



HAL
open science

Analyzing and explaining data-driven Artificial Intelligence Models by argumentation

Henri Trenquier

► **To cite this version:**

Henri Trenquier. Analyzing and explaining data-driven Artificial Intelligence Models by argumentation. Artificial Intelligence [cs.AI]. Université Paul Sabatier - Toulouse III, 2023. English. NNT : 2023TOU30355 . tel-04612661

HAL Id: tel-04612661

<https://theses.hal.science/tel-04612661v1>

Submitted on 14 Jun 2024

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

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

Doctorat de l'Université de Toulouse

préparé à l'Université Toulouse III - Paul Sabatier

Analyse et explication par des techniques d'argumentation
de modèles d'intelligence artificielle basés sur des données

Thèse présentée et soutenue, le 22 décembre 2023 par

Henri TRENQUIER

École doctorale

EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse

Spécialité

Informatique et Télécommunications

Unité de recherche

IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par

Leila AMGOUD et Philippe MULLER

Composition du jury

Mme Marie-Christine LAGASQUIE-SCHIEX, Présidente, Université Toulouse III - Paul Sabatier

M. Nicolas MAUDET, Rapporteur, Sorbonne Université

Mme Wassila OUERDANE, Examinatrice, Centrale Supélec

Mme Leila AMGOUD, Directrice de thèse, CNRS Toulouse - IRIT

Membres invités

M. Philippe MULLER, Université Toulouse III - Paul Sabatier

Abstract

Classification is a very common task in Machine Learning (ML) and the ML models created to perform this task tend to reach human comparable accuracy, at the cost of transparency. The surge of such AI-based systems in the public's daily life has created a need for explainability. *Abductive explanations* are one of the most popular types of explanations that are provided for the purpose of explaining the behavior of complex ML models sometimes considered as black-boxes. They highlight feature-values that are sufficient for the model to make a prediction. In the literature, they are generated by exploring the whole feature space, which is unreasonable in practice. This thesis tackles this problem by introducing explanation functions that generate abductive explanations from a sample of instances. It shows that such functions should be defined with great care since they cannot satisfy two desirable properties at the same time, namely existence of explanations for every individual decision (*success*) and correctness of explanations (*coherence*). This thesis provides a parameterized family of argumentation-based explanation functions, each of which satisfies one of the two properties. It studies their formal properties and their experimental behaviour on different datasets.

Résumé

La classification est une tâche très courante dans le domaine de l'apprentissage automatique et les modèles d'apprentissage automatique créés pour accomplir cette tâche tendent à atteindre une précision comparable à celle des humains, au détriment de leur transparence. L'apparition de ces systèmes intelligents dans le quotidien du public a créé un besoin d'explicabilité. Les *explications abductives* sont l'un des types d'explications les plus populaires qui sont fournies dans le but d'expliquer le comportement de modèles d'apprentissage complexes, parfois considérés comme des boîtes noires. Elles mettent en évidence les caractéristiques qui sont suffisantes pour que le modèle prédise une certaine classe. Dans la littérature, elles sont générées en explorant l'ensemble de l'espace des caractéristiques, ce qui n'est pas raisonnable en pratique. Cette thèse aborde ce problème en introduisant des fonctions d'explication qui génèrent des explications abductives à partir d'un échantillon arbitraire d'instances. Elle montre que de telles fonctions doivent être définies avec beaucoup de soin car elles ne peuvent pas satisfaire simultanément deux propriétés souhaitables, à savoir l'existence d'explications pour chaque décision individuelle (*success*) et l'exactitude des explications (*coherence*). Cette thèse fournit une collection de fonctions d'explication paramétrées basées sur l'argumentation, chacune satisfaisant l'une des ces deux propriétés. De plus, elle étudie leurs propriétés formelles ainsi que leur comportement expérimental sur différents ensembles de données.

Acknowledgements

The journey through my PhD has been an incredible one, and it wouldn't have been possible without a group of exceptional people who provided their unwavering support and guidance. First, I must express my heartfelt thanks to my supervisors. Your expert advice, constant encouragement, and patience have been crucial to my research. You've been pillars of strength, and I am deeply grateful for your commitment and dedication to my growth as a researcher.

A special shoutout to Philippe, who not only guided me through the intricacies of my research but also showed me the ropes in teaching. Your mentorship has been invaluable, Philippe. Your knack for making complex ideas accessible (and occasionally pointing out when my enthusiasm far exceeded my clarity) has been a highlight of my PhD experience.

To my family, thank you for your love, support, and belief in me, even when the subject of my research sounded like a foreign language to you. Your unwavering faith and the occasional questions about when I'd finally graduate have kept me grounded and focused.

And to my friends, who have been the best distraction a PhD student could ask for. Thank you for understanding my absences, celebrating the small victories with me, and for all the laughter and relief you provided during those intense moments of study.

In essence, this thesis would not exist without all of you. Your support has been my motivation and has made all the difference in this journey.

Contents

1	Introduction	13
1.1	Machine Learning (ML)	13
1.2	eXplainable Artificial Intelligence (XAI)	14
1.3	Contributions	16
1.4	Structure of the manuscript	17
2	Explainability in Artificial Intelligence	19
2.0.1	Supervised Learning with Classifiers	20
2.0.2	Defining concepts: Interpretable or Explainable?	21
2.1	What is there to explain?	24
2.1.1	Scope: Explain a model or a prediction?	24
2.1.2	Portability: Explain an opaque or a transparent system?	26
2.1.3	Focus on specific models' aspects	27
2.2	What is an explanation?	29
2.3	Explanation functions	30
2.3.1	Intrinsically interpretable models	31
2.3.2	Model-agnostic explainers	32
2.4	Explanation Desiderata	37
2.4.1	List of Desiderata	38
2.4.2	Quality assessment: Metrics	41
2.5	Conclusion	43
3	Argumentation	45
3.1	What is argumentation?	45
3.2	Argumentation process	47
3.3	Abstract argumentation framework	50
3.4	Argumentation in Artificial Intelligence and XAI	59
4	Theoretical Approach with argumentation	61
4.1	Introduction	61
4.2	Classification models	62

4.3	Abductive Explanations	64
4.4	Plausible abductive explanations	65
4.5	Coherence vs Existence of explanations	67
4.6	Parameterized Family of Explainers	70
5	Experimental study	81
5.1	Datasets	82
5.1.1	Titanic	82
5.1.2	Adult	83
5.1.3	Lending	83
5.1.4	Recidivism	84
5.1.5	Diabetes	85
5.2	Arguments Enumeration	85
5.2.1	Overview	86
5.2.2	In-depth analysis	87
5.2.2.1	Potential arguments generation	87
5.2.2.2	<i>Minimality</i> Guarantee	88
5.2.2.3	<i>Consensus</i> check	88
5.2.3	Complexity	88
5.2.4	Experiments	90
5.2.4.1	Number of arguments	90
5.2.4.2	Proposition validation	90
5.3	Argumentation System: Attack relations	91
5.3.1	Complexity	93
5.3.2	Experimental results	93
5.4	Extension Enumeration	94
5.4.1	Extension Enumeration problem equivalence in graph theory	96
5.4.2	Implementation: Solver choice	97
5.4.3	Stable Extensions Enumeration solutions: Comparative study	98
5.4.3.1	The pipeline’s bottleneck	98
5.5	Explanation Functions	99
5.5.1	Selection Functions	99
5.5.1.1	Selection functions: Implementation	100
5.5.2	Inference Rules	101
5.5.3	Experiments	102
5.5.3.1	Result example: <i>adult</i> dataset	102
5.5.3.2	Orders of magnitude	102
5.5.3.3	Success	103
5.6	Further experiments	104

5.7 Conclusion	105
6 Conclusion	107

Chapter 1

Introduction

“No, no! The adventures first, explanations take such a dreadful time.”

Lewis Carroll, *Through the Looking-Glass*

1.1 Machine Learning (ML)

The data revolution, driven by the internet, social media, and the Internet of Things (IoT), provides artificial intelligence (AI) with vast amounts of data. This data fuels remarkable advances in machine learning (ML), a subfield of AI, which aims to learn a targeted object property (e.g. the class of an object) from a vast quantity of data. In industry, AI automates processes, enhances decision-making, and improves supply chains. In (Ransbotham et al., 2017), the authors conducted a survey to capture insights from organizations all around the world about their interest in AI. They report that 46% of companies try to or already incorporate AI in their processes or offerings. They also mention an example about *Ping An Insurance Co. of China Ltd.* which offers loans in a few minutes thanks to an AI based customer scoring tool. They also launched an intelligent Investment Risk advisor “KYZ Risk” to help customers in their investment decisions. Airbus is also another example of company that uses AI as it leverages AI in the A350 program to find flaws in the production line. Thanks to AI methods, they are able to identify patterns in production problems. Manufacturers can also rely on AI-powered robots for sorting tasks thanks to classification methods and thus improve in precision and efficiency. The entertainment industry leverages AI for content, product recommendation or ad personalization. Large companies such as *Google* or *Amazon* are able to collect data on their users to optimize their profits. Recommender systems increase the time spent on streaming content like *Youtube* and *Netflix*, they make precise product suggestions for *Amazon* buyers and allow to propose personalized ads on the internet and social media. (Zhang et al., 2021) propose a wide overview on the main methods for recommender and the technical challenges. In the healthcare sector, AI related research is

very active, in particular for assisting with the complex problem of medical diagnosis since it is able to analyze complex medical data. (Bhavsar et al., 2021) made a survey of the most recent methods that use ML techniques for medical diagnosis. They enumerate a list of diseases studied between years 2015 and 2020. (Erickson et al., 2017) provide another survey on the use of Computer Vision for Medical Imaging e.g. radiology. They highlight the tremendous progress in accuracy and robustness of ML methods, especially thanks to Deep Neural Networks which avoids bias from data choice tasks. AI is also expected to assist nurses with paperwork and patient monitoring, and assist surgeons to perform less intrusive operations. Companies like Metadvice or Merative are actors in clinical decision support. AI contributes to drug discovery through predictive modeling. It helps alleviating long drug discovery processes by preventing potential failures and by predicting potential drugs' properties. The finance sector employs AI for fraud detection, algorithmic trading, and personalized financial recommendations. In Cyber Security, AI detects and mitigates threats in real-time, safeguarding sensitive data and critical infrastructure. Early 2020's have been very prolific years for generative AI. ChatGPT, with GPT3 and later GPT3.5 and GPT4, was released at the end of 2022 and showed incredible performances in text generation and already transforms some professions. GPT is a Large Language Model (LLM) that shows great performance in information restitution and summaries and are more relevant than search engines in some situations. Developers, for example, can use ChatGPT to search through long and cumbersome documentation or develop simple programs just by writing a well defined prompt to ChatGPT. Prompts are the input base of text-to-text or text-to-image models. With a clear and precise prompt, one can improve the quality of the output. The role of prompt engineer will soon be common in many companies that use Large Language Models like GPT. AI-generated art like images with DALL-E or Midjourney, music covers or videos also emerges as novel forms of creative expression. The landscape is studied in (Gozalo-Brizuela and Garrido-Merchan, 2023).

1.2 eXplainable Artificial Intelligence (XAI)

These recent advances in machine learning rely on inductive models, depending on parameters that are adjusted based on a set of training instances. Such models tend to be large for practical tasks, in the sense of having a lot of parameters, and may allow for non-linear interactions between the input features. Consequently, they are perceived as black-boxes whose behavior is difficult to grasp both from their designers' and users' point of view. This opacity has sparked a new subfield of AI, explainable AI (XAI), whose approaches provide ways to explain what black-box models do and why they do it (Burkart and Huber, 2021; Miller, 2018). XAI gained also interest of AI community because the European Union requires transparency of AI models in its General Data Protection Regulation (GDPR)

applied in 2018. One particular consequence of this law is the right for an explanation (Goodman and Flaxman, 2017; Doshi-Velez and Kim, 2017). Indeed, when automated decision systems use high level predictors to make predictions that can significantly impact users, the latter can demand explanations. XAI is not only an effort for regulation. In (Vilone and Longo, 2020), the authors listed four benefits of a better explainability of AI systems.

- *explain to justify*: Explaining decisions of AI systems is a way to argue in favor of their behavior. Explanations are necessary to make AI systems accepted as a reliable additional source of information.
- *explain to control*: The role of explanations is to enhance the transparency of models and make it easier to identify the causes and functioning of the systems, avoiding bias and guaranteeing fair outputs.
- *explain to improve*: explanations should help scholars improve the accuracy and efficiency of their models and help debugging and to identify potential flaws.
- *explain to discover*: explanations support the extraction of novel knowledge and the learning of relationships and patterns inferred by the AI models.

As a consequence, of the above requirements and benefits, a plethora of studies have been done on XAI in the last decade. Some of them focused on defining explanation models (Ribeiro et al., 2016; Lundberg and Lee, 2017; Ribeiro et al., 2018) and others on introducing metrics or properties for judging the quality of explanation models (Doshi-Velez and Kim, 2017).

Existing explanation models can be classified in three different ways. The first way distinguishes explainers that provide explanations for individual predictions (i.e. explaining the decision of a given instance like “Why the application of Bob for a job position was rejected?”), called local explanations (Ribeiro et al., 2016, 2018; Dhurandhar et al., 2018; Darwiche and Hirth, 2020), from models that provide explanations for classes independently of instances, called global explanations (Amgoud, 2021a; Ignatiev et al., 2019). The second way is based on the information used for generating explanations. Some models, like those studied in (Darwiche and Hirth, 2020; Ignatiev et al., 2019; Audemard et al., 2022), use the whole set of instances, called the feature space, while others like *Anchors* (Ribeiro et al., 2018) and *LIME* (Ribeiro et al., 2016) use only a subset of the feature space. The third way distinguishes models which look inside the ML model from those which consider the model as a black-box whose internal reasoning is left unspecified. The former provide insight into the internal decision-making process (Shih et al., 2018; Ignatiev and Marques-Silva, 2021; Ferreira et al., 2022). They are suitable for explaining interpretable ML models like decision trees (Quinlan, 1986) and Bayesian networks (Heckerman, 2008).

However, they may not be feasible for complex and non-interpretable ones like deep neural networks. The second family of explanation model considers a ML model as a black-box and provides explanations without looking inside it. It looks for correlations between input data and the predictions made by the ML model. This approach has been largely applied to non-interpretable models (Ribeiro et al., 2018; Dhurandhar et al., 2018; Biran and McKeown, 2017; Luss et al., 2019; Mittelstadt et al., 2019; Wachter et al., 2017) and also to interpretable ones (Darwiche and Hirth, 2020; Ignatiev et al., 2018; Ignatiev and Marques-Silva, 2021).

1.3 Contributions

The aim of this thesis is to explain the outcomes of black-box classifiers. We thus look for local explanations by checking the correlations between input data and the predictions of classifiers. One of the most studied types of explanations in this context is the so-called abductive explanation, which highlights the feature-values that are sufficient for making a given prediction. For example, a client was refused a loan because he is unemployed. Such explanations are generally generated from the whole feature-space (Darwiche and Hirth, 2020; Ignatiev et al., 2019; Audemard et al., 2022). While the approach is reasonable when models are interpretable, it is not tractable in case of black-boxes, see (Cooper and Marques-Silva, 2021), as it requires an exhaustive exploration of the feature space.

As a solution, the two prominent explanation functions Anchors (Ribeiro et al., 2018) and LIME (Ribeiro et al., 2016) and the argument-based function (Amgoud, 2021b) generate abductive explanations from a sample (i.e., subset) of instances, avoiding thus exploring the whole feature space. However, it has been shown in (Amgoud, 2021b; Narodytska et al., 2019a) that the explanations of Anchors/LIME may be globally inconsistent and thus incorrect. The third function ensures correct explanations but does not guarantee the existence of explanations for every instance. Furthermore, it is very cautious as it simply removes all conflicting explanations that may be generated from the considered sample.

This thesis investigates explanation functions that generate abductive explanations from a subset of feature space while satisfying desirable properties. Its contributions are fourfold:

The first consists of proving an impossibility result, which states that a function that generates abductive explanations from a subset of instances cannot guarantee both existence of explanations (*success*) and their correctness (*coherence*). This result sheds light on the reason behind violation of success by the argument-based function from (Amgoud, 2021b). It also resulted in the publication of (Amgoud et al., 2023a).

The second contribution consists of a parameterized family of argumentation-based explanation functions, each of which satisfies one of the two incompatible properties. The

approach starts by generating arguments in favour of classes, identifies attacks among them, uses stable semantics (Dung, 1995) for generating sets of arguments that can be jointly accepted, identifies *accepted arguments*, and uses the latter for defining the novel types of abductive explanations. Accepted arguments are defined in our approach using two parameters: *selection function* and *inference rule*. The former selects a subset of stable extensions and the latter selects (accepted) arguments from the chosen extensions. We define various instantiations of the two parameters, capturing different *criteria* for solving conflicts between arguments.

The third contribution is a formal analysis and a comprehensive comparison of the new functions. We show that the family encompasses the argument-based function, however the new functions that ensure correctness of explanations perform better as they explain more instances and more classes.

The fourth contribution is an experimental analysis of the functions on various datasets. The results confirm that abductive explanations that are generated from datasets (as done by Anchors) are generally incorrect. They show also that the new functions which guarantee correctness perform well as they explain quite an important proportion of instances.

All these contributions contributed to the publication of (Amgoud et al., 2023b).

1.4 Structure of the manuscript

The rest of this manuscript is divided into four chapters and a conclusion. The first one describes the current state of the art on XAI and tries to draw out the current trends in XAI. Along with describing popular methods, it identifies their major shortcomings. The second chapter recalls the argumentation framework that has been proposed in (Dung, 1995). The next chapter applies argumentation techniques for generating jointly coherent abductive explanations from samples. The following chapter presents an implementation of the proposed explanation models. Finally, the last part is devoted to the conclusion and perspectives.

Chapter 2

Explainability in Artificial Intelligence

Explainable Artificial Intelligence (XAI) is a fast growing research area. Indeed, as AI makes its way into every aspect of the public's daily life, it is paramount to make AI more transparent to its users. In order to help every possible AI user to create, improve or operate an AI system, we need to make a great effort into developing the transparency of this powerful tool. The many and various forms that AI can take require a tremendous work from the XAI community and at least as many solutions for regulating them. The research towards this goal is rapidly growing and producing a large amount of techniques to interpret or explain a wide range of AI models. Many authors have made an effort into organizing the literature and offering clear and structured overviews on the **interpretability** or **explainability** techniques. However, it is still possible to find nuances in the point of views of different authors. This is not surprising since the meaning of understanding itself is still being discussed in the Philosophical and Psychological literature. Therefore, it is difficult for the XAI community to form a well defined goal (Páez, 2019).

In this section, we try to draw up the state of the art of XAI. We hope to mention all aspects of Explainability and focus on the points that will be useful in our work. The goal is to give the reader an overview of the existing work and a feel for its vastness. We first present the basic notions of classifiers and explanation model. Then, we try to show the complexity of even defining the most basic notions: **interpretability** and **explainability**. The two following sections, section 2.1 and section 2.2 depict the important concepts of explanandum and explanation. Then, in section 2.3 we showcase a few popular explanation methods. The following section is a review focused on desired characteristics of explanations.

2.0.1 Supervised Learning with Classifiers

First of all, let us present an important prerequisite for the following state of the art. In this thesis, we will focus on a particular technique of Machine Learning: Supervised Learning. Supervised learning is a branch of machine learning where the algorithm learns from a labeled dataset, meaning that each input data point is associated with a corresponding target value or label. The goal of supervised learning is to build a predictive model that can map any input features to the correct labels, in order to make predictions on new, unseen data. **Classification** is a ML task that consists in mapping an input x to a discrete target y (i.e. a class, label or category) from a set of possible classes C . A *classifier* R is the type of ML model that performs the classification task. For example, in email classification, a classifier can determine whether an email is a spam or not, based on its content. An image classifier can classify animals pictures into the animal's family.

Figure 2.1 sums up the definition of a classifier:

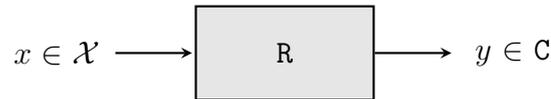


Figure 2.1: The classifier R predicts class y for the input x .

- **Input:** An input (x in Figure 2.1), also called instance, is a representation of the object to classify. The object is described by its characteristics. In ML, characteristics are called features (or attributes) and for each instance, each feature takes one specific value. These features can take various forms, for an image, a feature is usually a pixel, for a text it could be a word. $x = (x_1, \dots, x_n)$ are the corresponding values of these features. In the example of e-mail classification, the features can be the sender's e-mail address, the object of the e-mail and the content of the text. We call *input space* the space of all possible values for x .
- **Dataset:** A dataset \mathcal{Y} in supervised learning consists of a collection of data points, each comprising input feature-values and their corresponding *ground-truth* class. The ground-truth is the label usually given by human as the objective of classification. The dataset is divided into two subsets: the training dataset and the test dataset. The training dataset is used to train the model, while the test dataset is used to evaluate the model's performance.
- **Model:** A model is a mathematical or computational representation of the relationship between input features and classes. The model is represented by the function R in Figure 2.1. In supervised learning, the model learns from the training data and generates predictions for the labels of new data points. Models are usually defined

by their architecture. A model has a collection of parameters that are organised in a specific way depending on many parameters (type of data, task, etc...). During the training phase, the model’s parameters are tuned and optimized by a learning algorithm to make the data x and the model’s prediction c correspond.

Overall, supervised learning with classifiers is a powerful approach for solving a wide range of real-world problems, including image recognition, sentiment analysis, medical diagnosis, autonomous driving, and more, by leveraging large amounts of labeled data to build predictive models. As classifiers can be a central part of critical AI-based systems, it is necessary to have a good grasp of their behavior. For example, in autonomous driving, a ML model has to classify the objects detected around the car. In this context, it is important, in addition to a very high accuracy, to be able to explain the classifier’s behavior.

In order to study the classifiers, the XAI community creates explanation models, also called explainers or explanators to better understand ML models. From the XAI literature, we can count two major families of explanation methods. The first one, *model-specific* explainers, focus on dissecting the models and leverage this knowledge to give explanations. On the contrary, *model-agnostic* methods consider the model as a black-box and give explanations by studying the relations between the input and the output of the black-box classifiers. Figure 2.2 shows the basic idea behind model-agnostic methods.

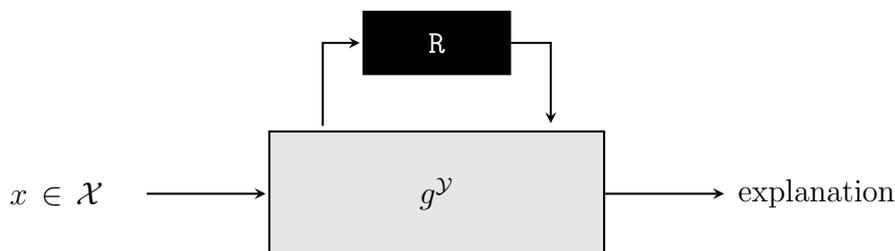


Figure 2.2: A model-agnostic explainer explains x

The model-agnostic explainer is free to query R on multiple inputs (including x) to build the explanation.

2.0.2 Defining concepts: Interpretable or Explainable?

There are two popular terms around the comprehension of Artificial Intelligence: Interpretable and Explainable. We can often read them in articles or presentations as “interpretable Machine Learning” (iML) or “eXplainable Artificial Intelligence” (XAI). These words are often used interchangeably but this trend may lead to a blur in the research goal.

We would like to propose a clear definition to the terms **Interpretability** and **Explainability** that are often used interchangeably by XAI community. We hope this allows the reader to have a clear understanding of our objectives in this research. We also hope to make a step towards an alignment of the different concepts and notions in XAI.

We first list, then discuss relevant ideas in most common definitions we will conclude each review with the definition that is the most aligned with our research.

Interpretability The term **Interpretability** is often used in the literature but no clear definition can be extracted. Here is a list of definition attempts that were proposed in the literature:

- “the degree to which a human can understand the cause of a decision” (Biran and Cotton, 2017; Miller et al., 2017)
- “the degree to which a human can constantly predict the model’s results” (Kim et al., 2016)
- ”The capacity to provide or bring out the meaning of an abstract concept” (Vilone and Longo, 2020)
- “Users can understand the contribution of individual features in the model: quantify the impact of each predictor” (Lou et al., 2012)
- “if their operations can be understood by a human, either through introspection or through a produced explanation” (Biran and Cotton, 2017)
- “the ability to explain or provide the meaning in understandable terms to a human” (Guidotti et al., 2018)
- “Provide qualitative understanding between the input variables and the response” (Ribeiro et al., 2016)
- “an AI model’s decision can be explained globally or locally (with respect to mechanistic understanding), and that the model’s purpose can be understood by a human actor (Páez, 2019)((i.e. functional understanding)” (Schwalbe and Finzel, 2023)

The two most recurring terms are *human* and *understand*. It is clear that to have some interpretability, the understanding of a human is necessary. However, something that is not clear is what should be understood. Is it a decision? a cause? a meaning? a model? its operations? The subject of interpretability is ill-defined. Two other blurry aspects of this term we could think of are the level of understanding of the subject by the human actor and the amount of effort required for this level of understanding.

However, *interpretability* is used in many ways. For example, (Molnar, 2022) and (Schwalbe and Finzel, 2023) agree that *interpretable Machine Learning* refers to the area of research or the discipline:

- “Area of research concerned with the creation of interpretable AI systems (interpretable models)” (Schwalbe and Finzel, 2023)
- “Extraction of relevant knowledge from a ML model concerning relationships either contained in data or learned by the model” (Molnar, 2022)

Interpretable model is a formulation where the subject is well defined. According to (Schwalbe and Finzel, 2023), **interpretable models** are “Machine Learning techniques that learn more structured representations, or that allow for tracing causal relationships. They are inherently interpretable, i.e., no additional methods need to be applied to explain them, unless the structured representations or relationship are too complex to be processed by a human actor at hand.”

In this definition, the *machine learning technique* is qualified as interpretable. The interpretability describes the technique in its globality. Another important aspect is the non need for explanations. When the subject is well defined, it seems that the definition is clearer. Thus we will reserve the term of *interpretability* and *interpretable* to Machine Learning models. For example, we will say that a model is interpretable (to some extent) or not. In this definition, the interpretability of a model should be evaluated by a human actor. The human should have a reasonable knowledge of the task and shall not make an effort to understand the model’s behavior. If the amount of effort necessary to interpret the model is too high and explanations become necessary to understand it, then the model should not be considered inherently interpretable.

Explainability The word is less often defined in the literature, we want to nuance it from interpretability. We first propose a few definitions for explainability:

- “The system either provides knowledge and explanations necessary for the user to carry out his or her task, or alternatively, the system carries out some action and then explains the need and reason for the action the system itself has taken to the user.” (Johnson and Johnson, 1993)
- “models that are able to summarize the reasons for neural network behavior, gain the trust of users, or produce insights about the causes of their decisions.” (Gilpin et al., 2018)
- “Explainability is associated with the notion of explanation as an interface between humans and a decision maker that is, at the same time, both an accurate proxy of the decision maker and comprehensible to humans.” (Barredo Arrieta et al., 2020)

In these three definitions, the idea of additional information is at issue. The term *decision* (or action) of the system is also always present in these definitions. Explainability seem to refer to the possibility for an ML model to provide information in addition to its normal result. This extra information is associated with the notion of explanation. It should induce transparency into the system and help the user understand the stakes of the model’s result. The behavior of the model in a specific work is described by the terms *reasons* or *causes*. The definitions also add information about the goals and desired aspects of explainability: *gain the trust of the users* or provide an *accurate proxy*.

To summarize, **Explainability** concerns the techniques that aim at providing additional information about a ML model’s output to the user. Obviously, this information should be relevant to the model’s task instance.

