 10, 2006.

Ana Lucic, Hinda Haned, and Maarten DeRijke. Why does my model fail? Contrastive local explanations for retail forecasting. *IJCAI 2019 Workshop on Explainable Artificial Intelligence*, 2019.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774, 2017.

Christophe Marsala. A fuzzy decision tree based approach to characterize medical data. In *Proc. of the IEEE Int. Conf. on Fuzzy Systems (FuzzIEEE'09)*, pages 1332–1337, 2009.

David Martens and Foster Provost. Explaining Data-Driven Document Classifications. *Mis Quarterly*, 38(1):73–99, 2014.

Robert McCrae. Creativity, divergent thinking, and openness to experience. *Journal of Personality and Social Psychology*, 6(52), 1987.

Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

Yao Ming, Huamin Qu, and Enrico Bertini. Rulematrix: Visualizing and understanding classifiers with rules. *IEEE Transactions on Visualization and Computer Graphics*, 25, 2018.

Brent Mittelstadt, Chris Russell, and Sandra Wachter. Explaining explanations in ai. In *Proc. of the Conf. on Fairness, Accountability and Transparency (FAT* '19)*, pages 279–288, 2019.

Christoph Molnar. *Interpretable Machine Learning*. 2019. URL https://christophm.github.io/interpretable-ml-book.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Proc. of the IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'16)*, pages 2574–2582, 2016.

Shane T. Mueller, Robert Hoffman, William Clancey, Abigail Emrey, and Gary Klein. Explanation in human-AI systems: A literature meta-review. synopsis of key ideas and publications and bibliography for explainable AI. *DARPA XAI Literature Review*, 2019.

Mervin E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM*, 2(4):19–20, 1959.

Nyaradzo Mvududu and Gibbs Y. Kanyongo. Using real life examples to teach abstract statistical concepts. *Teaching Statistics*, 33(1):12–16, 2011.

Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *Proc. of the European Conf. on Computer Vision (ECCV'18)*, pages 613–628, 2018.

Robert Nisbet, John Elder, and Gary Miner. *Handbook of Statistical Analysis and Data Mining Applications*. Academic Press, Inc., 2009.

Eimear O'Connor, Teresa McCormack, and Aidan Feeney. Do children who experience regret make better decisions? A developmental study of the behavioral consequences of regret. *Child Development*, 85(5):1995–2010, 2014.

Andrea Papenmeier, Gwenn Englebienne, and Christin Seifert. How model accuracy and explanation fidelity influence user trust in AI. *IJCAI 2019 Workshop on Explainable Artificial Intelligence*, 2019.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 582–597. 2016.

Fabian Pedregosa, Gael. Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Muller, Joel Nothamn, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vicent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Rafael Pinot, Laurent Meunier, Alexandre Araujo, Hisashi Kashima, Florian Yger, Cedric Gouy-Pailler, and Jamal Atif. Theoretical evidence for adversarial robustness through randomization. In *Advances in Neural Information Processing Systems 32*, pages 11838–11848. 2019.

Xavier Renard, Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, and Marcin Detyniecki. Detecting potential local adversarial examples for human-interpretable defense. In *Joint European Conf. on Machine Learning and Knowledge Discovery in Databases, ECML-PKDD'18, Workshop on Adversarial Learning (Nemesis)*, 2018.

Marco Tulio Ribeiro, Sameer Singh, and Carlos 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 (KDD'16)*, pages 1135–1144, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proc. of the AAAI Conf. on Artificial Intelligence*, 2018.

Neal J. Roese. Counterfactual thinking. *Psychological Bulletin*, 1(121), 1997.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215, 2019.

Stefan Rueping. *Learning Interpretable Models*. PhD thesis, University of Dortmund, 2006.

Chris Russell. Efficient search for diverse coherent explanations. In *Proc. of the Conf. on Fairness, Accountability, and Transparency (FAT* '19)*, pages 20–28, 2019.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proc. of the IEEE Int. Conf. on Computer Vision (ICCV'16)*, pages 618–626, 2016.

Boris Sharchilev, Yury Ustinovskiy, Pavel Serdyukov, and Maarten de Rijke. Finding influential training samples for gradient boosted decision trees. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018,1*, pages 4584–4592, 2018.

Galit Shmueli. To explain or to predict? *SSRN Electronic Journal*, 3(25), 2010.

Erik Strumbelj, Igor Kononenko, and Marko Robnik-Sikonja. Explaining instance classifications with interactions of subsets of feature values. *Data Knowledge Engineering*, 68:886–904, 2009.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proc. of the 32nd Int. Conf. on Machine Learning (ICML'15)*, pages 814–823, 2015.

Sarah Tan. Distill and compare: Auditing black-box models using transparent model distillation. In *AIES*, 2018.

Guanhong Tao, Shiqing Ma, Yingqi Liu, and Xiangyu Zhang. Attacks meet interpretability: Attribute-steered detection of adversarial samples. In *Advances in Neural Information Processing Systems 31*, pages 7717–7728. 2018.

Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1), 1996.

Ryan Turner. A model explanation system. *NIPS Workshop on Black Box Learning and Inference*, 2015.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In *Proc. of the Conf. on Fairness, Accountability, and Transparency (FAT*'19)*, pages 10–19, 2019.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box; automated decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2):841–887, 2018.

Anne Watson and Steve Shipman. Using learner generated examples to introduce new concepts. *Educational Studies in Mathematics*, 69(2):97–109, 2008. ISSN 00131954. doi: 10.1007/s10649-008-9142-4.

Adrian Weller. Transparency: Motivations and challenges. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pages 23–40, 2019.

Jason Yosinki, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Unserstanding neural networks through deep visualization. *Deep Learning Workshop, 31st Int. Conf. on Machine Learning*, 2015.

Long Yu and Jian Xiao. Trade-off between accuracy and interpretability: Experience-oriented fuzzy modeling via reduced-set vectors. *Computers & Mathematics with Applications*, 57(6): 885 – 895, 2009. Advances in Fuzzy Sets and Knowledge Discovery.

Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistics Society*, 2016.

Yu Lin Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankui Zhuo, and Subbarao Kambhampati. Plan explicability and predictability for robot task planning. *Proc of the IEEE Int. Conf. on Robotics and Automation (ICRA'15)*, pages 1313–1320, 2015.