It is difficult to establish a hierarchy between the notion of interpretability and explainability. For example, (Gilpin et al., 2018) argues that explainability of a model implies its interpretability. One could disagree and argue the contrary: if explanations are needed to understand a model, then it means that it is not interpretable. One could also argue that explainability is more directed to ML decision than the models themselves, and the concept of global explanation should be called interpretability techniques. These two concepts are well entangled and one could spend much more effort into defining clear borders between the terms. In this study we will reserve the word interpretability to models. Explainability, being the focus of this work, will be elaborated in the next sections. The concepts of **explanandum**, **explanation** and **explanator** that are at the core of explainability will be thoroughly studied in section 2.1, section 2.2 and section 2.3 respectively.

2.1 What is there to explain?

Schwalbe and Finzel (2023) refer to an *explanandum* as to “*what* is to be explained in an explanation system. This usually designates a model (e.g., a deep neural network)”, but there are many aspects of a model that brings opacity to the system. In this section, we describe two dual concepts that are fundamental in XAI: scope (subsection 2.1.1) and portability (subsection 2.1.2). The following subsection covers several other aspects of ML models that XAI researchers tackle.

2.1.1 Scope: Explain a model or a prediction?

The scope is a dual concept in XAI: it is either local or global. On the one hand, an explanation is considered local when the explanandum is a specific instance. This is the case in our brief presentation of model-agnostic explainer fig. 2.2 on page 21. In this figure, the classifier is queried by the explainer g^y on multiple instances that are *close* to x to

build the explanation of the prediction of input x by the classifier R . This collection of instances is called a distribution. Usually, the user expects an explanation that provide information on the behavior of the model **around** this input x by querying the model R on similar instances taken from the distribution. The locality scope allows to focus on predictions in specific contexts. Large models can have many features but it is not always relevant to explain their behavior in all situations. For this reason, humans prefer to have explanations on situations that are close to theirs. Let's take the example of Bob, a bank customer who wonders why his loan was rejected. Since the banker uses a ML-based loan risk advisor he should give him a reason. In this case the customer is only interested in his own situation and what he can do to increase his chances to get the loan. Nevertheless, this scope raises the following question: "To what extent is this explanation valid?". Indeed, this notion of distribution is not always precisely defined and does not constitute a good indication on the range of validity of each explanation. The banker told Bob that his income was too low for the loan. Does this mean that Alice, who has a bigger salary, could get the loan? Some notable Local explanation methods that address this situation are SHAP (Lundberg and Lee, 2017), LIME (Ribeiro et al., 2016), Anchors (Ribeiro et al., 2018) (that are detailed further, subsection 2.3.2) and RISE (Petsiuk et al., 2018), Sensivity analysis (Baehrens et al., 2010) and deconvNet (Zeiler and Fergus, 2013).

On the other hand, some explanations try to describe the model *globally*. Their purpose is to give a general idea of how the model works. For example, one can be interested in locating the decision boundaries of a model. The knowledge of boundaries can be useful to extract rules on predictions of the model: "As long as instance I remains within these decision boundaries it will be classified as class c ". This is the case of VIA (Thrun, 1994) and DeepRED (Zilke et al., 2016). A global explanation is usually most desired to gain trust in a model. The more we globally understand a model, the more trust it gains. This global understanding provides trust without searching corner-cases that we don't need to specifically cover. In this regard, the user can expect a coherence between all explanations. In other words, the different global explanations given by the same explanation function should not contradict each other, or the explainer cannot be trusted. Since ML models are usually quite big, and decision depend on more than 3 or 4 features, it is very difficult to identify decision borders and even impossible to make them understandable to humans.

One can also argue that with enough local explanations, it is possible to build a global comprehension of a model. This is the case for Anchors (Ribeiro et al., 2018). However, the global explanations are hard to evaluate since we cannot prove they cover all possible decision boundaries. Sometimes, predictions in some part of the input space can be more chaotic than another and may require more local analysis than other parts.

As explanations are vectors to induce trust in ML models, the choice between local and global explanations should be carefully studied. The result of this choice should impact

the context of use of the model. A globally safe model could be used by any user without worrying about border cases that may not have been seen. A locally safe model would be unsafe for unadvised users since they could use the model on corner-case inputs that have not been tested before.

2.1.2 Portability: Explain an opaque or a transparent system?

There is a very large amount of different models in the literature and some are more interpretable than others. When it is possible, it is useful to dive into the inner workings of each model. It is necessary to understand that the models belong to a full spectrum and can be more or less interpretable according to their nature, but also their size (or depth), the task they perform, the nature of the data etc. When explaining a ML model, some of the characteristics of the model can be meaningful information, or can allow more faithful explanations. For example, most simple ML models such as Bayesian Network, Decision Trees or Linear Regression models have very simple mechanics that can be easily leveraged to compute explanations. We call Model-Specific Explanation methods, the methods that leverage the knowledge of the structure or of the weights of the explanandum. Model specific explanation methods are numerous and concern most ML models of the literature. Some explanation methods rely on the gradient of the prediction function to build explanations such as the Sensitivity Analysis (Baehrens et al., 2010). When the problem's dimension is low enough, the gradient, indicating the direction of the local optimum can be showcased as a local explanation. Other methods can rely on the analysis of the weights for attention analysis or the backpropagation operation such as Deconvnet (Zeiler and Fergus, 2013), Backprop (Simonyan et al., 2014) or LRP (Bach et al., 2015).

On the other side of the spectrum, Deep Neural Networks can accumulate huge amounts of parameters. For example, the popular Large Language Model GPT4 accumulates 1.7 trillions parameters. In the case of excessively complex models, it is preferable to consider the latter as a black-box model and make abstraction of its inner mechanisms. Moreover, the choice of ignoring the model's information has multiple benefits. Firstly, to make abstraction of it and adapt the explanation method to a greater number of models. Secondly, an explanation function that is model-agnostic does not require the users any knowledge on the explanandum. This is the case for LIME and Anchor (Ribeiro et al., 2016, 2018) that are very popular explanation methods. Model agnostic methods also have drawbacks. Model agnostic explanation functions cannot leverage the inner mechanisms of the explanandum. Hence, it is harder to prove that the explanation derives from actual causes or only correlations. Some model-agnostic explanation functions require to build a proxy model, also known as a surrogate or mimic model. This proxy model is usually interpretable, is trained to behave like the explanandum and allows to extract explanations from it. The issue is how to guarantee faithfulness between the surrogate model and the

explained model. LIME is an example of explainer based on a surrogate model. We explain this solution later in section 2.3.2.

2.1.3 Focus on specific models' aspects

The XAI research community is devoted to knowing more about any aspect of any model. While the ultimate goal would be to find a miracle explainer that is model-agnostic, global and with strong guarantees, it is necessary to proceed by small specific steps. Researchers make small improvements and build tools and methods that aim at explaining specific aspects of the ML models. These aspects cover all parts of the creation of a ML model. The structure and symbolic processing pipeline, the training, the uncertainty of predictions and the data itself. Knowledge about all these aspects is useful to build a 360 degrees representation of complex models.

Processing The study of the processing pipeline aims at providing knowledge about the decision boundaries of the models. Solutions can be either model agnostic or model specific. Model agnostic methods count solutions such as RISE (Petsiuk et al., 2018), LIME (Ribeiro et al., 2016) and LRP (Bach et al., 2015). These are 3 feature attribution methods. Their goal is to provide information on the role of each feature in the model in the prediction. In other words, which characteristics of the input are the most responsible for the prediction. TREPAN (Craven and Shavlik, 1995) and Concept Tree (Renard et al., 2019) are also model agnostic methods that use decision tree extraction as proxy model to extract knowledge from the models. This knowledge is appropriate to provide rule based explanations such as contrastive, counterfactual or abduction based explanations. These explanations are very well received by non expert users (Miller, 2021).

Inner Representation Complex ML models such as Convolutional Neural Networks (CNN) for Computer Vision and transformers for Natural Language Processing (NLP) create a new representations of the input space. This new representation is called the latent space. This latent space may provide new information on either our own representations or the model's behavior. In both case it is an interesting aspect of complex models. Researchers tend to look for links between latent representations and human known semantic concepts. NetDissect (Bau et al., 2017) try to link internal elements of CNNs to human concepts such as color, texture etc. We can also mention Net2Vec (Fong and Vedaldi, 2018), TCAV (Testing Concept Activation Vectors) (Kim et al., 2018), ACE (Automatic Concept-based Explanation) (Ghorbani et al., 2019) that are similar methods. For these methods to be trustable, it seems important to guarantee that this information accurately represent the explanandum's behavior. This would be done by providing results of the correlation between the latent representation and the semantic concept.

Development (training) The model behavior eventually depends on two main components: the architecture and the training. It is natural to wonder what is the impact of the training data on the final prediction model. Firstly, evaluating the quality of the training data is a first step, but we will discuss this task in the next paragraph. The question in the training part is what is the effect of new samples on the final model. Influence Functions (Koh and Liang, 2020) is a popular attempt in explicating the influence of training samples on the training and the decision process.

Data The training data quality is of major importance for the quality of the prediction model. Explaining via data is a “ante-hoc” method that improves the global process of building the prediction model as well as extracting knowledge for explanations. In order to build a good model, it is important to have numerous data that cover well the input space and represents well the real world distribution of the problem. This means that models will train well and produce better results on inputs that are taken from real life scenarios. Datasets can also be synthetically extended to induce robustness. Study of data allows to know more about the input space representation that we built with the dataset. If this representation is of too high dimensionality, methods such as PCA (Jolliffe, 2002) or t-SNE (Maaten and Hinton, 2008) allow to project the data into 2D or 3D spaces. They can also be used to build simpler models with less features.

Uncertainty Knowing how certain a model is for each prediction could be a useful information to have in mind when a decision follows the prediction. This is a common behavior for humans to tell their degree of certainty when a decision is to be made. The role of classifiers is to choose a class for a given input. Actually, the raw output of the model is given by a logistic function that maps each class to a value between 0 and 1. The predicted class is the class that corresponds to the largest value. It is surely interesting to know if all probabilities were really close to each others (the model is unsure since all classes are equally likely) or not (one class is much more likely than the others). Although, the *uncertainty* information is available for black-box models, it is rarely leveraged for explanations. One issue about this value is that it suffers from a lack of calibration, especially for deep models (Guo et al., 2017). In the case of safety critical systems such as autonomous driving, the systems are lead to make decisions based on the predictions. When the decision is uncertain, the system should be able to switch to a safety fallback routine in order to reduce any risk. This issue is tackled by (Kumar et al., 2019; Henne et al., 2020). In the latter, the authors benchmark several methods to improve the estimation of uncertainty in Deep Neural Networks predictions.

2.2 What is an explanation?

Explanations are the interface between opaque or black-box intelligent systems and the human. Explanations showcase details and reasons on the causal relationship between the input of the system and the predicted output. In the field of XAI, researchers have been actively proposing ways to provide new information on AI-based systems' behavior. There are various solutions that apply in different circumstances.

According to the Cambridge Dictionary, an explanation is “the details or reasons that someone gives to make something clear or easy to understand”.

There are many ways to explain something to a person. Depending on the task, on the context, on the data, on the interlocutor, the explanation should vary. We present in this section four different ways to explain in the context of Machine Learning. We first describe abduction-based explanations which is a central notion of this thesis.

Abductive explanations Abductive, or abduction-based, explanations, are answers to the ‘Why was x classified as c_1 ?’ question. Abduction-based often search for the sufficient characteristic of an instance to be classified a certain way. Abduction-based explanations are the base of rule-based explanations as they can be directly translated into them. For example, if a sufficient characteristic to be granted a loan by an AI-based loan decision making system is ‘not having a loan’ and ‘earning \$50k per month’, then the corresponding rule would be ‘*if* you do not have a loan *and* your salary is \geq \$50k, *then* the loan is granted. Moreover, abductive explanations are constructed to have a minimal set of requirements to imply a prediction. (Ignatiev et al., 2018) studies the construction of such explanations. In (Ignatiev et al., 2019), the same author explores good properties of such rules and finds relationships with contrastive explanations.

Contrastive explanations Contrastive or counterfactual explanations aim to provide insight into why a model made a particular prediction by presenting an alternative scenario. These type of explanation has been actively motivated by Miller (2018) who extensively studied the literature in social sciences and philosophy to decide what is a *good* explanation for a human. The main take from social science is that good explanations for humans are *contrastive*. A person often wants to understand specific cases and their boundaries. The person will often ask ‘Why was c_1 predicted rather than c_2 ?’. For instance, in credit scoring, a counterfactual explanation may indicate the changes in income or credit history required for an applicant to be approved. If a loan application is denied, a contrastive explanation might suggest that the application would have been approved if the applicant’s income were \$10,000 higher. In (Wachter et al., 2017), the authors propose a counterfactual based explanation fonction and study how the proposition aligns with the GDPR’s guidelines.

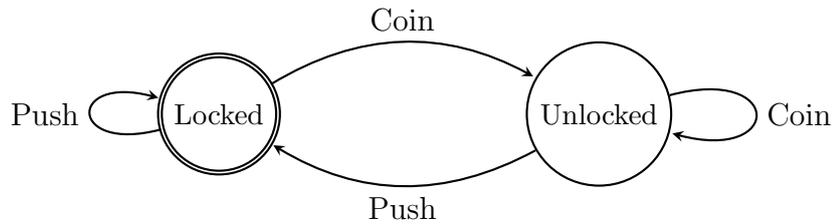


Figure 2.3: The state diagram (graph) for a turnstile, whose nodes represent states and edges represent transitions.

Feature importance explanations Feature importance explanations focus on identifying the most influential features or variables that contributed to a model’s decision. These explanations are valuable for understanding which factors had the most significant impact on a prediction. Feature importance is an interesting take in image classification where pixels alone do not have a particular meaning but together provide meaningful information. With feature importance, it is possible to showcase the importance of groups of features to show what part of an image was important for the classifier. This has been proven useful in model debugging. For example, (Ribeiro et al., 2016) found a bias in the training data with the example of a husky classified as a wolf because of the snowy background. Another example is in NLP, attention-based explanations show which words of a text are most important for the task. Feature importance can be measured using various techniques, such as permutation importance (feature’s values of an instance are permuted to measure the difference of output), SHAP values (Lundberg and Lee, 2017) (explained in section 2.3.2).

Graph-based explanations Graph-based explanations leverage graphical representations to illustrate relationships and dependencies within a model. These explanations are particularly valuable when dealing with structured data or models with complex interactions. Graph-based explanations can provide insights into how features or entities are connected and influence each other. (Zhang et al., 2018) evaluates knowledge hierarchy in Convolutional Neural Networks by leveraging graphs. Graph are a good tool to image interactions between several agents. It is a good representative tool that is used in many other domains such as engineering (e.g. state graphs Figure 2.3), social sciences, economics etc.

2.3 Explanation functions

In this section, we present explainability techniques. We begin with two intrinsically interpretable ML models that are relevant to our scope. They are interesting because they are self-explainable and they can be used as proxy models to explain larger ones such as

in *LIME*. The goal is to see for each model how it is interpreted and what explanations we can extract from them. Then, we present 3 model-agnostic explanation methods: *LIME*, *Anchors* and *SHAP*. For each method, we describe how the information is extracted from the black-box model, what kind of explanation are provided and their locality and finally, what are the guarantees for explanations.

2.3.1 Intrinsically interpretable models

Logistic regression Logistic regression is a classification model based on linear regression model. A linear regression model linearly combines the features which are weighted.

The Mathematical formula is as follows. Let f the linear regression model, p the number of features, $x = (x_0, \dots, x_p)$ an instance of the regression problem.

$$f(x) = \omega_0 + \sum_p^{j=1} \omega_j x_j \quad (2.1)$$

The training of this models consists in finding the weights that will minimize the error between the prediction and the ground-truth labels. The interpretation of a linear Regression model is very simple. Any feature x_j participates to the prediction according to its corresponding weight ω_j . Moreover, the features are not dependant to each other.

The logistic regression models leverages the regression model to determine the probability for an instance to belong to the positive class. First, the linear function separating the two classes is the same as the linear regression model. Then, the sigmoid function is applied to determine the probability.

$$P(y = +1|x = x) = \frac{1}{1 + \exp(-f(x))} \quad (2.2)$$

The explanation resides in the hyperplane separating the data. This hyperplane is defined by $f(x)$ it is thus as simple to interpret than the linear regression.

Logistic regression is a good model to classify linearly separable data, but performs poorly otherwise. In this sense, it can still be an interesting choice when the hypothesis of the linearity of the data is reasonable. In this case, one can benefit from its efficiency and transparency.

Decision Trees Decision trees are popular for their simplicity and interpretability and have the advantage of capturing non-linear relationships in the data. The training of tree models consists in separating the data into subsets according to discriminant features. The goal at each stage is to find the cutoff value for this feature that best separates the data according to their labels. In the tree structure, the prediction is leaves represent the class of the instance and the intern nodes are conjunctions of features that lead to a label.

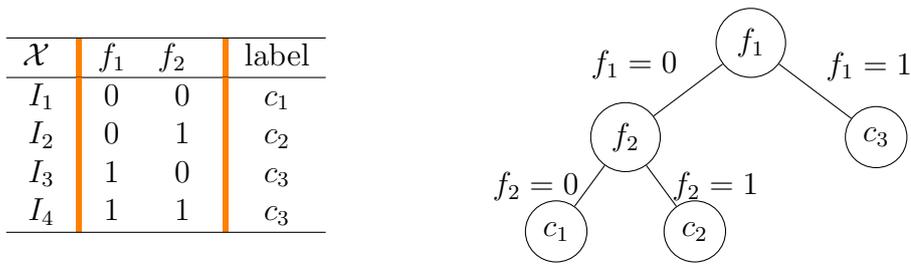


Figure 2.4: The decision tree classifier for dataset \mathcal{X}

A user can simply read the tree from the root and follow the branches corresponding to the predicted input in order to read the prediction. The most common learning process for decision trees is called top-down induction of decision trees (TDIDT). It is based on applying recursively the splitting of the learning dataset. Each set of data represents a node and a rule is used to split the data into two subsets, constituting the successor children. When all the instances in a set have the same label, the recursion is stopped.

Figure 2.4 shows a toy example of a decision tree classifier fit to predict the labels of instances from \mathcal{X} . From this tree, it is very simple to extract rule conjunctions to explain the classification of instances. For example, we can give the following reason for the classification of I_1 : **IF** $f_1 = 0$ **AND** $f_2 = 0$ **THEN** $Salary \leq 50K$. To explain the decision of I_3 and I_4 , we can give this explanation: **IF** $f_1 = 1$ **THEN** $Salary \leq 50K$.

There also exists techniques that use multiple trees such as Boosted trees or Bootstrap aggregated decision trees (random forests). They are called ensemble methods. Although these methods can show great results, the size of the models can make the interpretation more difficult.

2.3.2 Model-agnostic explainers

Model-agnostic explainers generally aim to globally mimic the behavior of a larger model or a function that approximate more specific aspects of the large model. A model aiming to mimic a large model while remaining interpretable is called a surrogate model or a proxy. There are two steps for these models: the process of training the simple model as accurately as possible with respect to the original model and then the task of explaining its own prediction.

LIME: LIME (Local Interpretable Model-Agnostic Explanations) is a popular method for explaining the predictions of machine learning models. It was introduced by Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin in (Ribeiro et al., 2016). LIME is a model-agnostic explanation method on individual predictions.

In this analysis, we provide a detailed explanation of LIME, its explanation function, and the guarantees it offers, as well as highlight some of its pros and cons based on other

studies.

LIME provides explanations by approximating the model's behavior in the local neighborhood of a specific data point. Here's how it works:

1. Selecting an Instance: LIME starts by selecting a data instance for which you want an explanation. They define $x \in \mathbb{R}^d$ the original representation of an instance to be explained. Then, an interpretable representation of x is also defined $x' \in \{0, 1\}^{d'}$ (e.g. superpixels or presence of words.)
2. The goal is to explain f , $f : \mathbb{R}^d \rightarrow \mathbb{R}$. To do this, LIME creates $g \in \mathcal{G}$, $g : \{0, 1\}^{d'} \rightarrow \mathbb{R}$ or $[0, 1]$ and \mathcal{G} is the set of potentially interpretable models (linear models, decision trees etc.) and $\Omega(g)$ is the complexity of the interpretable model. It is arbitrarily defined.
3. To create this set of interpretable models, LIME generates a dataset of perturbed versions of x by making perturbations based on a proximity metric π_x . π_x represents the locality around x and $\pi_x(z)$ is the proximity measure between z and x .
4. The perturbed dataset is used to probe the black-box classifier and records the outputs. The recordings are transformed to the interpretable representation and then used to fit interpretable surrogate models in the interpretable representation. In the article, the authors chose sparse linear models as interpretable models.
5. The surrogate model and explanation $\xi(x)$ is the solution to the optimisation problem:

$$\xi(x) = \operatorname{argmin}_{g \in \mathcal{G}} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

\mathcal{L} is a measure of the fidelity of g to f in this example, the authors use the square loss function weighted by the proximity function. We note that $\xi(x) \in \mathcal{G}$.

6. Finally, LIME provides explanations based on the interpretable model's coefficients. It identifies which features had the most significant influence on the prediction and quantifies their impact.

LIME offers several advantages but also some shortcomings. Firstly, LIME provides local explanations to individual predictions. The methods explicitly optimizes fidelity based on the \mathcal{L} function and interpretability with Ω . To a certain extent, the method ensures that the explanation is locally faithful to the black-box model's behavior around the selected instance. This means that the explanation should accurately reflect how the black-box model behaves for similar data points in the neighborhood. This fidelity guarantee is however limited by two aspects. The difference between the actual representation and *interpretable representation* of inputs in the model is a first hindrance to fidelity to the

original model. The second aspect is in the definition of the proximity measure π . (Zhang et al., 2019) showed that LIME presents several sources of uncertainty. Moreover, since LIME doesn't provide confidence intervals or uncertainty estimates for its explanations, it is difficult to measure the extent of the validity of an explanation and if similar data is also covered by the explanation.

Finally, (Sokol and Flach, 2020) perform a complete analysis based on their own taxonomy of desiderata. They conclude that LIME's interpretable model, being a simple and transparent one, ensures that the explanation itself is easy to understand and validate. Moreover, LIME's explanations may vary based on the choice of perturbed instances and their distribution. Different perturbation strategies can lead to different explanations. These observations concern two desired properties of explanation techniques, respectively, transparency and robustness. They are detailed later in section 2.4. The quality of LIME explanations can depend on user-defined parameters, such as the number of perturbed instances or the choice of the type of interpretable model.

In summary, LIME is a powerful tool for explaining individual predictions of machine learning models, offering local fidelity and interpretability. However, its sensitivity to perturbations and potential subjectivity in parameter choices are aspects to consider when using LIME. Researchers and practitioners can combine LIME with other explainability techniques to gain a better understanding of model behavior. (Cian et al., 2020) use GradCAM and LIME to explain predictions of a Convolutional Neural Network in a classification task. They conclude that using both methods can yield complementary insights. Simulatability evaluates if the explanation allow humans to predict the model's behavior in new situations (see section 2.4). This property was evaluated in a human centered test in (Hase and Bansal, 2020).

anchors: *Anchors* short for "High-Precision Model-Agnostic Explanations" (Ribeiro et al., 2018) is designed to offer precise and understandable explanations for individual predictions of machine learning models. It shares similarities with LIME but focuses on finding a minimal set of conditions (anchors) that are both sufficient and necessary for a particular prediction.

Suppose we want to explain the prediction of a classifier f on an instance x . Firstly, the authors define *rules*. *Rules* are sets of conditions $A = \{predicate_1, \dots, predicate_n\}$ on feature values that anchor the prediction. If the predicates are true for an input x , then $A(x) = 1$. Secondly, \mathcal{D}_ξ is distribution of perturbations from the explained instance x . Elements of \mathcal{D}_ξ , z , are derived from the rule and in the vicinity of x . A rule A becomes an anchor if all $z \in \mathcal{D}_\xi(z|A)$ (perturbed instances that verify A have the same prediction as x).

To compute anchors, the framework introduces two important metrics: Precision and Coverage. Precision represents the likelihood that perturbed instances z that satisfy the

rule A are predicted as the same class as the explained instance x . Equation 2.3 is the formal definition of the Precision metric.

$$prec(A) = \mathbf{E}_{\mathcal{D}_x(z|A)}[\mathbb{1}_{f(z)=f(x)}] \quad (2.3)$$

Since it is intractable to compute the precision exactly, the authors chose a statistical approach. For the rule to become an anchor, the anchor has to satisfy a precision threshold with high probability. The coverage of an anchor A is the probability that A applies to a sample in \mathcal{D}_\S . The formal definition is given in Equation 2.4

$$cov(A) = \mathbf{E}_{\mathcal{D}_x(z)}[A(z)] \quad (2.4)$$

The construction of anchors is made incrementally by starting from an empty rule and adding predicates. When there are enough predicates to satisfy the precision constraint, the best anchor is found. The coverage is naturally maximised since the anchor is as short as possible. A second, less greedy approach is proposed in the paper where candidate rules are searched using KL-LUCB, an instance of multi-armed bandit exploration problem (Kaufmann and Kalyanakrishnan, 2013). In this version, a set of anchors that satisfy highest precision are collected and the one with the best coverage is returned.

Anchor is a very popular explanation method. It offers several important advantages:

- It is model agnostic: Anchor is tailored to be used on any classifier.
- Anchor offers high precision explanation by finding minimal sets of conditions that accurately capture the decision boundaries of the black-box model.
- Anchor provides explanations that are both sufficient (guaranteeing correctness) and necessary (minimal) for the prediction. This ensures that the explanation locally reflects the true behavior of the model.
- The anchor conditions are interpretable and can be presented as simple rules, making them understandable to users.

On the other hand, the authors affirm that Anchor can be used to globally explain the model thanks to an optimized set of anchors. This statement should be taken with a pinch of salt. In Example 1, we see that Anchor can sometimes yield incoherent results.

Example 1 (Incoherent anchor explanations) *Anchor was used to explain two instances x_1 and x_2 .*

- *The anchor explanation A_1 returned for x_1 with a precision of 1 is*
if Capital Loss = 2 AND Marital Status = Married-civ-spouse AND Hours per week > 45.00 AND 28.00 < Age <= 37.00 THEN Salary <= 50K

- The anchor explanation A_2 , yielded for x_2 , with a precision of 1 states that
*textbfif Capital Loss = 2 AND Education = Bachelors AND Sex = Male AND Race = White AND Workclass = undefined AND Occupation = Machine-op-
inspct AND Relationship = Not-in-family AND 28.00 < Age <= 37.00 THEN Salary > 50K*

The A_1 rule has constraints on Capital Loss, Marital Status, Hours per week and Age features while A_2 has the same constraint on Capital Loss than A_1 and other constraints on different features. The example shows two anchor explanations with opposite conclusion that have a perfect precision but could, in theory, concern a single instance.

This issue of “Potentially conflicting anchors” was raised by the authors in (Ribeiro et al., 2018) along with other limitations such as overly specific anchors or the limits of perturbations. Evaluations of the Anchors method seem scarce. (Hase and Bansal, 2020) evaluated Anchors among LIME and other methods to evaluate their simulatability. The study evaluating the limits of Anchors is missing although Ribeiro et al. themselves acknowledge limitations of the methods such as possibly overly specific anchors when the instance is near a decision boundary, or possibly conflicting anchors.

In summary, Anchors is a model-agnostic method that aims to provide high-precision and interpretable explanations for individual predictions. Its guarantees of sufficiency and necessity make it a valuable tool for understanding complex machine learning models. However, its performance may vary depending on the specific use case and dataset, as well as lacking in guarantees for explanations.

SHAP: The SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) explanation function is a cutting-edge technique in machine learning interpretability that builds upon previous work in the field of cooperative game theory and Shapley values. The concept of Shapley values, initially introduced by (Shapley, 1953), was designed to allocate the contribution of each player in a cooperative game. In the context of machine learning, SHAP values are applied to explain individual predictions made by complex models, such as black-box models like deep neural networks or ensemble methods.

SHAP values aim to provide a clear and intuitive understanding of how each feature contributes to a specific prediction. They take into account all possible feature combinations and assess the impact of each feature’s inclusion or exclusion in these combinations. This exhaustive evaluation enables SHAP values to offer a comprehensive view of feature importance and interactions.

The theoretical Shapley Value Estimation consists in estimating an importance value to each feature by comparing results of models trained with and without this feature. The differences in output of these models is computed for all feature $S \subseteq F$ with F the set

of all features. This process is extremely costly. In order to tackle this issue, (Lundberg and Lee, 2017) proposes SHAP to locally approximate the Shapley values respecting three important properties. The idea is based on the use of a surrogate model called the explanation model g that approximates the original prediction model f .

- **Local accuracy:** this property requires that the explanation mode g matches the predictions of the original model f .
- **Missingness:** If a feature's value is equal to zero across the whole dataset, then the attributed impact on this feature should be nil as well.
- **Consistency:** consistency guarantees that the feature impact value cannot decrease if it contributes positively to inputs' prediction

To compute SHAP values, several methods have been developed in accordance with the Local accuracy, Missingness and Consistency propoerties. TreeSHAP (Lundberg et al., 2020) and KernelSHAP (Lundberg and Lee, 2017) being among the most widely used. TreeSHAP is tailored for tree-based models like decision trees and random forests, while KernelSHAP is applicable to any model by employing a kernel approximation technique.

SHAP and LIME have been studied together to evaluate their performances. (Gramegna and Giudici, 2021) makes a comparative study between LIME and SHAP in the context of credit risk estimation. They report that SHAP is showed better results to assign importance to features used by a black-box prediction model. In (Vega García and Aznarte, 2020) SHAP is successfully used to interpret the decision of an opaque ML model in the context of NO_2 pollution forecasting.

In summary, SHAP values represent a significant advancement in machine learning interpretability, building upon the foundation of cooperative game theory and Shapley values. They provide a nuanced understanding of feature importance and interactions, with well-defined guarantees of consistency, additivity, and fairness, making them a valuable tool for model explanation and decision-making transparency.

2.4 Explanation Desiderata

Explanations are the interface between the AI systems and humans. It is the final product of the explanation function. As any product, it should be considered with its own certifications, ergonomoy, and context of use. First of all, it is important to provide an advised context of use. Whether the explanation is meant to provide an intuition or a very faithful representation of the model's behavior, if the model should be used in sensitive environment such as autonomous driving, jurisdiction or it is made for entertainment or business purposes. Unfortunately, explanations tend to lack clarity on the guarantees their

results may provide. Certifications that explanation function provide should always be clearly discussed.

When using an explanation system, users can doubt the degree of truth in the information given by the explanation. They can wonder “Does this explanation reflects the actual behavior of the ML model?” or ”Is this information true all the time, in any situation?”. Also, if the context is a model used by non-expert users, it is important to provide a flexible explanation function that outputs simple explanations. On the other hand, if the context is more precise, it would not be required to provide simple explanation, expert would probably prefer more complex but more precise explanations. These notions around **Explainability** are examples of desired properties, or desiderata. The literature of XAI being very abundant, the definitions of each concept are numerous and not always aligned. In this section, we present a list of these desiderata and try to gather similar concepts under an umbrella term while expliciting the nuances between them. In a second phase, we present applied metrics that can be used to measure and compare the quality of explanations.

2.4.1 List of Desiderata

In this subsection, we develop a list of desiderata for the quality of explanation systems and explanations themselves. Desiderata are desired properties or metrics that should be expected from stakeholders for an explanation system. Nevertheless, we will see that it is quite difficult to develop standards. Indeed, most XAI researchers have influences of other fields of research. This has the effect of creating a very rich set of points of view from which stems a wide variety of notions and concepts which themselves can have multiple definitions with small variations. It is a fact that many research fields such as social sciences, Psychology, Mathematics Philosophy and more have tackled the issue of explainability and have developed their own desiderata. The objective here is to describe an overview of the desired characteristics for explanation methods as well as defining them and gathering the ones that are similar. For that, we summarize most notions presented in the this subsection. In Table 2.1, we regroup similar concepts under a representative concept in order to disambiguate the terms. Moreover, we mention to what object the desiderata predicates to.

In (Doshi-Velez and Kim, 2017), the authors lay out a taxonomy of evaluation approaches for explainability and interperatability. They divide the approaches into three categories: human-grounded, application-grounded and functionally-grounded. In this section, we provide a description and present relevant metrics for each category and we focus on functionally-grounded metrics.

Firstly, **human-grounded evaluation** is intended to qualify the degree of appreciation of an explanation by a human user. In this category, we can find two desired characteristics

Concept	predicates to	is similar to
Interpretability	Models	Understandability Simulatability Intelligibility Effectiveness Decomposability Transparency Predictability
Robustness	Models, Explanators	Stability Consistency
Portability	Explanators	Transferability Translucency
Faithfulness	Explanators, Explanations	Soundness Causality
Simplicity	Explanations	Complexity Selection Conciseness
Informativeness	Explanations	Relevancy Interestingness Triviality

Table 2.1: Disambiguation table for XAI desiderata

for explanations. *Interpretability*, is a notion that we presented previously and that is one of the most present desiderata in literature. In (Doshi-Velez and Kim, 2017; Lipton, 2017; Hase and Bansal, 2020), it is mentioned under the name of *simulatability* or *Predictability* in (Vilone and Longo, 2020). This concept involves the ability of the user to predict the model’s outputs thanks to the model’s interpretability or explanations. It is an evaluation metric used in practice in (Ribeiro et al., 2016). The concepts of *Understandability* (Vilone and Longo, 2020; Barredo Arrieta et al., 2020), *Intelligibility*, (Vilone and Longo, 2020; Bellotti and Edwards, 2001; Kulesza et al., 2013), *comprehensibility* (Fel and Vigouroux, 2020; Barredo Arrieta et al., 2020), *Effectiveness* (Schwalbe and Finzel, 2023) or *transparency* (Barredo Arrieta et al., 2020; Lipton, 2017) seem to be used to characterise intrinsic interpretability of ML models. It seems *simulatability* is an informative tool to evaluate a ML model. This metric’s interest depends on the quality of the experimentation and the choices of the authors, thus, it should be defined with a strict protocol (e.g. Should users know the model’s accuracy? Should the users predict the same amount of correct and incorrect outputs?).

Another important concept related to human users is *simplicity*. *Simplicity* (Lombrozo, 2007; Vilone and Longo, 2020) is also referred to as *complexity* in (Ribeiro et al., 2016). As this metric is based on human appreciation, the metric is usually decided arbitrarily. For example, in the latter article, the complexity of a decision tree can be its depth, or the number of elements in a rule. We detail these types of explanation in section 2.2. *Simplicity* characterizes explanations, and more precisely, the load of information contained in them. The knowledge advanced by an explanation should not overload the user’s mind and be difficult to understand. *Selection* is a close concept proposed in (Vilone and Longo, 2020) and add an idea of sufficiency in explanations to avoid bringing irrelevant information that blurs the explanation.

Application-grounded evaluation focuses on explainers applied to specific tasks. Depending on the context or task, the stakeholders can have different expectations for explanations. This is a characteristic that depends on the task or stakes of the problem. The metrics can evaluate the ability of explanation to identify errors cases, bias, or unknown

relationships between features. *Informativeness* is a redundant notion for evaluation of explanations. It describes the usefulness of information brought in play by the explanation to the users (Vilone and Longo, 2020). In the same article, the author also defines a similar concept, *Interestingness* which involves a degree of novelty in the explanations. This notion falls under this category because the quality of the information raised by the explanations should be judged by domain experts and for well defined tasks.

Functionally-grounded Evaluation is our main focus in this thesis. The desired characteristics of this category are more abstract and can apply to most explainers.

Portability refers to the range of models that an explainer can explain. In subsection 2.0.1, we introduced this concept with *model-agnostic* and *model-specific* explainers. If an explainer considers the ML model as a black-box, it can explain all similar models as long as the task and the feature space remain the same. On the contrary, *model-specific* models have a lower portability since they can only explain one type of model. This characteristic is also called *transferability* (Vilone and Longo, 2020) or *translucency* (Schwalbe and Finzel, 2023).

Faithfulness, according to (Schwalbe and Finzel, 2023; Jacovi and Goldberg, 2020; Vilone and Longo, 2020), “measures how accurately the behavior of the explainer conforms with that of the actual object of explanation.” Depending on the context, the definition can slightly vary to adapt to the method. For example, (Ribeiro et al., 2016) considers the accuracy only in the vicinity of a single instance, the one to explain. This notion also comes with its similar concepts or synonyms: *soundness* (Sokol and Flach, 2020; Kulesza et al., 2015; Vilone and Longo, 2020) or *causality* (Vilone and Longo, 2020). As models are more complex (i.e. have a larger feature space) or deeper structure, the explanations must concern more features implying a trade-off with *simplicity*.

Robustness is desired for ML models as well as their explainers. According to (Doshi-Velez and Kim, 2017), “*robustness* ascertains whether algorithms reach certain levels of performance in the face of parameter or input variation”. In AI-based systems should predict similar outputs for the same inputs with small perturbations. This is true for instances that are expected to be far from decision boundaries. In the literature, robustness can be tested on classifiers by searching adversarial examples which are instances with minimal alteration that get classified differently than the original instance. Other papers such as (Jacovi and Goldberg, 2020; Alvarez-Melis and Jaakkola, 2018; Wolf et al., 2020) define this concept in the same way. Other words such as *Stability* (Fel and Vigouroux, 2020; Molnar, 2022) and *Consistency* are also used to define this notion.

For a more diverse list of desiderata and notions, the reader can refer to (Vilone and Longo, 2020; Schwalbe and Finzel, 2023). These two articles make a very extensive review of the different concepts and notions around XAI. We decided to narrow the scope of this review to the most recurrent terms in the literature.

2.4.2 Quality assessment: Metrics

In the previous section, we presented a list of desired characteristics for explanations. In order to evaluate how an explanation or an explainer fulfills these desiderata, it is important to define metrics. With a clear metric, one can evaluate a characteristics of its explanation or explainer as well as compare it to other methods. However, it is difficult to define general metrics to evaluate and compare different explanations. The techniques to generate explanation are significantly different and often focus on different goals. This explains why few comparative studies are made to assess the existing solutions. In this section, we review a set of metrics that researchers use in their explanation functions to provide a certain degree of guarantee on aspects of the explanation. Most explainers aim at providing human readable explanations. These explanations can be assessed in human-centered experiments. Nevertheless, these studies would better fit in social science or psychological study. Thus, we focus on quantitative measures and what they represent. We show that these metrics are related to some desiderata.

Simplicity Although *simplicity* is a human-grounded concept, the *coverage* metric is able to quantify it to some extent. Evaluation of *coverage* involves assessing a quantity of instances concerned by an explanation. It is usually used for rule-based explanations. A good *coverage* implies several advantages for simplicity. Firstly, to have rules with better coverage allow to cover a bigger part of the input space. Moreover, a rule that has less conditions is more likely to be useful for other similar inputs. Thus, it is also a indicator of *simulatability*. Since Anchors (Ribeiro et al., 2018) provides rule-based explanations, coverage is used as an evaluation metric. Again, in this situation, the rules are local, so the coverage is also defined locally. The exact measures are given in section 2.3.2 along with the underlying context.

Faithfulness measures *Faithfulness* is an essential property of explanation models. It guarantees that the explanation is fit to represent the underlying model and, as a consequence, can be trusted.

In the case of surrogate models for explanation such as LIME (Ribeiro et al., 2016), fidelity is the objective function to be optimized. In this solution, the local fidelity is the mean square error between the explanandum and the surrogate model. This error is computed on a local distribution function and weighted by a distance function. More details are given in section 2.3.2.

In the case of rule-based explanations we also find an quantitative measure of Fidelity. **Precision** (Ribeiro et al., 2018) is a measure of the degree of correctness of a rule-based explanation. It is the probability that the rule correctly represent the model’s behavior on a set of input data. A rule that never agree with the model’s prediction should have

precision of zero. In the bank loan example, suppose a rule states that *if* the customer has salary lower to 20k per year *then* the loan is accepted. This rule has a low precision. Indeed, only rare cases such as high capital gain, could make the rule work.

It is important to note that the set of input has to be well defined to assess of the fidelity is global or local. In Ribeiro et al.’s work, the fidelity measure is always local. For LIME, the weighting distance function serves this purpose and in Anchors, the precision is calculated on the vicinity distribution of the explained instance.

Consistency and Stability Consistency or stability are blurry terms. They are often used interchangeably or to define different notions. According to (Molnar, 2022), *consistency* means to express how explanations for the same instance change for different models that have been trained on the same task. It can also mean how similar instances should be explained similarly. The consensus for this definition is the notion of robustness. A widely means used to measure this metric is the *Lipschitz continuity* (Definition 1). This is the case for (Agarwal et al., 2021), an analysis of the robustness of two explanation methods, including LIME.

Definition 1 A function $h, \mathbb{R}^{d_1} \mapsto \mathbb{R}^{d_2}$ for $d_1, d_2 \in \mathbb{N}$ is *L-Lipschitz* if there exists a universal constant $L \in \mathbb{R}^{>0}$, such that $\|h(x) - h(x')\|^2 \leq L\|x - x'\|^2, \forall x, x' \in \mathbb{R}^{d_1}$.

Finally there is also another definition: how the same explanation function can deliver different explanations on the same instance. This is the case with LIME (Ribeiro et al., 2016) which is known to deliver different explanations for the same instance. This issue was tackled in (Shankaranarayana and Runje, 2019) and (Zafar and Khan, 2019). In the former study, the stability is measured with the *Jaccard coefficient*. This coefficient Equation 2.5 measures the difference of two sets of data. In their case, they are sets of explanations given by the explainer.

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|} \quad (2.5)$$

In the latter, the authors use the standard deviation of the weights of the surrogate model across multiple models generated by LIME.

Complexity The complexity of an explanation depends on the form of the explanation. There is not a single metric to cover all of them. The metrics are often straightforward, such as the length of textual explanations or the number of features included. In (Ribeiro et al., 2016), surrogate models are used to approximate the underlying model. Thus, they define a *complexity* term Ω in the objective function to be minimized. Some examples are given for decision tree: the depth of the tree or linear models: the number of non zero features.

2.5 Conclusion

In this state of the art, we focused on all possible aspects of explainability that relate to classifiers and black-box approaches. We saw that explanation methods are abundant and study all aspects of ML models. The amount of studies is also an impediment to the development of unified evaluation frameworks for ML models, and for the explainers themselves. A popular approach for explaining ML models is the generation of local explanations and considering the model as a black-box. In our research, we focus on these assumptions to build a framework that generates *abductive* explanations using an arbitrary distribution. Additionally, we observed that explanation methods can be evaluated on *simulatability* and *simplicity* in human centered evaluations like in (Ribeiro et al., 2016; Hase and Bansal, 2020), compared to each other (Gramegna and Giudici, 2021; Cian et al., 2020), but that the formal explanation method evaluation studies are scarce. This is why we put an emphasis on providing formal guarantees with our explanations.

We argue that using formal methods and deliver exact explanations is more important than delivering simple statistical based explanations whose fidelity can be questioned. A solution to tackle the issue of reasoning with incomplete information is Argumentation. Argumentation provides a framework to evaluate objects called argument using a formally defined structure between those arguments. The takeaways could be strong guarantees on well defined explanations using incomplete information. Incomplete information is the base problem that ML tries to solve by learning to predict unseen information. Argumentation proposes several solution to tackle this issue that depend on the use cases, leaving room for adaptation to different task, data, contexts etc.

Chapter 3

Argumentation

In this chapter, we present argumentation. This presentation serves two purposes: to establish the state of the art of argumentation and to give the reader an idea of why argumentation can deliver interesting results in explaining AI models. In the previous chapter, we observed that the high dimensional scale of ML models was a major issue for building interpretable surrogate models. As our knowledge of models' predictions on the whole input space is incomplete, argumentation presents itself as a formal alternative to decide which explanations to accept (Bench-Capon and Dunne, 2007).

3.1 What is argumentation?

Argumentation is a natural cognitive process we, as humans, use in our communications, oral or written. Indeed, we use argumentation in order to support an idea, an action or a decision. We regularly engage in argumentative practices when we react to arguments put forward by others with the objective of influencing their point of view.

Before defining what argumentation is, let us start by presenting a short dialogue originally proposed in (Sartor, 1994) between a journalist, John and Mary.

Example 2 (Argumentation in a dialogue between Mary and John)

Mary: Newspapers have no right to publish information I .

John: Why?

Mary: Because it is about X 's private life. (α_1)

John: Information I is not private because X is a minister and any information concerning ministers is public. (α_2)

Mary: But X is no longer a minister since he resigned last month. (α_3)

John: You are right.

In this simple example showcasing a specimen of argumentation, Mary questions John's decision to "publish information I " by putting forward the two arguments α_1 and α_3 while John justifies his *disagreement* with Mary by advancing the argument α_2 .

The example also highlights a relationship between the arguments that seem to be attacking each others: α_1 is attacked by α_2 which, in turn, is attacked by α_3 .

In this example argumentation is presented as a reasoning process in which arguments function is increase or decrease the acceptability of a given point of view. Mary advances an affirmation. John asks the question why so Mary can explain her standpoint. In the following statements, Mary and John build arguments to challenge the acceptability of the original statement. At the end, John understands and accepts Mary's statement.

Let us recall below a definition of argumentation given in (van Eemeren et al., 1996).

Argumentation is a verbal and social activity of reason aimed at increasing (or decreasing) the acceptability of a controversial standpoint for the listener or reader, by putting forward a constellation of propositions intended to justify (or refute) the standpoint before a rational judge.

Let us analyze the above definition. Argumentation is a *verbal activity*, which is normally conducted in an ordinary language (such as French). A speaker or a writer, engaged in argumentation, uses words and sentences to state, to justify or to deny something.

Argumentation is a *social activity* since it is directed at other people. Of course, the social nature of argumentation is most clearly evident in a discourse between two or more interlocutors. All the same, even when people are conferring with themselves, contemplating the pros and cons of their own ideas, their conduct is basically social.

Argumentation is an *activity of reason* since it indicates that the arguer has some thoughts about the subject. Putting forward an argument means that the arguer attempts at showing that a "rational" account can be given of his or her position on the matter. In the short dialogue above, Mary has a reason behind her claim.

Argumentation always relates to a particular *opinion*, or *standpoint*, about a specific subject. The nature of the standpoint can vary. It may be propositional standpoints, i.e. things that are believed or known, like the case of Mary about publishing information I . A standpoint can also be an action to try to perform, a goal to try to achieve, etc.

The need for argumentation arises when opinions concerning the standpoint differ. By itself, holding a standpoint is not enough to initiate argumentation. Arguing makes sense only if there is a listener or reader who entertains doubts about an opinion or has a diverging opinion. Argumentation starts from the presumption, right or wrong, that the standpoint of the arguer is not immediately accepted, but controversial. In the previous dialogue, argumentation starts when John disagrees with Mary’s claim.

The last issue in the definition of (van Eemeren et al., 1996) concerns the goal of argumentation. Argumentation is intended to justify one’s standpoint, or to refute someone else’s. In an argumentative justification of a standpoint one is attempting to defend the standpoint by showing that it conveys an acceptable proposition; in an argumentative refutation one attacks the standpoint by showing that the proposition is unacceptable whereas the opposite, or contradictory, proposition is acceptable. Justifying or refuting a standpoint by way of argumentation, as in advancing standpoints, proceeds by putting forward respectively arguments pros (i.e. arguments in favor of the standpoint) and arguments cons (i.e. arguments against the standpoint).

3.2 Argumentation process

Whatever the problem to solve is (e.g. decision making or object classification), argumentation is seen as a three-steps process:

1. Constructing arguments in *favor/against* statements
2. Evaluating the acceptability of the arguments
3. Concluding

Generally speaking, an argument gives a reason for believing a statement, or choosing an action. It has three main components: i) a *support* which is a set of premises, ii) a *conclusion*, and iii) a *link* between the support and the conclusion.

In the dialogue between Mary and John Example 2, three arguments α_1, α_2 and α_3 have been uttered. For instance, the support of α_1 is “Information I is about X ’s private life” while its conclusion is “Newspapers have no right to publish information I ”. It is worth mentioning that an argument is not a proof meaning as it does not guarantee the “validity” of its conclusion. We can make a parallel observation with model-agnostic explanation functions. We saw that explanations given by LIME or Anchors had a limited validity. The rules given as explanation have similarities with arguments: rule has a set premises (or predicates) and a conclusion (the prediction). As we will show in the subsequent section, the conclusions of α_1 and α_3 are valid at the end of the dialogue while

the conclusion of α_2 is not. We would like to leverage the same process to evaluate the validity of explanations of ML models.

The second basic component of an argument framework is the notion of attack. The idea is that arguments may be conflicting. In the above dialogue, for instance, it is clear that argument α_2 attacks α_1 and that α_3 attacks α_1 . It has been acknowledged in the literature that an argument can be attacked by another argument for three main reasons:

Conclusion-Conclusion: The idea here is that the two arguments have contradictory conclusions. This is exactly what happens between the two arguments β_1 and β_2 presented respectively by Tisias and Corax. According to the legend, Tisias was a student of Corax. Corax would have agreed to teach his argumentation techniques to Tisias, and to be paid according to the result of Tisias's first lawsuit. If he won, the deal stated that he should pay Corax for the lectures, otherwise, he would not pay him. At the end of his studies, Tisias sued his Master Corax. The idea he supported was he should not pay Corax. Tisias presented the following argument in favor of not paying Corax:

β_1 : Tisias should not pay Corax since there are two situations: i) If Tisias wins, then according to the judges he will not pay Corax. ii) If he loses, then according to the deal he made with Corax, he will not pay as well.

β_2 : Tisias should pay Corax since: i) If Tisias wins, then according to his deal with Corax, he should pay him. ii) If he loses, then according to the judges, he should pay Corax as well.

The conclusion of β_1 is "Tisias should not pay Corax" while the conclusion of β_2 is "Tisias should pay Corax".

Conclusion-Support: An argument may also attack another argument if its conclusion undermines a premise of the second argument. In the dialogue between Mary and John, the conclusion of α_2 conflicts with a premise used in the support of α_1 .

Conclusion-Link: An argument may challenge the link or the connection between the premises and the conclusion of another argument. Let us consider the following example borrowed from (Chesnevar and Simari, 2005).

δ_1 : Tweety flies because all the birds I have seen fly.

δ_2 : I have seen Opus, it is a bird and it does not fly.

The conclusion of the argument δ_2 contradicts the link between the support of δ_1 , i.e. "all the birds I have seen fly" and its conclusion "Tweety flies".

Since arguments are conflicting, it is important to know which ones to rely on for inferring conclusions or choosing actions. This amounts to evaluate the quality of the different arguments. For instance, in the dialogue between Mary and John, one expects to know at the end of the dialogue whether newspapers have or not right to publish information I . For that purpose, the three arguments α_1 , α_2 and α_3 should be evaluated.

For that purpose, a plethora of methods, called semantics have been proposed in the literature. They are roughly classified into three families: extension semantics, gradual semantics and ranking semantics. Initiated in (Dung, 1995), extension semantics look for sets of arguments (called extensions) that can be jointly accepted. Then, a dialectical status is assigned to each argument according to their membership in the identified extensions. Introduced in (Cayrol and Lagasque-Schiex, 2005), gradual semantics focus on individual arguments, and ascribe to each of them a value taken from an ordered scale representing its strength. The third family, ranking semantics, has been proposed in (Amgoud and Ben-Naim, 2013). Its semantics rank arguments from the strongest to the weakest ones. In this thesis, we focus only on extension semantics.

In Dung’s argumentation framework, arguments are abstract and considered as atomic. This setting allows the analysis to focus on the relationships of arguments with other arguments and avoids lingering over argument quality or analysis. For XAI, it is a good assumption to make explanations abstract and apply the framework to a wider range of applications. For example, the abstraction of arguments fits the idea of model agnostic explainer section 2.3 because it does not require specific formats for the arguments. However, the reader can report to (Besnard and Hunter, 2008b) for an introduction on logical argumentation. Logical argumentation is a sub-field of argumentation in which arguments are defined with classical logic. It is one way to define the arguments and identify the attacks and supports (steps 1 and 2) between them. We will see in chapter 4 how we manage to define arguments to serve our purposes.

When arguments and attack relations are defined, the structure of the arguments themselves is no longer needed to find which arguments are the strongest or which ones are acceptable. The rest of the process completely disregards the structure of arguments and attacks.

The argumentation framework is at the center of the argumentation process. It is composed of a set of arguments and a set of binary relations between these arguments. Since the arguments are abstract objects, we do not use their structure or content to evaluate. They are represented by nodes in the graph. The binary relations are represented as the edges between nodes. They can be directed or not directed depending on the form of attacks. Finally, we suppose that any graph topology can exist. Given a graph, the analysis of the arguments is done with methods called semantics.

3.3 Abstract argumentation framework

An argumentation framework is a set of arguments and a binary relation encoding attacks among those arguments.

Definition 2 (Argumentation framework) *An argumentation framework is a pair $\text{AF} = (\mathcal{A}, \mathcal{R})$ where \mathcal{A} is a set of arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is an attack relation. For $\alpha, \beta \in \mathcal{A}$, $(\alpha, \beta) \in \mathcal{R}$ means that the argument α attacks the argument β .*

It is worth mentioning that in (Dung, 1995), no indication is given about \mathcal{A} and \mathcal{R} , neither on their origin nor on how they should be elicited. Moreover, the framework completely abstracts from the application to which it can be applied.

Each argumentation framework can be represented by a directed graph whose nodes are the arguments of \mathcal{A} and arcs are the different attacks of \mathcal{R} .

Definition 3 (Graph of an argumentation framework) *The graph associated with an argumentation framework $\text{AF} = (\mathcal{A}, \mathcal{R})$ is $\mathcal{G}_{\text{AF}=(\mathcal{A},\mathcal{R})} = (\mathbf{V}, \mathcal{X})$, where $\mathbf{V} = \mathcal{A}$ and $\mathcal{X} = \mathcal{R}$.*

Let us consider the argumentation framework associated to the dialogue between Mary and John.

Example 3 (Dialogue between Mary and John) *The argumentation framework associated to the dialogue between Mary and John is the pair $\text{AF}_1 = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A} = \{\alpha_1, \alpha_2, \alpha_3\}$ and $\mathcal{R} = \{(\alpha_2, \alpha_1), (\alpha_3, \alpha_2)\}$. The graph associated with this framework is depicted in the figure below.*

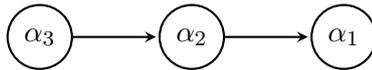


Figure 3.1: Graph of AF_1

Let us now define the argumentation framework that captures the dialogue between Tisias and his master Corax.

Example 4 (Dialogue between Tisias and Corax) *The argumentation framework associated to the dialogue between Tisias and Corax is the pair $\text{AF}_2 = (\mathcal{A}, \mathcal{R})$ where $\mathcal{A} = \{\beta_1, \beta_2\}$ and $\mathcal{R} = \{(\beta_1, \beta_2), (\beta_2, \beta_1)\}$. The graph associated with this framework is depicted in the figure below.*

Let us now recall the key notions of (indirect) attack and defence as proposed by (Dung, 1995).

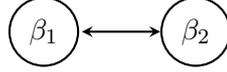


Figure 3.2: Graph of AF_2

Definition 4 (Indirect attack - Indirect defense - Strict defense) Let $AF = (\mathcal{A}, \mathcal{R})$ be an argumentation framework, $\alpha, \beta \in \mathcal{A}$, and $\mathcal{B} \subseteq \mathcal{A}$.

- β indirectly attacks α iff there exists a finite sequence of arguments $\alpha_0, \dots, \alpha_{2n+1}$ such that:
 - $\alpha = \alpha_0$ and $\beta = \alpha_{2n+1}$
 - $\forall i, 0 \leq i \leq 2n, (\alpha_{i+1}, \alpha_i) \in \mathcal{R}$.
- β indirectly defends α against the argument α iff there exists a finite sequence of arguments $\alpha_0, \dots, \alpha_{2n}$ such that:
 - $\alpha = \alpha_0$ and $\beta = \alpha_{2n}$
 - $\forall i, 0 \leq i < 2n, (\alpha_{i+1}, \alpha_i) \in \mathcal{R}$.

In (Dung, 1995), different acceptability semantics have been proposed for evaluating interacting arguments. The basic idea behind these semantics is the following: for a rational agent, an argument is acceptable if he can defend this argument against all attacks on it. All the arguments acceptable for a rational agent will be gathered in a so-called *extension*. An extension must satisfy a *consistency* requirement and must *defend* all its elements.

Definition 5 (Conflict-free, Defence) Let $AF = (\mathcal{A}, \mathcal{R})$ and $\mathcal{B} \subseteq \mathcal{A}$.

- \mathcal{B} is conflict-free iff $\nexists \alpha, \beta \in \mathcal{B}$ such that $(\alpha, \beta) \in \mathcal{R}$.
- \mathcal{B} defends an argument α iff $\forall \beta \in \mathcal{A}$, if $(\beta, \alpha) \in \mathcal{R}$, then $\exists \delta \in \mathcal{B}$ such that $(\delta, \beta) \in \mathcal{R}$.
- \mathcal{B} strictly defends α iff $\forall \beta \in \mathcal{A}$ such that $(\beta, \alpha) \in \mathcal{R}$, $\exists \delta \in \mathcal{B}$ such that $(\delta, \beta) \in \mathcal{R}$ and $(\beta, \delta) \notin \mathcal{R}$.

The fundamental semantics in (Dung, 1995) is the one that features admissible extensions. The other semantics, like preferred and stable, are based on it.

Definition 6 (Acceptability semantics (Dung, 1995)) Let $AF = (\mathcal{A}, \mathcal{R})$ and \mathcal{B} be a conflict-free set of arguments, and let $\mathcal{F}: 2^{\mathcal{A}} \mapsto 2^{\mathcal{A}}$ be a function such that $\mathcal{F}(\mathcal{B}) = \{\alpha \mid \mathcal{B} \text{ defends } \alpha\}$.

- \mathcal{B} is a naive extension iff it is maximal (wrt set- \subseteq).

- \mathcal{B} is an admissible extension iff $\mathcal{B} \subseteq \mathcal{F}(\mathcal{B})$.
- \mathcal{B} is a complete extension iff $\mathcal{B} = \mathcal{F}(\mathcal{B})$.
- \mathcal{B} is a grounded extension iff it is the minimal (wrt set- \subseteq) complete extension.
- \mathcal{B} is a preferred extension iff it is a maximal (wrt set- \subseteq) complete extension.
- \mathcal{B} is a stable extension iff it is a preferred extension that attacks any argument in $\mathcal{A} \setminus \mathcal{B}$.

Let $\sigma_x(\mathbf{AF})$ be the set of all extensions of \mathbf{AF} under the semantics x where $x \in \{a, c, g, p, s\}$. Notations a, c, g, p, s respectively stand for ‘admissible’, ‘complete’, ‘grounded’, ‘preferred’ and ‘stable’.

Let us illustrate the previous definition on the two argumentation frameworks \mathbf{AF}_1 and \mathbf{AF}_2 .

Example 5 (Examples 3 and 4 cont.) *The argumentation framework \mathbf{AF}_1 has three admissible extensions: $\mathcal{E}_1 = \emptyset$, $\mathcal{E}_2 = \{\alpha_3\}$ and $\mathcal{E}_3 = \{\alpha_1, \alpha_3\}$. Note that the set $\{\alpha_2\}$ is conflict-free but is not an admissible extension since it does not defend the argument α_2 against α_3 . It can also be checked that the set \mathcal{E}_2 is not a complete extension while \mathcal{E}_3 is. The set \mathcal{E}_3 is the only preferred extension of the framework \mathbf{AF}_1 . It is also the grounded extension and the only stable extension of that framework.*

The argumentation framework \mathbf{AF}_2 has three admissible extensions: $\mathcal{E}_1 = \emptyset$, $\mathcal{E}_2 = \{\beta_1\}$ and $\mathcal{E}_3 = \{\beta_2\}$. The sets \mathcal{E}_2 and \mathcal{E}_3 are both preferred and stable. Note that \mathcal{E}_1 is the grounded extension of \mathbf{AF}_2 .

The following property summarizes the properties of the previous acceptability semantics for a given argumentation framework.

Property 1 *Dung (1995)*

Let $\mathbf{AF} = (\mathcal{A}, \mathcal{R})$ be an argumentation framework.

- Each admissible extension is included in a preferred extension.
- The grounded extension is included in each preferred extension.
- Each stable extension is a preferred one, but the reverse is not true.
- The framework $\mathbf{AF} = (\mathcal{A}, \mathcal{R})$ has at least one preferred extension, always a unique grounded extension, and maybe zero or several stable extensions.
- When \mathcal{R} is finite, the grounded extension is exactly the set $\bigcup \mathcal{F}^{i \geq 1}(\emptyset)$.

In (Amgoud and Cayrol, 2002b), Amgoud and Cayrol have shown that the grounded extension of a finite argumentation framework contains all the arguments that are not attacked, and also the arguments which are defended directly or indirectly by non-attacked arguments. Thus when \mathcal{R} is finite, the grounded extension is defined as follows: $\bigcup \mathcal{F}^{i \geq 1}(\emptyset) = \mathcal{C}_{\mathcal{R}} \cup [\bigcup \mathcal{F}^{i \geq 1}(\mathcal{C}_{\mathcal{R}})]$ where $\mathcal{C}_{\mathcal{R}} = \{\alpha \in \mathcal{A} \mid \nexists \beta \in \mathcal{A} \text{ s.t. } (\beta, \alpha) \in \mathcal{R}\}$. Moreover, the grounded extension strictly defends all its elements.

Proposition 1 *Amgoud and Cayrol (2002b)*

Let $\text{AF} = (\mathcal{A}, \mathcal{R})$ be an argumentation framework with \mathcal{R} being finite. Let GE be the grounded extension of AF .

- *For all $\alpha \in \text{GE}$, GE strictly defends α .*
- *For all $\alpha \in \text{GE}$, α is indirectly defended by arguments of $\mathcal{C}_{\mathcal{R}}$ against all its attackers.*

Note that an argument that is indirectly defended against all its attackers by arguments of $\mathcal{C}_{\mathcal{R}}$ is not necessarily acceptable (i.e. does not necessarily belong to the grounded extension of the argumentation framework). Let us consider the following example:

Example 6 *Let $\text{AF}_3 = (\mathcal{A}, \mathcal{R})$ be an argumentation framework such that $\mathcal{A} = \{\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$, and \mathcal{R} is as depicted in the figure below.*

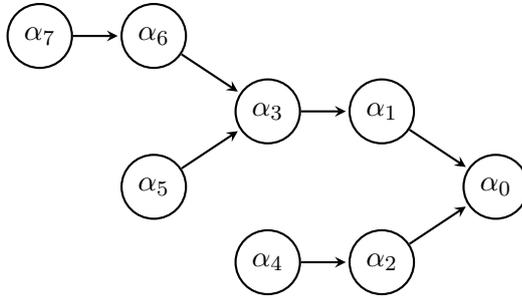


Figure 3.3: Graph of AF_3

It can be checked that $\mathcal{C}_{\mathcal{R}} = \{\alpha_4, \alpha_5, \alpha_7\}$ and that the grounded extension of AF_3 is $\text{GE} = \{\alpha_1, \alpha_4, \alpha_5, \alpha_7\}$. Note that the argument α_0 is attacked by two arguments α_1 and α_2 . It is indirectly defended by α_7 against α_1 , and by α_4 against α_2 . The two arguments α_4 and α_7 are both in $\mathcal{C}_{\mathcal{R}}$, while α_0 is not in the set GE because it is indirectly attacked by α_5 which is in $\mathcal{C}_{\mathcal{R}}$.

Stable semantics has been defined in (Dung, 1995) for capturing some results in nonmonotonic logics literature (e.g. (Gelfond and Lifschitz, 1990; Reiter, 1980)). The idea behind this semantics is that a set of arguments is “acceptable” if it attacks any argument

that is outside the set. This condition makes stable semantics too demanding and the existence of stable extensions not guaranteed for every argumentation framework. Indeed, it may be the case that an argumentation framework has no stable extensions as shown in the following example.

Example 7 Let $\text{AF}_4 = (\mathcal{A}, \mathcal{R})$ be an argumentation framework where $\mathcal{A} = \{\beta_1, \beta_2, \beta_3\}$. Let also the attack relation be as depicted in the figure below.

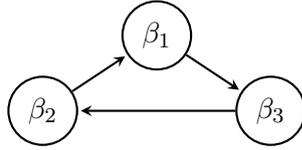


Figure 3.4: Graph of AF_4

The argumentation framework AF_4 has no stable extension while it has a preferred extension which is the empty set.

Preferred semantics has been introduced in order to palliate the limits of stable one. As stated in Property 1, each argumentation framework has preferred extensions. The latter are maximal sets of arguments that can defend themselves against any attack. The following example shows that an argumentation framework may have preferred extensions that are not stable.

Example 8 Let $\text{AF}_5 = (\mathcal{A}, \mathcal{R})$ be an argumentation framework where $\mathcal{A} = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7\}$ and the attack relation \mathcal{R} is as depicted in the figure below.

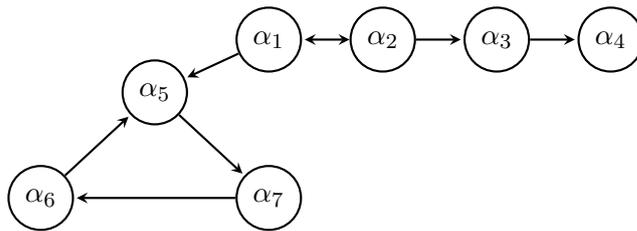


Figure 3.5: Graph of AF_5

The framework AF_5 has two preferred extensions: $\mathcal{E}_1 = \{\alpha_1, \alpha_3, \alpha_7\}$ and $\mathcal{E}_2 = \{\alpha_2, \alpha_4\}$. It can be checked that the set \mathcal{E}_1 is a stable extension while \mathcal{E}_2 is not stable since it does not attack the arguments α_5, α_6 and α_7 .

When the preferred and the stable extensions of an argumentation framework coincide, that framework is said to be *coherent*.

Definition 7 (Argumentation Framework coherence) *Let $\text{AF} = (\mathcal{A}, \mathcal{R})$, AF is coherent iff $\sigma_p(\text{AF}) = \sigma_s(\text{AF})$.*

The absence of stable extensions in a framework is mainly due to the existence of odd-length cycles in the graph associated with that framework. The following result summarizes the main properties of the different acceptability semantics depending on the existence of cycles in the graph of an argumentation framework.

Property 2 *Dung (1995); Dunne and Bench-Capon (2002)*

Let $\mathcal{G}_{\text{AF}=(\mathcal{A},\mathcal{R})}$ be the graph associated with the argumentation framework $\text{AF} = (\mathcal{A}, \mathcal{R})$.

- *If \mathcal{G}_{AF} has no cycles, then AF has a unique preferred extension which is also the grounded extension and the unique stable extension of AF .*
- *If \mathcal{G}_{AF} has no odd-length cycles, then AF is coherent.*
- *If \mathcal{G}_{AF} has no even-length cycles, then AF has a unique preferred extension.*
- *If $\text{AF} = (\mathcal{A}, \mathcal{R})$ has no stable extension, then \mathcal{G}_{AF} has at least one odd-length cycle.*
- *If the empty set is the unique preferred extension of AF , then \mathcal{G}_{AF} has at least one odd-length cycle.*

Dung’s acceptability semantics handle differently odd-length cycles and even-length ones. Let us consider again the argumentation framework AF_2 of Example 4. We have shown that this framework has two extensions which are both preferred and stable. If that framework is extended with a new argument, say β_3 as in Example 7, the new framework AF_4 has no stable extensions. In (Baroni et al., 2005), the authors claim that both cycles should be handled in a similar way. Consequently, six new acceptability semantics have been proposed. Like Dung’s ones, they return conflict-free subsets of arguments, but not necessarily maximal (for set inclusion) ones. In (Caminada, 2006) a weaker version of stable semantics, called *semi-stable*, has been proposed. It has been shown that any stable extension is a semi-stable one, and any semi-stable extension is a preferred one. The very last acceptability semantics that has been proposed in the literature is the so-called *ideal* semantics (Dung et al., 2007a). This semantics computes only one extension which is an admissible extension that is contained in every preferred extension.

The different acceptability semantics define exactly which arguments may hold together. The *status* of a given argument is defined as follows:

Definition 8 (Argument status) Let $AF = (\mathcal{A}, \mathcal{R})$ be an argumentation framework and $\sigma_x(AS)$ its set of extensions under a given semantics x . Let $\alpha \in \mathcal{A}$.

1. α is skeptically accepted iff $\alpha \in \bigcap_{\mathcal{E}_i \in \sigma_x(AF)} \mathcal{E}_i$
2. α is credulously accepted iff $\alpha \in \bigcup_{\mathcal{E}_i \in \sigma_x(AF)} \mathcal{E}_i$.
3. α is rejected iff $\alpha \notin \bigcup_{\mathcal{E}_i \in \sigma_x(AF)} \mathcal{E}_i$.

Example 9 (Examples 3 and 4 cont.) It is easy to check that the two arguments, α_1 and α_3 , presented by Mary are both skeptically accepted under preferred, stable and grounded semantics while the argument α_2 uttered by John is rejected. Regarding the two arguments β_1 and β_2 uttered by Tisias and Corax, they are both credulously accepted under preferred and stable semantics, and are both rejected under grounded semantics.

Since the empty set is an admissible extension of any argumentation framework, it is natural to not consider admissible semantics for computing the status of arguments. Otherwise, all the arguments would be either rejected or credulously accepted.

An important question is whether to consider skeptical acceptance of arguments or credulous one. The choice depends broadly on whether the argumentation framework at hand is used for *theoretical* reasoning or for *practical* reasoning purposes. Theoretical reasoning is concerned with deciding what to believe, while practical reasoning is concerned with deciding what to do. In (Harman, 2004), the philosopher Gilbert Harman distinguishes two main differences between the two kinds of reasoning: i) wishful thinking, and ii) reasonableness of arbitrary choices. In order to better explain the two issues, Gilbert Harman has presented the following example about Albert who thinks about which route to take:

Albert thinks about what route to take to get to Boston. He thinks that, while the direct western route is faster, the scenic eastern route is longer but more enjoyable with less traffic.

It is clear that the reasoning Albert goes through in settling on what route to take is practical since he is deciding what to do. Let us now consider the case of Albert's friend Betty.

Betty tries to decide what route Albert will take. She thinks about what Albert has done before, what Albert likes in a route, and how much of a hurry Albert is in.

Betty's reasoning is theoretical. She is trying to arrive at a belief about what Albert will do.

The first important difference between theoretical and practical reasoning has to do with wishful thinking, which is acceptable in practical reasoning and not in theoretical reasoning. Albert's preference for the eastern route can give him a practical reason to take the eastern route rather than the western route. But Betty's preference for Albert to be taking the eastern route does not in the same way give her a theoretical reason to believe that he is taking the eastern route. In an argumentation context, this means that in practical reasoning, one may prefer a credulously accepted argument to another while this is not allowed in theoretical reasoning.

The second important difference between theoretical and practical reasoning has to do with the reasonableness of arbitrary choices. Suppose Albert is trying to decide whether to take the eastern route or the western route and he finds that nothing favors one route over the other. This means that each option is supported by a credulously accepted argument, and both arguments are not skeptically accepted. Then it is reasonable for him to decide arbitrarily to take one of the two routes. If it is urgent for him to get to Boston, it would be a mistake for him to suspend judgment in this case. On the other hand, if Betty is trying to decide which route Albert is taking and there is no particular reason to think he is going one way rather than the other, meaning that she has two credulously accepted arguments and no skeptically accepted one, then it is not reasonable for her to decide arbitrarily that he is taking one route rather than the other. In the theoretical case, Betty should suspend judgment. In sum, skeptical acceptance is used for theoretical reasoning purposes while credulous acceptance is required for practical reasoning.

Computational considerations

In the literature, extension semantics have been used in most argumentation-based systems dealing with defeasible information. They provide very promising results as they cover most non-monotonic reasoning approaches. However these semantics suffer from a serious problem which is computational cost. Indeed, the problems of finding and enumerating extensions are computationally challenging as summarized in Table 3.1.

In addition to this study, (Kröll et al., 2017) studies the complexity and tractability of the problem of enumerating all extensions. They study the cases of several semantics including *naive* and *stable* semantics. They find that there exists an enumeration algorithm that output each extension with polynomial delay and polynomial space.

As a consequence of the above complexity results, several solvers have been developed in the literature for enumerating extensions in an efficient way and checking the status of arguments.

In 2015, the argumentation community organized the first competition to reward the best solvers for argumentation tasks on big and complex datasets. The first edition took place with the workshop Theory and Application of Formal Argument (TAFA'15) as a joint

Problem	Question	Complexity
ADM(AF, S)	Is S admissible?	P
STAB-EXT(AF, S)	Is S a <i>stable</i> extension?	P
PREF-EXT(AF, S)	Is S a <i>preferred</i> extension?	CO-NP-complete
HAS-STAB(AF)	Does AF has any stable extension?	NP-complete
CA(AF, S)	Is x in some <i>preferred</i> extension?	NP-complete
IN-STAB(AF, S)	Is x in some <i>stable</i> extension?	NP-complete
ALL-STAB(AF, S)	Is x in <i>every</i> stable extensions?	CO-NP-complete
SA(AF, S)	Is x in <i>every</i> preferred extension?	$\prod_2^{(p)}$ -complete
COHERENT(AF)	Is the system AF coherent?	$\prod_2^{(p)}$ -complete

Table 3.1: Complexity of decision problems in finite argumentation systems (Dunne, 2007)

event with the 24th International Joint Conference on Artificial Intelligence (IJCAI'15). Currently, the competition takes place every two years. In every edition, there is a list of tasks that the competitors have to tackle. The Competition is organised into 7 tracks, one for each semantics σ :

- Complete Semantics
- Preferred Semantics
- Stable Semantics
- Semi-stable Semantics (Caminada et al., 2012)
- Stage Semantics (Verheij, 1996)
- Grounded Semantics
- Ideal Semantics (Dung et al., 2007b)

Each track includes these four reasoning problems:

- (SE- σ) Given an abstract argumentation framework, determine some extension
- (EE- σ) Given an abstract argumentation framework, determine all extensions
- (DC- σ) Given an abstract argumentation framework and some argument, decide whether the given argument is credulously inferred
- (DS- σ) Given an abstract argumentation framework and some argument, decide whether the given argument is skeptically inferred

Semi-stable, stage and ideal semantics have been added in 2017. The static (main) track remains the same every two years since 2015. However, a Dynamic Track has opened

whose goal is to reach the new solution for a slightly changing argumentation framework without computing the whole set of extensions model from scratch. Here is a list of notable contenders of each edition:

- CoQuiAAS (Lagniez et al., 2015)
- ASPARTIX (Dvořák et al., 2020)
- CEGARTIX (Dvořák et al., 2014)
- ArgSemSAT (Cerutti et al., 2014)
- Pyglaf (Alviano, 2021)

CoQuiAAS, CEGARTIX and ArgSemSAT have participated in the three first edition in all tracks and got ranked in the top 5 in almost all tracks. ASPARTIX, since 2015 and Pyglaf since 2017, have participated in all editions. ASPARTIX have shown the best performance in the EE-Stable tracks in 2015 and 2017. They scored 2nd place in 2019. In this thesis, we use ASPARTIX for enumerating stable extensions as we will see in chapter 5.

3.4 Argumentation in Artificial Intelligence and XAI

Argumentation has become an Artificial Intelligence keyword for the last decades. In its essence, argumentation can be seen as a particularly useful and intuitive paradigm for doing non-monotonic reasoning. The advantage of argumentation is that the reasoning process is composed of modular and quite intuitive steps, and thus avoids the monolithic approach of many traditional logics for defeasible reasoning.

Another interesting property of the argumentation approach is that it can be given dialectical proof procedures that are quite close to the process by which humans would discuss an issue. The similarity with human-style discussions gives formal argumentation an advantage that can be useful in many contexts. Argumentation techniques are used to specify *reasoning*, such as belief revision (e.g. (Rotstein et al., 2008)), handling inconsistency in knowledge bases (e.g. (Amgoud and Cayrol, 2002a; Besnard and Hunter, 2001, 2008a; Garcia and Simari, 2004; Governatori et al., 2004; Simari and Loui, 1992)), decision making under uncertainty (e.g. (Amgoud and Prade, 2006; Bonet and Geffner, 1996; Fox and McBurney, 2002; Fox and Parsons, 1997; Gordon and Karacapilidis, 1997)), merging information coming from different sources (e.g. (Amgoud and Kaci, 2007; Amgoud and Parsons, 2002; Brena et al., 2005)), practical reasoning (e.g.(Amgoud, 2003; Atkinson et al., 2004; Rahwan and Amgoud, 2006)), and goal generation (e.g. (Hulstijn and van der Torre, 2004)). Argumentation is also gaining increasing interest in multi-agent systems

research community, namely for modeling *multi-agent interaction*. Since the seminal work by Walton and Krabbe (Walton and Krabbe, 1995) on the different categories of dialogue, different argumentation-based systems have been proposed for persuasion dialogues (e.g. (Amgoud et al., 2000a; Prakken, 2006)), negotiation (e.g. (Amgoud et al., 2000b; Amgoud and Prade, 2004; Kakas and Moraitis, 2006; Kraus et al., 1998; Parsons and Jennings, 1996)), and inquiry dialogues (e.g. (Parsons et al., 2003; Black and Hunter, 2007)). More recently, argumentation is largely used in machine learning. For instance, (Amgoud and Serrurier, 2008; Cocarascu and Stylianou, 2020; Alcaraz, 2023) proposed novel classification models that are based on arguments. Their explanations are defined in dialectical ways as fictitious dialogues between a proponent (supporting an output) and an opponent (attacking the output) following (Dung, 1995). The authors in (Čyras et al., 2019a,b; Rago et al., 2018) followed the same approach for defining explainable multiple criteria of decision systems, recommendation systems or scheduling systems. In the above, papers, an argument is simply an instance and its label. In other papers including (Potyka et al., 2022; Rago et al., 2023; Proietti and Toni, 2023), argumentation is rather used for explaining machine learning models like random forest and neural networks.

Chapter 4

Theoretical Approach with argumentation

4.1 Introduction

As said in the introduction, recent advances in many AI fields rely on inductive models, depending on parameters that are adjusted based on a set of training instances. Such models tend to be large for practical tasks, in the sense of having a lot of parameters, and may allow for non-linear interactions between input features. Consequently, they are perceived as *black-boxes* whose behaviour is difficult to grasp both from their designers and users' point of view.

In this chapter, we focus on black-box classifiers and provide ways for explaining their outcomes for instances, i.e. providing local explanations.

We investigate one of the most studied types of explanation, the so-called *abductive explanations*, which highlight feature-values that are sufficient for making a given prediction. For example, a client was refused a loan *because he is unemployed*. Such explanations are generally generated from the whole feature space e.g. (Darwiche and Hirth, 2020; Ignatiev et al., 2019; Audemard et al., 2022; Amgoud, 2021a). While the approach is reasonable when models are interpretable, like Decision Trees or Random Forests, it is not tractable in case of black-boxes, see (Cooper and Marques-Silva, 2021), as it requires an exhaustive exploration of the feature space.

As a solution, the two prominent explanation functions Anchors (Ribeiro et al., 2018) and LIME (Ribeiro et al., 2016) and the argument-based function (Amgoud, 2021b) generate abductive explanations from a sample (i.e., subset) of instances, avoiding thus exploring the whole feature space. However, it has been shown in (Amgoud, 2021b; Narodytska et al., 2019b) that the explanations of Anchors/LIME may be globally inconsistent and thus incorrect. The third function ensures correct explanations but does not guarantee the existence of explanations for every instance. Furthermore, it is very cautious as it simply

removes all conflicting explanations that may be generated from the considered sample.

This chapter investigates explanation functions that generate abductive explanations from a subset of feature space while satisfying desirable properties.

Firstly, we define two principles: existence of explanations, *success*, and *coherence* between the explanations. We show that a function that generates abductive explanations from a subset of instances cannot guarantee both principles. This result sheds light on the reason behind violation of success by the argument-based function from (Amgoud, 2021b).

Secondly, we propose a parameterized family of argumentation-based explanation functions, each of which satisfies one of the two incompatible properties. To create these functions, we generate arguments in favor of classes, then identify attacks among these arguments, we enumerate sets of arguments that can be jointly accepted using the stable semantics (Dung, 1995). Finally, we can use the latter to identify *accepted arguments* and define novel types of abductive explanations. In our approach, we choose the policy of acceptance of arguments using two parameters: *selection function* and *inference rule*. The former selects a subset of stable extensions and the latter selects (accepted) arguments from the chosen extensions. We define various instantiations of the two parameters, capturing different *criteria* for solving conflicts between arguments.

Finally, we propose a formal analysis and a comprehensive comparison of our collection of functions. We show that not only the family encompasses the argument-based function and ensures correctness of explanations but also performs better as the functions explain more instances and more classes.

4.2 Classification models

We consider a classification theory as a tuple made of a finite set of *features*, a function which returns the *domain* of every feature and a finite set of *classes*. The theory represents the model's environment.

Definition 9 (Theory) *A theory is a tuple $T = \langle \mathcal{F}, \text{dom}, \mathcal{C} \rangle$ s.t.*

- \mathcal{F} is a finite set of features,
- dom is a function on \mathcal{F} such that, for every $f \in \mathcal{F}$, $\text{dom}(f)$ is countable (discrete domains),
- \mathcal{C} is a finite set of possible distinct classes with $|\mathcal{C}| > 1$.

We introduce next the useful notion of literal, which is an assignment of value to a feature in \mathcal{F} .

Definition 10 (Literal) Let $\mathbb{T} = \langle \mathcal{F}, \text{dom}, \mathbb{C} \rangle$ be a theory. A literal is a pair (f, v) where $f \in \mathcal{F}$ and $v \in \text{dom}(f)$. Let $\text{Lit}(\mathbb{T})$ denote the set of all possible literals of \mathbb{T} .

A set of literals is *consistent* if it does not contain two literals having the same attribute but distinct values.

Definition 11 (Consistency) A set $L \subseteq \text{Lit}(\mathbb{T})$ is consistent iff $\nexists (f, v), (f', v') \in L$ such that $f = f'$ and $v \neq v'$. Otherwise, L is said to be inconsistent.

We call *instance* any assignment of values to all the features. Therefore, an instance is always a consistent set of features.

Definition 12 (Instance) Let $\mathbb{T} = \langle \mathcal{F}, \text{dom}, \mathbb{C} \rangle$ be a theory. An instance is a subset I of literals such that every attribute $f \in \mathcal{F}$ appears exactly once in I . Let $\mathcal{X}_{\mathbb{T}}$ denote the set of all instances of \mathbb{T} , called *feature space*.

A classification model, or classifier, is a surjective function mapping every instance into a single prediction.

Definition 13 (Classifier) Let $\mathbb{T} = \langle \mathcal{F}, \text{dom}, \mathbb{C} \rangle$ be a theory. A classifier on \mathbb{T} is a surjective function \mathbb{R} from $\mathcal{X}_{\mathbb{T}}$ to \mathbb{C} , i.e. $\mathbb{R} : \mathcal{X}_{\mathbb{T}} \mapsto \mathbb{C}$.

To better clarify each of these five basic notions, let us consider the following example:

Example 10 Consider a theory made of two binary features f_1, f_2 and three classes c_1, c_2, c_3 . The table below summarizes the predictions made by a classifier \mathbb{R} .

$\mathcal{X}(\mathbb{T})$	f_1	f_2	$\mathbb{R}(I_i)$
I_1	0	0	c_1
I_2	0	1	c_2
I_3	1	0	c_3
I_4	1	1	c_3

In this example,

- $\mathbb{T} = \langle \mathcal{F}, \text{dom}, \mathbb{C} \rangle$ with
 - $\mathcal{F} = \{f_1, f_2\}$
 - $\text{dom}(f_1) = \text{dom}(f_2) = \{0, 1\}$
 - $\mathbb{C} = \{c_0, c_1, c_2\}$
- all the possible literals are $\{(f_1, 0), (f_1, 1), (f_2, 0), (f_2, 1)\}$
- the set $\{(f_1, 0), (f_1, 1)\}$ is not consistent whereas $\{(f_1, 0), (f_2, 1)\}$ is.
- the set $\{(f_1, 0), (f_2, 1)\}$ of literals is an instance since it covers all features.

4.3 Abductive Explanations

An explanation function answers questions of the form: “Why does the classifier \mathbf{R} assign class c to instance x ?”. The answer to this type of question is called an abductive explanation. Since they have been studied for a long time, e.g. (Dimopoulos et al., 1997; Kakas and Riguzzi, 2000), they are a good candidate to explain classifiers. (Shih et al., 2018; Ignatiev et al., 2018; Darwiche and Hirth, 2020) are examples of use of abductive explanation for explainability. In these papers, an abductive explanation is defined as a subset-minimal set of literals that is sufficient for predicting the class of an instance.

Definition 14 (Abductive Explainer) *Let \mathbf{R} be a classifier and \mathbf{T} a theory. An abductive explainer is a function \mathbf{g}_a mapping every $I \in \mathcal{X}_{\mathbf{T}}$ into the set of any L verifying the following:*

- a) $L \subseteq I$,
- b) $\forall I' \in \mathcal{X}_{\mathbf{T}} \setminus \{I\}$ such that $L \subseteq I'$, $\mathbf{R}(I') = \mathbf{R}(I)$,
- c) $\nexists L' \subset L$ such that L' satisfies the above conditions.

The set of literals L is called abductive explanation.

Every instance may have one or several abductive explanations as shown in the following example.

Example 11 *Consider a theory made of two binary features f_1, f_2 and three classes c_1, c_2, c_3 . The table below summarizes the predictions made by a classifier \mathbf{R} .*

$\mathcal{X}(\mathbf{T})$	f_1	f_2	$\mathbf{R}(I_i)$
I_1	0	0	c_1
I_2	0	1	c_2
I_3	1	0	c_3
I_4	1	1	c_3

The abductive explanations of I_1, I_2, I_3, I_4 are given below.

- $\mathbf{g}_a(I_1) = \{L_1\}$ $L_1 = \{(f_1, 0), (f_2, 0)\}$
- $\mathbf{g}_a(I_2) = \{L_2\}$ $L_2 = \{(f_1, 0), (f_2, 1)\}$
- $\mathbf{g}_a(I_3) = \mathbf{g}_a(I_4) = \{L_3\}$ $L_3 = \{(f_1, 1)\}$

It has been shown in (Cooper and Marques-Silva, 2021) that the problem of finding one abductive explanation and testing whether a set of literals is an abductive explanation are not tractable.

Property 3 (Cooper and Marques-Silva (2021)) *Testing whether a set L of literals is an abductive explanation is $CO - NP$ - complete and the complexity of finding one abductive explanation is FP^{NP} .*

The above high complexities are due to the conditions $b)$ in definition 14 on the preceding page. It states that generating an abductive explanation for the prediction of an instance requires testing a set inclusion on the whole input space. Note also that when the classifier is a black-box, this condition is not reasonable due to the obligation of querying the prediction on the huge size of the input space, and the complexity of querying black-box classifiers like deep neural networks.

4.4 Plausible abductive explanations

We have seen previously that generating an abductive explanation requires exploring the whole feature space, which is very costly in case of black-box classifiers. To solve this problem, we propose to generate explanations using a sample of the input space. The idea is to consider a subset of instances which may be chosen in different ways. It may be the dataset on which the classifier has been trained, or the dataset on which the classifier has shown the best performances, etc.. Whatever its source, the sample (i.e. dataset) should satisfy a property stating that every class in the set \mathbf{C} of the theory should be represented in the sample. This condition ensures a quite well-balanced sample. In what follows, we call explanations generated from samples, plausible abductive explanations.

Definition 15 (Plausible Explainer) *Let R be a classifier, T a theory, $\mathcal{Y} \subseteq \mathcal{X}_T$. A plausible explainer is a function g_p mapping every $I \in \mathcal{Y}$ into the set of any L verifying the following:*

- a) $L \subseteq I$,*
- b) $\forall I' \in \mathcal{Y} \setminus \{I\}$ such that $L \subseteq I'$, $R(I') = R(I)$,*
- c) $\nexists L' \subset L$ such that L' satisfies the above conditions.*

The set L is called plausible abductive explanation.

Let us illustrate the definition on an example.

Example 12 *Let us consider the theory made of four binary features and three classes. Assume a classifier R which provides the predictions below for the seven instances in \mathcal{Y} .*

\mathcal{Y}	f_0	f_1	f_2	f_3	$\mathbf{R}(I_i)$
I_1	0	0	1	0	c_1
I_2	0	0	1	1	c_1
I_3	0	1	0	0	c_0
I_4	0	1	0	1	c_2
I_5	0	1	1	1	c_1
I_6	1	1	0	1	c_2
I_7	1	1	1	0	c_2

The function \mathbf{g}_p returns the following explanations.

- $\mathbf{g}_p(I_1) = \{L_5, L_7\}$ $L_1 = \{(f_2, 0), (f_3, 0)\}$
 - $\mathbf{g}_p(I_2) = \{L_2, L_5, L_7\}$ $L_2 = \{(f_2, 1), (f_3, 1)\}$
 - $\mathbf{g}_p(I_3) = \{L_1, L_6\}$ $L_3 = \{(f_0, 1)\}$
 - $\mathbf{g}_p(I_4) = \{L_4\}$ $L_4 = \{(f_2, 0), (f_3, 1)\}$
 - $\mathbf{g}_p(I_5) = \{L_2, L_7\}$ $L_5 = \{(f_1, 0)\}$
 - $\mathbf{g}_p(I_6) = \{L_3, L_4\}$ $L_6 = \{(f_0, 0), (f_1, 1), (f_3, 0)\}$
 - $\mathbf{g}_p(I_7) = \{L_3, L_8\}$ $L_7 = \{(f_0, 0), (f_2, 1)\}$
- $L_8 = \{(f_1, 1), (f_2, 1), (f_3, 0)\}$

Unlike abductive explanations (definition 14 on page 64), plausible explanations are generated under incomplete information, that's to say under a portion of the input space only. This is why we refer to them as plausible since they may not hold under complete information as shown in the example below.

Example 12 (Cont.) Assume that the prediction of the instance $I_8 \in \mathcal{X}_T$ below is $\mathbf{R}(I_8) = c_1$.

	f_0	f_1	f_2	f_3	$\mathbf{R}(I_8)$
I_8	1	1	0	0	c_1

Note that $L_1 \in \mathbf{g}_p(I_3)$ while L_1 cannot be a subset of an abductive explanation (i.e., $L_1 \notin \mathbf{g}_a(I_3)$).

However, we show that every abductive explanation of an instance is a superset of a plausible explanation of the same instance. This shows that plausible explanations are approximations of and shorter than abductive ones.

Proposition 2 Let T be a theory and $\mathcal{Y} \subseteq \mathcal{X}_{\mathsf{T}}$. For every $I \in \mathcal{Y}$, if $L \in \mathfrak{g}_a(I)$, then $\exists L' \subseteq L$ such that $L' \in \mathfrak{g}_p(I)$.

Proof Let T be a theory, $\mathcal{Y} \subseteq \mathcal{X}_{\mathsf{T}}$ and $I \in \mathcal{X}_{\mathsf{T}}$. Let $L \in \mathfrak{g}_a(I)$. By definition, $\forall I' \in \mathcal{X}_{\mathsf{T}}$ such that $L \subseteq I'$, $\mathsf{R}(I') = \mathsf{R}(I)$. Since $\mathcal{Y} \subseteq \mathcal{X}_{\mathsf{T}}$, then $\forall I' \in \mathcal{Y}$ such that $L \subseteq I'$, $\mathsf{R}(I') = \mathsf{R}(I)$ (1). Thus, $\exists L' \subseteq L$ such that L' is a minimal (for set inclusion) set verifying (1). So, $L' \in \mathfrak{g}_p(I)$. ■

As expected, reducing the exploration space when generating abductive explanations leads to an important gain in computational complexity as shown in the following propositions.

Proposition 3 Cooper and Amgoud (2023)

- Testing whether a set of literals L is a plausible abductive explanation can be achieved in polynomial time,
- Finding a plausible abductive explanation can be achieved in polynomial time.

Indeed, the generation of a plausible explanation depends simply on the number of instances in the sample and the number of features in the theory. While this gain is very important, we show next that the plausible explainer suffers from a tricky issue.

4.5 Coherence vs Existence of explanations

In (Amgoud and Ben-Naim, 2022), the authors introduced a set of principles for explanation functions that interpret the global behaviour of a classifier, i.e., those that explain classes instead of instances. Every principle is seen as a desirable property that should be satisfied. In what follows we adapt two of them to functions explaining instances from samples.

Definition 16 Let R be a classifier, T a theory and $\mathcal{Y} \subseteq \mathcal{X}_{\mathsf{T}}$. A refined plausible explainer is a function \mathfrak{g} mapping every $I \in \mathcal{Y}$ into $\mathfrak{g}(I) \subseteq \mathfrak{g}_p(I)$.

The first principle, called *success*, states that any refined plausible explainer should return at least one explanation to every instance. It ensures feedback for end-users.

Principle 1 (Success) A refined plausible explainer \mathfrak{g} satisfies *success* iff for any classifier R , any theory T , any $\mathcal{Y} \subseteq \mathcal{X}_{\mathsf{T}}$, and any $I \in \mathcal{Y}$, we have that $\mathfrak{g}(I) \neq \emptyset$.

The second principle, called *coherence*, states that the explanations of instances labelled with different classes should be inconsistent. This property prevents the following three undesirable situations: Assume two instances $I, I' \in \mathcal{Y}$ such that $\mathsf{R}(I) \neq \mathsf{R}(I')$. Assume also that L is an explanation for I and L' is an explanation for I' . We may have the following:

- i) $L = L'$,
- ii) $L \subset L'$,
- iii) $L \not\subseteq L'$ and $L \cup L'$ is consistent.

It is clearly not reasonable to predict different classes on the basis of the same set of information i), ii). For the third case, assume L and L' stand respectively for: Age ≤ 45 , salary $\leq 50K$ and $\mathbf{R}(I)$ and $\mathbf{R}(I')$ stand for accepting and rejecting a loan respectively. The two explanations are incompatible since they both match a profile of a customer whose age is 30 and salary is 40K. The first rule states that this customer should have the loan while the second predicts rejection.

Principle 2 (Coherence) *A refined plausible explainer \mathbf{g} satisfies coherence iff for any classifier \mathbf{R} , any theory \mathbf{T} , any $I, I' \in \mathcal{Y}$, if $\mathbf{R}(I) \neq \mathbf{R}(I')$, then $\forall L \in \mathbf{g}(I), \forall L' \in \mathbf{g}(I'), L \cup L'$ is inconsistent.*

It is well-known in the literature that the function \mathbf{g}_a provides at least one explanation for each instance in the theory's feature space. From Proposition 2, it follows that the same holds for the plausible explainer \mathbf{g}_p , thus \mathbf{g}_p satisfies success.

Proposition 4 *For any theory \mathbf{T} , any $\mathcal{Y} \subseteq \mathcal{X}_{\mathbf{T}}$, any classifier \mathbf{R} , and any $I \in \mathcal{Y}$, $\mathbf{g}_p(I) \neq \emptyset$.*

Proof Let $I \in \mathcal{Y}$. From (Amgoud, 2021a), $\mathbf{g}_a(I) \neq \emptyset$. Thus, $\exists L \in \mathbf{g}_a(I)$. From Proposition 2, $\exists L' \subseteq L$ such that $L' \in \mathbf{g}_p(I)$. ■

The situation is different for the second principle. Indeed, the following example shows that the plausible explainer \mathbf{g}_p violates coherence, and may provide **erroneous** explanations.

Example 12 (Cont.) Consider the two instances I_2 and I_3 . Note that $\mathbf{R}(I_2) \neq \mathbf{R}(I_3)$ while $L_5 \in \mathbf{g}_p(I_2)$, $L_1 \in \mathbf{g}_p(I_3)$ and $L_1 \cup L_5$ is consistent. Consequently, there exists $I' \in \mathcal{X}_{\mathbf{T}}$ such that $L_1 \cup L_5 \subseteq I'$. Since I' has a single class, then at least one of the two explanations (L_1, L_5) is incorrect.

In what follows, we show that the two principles (success, coherence) are *incompatible* when explanations are generated from a dataset or more generally from a subset of instances. In other words, there is no (refined) plausible explainer that can satisfy the two principles at the same time for every classifier, every theory, and every subset of the feature space.

Theorem 1 *There is no refined plausible explainer that satisfies both coherence and success.*

Proof The two properties of Coherence and Success are *compatible* iff there exists a refined plausible explainer, say \mathbb{L} , which satisfies both properties. Recall also that \mathbb{L} satisfies Coherence (resp. Success) iff the property holds for every theory, every dataset and every classifier. To show that Coherence and Success are *not compatible*, it is sufficient to show that such a function \mathbb{L} does not exist.

Assume that \mathbb{L} is a refined plausible explainer that satisfies both Coherence and Success. Consider the theory below made of two binary features f_1, f_2 , and a binary classifier \mathbb{R} . The table below summarizes the predictions made by the classifier for the simple dataset \mathcal{Y} below made of two instances.

\mathcal{Y}	f_1	f_2	$\mathbb{R}(I_i)$
I_1	0	1	0
I_2	1	0	1
I_3	0	0	2

The function \mathbf{g}_p returns the following plausible explanations.

- $\mathbf{g}_p(I_1) = \{L_1\}$ $L_1 = \{(f_2, 1)\}$
- $\mathbf{g}_p(I_2) = \{L_2\}$ $L_2 = \{(f_1, 1)\}$
- $\mathbf{g}_p(I_3) = \{L_3\}$ $L_3 = \{(f_1, 0), (f_2, 0)\}$

Since \mathbb{L} is a refined plausible explainer, then from definition 16 on page 67 it holds that:

$$\forall i \in \{1, 2, 3\}, \quad \mathbb{L}(I_i) \subseteq \mathbf{g}_p(I_i) \quad (\text{A1}).$$

Since \mathbb{L} satisfies Success, then $\mathbb{L}(I_1) \neq \emptyset$ and $\mathbb{L}(I_2) \neq \emptyset$. Thus, $\forall i \in \{1, 2\}, \mathbb{L}(I_i) = \mathbf{g}_p(I_i)$. However, $L_1 \cup L_2$ is consistent while $\mathbb{R}(I_1) \neq \mathbb{R}(I_2)$, thus \mathbb{L} violates Coherence.

Let us now start by coherence. From coherence of \mathbb{L} , $\nexists L, L' \in \mathbb{L}(I_1) \cup \mathbb{L}(I_2)$ such that $L \cup L'$ is consistent, $L \in \mathbb{L}(I_i)$, $L' \in \mathbb{L}(I_j)$, and $\mathbb{R}(I_i) \neq \mathbb{R}(I_j)$ (A2). From (A1), $\mathbb{L}(I_1) \cup \mathbb{L}(I_2) \subseteq \mathbf{g}_p(I_1) \cup \mathbf{g}_p(I_2)$. But, $\mathbf{g}_p(I_1) \cup \mathbf{g}_p(I_2) = \{L_1, L_2\}$, then from (A2) either $L_1 \notin \mathbb{L}(I_1) \cup \mathbb{L}(I_2)$, in which case $\mathbb{L}(I_1) = \emptyset$, or $L_3 \notin \mathbb{L}(I_1) \cup \mathbb{L}(I_2)$, in which case $\mathbb{L}(I_2) = \emptyset$. Thus, \mathbb{L} violates Success. ■

To sum up, the previous result shows that generating abductive explanations from a subset of feature space is a tricky issue. A refined plausible explainer can either, like \mathbf{g}_p always guarantee explanations for every instance but they may be wrong, or provide correct explanations for only a subset of instances. The following section defines in a **unified setting** various functions for each of the two policies.

4.6 Parameterized Family of Explainers

Throughout this section we consider an arbitrary but fixed subset $\mathcal{Y} \subseteq \mathcal{X}_{\mathbf{T}}$ of instances of theory $\mathbf{T} = \langle \mathcal{F}, \text{dom}, \mathbf{C} \rangle$. We define a novel parameterized family of refined explanation functions. The family is based on argumentation theory and thus follows these steps: it starts by generating arguments from \mathcal{Y} , identifies attacks among them, uses a semantics for generating sets of arguments that can be jointly accepted, identifies accepted arguments, and uses the latter for defining novel types of abductive explanations.

In our approach, arguments support classes, in the sense that they provide minimal sets of literals that determine a class. They are thus independent from instances. An advantage of not considering instances is to reduce the number of arguments that can be built from \mathcal{Y} . Furthermore, explanations of an instance are explanations of its predicted class.

Definition 17 (Argument) *An argument built from \mathcal{Y} is a pair $\langle L, c \rangle$ such that:*

- $L \subseteq \text{Lit}(\mathbf{T})$ and $c \in \mathbf{C}$,
- $\exists I \in \mathcal{Y}$ such that $L \subseteq I$,
- $\forall I \in \mathcal{Y}$ such that $L \subseteq I$, $\mathbf{R}(I) = c$,
- $\nexists L' \subset L$ that verifies the above conditions.

L and c are called respectively support and conclusion of the argument. $\text{Arg}(\mathcal{Y})$ denotes the set of arguments built from \mathcal{Y} .

The second condition of the above definition ensures that arguments are extracted from instances of the set \mathcal{Y} . It discards any fallacious argument whose support is not included in any instance of \mathcal{Y} and thus satisfies the third condition in a vacuous way. The third condition states that the support L is correlated to the conclusion c .

Example 12 (Cont.) Eight arguments are generated from \mathcal{Y} :

- $a_1 = \langle L_1, c_0 \rangle$ $a_2 = \langle L_6, c_0 \rangle$
- $a_3 = \langle L_2, c_1 \rangle$ $a_4 = \langle L_5, c_1 \rangle$ $a_5 = \langle L_7, c_1 \rangle$
- $a_6 = \langle L_3, c_2 \rangle$ $a_7 = \langle L_4, c_2 \rangle$ $a_8 = \langle L_8, c_2 \rangle$

Notice that the support of every argument is a plausible abductive explanation of one or more instances in \mathcal{Y} . Before presenting the result, let us first introduce two notations.

Notations: Let $\mathcal{E} \subseteq \text{Arg}(\mathcal{Y})$. We denote by $\text{cov}_i(\mathcal{E})$ the set of instances covered by \mathcal{E} , ie., $\text{cov}_i(\mathcal{E}) = \{I \in \mathcal{Y} \mid \exists \langle L, c \rangle \in \mathcal{E} \text{ and } L \subseteq I\}$, and by $\text{cov}_c(\mathcal{E})$ the set of classes covered by \mathcal{E} , ie., $\text{cov}_c(\mathcal{E}) = \{c \in \mathbf{C} \mid \exists \langle L, c \rangle \in \mathcal{E}\}$.

Proposition 5 *The following properties hold.*

- For every $\langle L, c \rangle \in \text{Arg}(\mathcal{Y})$, the set L is consistent,
- $L \in \bigcup_{I \in \mathcal{Y}} \mathbf{g}_p(I)$ iff $\langle L, c \rangle \in \text{Arg}(\mathcal{Y})$,
- For every $I \in \mathcal{Y}$, $\exists \langle L, \mathbf{R}(I) \rangle \in \text{Arg}(\mathcal{Y})$ such that $L \subseteq I$,
- $\text{cov}_i(\text{Arg}(\mathcal{Y})) = \mathcal{Y}$,
- $\text{cov}_c(\text{Arg}(\mathcal{Y})) = \mathbf{C}$ iff $\{\mathbf{R}(I) \mid I \in \mathcal{Y}\} = \mathbf{C}$,
- The set $\text{Arg}(\mathcal{Y})$ is finite.

Proof Let $\mathcal{Y} \subseteq \mathcal{X}(\mathbb{T})$ and $I \in \mathcal{Y}$. Since I is consistent, then every $L \subseteq I$, L is consistent, which shows the first property. Thus, there exists $L \subseteq I$ that verifies Def. 9, which shows the second property. The third and the last properties follow from the second one.

Assume that $\text{cov}_c(\text{Arg}(\mathcal{Y})) = \mathbf{C}$, thus $\{c \in \mathbf{C} \mid \exists \langle L, c \rangle \in \text{Arg}(\mathcal{Y})\} = \mathbf{C}$ (1). Note that $\{\mathbf{R}(I) \mid I \in \mathcal{Y}\} \subseteq \mathbf{C}$ follows from the definition of a classifier. Let $c \in \mathbf{C}$ and let's show that $c \in \{\mathbf{R}(I) \mid I \in \mathcal{Y}\}$. From (1), $\exists \langle L, c \rangle \in \text{Arg}(\mathcal{Y})$. By Def. 9, $\exists I \in \mathcal{Y}$ such that $L \subseteq I$ and $\mathbf{R}(I) = c$. So, $c \in \{\mathbf{R}(I) \mid I \in \mathcal{Y}\}$. Assume now that $\{\mathbf{R}(I) \mid I \in \mathcal{Y}\} = \mathbf{C}$ (2), and let's show that $\text{cov}_c(\text{Arg}(\mathcal{Y})) = \mathbf{C}$. From (2) we have $\forall c \in \mathbf{C}$, $\exists I \in \mathcal{Y}$ such that $\mathbf{R}(I) = c$. From the second property, $\exists \langle L, \mathbf{R}(I) \rangle \in \text{Arg}(\mathcal{Y})$. Assume $L \in \mathbf{g}_p(I)$, then $\langle L, \mathbf{R}(I) \rangle \in \text{Arg}(\mathcal{Y})$ as L satisfies all the conditions of Def.9. Let $\langle L, c \rangle \in \text{Arg}(\mathcal{Y})$. Hence, by definition 9, $\exists I' \in \mathcal{Y}$ such that $L \subseteq I'$, $\forall I'' \in \mathcal{Y} \setminus \{I'\}$ such that $L \subseteq I''$, $\mathbf{R}(I'') = \mathbf{R}(I)$, and $\nexists L' \subset L$ such that L' satisfies the above conditions. So, $L \in \mathbf{g}_p(I)$. ■

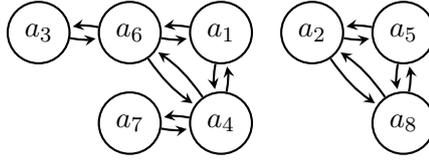
Arguments may be conflicting, particularly when they violate the coherence property, i.e., their supports are consistent but their conclusions are different.

Definition 18 (Attack) *Let $a = \langle L, c \rangle$, $a' = \langle L', c' \rangle \in \text{Arg}(\mathcal{Y})$. We say that a attacks a' iff $L \cup L'$ is consistent and $c \neq c'$. We denote by $\text{Att}(a)$ the set of all attackers of a .*

Property 4 *The attack relation is symmetric and irreflexive.*

Proof Symmetry follows straightforwardly from the definition. Irreflexivity follows from the fact that for every argument $\langle L, c \rangle$, the set L is consistent (see Property 5) and an argument supports a single class. ■

Example 12 (Cont.) The attacks between the eight arguments are depicted in the figure below.



Arguments and their attack relations form an argumentation system as follows.

With a symmetric attack relation, naive extensions coincide with stable extensions (definition 6 on page 51). A later objective is to extend the framework with new, asymmetrical attack relation. In this sense, we only focus on stable extensions rather than naive extensions.

Definition 19 (Argumentation system) An argumentation system built from \mathcal{Y} is a pair $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$ where $\mathcal{R} \subseteq \text{Arg}(\mathcal{Y}) \times \text{Arg}(\mathcal{Y})$ such that for $a, b \in \text{Arg}(\mathcal{Y})$, $(a, b) \in \mathcal{R}$ iff a attacks b (in the sense of Def. 18).

Since arguments are conflicting, they should be evaluated using a semantics. In this paper, we consider the stable semantics that has been recalled in section 3.3 an extension-based semantics introduced in (Dung, 1995), namely *stable* semantics. It computes sets of arguments that can be jointly accepted. Each set is called a *stable extension* and represents a set of compatible plausible explanations.

Example 12 (Cont.) The AS depicted in the above figure has nine stable extensions.

- $\mathcal{E}_1 = \{a_1, a_2, a_3, a_7\}$ $\mathcal{E}_2 = \{a_1, a_3, a_5, a_7\}$
- $\mathcal{E}_3 = \{a_1, a_3, a_7, a_8\}$ $\mathcal{E}_4 = \{a_2, a_3, a_4\}$
- $\mathcal{E}_5 = \{a_3, a_4, a_5\}$ $\mathcal{E}_6 = \{a_3, a_4, a_8\}$
- $\mathcal{E}_7 = \{a_2, a_6, a_7\}$ $\mathcal{E}_8 = \{a_5, a_6, a_7\}$ $\mathcal{E}_9 = \{a_6, a_7, a_8\}$

An argumentation system has one stable extension if the attack relation is empty and multiple extensions otherwise.

Proposition 6 Let $\mathcal{Y} \subseteq \mathcal{X}_T$ and $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$.

- $\sigma(AS) \neq \emptyset$,
- $\sigma(AS) = \{\text{Arg}(\mathcal{Y})\}$ iff $\mathcal{R} = \emptyset$.

Proof From Property 4, \mathcal{R} is symmetric and irreflexive. From Coste-Marquis et al. (2005), $\sigma(AS)$ contains all subset-maximal conflict-free sets of arguments, called naive extensions. Since \mathcal{R} is irreflexive, then $\sigma(AS) \neq \emptyset$. The second property follows from the definitions of \mathcal{R} and stable extension. ■

Let us now turn to the evaluation of individual arguments. Accepted arguments are defined in our approach using two parameters: *selection function* and *inference rule*. The former selects a subset of stable extensions and the latter selects arguments from the chosen extensions. We define various instantiations of the two parameters, capturing different *criteria* for solving conflicts between arguments.

Definition 20 (Selection Functions) *Let $\Sigma = \{\mathcal{E}_1, \dots, \mathcal{E}_k\}$ such that for any $i \in \{1, \dots, k\}$, $\mathcal{E}_i \subseteq \text{Arg}(\mathcal{Y})$. We define below selection functions:*

- $\text{Max}(\Sigma) = \Sigma$
- $\text{Card}(\Sigma) = \{\mathcal{E} \in \Sigma \mid \forall \mathcal{E}' \in \Sigma, |\mathcal{E}| \geq |\mathcal{E}'|\}$
- $\text{Incl}_i(\Sigma) = \{\mathcal{E} \in \Sigma \mid \text{cov}_i(\mathcal{E}) \text{ is subset-maximal}\}$
- $\text{Card}_i(\Sigma) = \{\mathcal{E} \in \Sigma \mid \forall \mathcal{E}' \in \Sigma, |\text{cov}_i(\mathcal{E})| \geq |\text{cov}_i(\mathcal{E}')|\}$
- $\text{Incl}_c(\Sigma) = \{\mathcal{E} \in \Sigma \mid \text{cov}_c(\mathcal{E}) \text{ is subset-maximal}\}$
- $\text{Card}_c(\Sigma) = \{\mathcal{E} \in \Sigma \mid \forall \mathcal{E}' \in \Sigma, |\text{cov}_c(\mathcal{E})| \geq |\text{cov}_c(\mathcal{E}')|\}$
- $\text{Mix}(\Sigma) = \text{Card}_c(\text{Card}_i(\Sigma))$

Applied to the set of stable extensions of an argumentation system, the function **Max** returns all the extensions, **Card** selects the extensions that contain more arguments, the two functions **Incl_i**, **Card_i** focus on the instances covered by the extensions and choose extensions with more instances. These functions promote the Success principle, which requires an explainer to have at least one explanation for each instance. The functions **Incl_c**, **Card_c** focus on classes being justified by arguments. As we will see later, these principles promote explaining a large number of classes. This is useful when explanations are provided for classifier designers as they describe classifier's behaviour. Finally, the function **Mix** combines **Card_i** and **Card_c**, indeed, it starts by selecting the extensions that cover more instances, then it refines the result by selecting extensions that explain more classes.

Example 12 (Cont.) Recall that $\sigma(AS) = \{\mathcal{E}_1, \dots, \mathcal{E}_9\}$.

- $\text{cov}_i(\mathcal{E}_1) = \{I_2, I_3, I_4, I_5, I_6\}$
- $\text{cov}_i(\mathcal{E}_2) = \{I_1, I_2, I_3, I_4, I_5, I_6\}$
- $\text{cov}_i(\mathcal{E}_3) = \{I_2, I_3, I_4, I_5, I_6, I_7\}$ with $\text{cov}_c(\mathcal{E}_1) = \text{cov}_c(\mathcal{E}_2) = \text{cov}_c(\mathcal{E}_3) = \{c_0, c_1, c_2\}$
- $\text{cov}_i(\mathcal{E}_4) = \{I_1, I_2, I_3, I_5\}$ $\text{cov}_c(\mathcal{E}_4) = \{c_0, c_1\}$

- $\text{cov}_i(\mathcal{E}_5) = \{I_1, I_2, I_5\}$ $\text{cov}_c(\mathcal{E}_5) = \{c_1\}$
- $\text{cov}_i(\mathcal{E}_6) = \{I_1, I_2, I_5, I_7\}$ $\text{cov}_c(\mathcal{E}_6) = \{c_1, c_2\}$
- $\text{cov}_i(\mathcal{E}_7) = \{I_3, I_4, I_6, I_7\}$ $\text{cov}_c(\mathcal{E}_7) = \{c_0, c_2\}$
- $\text{cov}_i(\mathcal{E}_8) = \{I_1, I_2, I_4, I_5, I_6, I_7\}$ $\text{cov}_c(\mathcal{E}_8) = \{c_1, c_2\}$
- $\text{cov}_i(\mathcal{E}_9) = \{I_4, I_6, I_7\}$ $\text{cov}_c(\mathcal{E}_9) = \{c_2\}$

The selection functions return the following extensions:

- $\text{Card}(\sigma(AS)) = \text{Card}_c(\sigma(AS)) = \text{Incl}_c(\sigma(AS)) = \{\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3\}$
- $\text{Incl}_i(\sigma(AS)) = \text{Card}_i(\sigma(AS)) = \{\mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_8\}$
- $\text{Mix}(\sigma(AS)) = \{\mathcal{E}_2, \mathcal{E}_3\}$

We show next the links between the selection functions.

Proposition 7 *Let Σ be a non-empty set of subsets of arguments of $\text{Arg}(\mathcal{Y})$. The following inclusions hold:*

- $\text{Card}(\Sigma) \subseteq \text{Max}(\Sigma)$
- $\text{Mix}(\Sigma) \subseteq \text{Card}_i(\Sigma) \subseteq \text{Incl}_i(\Sigma) \subseteq \text{Max}(\Sigma)$
- $\text{Card}_c(\Sigma) \subseteq \text{Incl}_c(\Sigma) \subseteq \text{Max}(\Sigma)$

Proof The inclusions $\alpha(\Sigma) \subseteq \text{Max}(\Sigma)$, with $\alpha \in \{\text{Card}, \text{Incl}_i, \text{Incl}_c\}$, are obvious from the definitions. Let's show that $\text{Mix}(\Sigma) \subseteq \text{Card}_i(\Sigma) \subseteq \text{Incl}_i(\Sigma)$. The inclusion $\text{Mix}(\Sigma) \subseteq \text{Card}_i(\Sigma)$ follows straightforwardly from the definition of Mix . Assume now that some $\mathcal{E} \in \Sigma$ s.t. $\mathcal{E} \in \text{Card}_i(\Sigma)$ and $\mathcal{E} \notin \text{Incl}_i(\Sigma)$. Thus, $\forall \mathcal{E}' \in \Sigma, |\text{cov}_i(\mathcal{E})| \geq |\text{cov}_i(\mathcal{E}')|$ (1). Since $\mathcal{E} \notin \text{Incl}_i(\Sigma)$, then $\exists \mathcal{E}' \in \Sigma$ such that $\text{cov}_i(\mathcal{E}) \subset \text{cov}_i(\mathcal{E}')$, so $|\text{cov}_i(\mathcal{E})| < |\text{cov}_i(\mathcal{E}')|$, which contradicts (1). The proof of the last property is similar. ■

The selection functions may still return several extensions, hence we need to identify the strongest arguments which will yield explanations. For that purpose, we introduce two inference rules that provide strong arguments from extensions.

Definition 21 (Inference Rules) *Let Σ be a non-empty set of subsets of arguments of $\text{Arg}(\mathcal{Y})$ and $a \in \text{Arg}(\mathcal{Y})$. We define the following inference rules:*

- Universal inference: $\Sigma \vdash^{\forall} a$ iff $a \in \bigcap_{\mathcal{E} \in \Sigma} \mathcal{E}$.
- Existential inference: $\Sigma \vdash^{\exists} a$ iff $\exists \mathcal{E} \in \Sigma$ s.t. $a \in \mathcal{E}$.

\mathcal{Y}	I_1	I_2	I_3	I_4	I_5	I_6	I_7
$\mathbf{g}^{\text{Max},\forall}$	\emptyset						
$\mathbf{g}^{\text{Card},\forall}$	\emptyset	$\{L_2\}$	$\{L_1\}$	$\{L_4\}$	$\{L_2\}$	$\{L_4\}$	\emptyset
$\mathbf{g}^{\text{Incl}_i,\forall}$	\emptyset	\emptyset	\emptyset	$\{L_4\}$	\emptyset	$\{L_4\}$	\emptyset
$\mathbf{g}^{\text{Card}_i,\forall}$	\emptyset	\emptyset	\emptyset	$\{L_4\}$	\emptyset	$\{L_4\}$	\emptyset
$\mathbf{g}^{\text{Incl}_c,\forall}$	\emptyset	$\{L_2\}$	$\{L_1\}$	$\{L_4\}$	$\{L_2\}$	$\{L_4\}$	\emptyset
$\mathbf{g}^{\text{Card}_c,\forall}$	\emptyset	$\{L_2\}$	$\{L_1\}$	$\{L_4\}$	$\{L_2\}$	$\{L_4\}$	\emptyset
$\mathbf{g}^{\text{Mix},\forall}$	\emptyset	$\{L_2\}$	$\{L_1\}$	$\{L_4\}$	$\{L_2\}$	$\{L_4\}$	\emptyset

Table 4.1: The outcomes of all functions $\mathbf{g}^{\alpha,\forall}$ in Example 12.

The next result shows the links between the two rules.

Proposition 8 *Let Σ be a non-empty set of subsets of arguments of $\text{Arg}(\mathcal{Y})$ and $a \in \text{Arg}(\mathcal{Y})$. The following hold:*

- $\Sigma \sim^\forall a \Rightarrow \Sigma \sim^\exists a$
- *If $|\Sigma| = 1$, then $\Sigma \sim^\forall a \iff \Sigma \sim^\exists a$.*

Proof The first part is straightforward from the inference definitions, with decreasing constraints on a . For the second part, if there is only one extension E , then intersection is that extension and the rest follows. ■

Selection functions and inference rules are combined for defining *accepted* arguments. Each pair gives birth to a criterion for declaring an argument as accepted.

Definition 22 (Accepted Arguments) *Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$, α be a selection function and β an inference rule. An argument $a \in \text{Arg}(\mathcal{Y})$ is accepted, denoted by $AS \sim^{\alpha,\beta} a$, iff $\alpha(\sigma(AS)) \sim^\beta a$.*

We show that accepted arguments under the function **Max** (which retains all extensions) are non-attacked ones if **Max** is combined with the universal rule and they are all arguments of the system when **Max** is combined with the existential rule.

Proposition 9 *Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$ and $a \in \text{Arg}(\mathcal{Y})$.*

- $AS \sim^{\text{Max},\forall} a \iff \text{Att}(a) = \emptyset$
- $\{a \in \text{Arg}(\mathcal{Y}) \mid AS \sim^{\text{Max},\exists} a\} = \text{Arg}(\mathcal{Y})$

Before proving these results, let us recall a property given in (Coste-Marquis et al., 2005).

Property 5 (Coste-Marquis et al., 2005) Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$ and $a \in \text{Arg}(\mathcal{Y})$. If \mathcal{R} is symmetric and irreflexive, then the following hold:

- $\forall a \in \text{Arg}(\mathcal{Y}), a \in \bigcup_{\mathcal{E} \in \sigma(AS)} \mathcal{E}$
- $a \in \bigcap_{\mathcal{E} \in \sigma(AS)} \mathcal{E}$ iff $\text{Att}(a) = \emptyset$

This property allows to prove Proposition 9:

Proof The first result follows from Property 5.

Assume that $AS \sim^{\alpha, A} a$. There are two cases: i) $\text{Att}(a) = \emptyset$, thus from the first result we have $AS \sim^{\text{Max}, \forall} a$. ii) $\text{Att}(a) \neq \emptyset$. By definition, $\exists \mathcal{E} \in \sigma(AS)$ such that $a \in \mathcal{E}$ and $\forall b \in \text{Att}(a), b \notin \bigcup_{\mathcal{E} \in \sigma(AS)} \mathcal{E}$, which contradict Property 5.

The third result follows from Property 5, namely the fact that $\forall a \in \text{Arg}(\mathcal{Y}), a \in \bigcup_{\mathcal{E} \in \sigma_n(AS)} \mathcal{E}$. ■

Below are links between accepted arguments returned using the same inference rule but distinct selection functions.

Proposition 10 Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$, $a \in \text{Arg}(\mathcal{Y})$, and α, α' be two selection functions. If $\alpha(\Sigma) \subseteq \alpha'(\Sigma)$, then:

- $AS \sim^{\alpha', \forall} a \Rightarrow AS \sim^{\alpha, \forall} a$
- $AS \sim^{\alpha, \exists} a \Rightarrow AS \sim^{\alpha', \exists} a$

Proof Let α, α' be two selection functions such that $\alpha(\Sigma) \subseteq \alpha'(\Sigma)$. Assume that $AS \sim^{\alpha', \forall} a$. Then, $\forall \mathcal{E} \in \alpha'(\Sigma), a \in \mathcal{E}$. Since $\alpha(\Sigma) \subseteq \alpha'(\Sigma)$, then $\forall \mathcal{E} \in \alpha(\Sigma), a \in \mathcal{E}$, so $AS \sim^{\alpha, \forall} a$. Assume now that $AS \sim^{\alpha, \exists} a$. Then, $\exists \mathcal{E} \in \alpha(\Sigma)$ such that $a \in \mathcal{E}$. Since $\mathcal{E} \in \alpha'(\Sigma)$, then $AS \sim^{\alpha', \exists} a$. ■

Below is a complete list of links between sets of accepted arguments returned by pairs of selection principles and inference rules.

Proposition 11 The following implications hold.

- $AS \sim^{\alpha, \forall} a \Rightarrow AS \sim^{\alpha, \exists} a, \forall \alpha$
- $AS \sim^{\text{Max}, \forall} a \Rightarrow AS \sim^{\text{Card}, \forall} a$
- $AS \sim^{\text{Max}, \forall} a \Rightarrow AS \sim^{\text{Incl}_1, \forall} a \Rightarrow AS \sim^{\text{Card}_1, \forall} a \Rightarrow AS \sim^{\text{Mix}, \forall} a$
- $AS \sim^{\text{Max}, \forall} a \Rightarrow AS \sim^{\text{Incl}_c, \forall} a \Rightarrow AS \sim^{\text{Card}_c, \forall} a$

- $AS \sim^{\text{Card},\exists} a \Rightarrow AS \sim^{\text{Max},\exists} a$
- $AS \sim^{\text{Mix},\exists} a \Rightarrow AS \sim^{\text{Card}_i,\exists} a \Rightarrow AS \sim^{\text{Incl}_i,\exists} a \Rightarrow AS \sim^{\text{Max},\exists} a$
- $AS \sim^{\text{Card}_c,\exists} a \Rightarrow AS \sim^{\text{Incl}_c,\exists} a \Rightarrow AS \sim^{\text{Max},\exists} a$

Proof The properties follow from Propositions 5, 6 and 8. ■

We are now ready to define our new parameterized family of plausible explanation functions. For a given instance I , they return the support of any argument in favour of $R(I)$ inferred by following one of the principles defined above. The support of the argument should be part of the instance I .

Definition 23 (Explanation Functions) *Let T be a theory, $\mathcal{Y} \subseteq \mathcal{X}_{\mathsf{T}}$, R a classifier, α a selection function and β an inference rule. An explainer is a function $\mathbf{g}^{\alpha,\beta}$ mapping every instance $I \in \mathcal{Y}$ into a set of subsets of literals such that every $L \in \mathbf{g}^{\alpha,\beta}(I)$ satisfies the following:*

- $AS \sim^{\alpha,\beta} \langle L, R(I) \rangle$ where $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$,
- $L \subseteq I$.

Example 12 (Cont.) Table 4.1 summarizes the explanations of the seven instances provided by every function which uses the universal inference rule. Note that the new functions explain more instances than the argument-based function (which is equivalent to $\mathbf{g}^{\text{Max},\forall}$) from (Amgoud, 2021b).

We show that all the above defined explanation functions are refined plausible explainers, i.e., they return subsets of explanations computed by the function \mathbf{g}_p (see Definition 16).

Proposition 12 *Let T be a theory, $\mathcal{Y} \subseteq \mathcal{X}_{\mathsf{T}}$ and R a classifier. For every selection function α , every inference rule β , every $I \in \mathcal{Y}$, it holds that $\mathbf{g}^{\alpha,\beta}(I) \subseteq \mathbf{g}_p(I)$.*

Proof Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$ be an argumentation system built from $\mathcal{Y} \subseteq \text{Inst}(\mathsf{T})$. Assume that $L \in \mathbf{g}^{\alpha,\beta}(I)$. Thus, $AS \sim^{\alpha,\beta} \langle L, R(I) \rangle$ and $L \subseteq I$. So, $L \in \mathbf{g}_p(I)$. ■

The following results show the links between the various explanation functions.

Proposition 13 *Let $I \in \mathcal{X}_{\mathsf{T}}$.*

- $\mathbf{g}^{\alpha,\forall}(I) \subseteq \mathbf{g}^{\alpha,\exists}(I)$ for any selection function α
- $\mathbf{g}^{\text{Max},\forall}(I) \subseteq \mathbf{g}^{\text{Card},\forall}(I)$

- $g^{\text{Max},\forall}(I) \subseteq g^{\text{Incl}_i,\forall}(I) \subseteq g^{\text{Card}_i,\forall}(I) \subseteq g^{\text{Mix},\forall}(I)$
- $g^{\text{Max},\forall}(I) \subseteq g^{\text{Incl}_c,\forall}(I) \subseteq g^{\text{Card}_c,\forall}(I)$
- $g^{\text{Card},\exists}(I) \subseteq g^{\text{Max},\exists}(I)$
- $g^{\text{Mix},\exists}(I) \subseteq g^{\text{Card}_i,\exists}(I) \subseteq g^{\text{Incl}_i,\exists}(I) \subseteq g^{\text{Max},\exists}(I)$
- $g^{\text{Card}_c,\exists}(I) \subseteq g^{\text{Incl}_c,\exists}(I) \subseteq g^{\text{Max},\exists}(I)$

Proof The properties follow from Proposition 11. ■

It is worth mentioning that the explanation function $\mathbf{g}^{\text{Max},\exists}$ corresponds exactly to the plausible explanation function \mathbf{g}_p .

Property 6 *It holds that $\mathbf{g}^{\text{Max},\exists} = \mathbf{g}_p$.*

The function $\mathbf{g}^{\text{Max},\forall}$ coincides with the function \mathbf{g}^* introduced in (Amgoud, 2021b). We show that this function is very cautious as it discards any explanation which is incoherent with at least one other explanation.

Proposition 14 *Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$ and $I \in \mathcal{Y}$. $\mathbf{g}^{\text{Max},\forall}(I) = \{L \in \mathbf{g}_p(I) \mid \forall L' \in \mathbf{g}_p(I'), \text{ if } R(I) \neq R(I') \text{ then } L \cup L' \text{ is inconsistent}\}$.*

Proof Follows from Proposition 9 and the fact that every $L \in \mathbf{g}_p(I)$ gives birth to an argument $\langle L, \mathcal{R}(I) \rangle \in \text{Arg}(\mathcal{Y})$. ■

We show next how the various explainer behave with respect to the two principles of Coherence and Success.

Theorem 2 *Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$, α be a selection function and β an inference rule. If $|\sigma(AS)| > 1$, then:*

- $\mathbf{g}^{\alpha,\beta}$ satisfies Success iff $\alpha = \text{Max}$ and $\beta = \exists$.
- $\mathbf{g}^{\alpha,\beta}$ satisfies Coherence iff $\beta = \forall$.

The above result shows that $\mathbf{g}^{\text{Max},\exists}$ (or \mathbf{g}_p) is the **only** function which satisfies success and all the other functions that are based on the existential inference rule violate both success and coherence. Consequently, those functions are not reasonable. Note that in this paper, we investigated the different possibilities for the purpose of *completeness* and proving formally which function is not suitable and why it is not.

Coherence is guaranteed by **all the functions that are based on the universal inference rule**. Furthermore, $\mathbf{g}^{\text{Card},\forall}$, $\mathbf{g}^{\text{Mix},\forall}$ and $\mathbf{g}^{\text{Card}_c,\forall}$ are more informative than the

other functions that use the same inference rule. Indeed, they provide more explanations for instances, and can thus **explain more instances**. However, the three functions may return different outcomes as they follow **different strategies**. $g^{\text{Card},\forall}$ is less interesting than the two others. It selects the extensions that contain more arguments, but the latter may support the same class as in our running example (the arguments in the extension \mathcal{E}_1 are all in favour of the class c_0). Hence, any instance whose prediction is c_1 gets an empty set of explanations.

The function $g^{\text{Mix},\forall}$ **maximizes the number of instances** for which explanations are provided, this is important in domains like healthcare or banking where explanations are generally requested by end-users.

$g^{\text{Card},\forall}$ **maximizes the number of explained classes**. It is suitable for understanding the global behaviour of a classifier, especially for a problem with a lot of classes.

Chapter 5

Experimental study

In this chapter, we investigate how the proposed theoretical argument-based explainer works in practice. Our solution is a model-agnostic explainer. In subsection 2.0.1, Figure 2.2 shows a simple representation of a model-agnostic explainer structure. The (black-box) prediction model R is probed on the dataset \mathcal{Y} and the predictions are the new labels for our explainer $g^{\mathcal{Y}}$.

For each step of the explanation process, we explain the theoretical base and the practical algorithms. Then we evaluate the computational complexity of the solution. Finally each step is evaluated experimentally. Experiments serve the purpose of giving an intuition on the behavior of the different elements at stake for the construction of explanations. The end goal is to propose a novel explanation function that can be used for any classifier. The experiments also serve the purpose of observing the limitations of our proposal.

The experiments are performed on a collection of datasets that we will use for multiple purposes and in different shapes throughout this chapter. The collection is composed of datasets widely used by the research communities for testing proofs of concept. They vary in terms of size of training data, and number of features. We detail the different datasets in section 5.1

The framework was implemented following the theoretical work. We can divide the process into 2 main parts:

- Construction of the Argumentation System:
Two steps lead to this result: The enumeration of arguments and the enumeration of the attacks. The result of both sets of arguments and attacks is a graph representing the Argumentation Framework.
- Enumeration and analysis of the extensions:
First, the enumeration of the stable extensions, then the selection of extension and finally the selection of arguments according to a chosen inference principle. The two

final steps depend on the parameters of the chosen explanation function.

A representation of the structure is given in Figure 5.1.

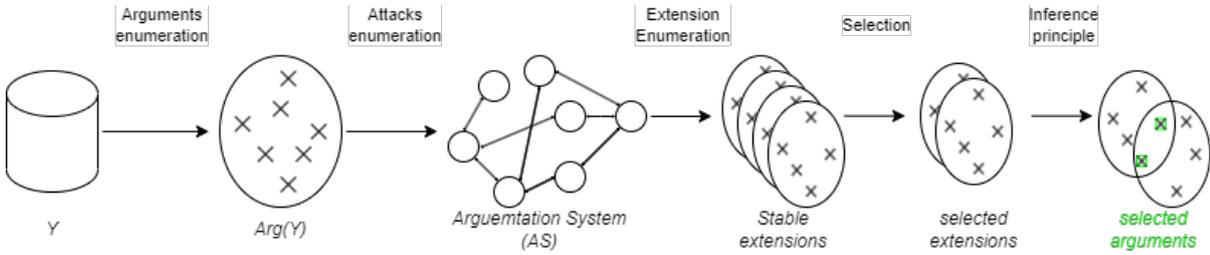


Figure 5.1: Overview of the implemented process

The argument-based explanation method requires a set of instances based on which the arguments will be enumerated. This procedure will be detailed in section 5.2. All these arguments will define a set of vertices. It is clear that the set of instances is far from covering the whole input space and the information e.g. arguments drawn from it will generate incoherences. That is why we identify conflicts between them, and encode them as a set of edges. In section 5.3 on page 91, we explain how edges in the graph are generated from the set of arguments according to the definition of the attack relation. This graph is called an Argumentation System (AS), the vertices are the arguments and the edges represent attacks between arguments. The following section describes how we implement possible techniques and the limits to finding coherent positions. A set of arguments is coherent if it is conflict-free. Finally, in section 5.5 we show how the selection functions and the inference rules allow the user to strategically aggregate extensions according to their needs and requirements.

5.1 Datasets

In the following sections, we test the implemented solution on five datasets. These datasets are well-known datasets often used to experiment ML explanation techniques (Ribeiro et al., 2018). Because these datasets have few features and a reasonable input space size, they are a good fit for our experiments. In this section we describe the datasets' structure as well as their sizes. We also give precisions on how we adapted them for our experiments. All datasets are available on Kaggle.

5.1.1 Titanic

The titanic dataset (Cukierski, 2012) is a toy dataset largely used to test ML models. The data describes the population of the Titanic at the time of the shipwreck in 1912

Feature	Definition	Categorical	Values
survival	Survival	Yes	2
pclass	Ticket class	Yes	3
sex	Sex	Yes	2
Age	Age in years	No	4
sibsp	# of siblings spouses aboard the Titanic	No	3
parch	# of parents children aboard the Titanic	Yes	7
ticket	Ticket number	No	-
fare	Passenger fare	No	4

Table 5.1: Features of the titanic dataset

resulting in the death of 1502 out of 2224 passengers and crew. Although luck was involved in surviving, the data shows that some groups of people showed better survival rates than others, for example, 73% of women survived against 19% for men. The dataset is composed of a training set of 819 instances and a test set of 418 instances. Table 5.1 lists the features, their definition. We also indicate if the feature is categorical, and if not it is discretized. The grey rows are the features not used in the experiments and the red one is the class to predict. Here, the ticket number is a unique value for each passenger, so it is not a useful information for the prediction. In this setting, we can calculate the size of the input space: 2016 possible instances.

In some experiments, we use portions of the full dataset. This is done by generating all possible instances in random order and dividing the set into 8 portions. Otherwise, we use instances from the training set as base dataset to generate explanations.

5.1.2 Adult

The adult dataset, available in Ribeiro’s Anchors implementation (Ribeiro et al., 2018), was first introduced in (Kohavi, 1996). The database collects 45222 clean observations in which adults are characterized by 15 features including the prediction goal: the person’s annual income is over or below 50 000 dollars. We detail the different features in table 5.2 on the following page. Again, the grey features are not used in experiments. The reason for eliminating these features is to keep the size of the input space reasonably low and be able to perform all experiments. In this setting, the input space contains 408240 possible observations.

5.1.3 Lending

The *lending* dataset 1 was also used to benchmark Anchors in (Ribeiro et al., 2018). The goal of the *Lending Club loan* dataset is to predict if a lended DVD will be returned correctly. This dataset counts 115 features. In this description we will only describe the features (table 5.3 on the next page) used in our experiments. There are 42538 instances

Feature	Definition	Categorical	Values
age	Age	No	4
workclass	Occupation	Yes	9
fnlwgt	unknown	No	7
education	Highest level of education	Yes	6
education-num	unknown	No	4
marital-status	Marital status	Yes	2
relationship	Relationship	Yes	3
race	Race	Yes	3
sex	Gender	Yes	2
capital-gain	Capital gain	Yes	2
capital-loss	Capital loss	Yes	4
hours-per-week	Level of schooling	No	4
native-country	Native country	Yes	3
income	if annual income is <50K or not	No	4

Table 5.2: Features of the adult dataset

Feature	Definition	Categorical	Values
home_ownership	The ownership status of the applicant’s residence	Yes	5
pub_rec_bankruptcies	Number of bankruptcies listed in the public record	Yes	4
loan_amnt	The amount of the loan the applicant received	No	4
annual_inc	Annual Income	No	4
desc	-	No	4
inq_last_6mths	Inquiries into the applicant’s credit during the last 6 months	No	3
revol_util	-	No	4
last_fico_range_high	-	No	3
last_fico_range_low	-	No	2
late_payment	Good or bad loan	Yes	2

Table 5.3: Features of the lending dataset

in the dataset. With the restriction of the inputs, we lower the input space size to 960 possible observations.

5.1.4 Recidivism

The *rcdv* dataset was published in (Schmidt and Witte, 1989) for an attempt to predict recidivism. The data contains the information about inmates released from North Carolina prisons in 1980. There also exists a dataset for 1978. The set is composed of 18 features including the prediction category. In our setting, we only use 12 features, 11 of which are categorical. The Age feature was discretized. In total, the input space has 24576 possible observations. Again, to keep the input space small enough, gray highlighted features are not considered.

Feature	Definition	Categorical	Values
WHITE	Race	Yes	2
ALCHY	Alcoholism	Yes	2
JUNKY	Sex	Yes	2
SUPER	Supervised Release	Yes	2
MARRIED	Married	Yes	2
FELON	Felony	Yes	2
WORKREL	Work Release	Yes	2
PROPTY	Crime against Property	Yes	2
PERSON	Crime against Person	Yes	2
MALE	Gender	Yes	2
PRIORS	Number of priors incarcerations	Yes	4
SCHOOL	Level of schooling	No	4
RULE	Prison violations	Yes	3
AGE	Age in months	No	4
TSERVD	Months served	No	4
FOLLOW	Duration of followup period	No	-
RECID	Recidivism	Yes	2
TIME	months until recidivism	No	-

Table 5.4: Features of the rcdv dataset

Feature	Definition	Categorical	Values
Pregnancies	Number of times pregnant	No	14
Glucose	Plasma glucose concentration	No	4
BloodPressure	Diastolic blood pressure (mm Hg)	No	3
SkinThickness	Triceps skin fold thickness (mm)	No	4
Insulin	2-Hour serum insulin (mu U/ml)	No	4
BMI	Body mass index	No	4
DiabetesPedigreeFunction	Diabetes pedigree function	No	4
Age	Age (years)	No	4
Outcome	Class variable (0 or 1)	Yes	2

Table 5.5: Features of the diabetes dataset

5.1.5 Diabetes

The *diabetes* dataset 2, was introduced in (Smith et al., 1988) for AI-based diagnostic of diabetes. The dataset is fairly simple, counting 768 samples characterizing females over 21 years old with 9 features table 5.5. The whole input space contains 172032 different observations. The features colored in gray were not accounted for in experiments to keep the input space small enough.

5.2 Arguments Enumeration

The arguments are the base of this framework. Arguments can be seen as minimal rules to assign a class to an instance of the data. The definition of argument is given in the theoretical framework definition 17 on page 70. Arguments are pairs $\langle L, c \rangle$, for which, any

Dataset	features	observations	input space
<i>titanic</i>	8 (6)	1237	2016
<i>adult</i>	14 (8)	45222	408240
<i>lending</i>	115 (5)	42538	960
<i>rcdv</i>	18 (12)	9549	24576
<i>diabetes</i>	9 (5)	768	172032

Table 5.6: Datasets summary (features used in experiments are in parthesis)

instance containing L will be predicted to class c by the black-box model. Arguments are extracted from a database of instances \mathcal{Y} , and each instance is predicted by a black-box classifier \mathbf{R} . The arguments are generated using these instances and predictions with the aim to explain the predictions of \mathbf{R} . This section depicts how the arguments are enumerated according to this definition. In the first subsection, we present an overview of the implementation and the main challenges. In the second subsection, we explain each solution in depth.

5.2.1 Overview

Based on definition 17 on page 70, we divide the matter into 3 sub-problems:

1. Check all possible arguments $\langle L, c \rangle$
2. Verify *consensus* (Equation 5.1) for each argument (e.g. 3rd item in the definition)

$$\forall I \in \mathcal{Y} \text{ such that } L \subseteq I, \mathbf{R}(I) = c \quad (5.1)$$

3. Verify *minimality* (Equation 5.2) for each argument (e.g. 4th item in the definition)

$$\nexists L' \subset L \text{ that verifies the above conditions} \quad (5.2)$$

The first two items of definition 17 on page 70 (i.e. $L \subseteq \text{Lit}(\mathbf{T})$ and $c \in \mathbf{C}$, and $\exists I \in \mathcal{Y}$ such that $L \subseteq I$), are trivially verified with the construction of arguments whereas the conditions of sub-problems (2) and (3) demand a more careful treatment for their verification. The sub-problem (1) is easily resolved by enumerating all combinations of all arguments' support $L \in \mathcal{Y}$. Then, for each enumerated argument, it is necessary to do both checks: *minimality* and *consensus*. The *minimality* is secured by generating arguments by increasing length. Thus, we can check for an argument of length n if it is a superset of an argument of length inferior to n . The *consensus* is guaranteed by finding all instances $I \in \mathcal{Y}$ that include L and verifying that they are all identically predicted.

5.2.2 In-depth analysis

In this section, we explain the different steps more thoroughly. The pseudo-algorithm is presented in algorithm 1. We will explain the choices made and the different data structures that are necessary. The first part will cover the issue of enumerating all possible arguments, then we explain how to guarantee *minimality* and finally the *consensus* in a third part.

5.2.2.1 Potential arguments generation

Algorithm 1 Argument generator (length k literals)

Input: \mathcal{Y} dataset, Arg_{k-1} the set of arguments of length $1..k-1$

Parameter: n length of arguments, y the prediction vector

Output: Arg_k the set of arguments of length n

```

1: for  $I \in \mathcal{Y}$  do
2:   for  $L \in \text{combinations}(I, n)$  do
3:     if  $L \in \text{Arg}_{k-1}$  then
4:       pass
5:     end if
6:     if  $\text{is\_minimal}(L)$  then
7:        $\text{args}_L = \text{get\_instances\_including\_L}(L)$ 
8:       if  $y(\text{args}_L) = y(L)$  then
9:         append  $L$  to  $\text{Arg}_k$ 
10:      end if
11:    end if
12:  end for
13: end for
14: return  $\text{Arg}_k$ 

```

A potential argument $\langle L, c \rangle$ is any argument which L can be found in an instance $I \in \mathcal{Y}$. We use the function $\text{combinations}(I, n)$ to enumerate all combinations of literals in I of length n . The dataset \mathcal{Y} is encoded using One-Hot (OH) encoding. This is possible because all features are either categorical or discretized. It allows to easily process instances of data. For example, the instance $I_0 = \langle f_0, 1 \rangle, \langle f_1, 0 \rangle, \langle f_2, 2 \rangle$ is encoded as in eq. (5.3)

$$\begin{aligned}
 I_{0_{enc}} &= (f_00 = 0, f_01 = 1, f_10 = 1, f_11 = 0, f_20 = 0, f_21 = 0, f_22 = 1) \\
 &= (0, 1, 1, 0, 0, 0, 1)
 \end{aligned} \tag{5.3}$$

These are the solutions for the two first lines of code in algorithm 1. However, this solution presents a minor flaw. Many potential arguments can appear several times in the dataset. In order to avoid the costly treatment of *consensus* and *minimality* verification, we use a set object args_checked in which we add every new potential argument and verify if it was not already treated (lines 3-4).

5.2.2.2 *Minimality* Guarantee

The 4th item in definition 17 on page 70 states that $\nexists L' \subset L$ that verifies other items in the definition. In order to guarantee this point, we choose to generate arguments by length. Starting from minimum length (e.g. one literal) and adding another literal when all arguments of length 1 are found. Thus, by induction, we can guarantee that all arguments are minimal.

Suppose $\text{Arg}_k(\mathcal{Y})$ contains all arguments of length 1 to k . We now generate arguments of length $k + 1$. Let L a potential argument, if $\exists L' \subset L \in \text{Arg}_k(\mathcal{Y})$, then L is not added to $\text{Arg}_{k+1}(\mathcal{Y})$. This verification is done in the *is_minimal* sub-routine (algorithm 2). If not, L is candidate for the *consensus* check. The downside of this method resides in the obligation to check that every subset of L is not in the list of arguments.

Algorithm 2 *is_minimal*

Input: argument L

Parameter: previously found arguments Arg_{k-1}

Output: True if L is minimal

```
1: for  $k \in 1..len(L) - 1$  do
2:   for  $subset \in combinations(L, k)$  do
3:     if  $subset \in \text{Arg}_{k-1}$  then
4:       return False
5:     end if
6:   end for
7: end for
8: return True
```

5.2.2.3 *Consensus* check

Our solution to collect all instances that contain an argument works by using a hash table. For each feature-value pair, the table keeps a set of instances including this pair, accessible in constant time. It is called *ibyfv* for instance by feature-value in algorithm 3 on the next page. This list of instances is later used in algorithm 1 on the preceding page to collect the predictions of every instance. If all instances are predicted to the same class by the model, the argument L is added to the set of arguments Arg_k .

5.2.3 Complexity

The goal is to enumerate all arguments. For this process, it is mandatory to check all possible subsets of feature values of each instance. We try to loop only once over the dataset \mathcal{Y} for each length and verify the *minimality* and *consensus* properties. Let T a theory, n the number of rows, m the number of different possible feature-value pairs.

Algorithm 3 `get_instances_including_L`

Input: argument L **Parameter:** $ibylv$ dictionary of instances by feature-value**Output:** True if $arg_{potential}$ is minimal

```
1:  $s = \text{set}()$ 
2: for  $l \in L$  do
3:    $s.\text{intersect}(ibylv(l))$ 
4: end for
5: return  $s$ 
```

According to algorithm 1 on page 87, the whole code is in two loops. They allow to address every potential argument in the dataset \mathcal{Y} . In the worst case scenario, the inside of the loops will be repeated as many times as the maximum number of potential arguments. We study here the worst case complexity.

$$|\text{Arg}_k(\mathcal{Y})| = O\left(\binom{m}{k} * n\right) \quad (5.4)$$

Thus,

$$|\text{Arg}(\mathcal{Y})| = O\left(\sum_{k=1}^m \binom{m}{k} * n\right) = O(2^m * n) \quad (5.5)$$

Inside the main loop, there are two additional costly actions:

- *Minimality* check (algorithm 2 on the facing page)
- *Consensus* check (algorithm 3)

As described in the pseudo-code algorithm algorithm 2 on the facing page, checking if a set of feature-value pairs e.g. the potential argument is minimal, we need to verify if every subset of this potential argument is not already an argument. The worst case complexity for *minimality* check is

$$O\left(\sum_{i=1}^{k-1} \binom{k-1}{i}\right) \quad (5.6)$$

times the complexity of the *set.in* python operation which is equal to $O(k)$ in the worst case. Thus, the worst case complexity for *minimality* check:

$$O((2^{k-1}) * k) \quad (5.7)$$

The worst case complexity for the multiple sets intersection is

$$(n - 1) * O(l) \quad (5.8)$$

where l is $\max(\text{len}(s1), \dots, \text{len}(sn))$ according to the python documentation.

In conclusion, the complexity for the generation of all arguments is

$$O\left(\sum_{k=1}^m \binom{m}{k} * 2^{k-1} * k * n\right) = O(3^m * m * n) \quad (5.9)$$

In practice, the experiments remain quite far from the worst case scenario. There is a balancing effect depending on the size of the dataset. A smaller dataset observes lesser combinations of feature values while a larger one is likely to have many redundant potential arguments. In the next sub-section, in addition to simple experiments to give an intuition on the number of arguments, we show the computation time for several datasets, varying the number of feature-values or the number of instances.

5.2.4 Experiments

We display the number of arguments found for several setups. The goal is to build an intuition on how quickly the number of arguments increases with the number of instances in the dataset \mathcal{Y} . We lead experiments on different number of inputs and different datasets.

5.2.4.1 Number of arguments

For the *diabetes* and *adult* datasets, we plot the number of arguments for 25, 50, 75, 100, 200, 300 and 500 first instances of the training dataset. They define our input dataset \mathcal{Y} .

Figure 5.2 shows that the number of arguments grows with the number of instances. We also observe that the number of arguments can easily exceed the number of instances to explain.

5.2.4.2 Proposition validation

It is possible to probe the whole input space on short datasets. For this experiment, we only use the features selected and listed in section 5.1. We take advantage of this to observe how the number of arguments grows when the knowledge of the input space is greater. We enumerate the arguments for 1/8, 2/8 ... 8/8 of the whole input space by synthetically generating the inputs and predicting their class with the black-box classifier. Figure 5.3 confirms this trend. We can see that the number of arguments decreases when we get closer to the input space, and their length grows.

In order to explain this behavior, we can use proposition 15.

Proposition 15 *Let \mathbb{T} be a theory and $\mathcal{Y} \subseteq \mathcal{X}(\mathbb{T})$. For every $I \in \mathcal{Y}$, if $L \in \mathbf{g}_a(I)$, then $\exists L' \subseteq L$ such that $L' \in \mathbf{g}_p(I)$.*

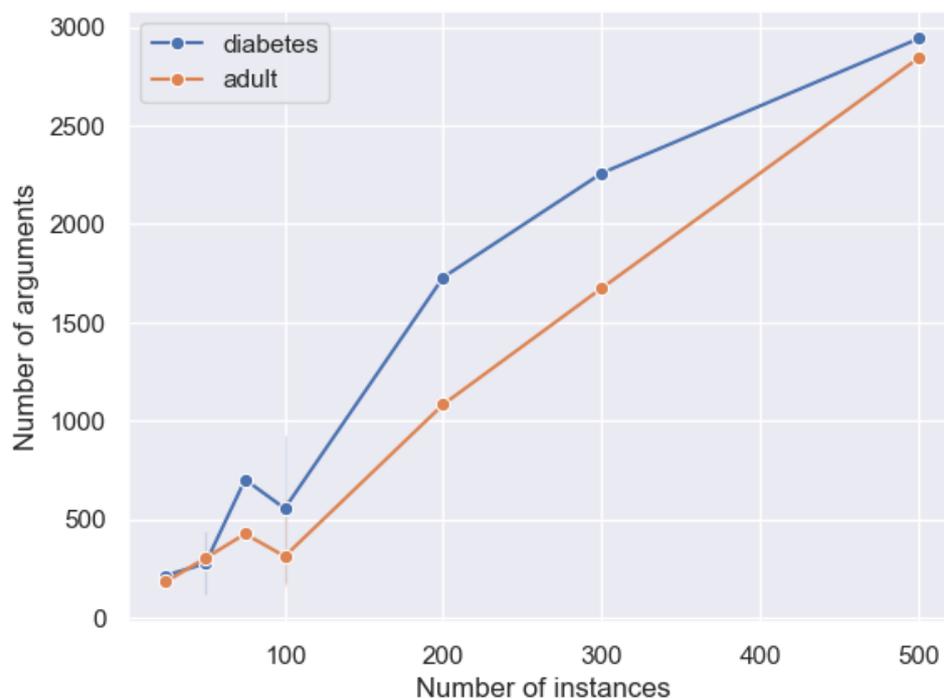


Figure 5.2: Arguments enumerated for portions of the training set

From the proposition, we can deduce that $|\text{Arg}(\mathcal{F})| \leq |\text{Arg}(S)|, \forall S \subseteq \mathcal{F}$. The result of the experiment with the whole dataset illustrates well a consequence of this property. We can conclude that the number of rules we can extract from 1 or 2/8 of the full input space is much higher than when we approach the full input space. This images the difference between the plausible (definition 15 on page 65) and the absolute explanation functions (definition 14 on page 64). We can make the hypothesis that because of the lack of knowledge, we assume too many plausible explanations. With better knowledge, the argument enumeration process generates less arguments because they are more constrained by the rest of the instances. To verify this, it is interesting to study the attack relations between arguments and how they evolve with more or less knowledge.

5.3 Argumentation System: Attack relations

After the enumeration of all arguments, it is likely to find incoherences between arguments (example 13). This is the case when two arguments have consistent support but contrary conclusions (definition 18 on page 71). In this section, we explain how we detect these incoherences.

Example 13 Suppose $a_1 = \langle L_1, c_1 \rangle$ and $a_2 = \langle L_2, c_2 \rangle$ are two arguments generated.

$a_1 = \langle \{ \text{Sex}=\text{Female}, \text{Workclass}=\text{State-gov} \}, > \$50k \rangle$

$a_2 = \langle \{ \text{Sex}=\text{Female}, \text{Relationship}=\text{Not-in-family} \}, < \$50k \rangle$

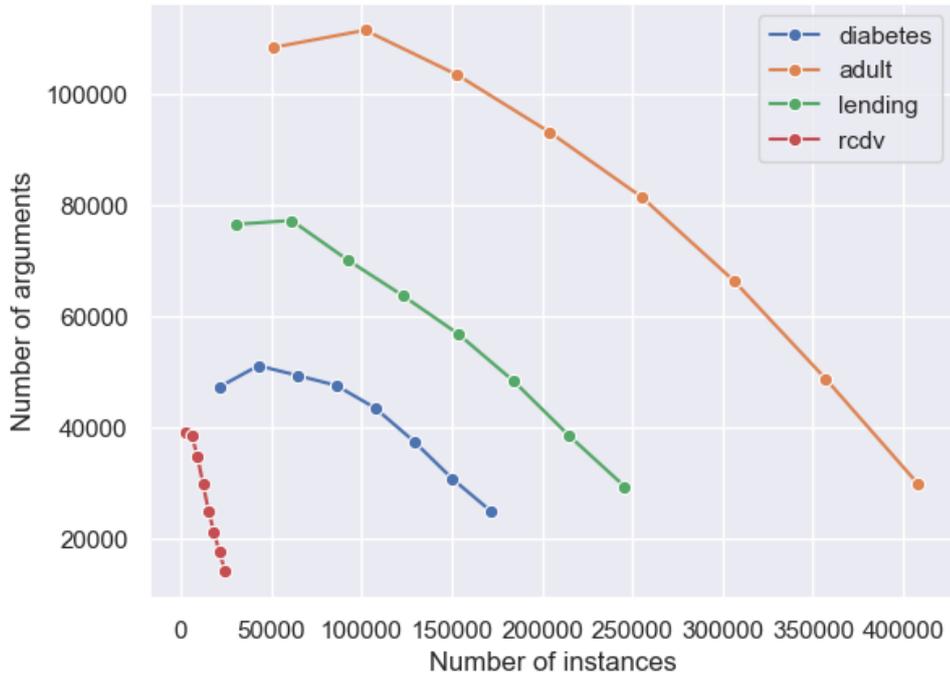


Figure 5.3: Arguments enumerated using portions of the input space

It is possible to find an instance I such that $L_1 \cup L_2 \subseteq I$. However according to a_1 , I would be predicted to c_1 and according to a_2 , it would be predicted to c_2 . This is incoherent.

To build the set of attacks, we check for each possible pair of arguments of different class if they are consistent. The consistency function is given in algorithm 4.

Algorithm 4 is_consistent

Pre-condition: $arg1$ and $arg2$ have different conclusions.

Input: arguments $arg1$ and $arg2$ given as sets of literals.

Parameter: $feature$ is a mapping corresponding each feature value to its feature

Output: True if $arg1$ and $arg2$ are consistent.

- 1: Let $union_{args} = sorted(union(arg1, arg2))$
 - 2: **for** $k \in 0..len(union_{args}) - 1$ **do**
 - 3: **if** $features[k] == features[k + 1]$ **then**
 - 4: **return** False
 - 5: **end if**
 - 6: **end for**
 - 7: **return** True
-

The union step allows to “merge” literals that are the same for both arguments. The sort allows to order the literals by feature. Then the for loop checks if two (different) literals have the same feature but different values. If so, the arguments are inconsistent.

The pseudo-code for the attacks enumeration from the list of arguments is given in algorithm 5 on the next page.

Algorithm 5 attacks enumeration

Input: arguments $args_0$ and $args_1$ lists of arguments for class 0 and class 1.

Output: List of attacks as pairs of arguments.

```
1:  $R_{atk} = []$ 
2: for  $a1, a2 \in product(args_0, args_1)$  do
3:   if  $is\_consistent(a1, a2)$  then
4:      $R_{atk} = R_{atk} \cup (a1, a2)$ 
5:   end if
6: end for
7: return  $R_{atk}$ 
```

There are two conditions for an attack. The first one is that both arguments have the different classes. By taking elements of the cartesian product (line 2) we can select all pairs of arguments with different conclusion c . The second condition is the consistency. The function *is_consistent* is in charge of this verification.

5.3.1 Complexity

There are two steps in the function. Let's suppose the maximum length of arguments is m . Firstly, we use a product of two sets of arguments (one for each class): for n_0, n_1 arguments, we have $n_0 * n_1$ calls of *is_consistent*. Secondly, the *is_consistent* function takes at most $2 * m$ verifications.

In conclusion, the worst case complexity for the attacks enumeration procedure is $O(m^3)$

5.3.2 Experimental results

We show in the in fig. 5.4 on the following page the total number of attacks in the AS according to the number of instances. In this case, we use data from the training set, keeping the number of instances low. In fig. 5.5 on page 95, we repeat the experiment for the whole input space.

These two experiments allow to view the differences in consistency between the information extracted from a small portion and a large portion of the input space. From these graphs, we can see that the maximum number of attacks is before 1/8 for all datasets. After, the global number of attacks keeps decreasing until reaching 0 attacks. The AS reached zero attack when the arguments extracted are all coherent with each other. This is the case for the absolute abductive explanation function (definition 14 on page 64).

In fig. 5.6 on page 96 and fig. 5.7 on page 97, we showcase the number of attacks per argument. These experiments show the inconsistency of knowledge in the dataset. An attack between two arguments highlights a contradiction between both arguments. Instead of showing the Argument System's (AS) graph, we prefer to show the box plots of attacks

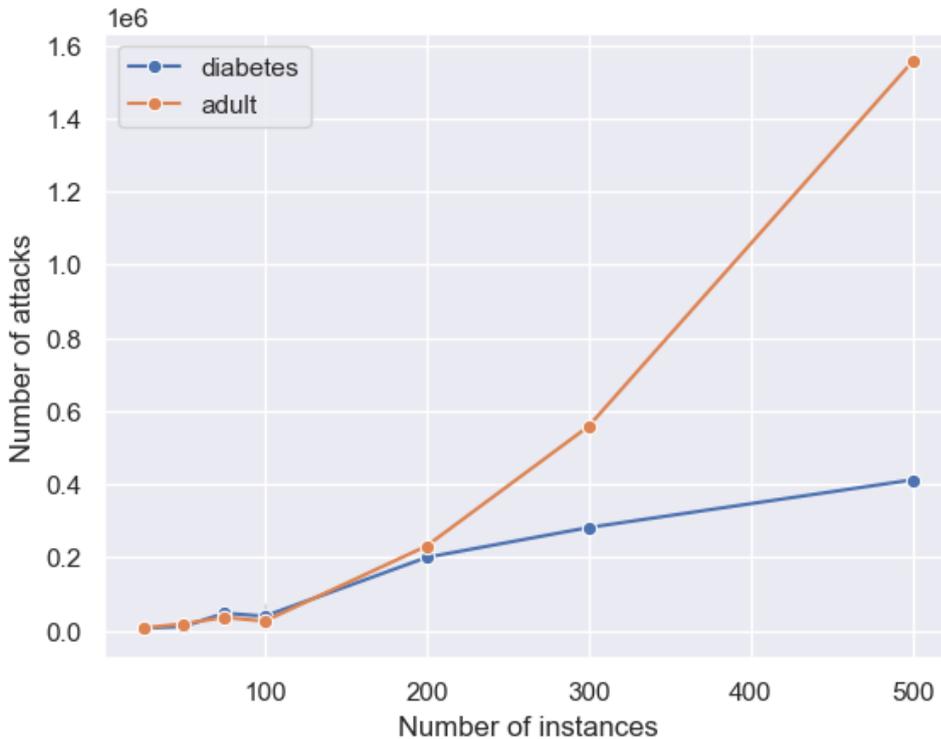


Figure 5.4: Number of attacks (in millions) on arguments enumerated using portions of the training set

per arguments in several settings. In fig. 5.6 on page 96, we study the AS for 25, 50, 75, 100, 200, 300 and 500 instances from the training dataset. Although we don't show the outliers for the box plots, we note that in this setting, no argument is left unattacked. For fig. 5.7 on page 97, we repeat the experiment on the 8 portions of the full input space.

This experiment shows for each dataset, the number of times an argument can be inconsistent with another argument. The results are similar to the global number of attacks per instances. We can see that for small initial datasets, the number of attacks keeps increasing. The number of attacks increases until reaching its maximum (located before 1/8 of the full-input space) and then slowly decreases to reach zero attack per argument.

5.4 Extension Enumeration

In argumentation, (Dung, 1995) introduced a means to evaluate conflicts between arguments using an extension based semantics. This semantics is explicated in definition 6 on page 51.

From the argumentation system, it is possible to enumerate extensions. This is the most challenging computational problem of this framework. By enumerating all extensions, it is possible to find the most relevant sets of conflict-free sets of arguments to explain decisions. We will first present the problem of listing all stable extension adapted to our case, then we will present the selected solution.

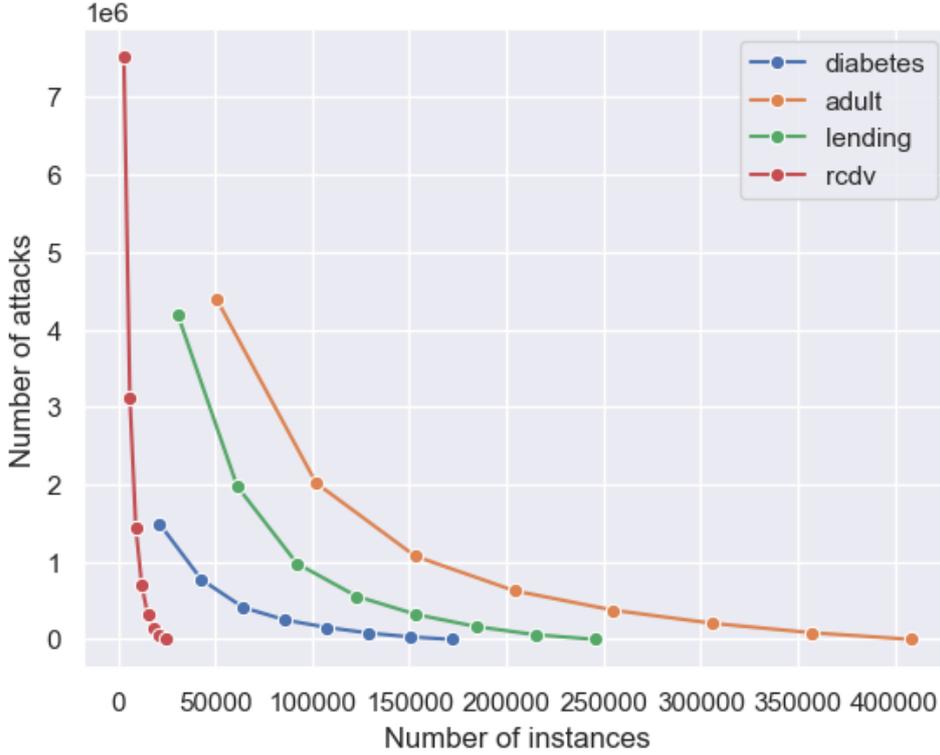


Figure 5.5: Number of attacks (in millions) on arguments enumerated using portions of the input space

Now that the Argument System has been computed from the dataset \mathcal{Y} our goal is to find sets of argument that are coherent with each other. *Coherence* is one out of two important guarantees for this explainer. We recall the definition of *Coherence* in Principle principle 2 on page 68

Principle 3 (Coherence) *A refined plausible explainer \mathbf{g} satisfies coherence iff for any classifier \mathbf{R} , any theory \mathbf{T} , any $I, I' \in \mathcal{Y}$, if $\mathbf{R}(I) \neq \mathbf{R}(I')$, then $\forall L \in \mathbf{g}(I), \forall L' \in \mathbf{g}(I'), L \cup L'$ is inconsistent.*

In other words, a coherent position is a set of arguments which is conflict-free. Thus, an explainer would be coherent if and only if all returned explanation would belong to the same coherent position. In argumentation, this position is also called a *naive extension* or in our case, because the attack relation is symmetric, a *stable extension*.

The next phase of the explanation process consists in enumerating extensions of the stable semantics. Later, from the set of all extensions, a collection of extensions is then selected according to the selection function and aggregated according to the inference rule. This selection function and the inference rule can be chosen according to the user's needs. We detail these extensions in section 5.5.

We remind the reader of the Stable Semantics is in definition 24 on the next page.

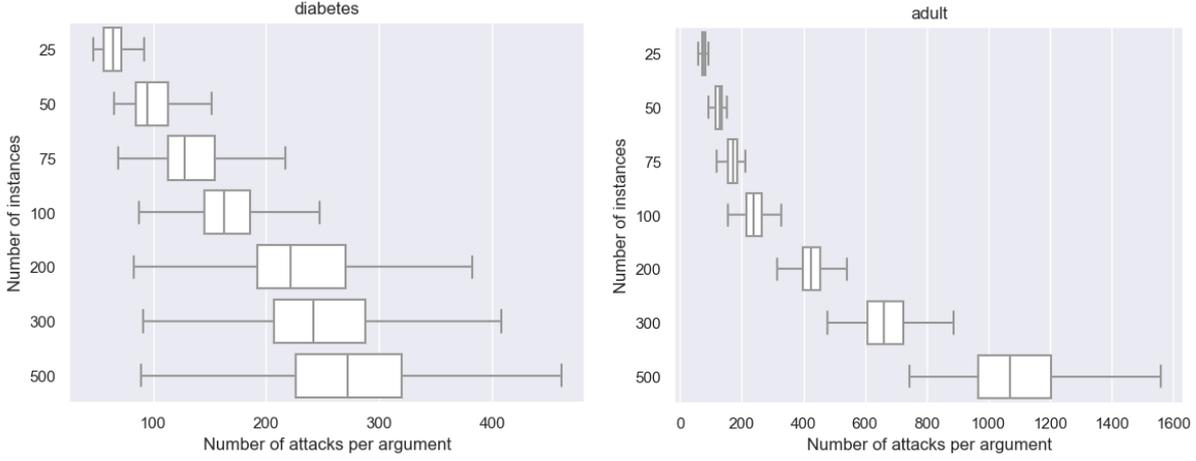


Figure 5.6: Number of attacks per arguments enumerated using a portion of the training set

Definition 24 (Stable Semantics) Let $AS = \langle \text{Arg}(\mathcal{Y}), \mathcal{R} \rangle$ and $\mathcal{E} \subseteq \text{Arg}(\mathcal{Y})$.

- \mathcal{E} is conflict-free iff $\nexists a, b \in \mathcal{E}$ such that $(a, b) \in \mathcal{R}$.
- \mathcal{E} is a stable extension iff it is conflict-free and $\forall a \in \text{Arg}(\mathcal{Y}) \setminus \mathcal{E}, \exists b \in \mathcal{E}$ such that $(b, a) \in \mathcal{R}$.

Let $\sigma(AS)$ denote the set of all stable extensions of AS .

In this section we explain the problem of enumerating and processing the extension to return a coherent set of explanations.

5.4.1 Extension Enumeration problem equivalence in graph theory

The extension enumeration problem is a well-known problem in Argumentation as well as in graph theory. The corresponding problem in Graph theory is listing all Maximal Independent sets. Independent sets are sets of vertices in a graph, no two of which are adjacent. The maximality, or inclusion-wise maximality, means that there are no superset of a maximal set that is also an independent set. This problem has been studied for many years in the scope of graph coloring, which is a well known NP-Complete problem. The listing of all maximal independent sets is a subroutine of this algorithm. This is also the complementary problem of listing all maximal cliques in a graph. In (Eppstein, 2000), the authors show an upper bound for this problem. The difficulty of this problem does not stem from the search of maximal independent sets but rather from their number. The number of maximal independent sets in an n -vertex graph is bounded by the limit $3^{n/3}$, found in (Moon and Moser, 1965).

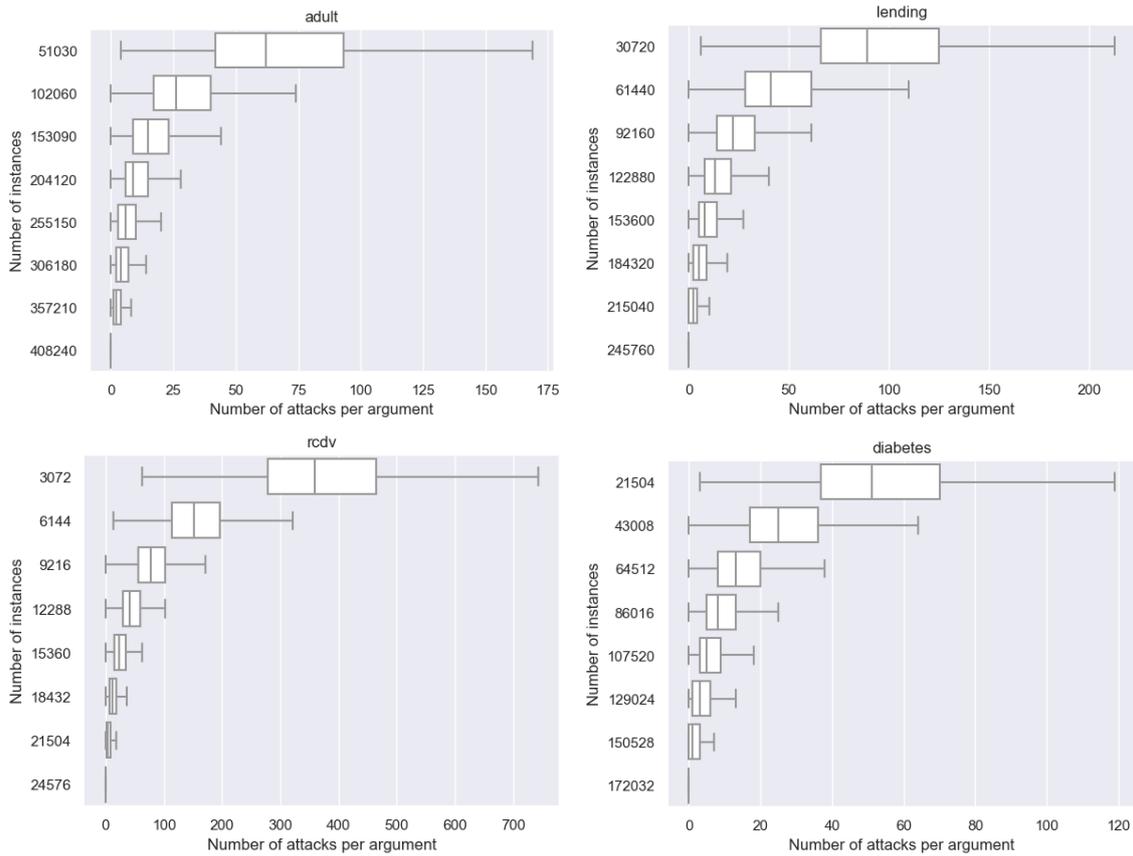


Figure 5.7: Number of attacks per arguments enumerated using a portion of the input space

5.4.2 Implementation: Solver choice

We found 4 alternatives for the enumeration of extensions.

- **NetworkX:** This library implements an algorithm for listing all maximal cliques in a graph proposed in (Zhang et al., 2005).
- **NetworkKit:** This library implements a solution proposed in (Eppstein and Strash, 2011) for listing maximal cliques that is optimized to work with large sparse graphs.
- **graph-tool:** This library implements the first algorithm proposed in (Bron and Kerbosch, 1973) for listing all maximal cliques, with a few improvements.
- **Aspartix:** Aspartix is a tool based on the clingo answer set solver.

NetworkX is an efficient python library for graphs. Aspartix is an implementation for computational argumentation. It was presented several times for the ICCMA challenge and the results are always very relevant. Since it is an older project, it is well documented and easier to set up. In order to make our implementation versatile, we decided to let the user able to choose between two solutions. *NetworkX* is an ‘easy’ solution for the

$ \mathcal{Y} $	$ \text{Arg} $	$ \mathcal{R} $	atks/node	nx	nk	gt	aspartix	$ \sigma(AS) $
50	97	818	16.86	1.28	0.5	1.32	2.618	156022
60	91	715	15.7	1.51	0.56	1.44	2.812	178491
70	112	1006	17.96	13.6	5.6	15.4	26	1568822
80	112	949	16.98	20.2	9.16	25.98	43.696	2777435
90	116	930	16.03	51	22.9	63.7	114.47	7317620
100	121	927	15.3	106.5	ME	145	259	16174014
150	138	1050	15.2	912		1260	2076	

Table 5.7: Performances comparaisn (in seconds) for solvers, NetworkX (nx), NetworKit (nk), graph-tool (gt) and Aspartix

framework. It is efficient and reliable, there is also no treatment necessary as the AS is already encoded with this library. Using *NetworkX* is a good solution for testing with reasonably small argumentation systems. However, with bigger graphs, we encounter memory issues. Thus, the best alternative is *Aspartix with clingo*. This is an efficient implementation and the extensions can be listed in a text file without memory issues. However, it is necessary to extract the argumentation system and solve the enumeration problem outside of the python framework. In the end, both solutions allow the user to create an extension generator that will be available for the explanation function section 5.5.

5.4.3 Stable Extensions Enumeration solutions: Comparative study

In order to have a better feel on how each solution handles our problem, we gave the argumentation system from our simplest dataset: Titanic. We made each solution complete the task on a range of Argumentation Systems generated with different number of instances. The results are gathered in table 5.7.

NetworKit (nk) takes excessive an amount of memory (Memory Error) for only 100 instances. NetworkX is also confronted to Memory Errors but for bigger Argumentation Systems. NetworkX seems to be the most fitting and practical option in our test setup. However, Aspartix is more promising for bigger experiments because it does not seem to have a memory cap. Graph-tool’s performances are roughly the same as NetworkX but a bit slower on this task.

5.4.3.1 The pipeline’s bottleneck

The coherence is guaranteed by the analysis of all extensions. Depending on the chosen selection function (section 4.6) we need to compute the coverage of each extension.

The coverage computation for one extension is in $O(n)$ with n the number of arguments in the class. But either for the coverage by instance or the coverage by class, the cost is

multiplied by the large number of extension, making the implementation hard to use for large ASs.

5.5 Explanation Functions

In section 4.6, we defined a collection of parameterized explanation functions. These functions are implemented in this framework. Given the set of all extensions given by the solver (section 5.4), the explanation function returns a set of arguments. This set of arguments holds the arguments that will be used for explaining the instances of the dataset. They are chosen by the user using two parameters.

- **Selection function:** The selection functions are defined in definition 20 on page 73. The selection function is the policy for choosing which extensions are relevant for explaining our dataset.
- **Inference rule:** The inference rules are defined in definition 21 on page 74. The inference rule decides how to aggregate the extensions selected by the selection function.

5.5.1 Selection Functions

The selection of extension is made by enumerating all extensions and keeping the extensions according to the policy *alpha*. In the following list, we give an intuition for all selection policies.

- The **Max** selection function takes all possible stable extension from the argumentation system.
- The **Card** function takes the extension that has the maximum number of arguments. If several extensions have the same, maximum, amount of arguments, they are all returned.
- The **Incl_i** takes all extension whose set of covered instances of the dataset is maximal (for set-inclusion).
- The **Card_i** function takes the extensions that have the maximum coverage over the dataset. The coverage is the cardinal of the set of arguments covered by the extension. If there are several extension with the same coverage cardinal, they are all selected.
- The **Incl_c** function takes all extensions which set of covered classes is maximal (for set-inclusion).

- The Card_c function takes extensions that have the largest amount of classes covered by the arguments.
- The Mix selection is given by applying the selection Card_c on the result of another selection Card_i .

5.5.1.1 Selection functions: Implementation

We remarked that the enumeration of extension is an expensive process. As a consequence, the extraction of explanation from the large collection also is a computational challenge. There are two ways to process the extensions. The extensions are fed as a stream to avoid memory overload. For example, it is inefficient or even impossible to compute the intersection or union of millions of sets at the same time. In this subsection, we expose our solution to return the selection of extension for each policy α for $\alpha \in \{\text{Max}, \text{Card}, \text{Card}_i, \text{Card}_c, \text{Incl}_i, \text{Incl}_c\}$.

In the case of Max , no treatment is necessary, we only need to apply the inference rule. In the case Card , we simply iterate over extensions and check the length of the extensions.

For Card_i and Card_c , we use hash tables, generated during the argument enumeration, to retrieve the cov_i or cov_c of each argument in the extension. The cardinals are the length of the union of the all cov_i or cov_c . Thank to the hash tables, we can compute in $O(1)$ the coverage for each argument. The complexity comes from repeating the hash search and aggregating the coverages. With $|ext|$ the length of an extension and m the maximum size of the arguments' coverage, the complexity to compute Card_i for one extension is $O(|ext| * m)$

For Incl_i and Incl_c , the solution is more complex. We explain it with algorithm 6 on the next page. The coverage hash table cov designates the class coverage for Incl_c and the instance coverage for Incl_i . The complexity inside the first for loop depends on the complexity of the cov_{ext} construction (same complexity as for Card_i and Card_c) and the complexity of the nested loop. The complexity of the inner loop is majored by the number of passing through the first loops times the complexity of the two set comparisons ($O(\text{len}(s_1) * \text{len}(s_2))$ in the worst case). We can write the complexity of algorithm 6 on the facing page as in eq. (5.10) with m the maximum size of the arguments' coverage and M the maximum size of the extension's coverage.

$$O\left(\sum_{e=0}^{|\sigma(AS)|} [|\sigma(AS)| * m * e * M^2]\right) = O(|\sigma(AS)|^3 * m * M^2) \quad (5.10)$$

We sum up the complexities of each selection function in table 5.8 on page 102 with $\sigma(AS)$ the set of all stable extensions in the AS, m the maximum size of the arguments'

Algorithm 6 Selection process for $Incl_i$ and $Incl_c$

Input: cov the hash table of coverage per argument.**Input:** $\sigma(AS)$ the set of all extensions.**Input:** cap the maximum number of extension to select.**Output:** List of selected extensions.

```
1:  $coverages_{ext} = dict()$ 
2: for  $ext \in \sigma(AS)$  do
3:   if  $len(coverages_{ext}) > cap$  then
4:     return  $coverages_{ext}$ 
5:   end if
    $coverages_{arg} = \emptyset$ 
6:   for  $arg \in ext$  do
7:     Add  $cov[arg]$  to  $coverages_{arg}$ 
8:   end for
9:    $cov_{ext} = \bigcup(coverages_{arg})$ 
10:   $remove = []$ 
11:   $add_{ext} = Bool(True)$ 
12:  for  $c \in coverages_{ext}$  do
13:    if  $c \subset cov_{ext}$  then
14:      Add  $c$  to  $remove$ 
15:    else if  $cov_{ext} \subset c$  then
16:       $add_{ext} = False$ 
17:      break
18:    end if
19:  end for
20:  for  $e \in remove$  do
21:    Remove  $e$  from  $coverage_{ext}$ 
22:  end for
23:  if  $add_{ext}$  then
24:    Add  $cov_{ext}$  to  $coverages_{ext}$ 
25:  end if
26: end for
27: return  $coverages_{ext}$ 
```

coverage, M the maximum size of the extension's coverage and E the cardinal of the largest extension.

5.5.2 Inference Rules

Inference rules are defined in the definition 21 on page 74. The `apply_inference` function returns the set of arguments that are accepted as explanations. The function takes an inference principle as input to aggregate the different extension that were previously selected. The possible inference principles are given below.

- **universal**: the function returns the intersection of all selected extensions.

Function	$Max / Card$	$Card_i / Card_c$	$Incl_i / Incl_c$
Complexity	$O(\sigma(AS))$	$O(\sigma(AS) * E * m)$	$O(\sigma(AS) ^3 * m * M^2)$

Table 5.8: Complexities of selection functions

Instance	Arguments	Sample explanation
0	0	None
1	5	$\langle \{ \text{Sex=Female, Relationship=Not-in-family, Workclass=State-gov} \}, >\$50k \rangle$
2	8	$\langle \text{Sex=Female, Age='Age > 48.00', Capital Loss=0, Hours per week='Hours per week <= 40.00', Capital Gain=0} \rangle, <\$50k \rangle$

Table 5.9: Sample results for $g^{Card_i, \forall}$. For instance n, we found x coherent arguments. One argument is showcased in the last column.

- **existence**: the parameter returns the union of all selected extension.

5.5.3 Experiments

In this section, we show a panel of explanations that we computed using our explanation functions. Since the coherence is proven for universal inference, we detail the scores in terms of success.

5.5.3.1 Result example: *adult* dataset

In this experiment, the explanation framework is used to explain a dataset of 100 instances on the adult dataset. Once the arguments are generated and the argumentation system defined, the solver enumerates all stable extensions. From the results file of the solver, a generator of all extensions is created and used to make the selection of extensions. Then, the set of selected extensions is aggregated according to the inference principle. Table 5.9 shows a sample of 3 explanations given by the $g^{Card_i, \forall}$ explanation function for instances 0, 1 and 2 of the training set. For the first instance, no coherent explanation was found. For the second and third instances, we found respectively 5 and 8 arguments to explain the instance. One of the argument is proposed on the last column.

5.5.3.2 Orders of magnitude

Now that the full explanation pipeline is built, we are interested in the order of magnitude of the different elements of the framework (instances, arguments, attacks, extensions). We consider one experiment per dataset and only a minimum amount of features (only the white lines of the dataset feature tables are used). In table 5.10 on the next page, we report the number of instances $|\mathcal{Y}|$ in the dataset \mathcal{Y} , the number of attacks $|\mathcal{R}|$ and the number of stable extensions found by the solver $|\sigma(AS)|$.

Dataset	$ \mathcal{Y} $	$ \mathbf{Arg}(\mathcal{Y}) $	$ \mathcal{R} $	$ \sigma(AS) $
adult	100	162	2639	$3.2 \cdot 10^6$
diabetes	50	114	816	$5.6 \cdot 10^6$
titanic	100	121	927	$1.6 \cdot 10^7$
rcdv	50	241	5966	$6.8 \cdot 10^7$
lending	50	111	573	$3.2 \cdot 10^7$

Table 5.10: Orders of magnitude of the number of instances $|\mathcal{Y}|$, the number of arguments $|\mathbf{Arg}(\mathcal{Y})|$, the number of attacks $|\mathcal{R}|$ and the number of extensions $|\sigma(AS)|$ for each experiment.

5.5.3.3 Success

In this subsection, we observe *Success* defined in principle 1 on page 67. Although *success* is a binary principle, it is possible to measure how close an explanation function is to validate it. We use the *coverage* (as defined in section 2.3.2) of the function over the dataset \mathcal{Y} . We compute explanations given by our explanation function $g^{\alpha,\beta}$ and vary the parameters $\alpha \in \{Max, Card, Card_i, Card_c, Incl_i, Incl_c, Mix\}$ and $\beta \in \{\forall, \exists\}$. In table 5.11 on the next page, we sum up the coverage values of each experiment. We also report the number of extensions selected by the α policy (in the # columns). Note that the *Mix* selection function yields the same results as *Card_i* since the *Card_c* selection function selects near all extensions. Indeed, in a binary classification problems, extensions with a high *Card_i* always cover both classes. In the case of g^{Incl_i} , it is important to note that we capped the number of selected extensions to avoid memory errors. As a consequence, the coverage results are not exact. However, the results provide upper (for $\beta = \forall$) and lower (for $\beta = \exists$) bounds for the real coverage values.

By observing the results, we can confirm that:

- $g^{Max,\exists}$ always has perfect coverage, which is a consequence of *Success*, confirming theorem 2 on page 78.
- $g^{card_i,\forall}$ always maximizes the coverage in the case of *Coherent* explanation functions.

Furthermore, we can make a few observations:

- For $g^{Max,\cdot}$, that selects all extensions, the explanation function returns no explanation for the \forall inference. This is due to the fact that all arguments are attacked at least once. In opposition, for the \exists inference, \mathcal{Y} is fully covered. These observations can be validated by proposition 9 on page 75.
- For selections with large amounts of extensions, like $\{Max, Card_c, Incl_c\}$, the functions tend to behave like $g^{Max,\cdot}$. Since almost all extensions seem to be chosen (the number of selected extensions is very close to $\#g^{Max,\cdot}$, it is very unlikely that an

Dataset	adult			diabetes			titanic			rcdv			lending		
$ \mathcal{D} $	100			50			100			50			50		
	\forall	\exists	#												
$g^{Max.}$	0	100	3208169	0	100	5598491	0	100	15854797	0	100	67683419	0	100	31884985
$g^{Card.}$	45	45	1	68	68	1	53	53	1	46	46	1	52	52	1
$g^{Card_i.}$	69	91	36	84	100	5	83	85	10	74	98	170	84	88	6
$g^{Card_c.}$	0	100	3208167	0	100	5598450	0	100	15854795	0	100	67683417	0	100	31884983
$g^{Incl_i.}$	0	100	11572	0	100	22906	0	100	116185	0	100	9216	0	100	76007
$g^{Incl_c.}$	<22	>84	> 2 · 10 ⁵	0	100	> 2 · 10 ⁵	<6	>81	> 2 · 10 ⁵	<6	>90	> 2 · 10 ⁵	<2	>90	> 2 · 10 ⁵
$g^{Mix.}$	69	91	36	84	100	5	83	85	10	74	98	170	84	88	6

Table 5.11: Success: Coverage values

argument remains out of the union (for \exists inference) or in the intersection (in the case of \forall inference).

- We note that $cov_i(Incl_i(\sigma(AS))) = cov_i(Max(\sigma(AS)))$ in all experiments. Is it always true? Let $Incl_i(\Sigma)$, the set of all extensions ϵ which $cov_i(\epsilon)$ is subset-maximal. Σ a set of extensions. Let ϵ^* a new extension, if $cov_i(\epsilon^*)$ is subset-maximal,

$$cov_i\left(\bigcup_{\epsilon \in Incl_i(\Sigma) \cup \epsilon^*}\right) = cov_i\left(\bigcup_{\epsilon \in Incl_i(\Sigma \cup \epsilon^*)}\right) \quad (5.11)$$

On the contrary, if $cov_i(\epsilon^*)$ is not subset-maximal, then $\exists \epsilon' \in \Sigma, cov_i(\epsilon^*) \subset cov_i(\epsilon')$. Thus, because cov_i is distributive,

$$cov_i\left(\bigcup_{\epsilon \in Incl_i(\Sigma) \cup \epsilon^*}\right) = \bigcup_{\epsilon \in Incl_i(\Sigma) \setminus \epsilon'} cov_i(\epsilon) \cup cov_i(\epsilon') \cup cov_i(\epsilon^*) = cov_i\left(\bigcup_{\epsilon \in Incl_i(\Sigma)}\right) \quad (5.12)$$

By induction, with a trivial initialization, we show that it is always true.

5.6 Further experiments

Our experiments focused on analysing the behavior of the concepts introduced in chapter 4 with an arbitrary set of instances. Additionally, we saw that the size of the argumentation system, has an important impact on the number of extensions. Firstly, further experiments could investigate how the input set of instances impacts the creation of arguments. It would be interesting to measure the impact of the distribution of the initial dataset. As ML models are trained on real world data, they are probably more predictable on a real-world distribution. The goal would be to find a dataset that presents less inconsistencies to make the AS more sparse. We could take advantage of this sparsity to compute extensions more efficiently. This research lacks an evaluation of the explanation themselves. The explanations could be compared to either a state-of-the art explanation function such as *Anchors* or to an intrinsically interpretable model such as a random forest classifier from which we can easily extract rules and compare them to our explanations.

5.7 Conclusion

The implementation of the theoretical proposition was challenging mainly because of the complexity stakes of the problem raised. We were able to propose a framework that works on binary classifier with categorical data. And we lead experiments on 5 well-known dataset of the field's literature. The experiments were able to confirm a few observations that we raised in the theory as well as showing behavior of the argumentation system on different datasets and input space sizes. There are still improvements to be made. For example, like the ICCMA's dynamic track section 3.3 shows, extensions can be computed dynamically. With explanations are updated as new inputs are predicted and added to the Argumentation Framework, the global process could be more adapted to real-world use and avoid running the whole pipeline for new sets of instances.

Chapter 6

Conclusion

In this thesis, we addressed the issue of explaining machine learning models. Firstly, we described the landscape of explainability in AI and focused on a particular aspect: model-agnostic explainers for classifiers. Secondly, we presented the basics of abstract argumentation. Our goal being to use it as a tool to build explanations from a small set of data labeled by the prediction model. Based on this abstract framework we formally defined two essential principles that explainers should guarantee: **coherence** and **success**. To showcase these principles, we propose a family of parameterizable explanation functions which are tailored to guarantee one or the other property. Unfortunately, we proved that no refined plausible explainer can satisfy both principles. Finally, an experimental implementation confronted the theoretical framework to five real-world datasets. In addition to empirically verifying the explanation framework, the experiments provided interesting results. In this last chapter, we summarize the results of this thesis and try to answer the initial research questions.

How can argumentation can be leveraged to explain black-box models?

To achieve these results, we leveraged argumentation to build a strong formal framework and define precisely our metrics. We saw that argumentation was a powerful tool to study decisions under inconsistent information. Thanks to the modular characteristic of argumentation, we were able to find a policy that fit our goals. We proposed a family of parameterizable explanation functions that could be chosen according to the goal of the user. We showed that the framework could be applied on benchmark datasets to provide explanations on the initial dataset. We also showed that, thanks to the framework's simplicity, one can use intermediate results such as the argumentation system as a baseline for further studies.

What desirable properties of explanations can we guarantee?

In this thesis, we addressed the issue of coherence in explanations given by explainers. We defined **coherence** as the impossibility that two explanation could contradict each other

in any situation. We also proposed the **success** that is a guarantee to yield explanations. These strong guarantees are valid on any classifier, theory or initial dataset. Thanks to the experiments, we showed the importance **coherence** in explanation systems and that it was improbable to propose globally coherent explanations without considering the whole input space. By trading off the coherence guarantees, it is possible to achieve better **success**. Thanks to the modularity of our framework, it is possible to adapt the explanation function to different contexts and goals. The user can either prioritize the explanation of individual instances, or prefer explanation for whole classes.

What are the computational stakes of providing truthful explanations?

As expected with previous knowledge and confirmed by our experimentations, the method is not scalable. There are two steps that require a lot of resources. The first one is the generation of arguments. This process has a complexity exponential in the number of features. The other part of the process that is very costly is the enumeration of all extensions. This problem is well-known in the literature and researchers are actively working on improving techniques for this task.

As a conclusion, we proposed a novel approach to explainability on Machine Learning models. Our approach is a model-agnostic explainer that provides strong guarantees on the explanations. The framework provides abductive explanations on a the set of instances defined by the user and this set defines the extent of the formal guarantees. The user can decide which explanation function, along with what guarantees he needs for his task. Moreover, the framework is based on argumentation. This field of research is very promising and has a lot to offer for explaining and providing a formal setting to define well defined guarantees on explanations.

Discussion and Perspectives: The great challenge of explaining black-box models comes from the problem of building knowledge from the black-box model by probing it with instances. Today, perturbation based distributions are widely used to probe the model around the instance to explain. In our setting, we decided to use a fixed dataset that will serve as knowledge base to our explanations. The argumentation framework allowed us to generate explanations based on the knowledge given by the dataset. As a consequence, this dataset represents the extent of the guarantees of our principles success and coherence.

In chapter 4 and chapter 5 we showed that scalability is a major limitation of our framework. We saw previously that the amount of extensions was incredibly high and the enumeration was a costly task. The enumeration of arguments is also a task that require a lot of resources.

Making the framework work dynamically could be an interesting upgrade. This would ease both problems that we mentioned previously. First, with a dynamic framework, it would be possible to extend the initial dataset that guarantees **success** and **coherence** as the user generates the explanations. The computational argument competition ICCMA opened a dynamic track in 2019, allowing the solvers to generate extensions as the argumentation system is altered. With a dynamic framework the generation of explanation could be more efficient by using the previous rules.

Although our framework is model-agnostic, it only works on categorical classifiers. There already exists several limitations to this type of model. Firstly, the explanations that are generated only work with meaningful features, and secondly, data structures with many features make explanations in the form of abductive explanations less relevant for explaining to human stakeholders since the explanations would contain too many conditions.

We think that the framework could be used as a wrapper to other rule-based explanation functions. For example, using Anchors (Ribeiro et al., 2018) to generate arguments and our framework to study the coherence and success, the framework could detect attacks of arguments and certify *coherence*.

The acceptability of arguments is a binary concept. However, argumentation proposes different semantics such as ranking or weighting semantics. We could for example combine the uncertainty of models to weight arguments. The use of these semantics could make the framework propose more nuanced explanations and less complex generation process, at the cost of coherence.

Bibliography

- All Lending Club loan data, a. URL <https://www.kaggle.com/datasets/wordsofthewise/lending-club>. Cited on page 83.
- Diabetes Dataset, b. URL <https://www.kaggle.com/datasets/mathchi/diabetes-data-set>. Cited on page 85.
- S. Agarwal, S. Jabbari, C. Agarwal, S. Upadhyay, S. Wu, and H. Lakkaraju. Towards the Unification and Robustness of Perturbation and Gradient Based Explanations. In *Proceedings of the 38th International Conference on Machine Learning*, pages 110–119. PMLR, July 2021. URL <https://proceedings.mlr.press/v139/agarwal21c.html>. ISSN: 2640-3498. Cited on page 42.
- B. Alcaraz. AJAR: An Argumentation-based Judging Agents Framework for Ethical Reinforcement Learning. 2023. Cited on page 60.
- D. Alvarez-Melis and T. S. Jaakkola. On the Robustness of Interpretability Methods, June 2018. URL <http://arxiv.org/abs/1806.08049>. arXiv:1806.08049 [cs, stat]. Cited on page 40.
- M. Alviano. The pyglaf argumentation reasoner (ICCMA2021). *CoRR*, abs/2109.03162, 2021. URL <https://arxiv.org/abs/2109.03162>. arXiv: 2109.03162. Cited on page 59.
- L. Amgoud. A formal framework for handling conflicting desires. In *Proceedings of the 7th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'03)*, pages 360–365. Springer, 2003. Cited on page 59.
- L. Amgoud. Explaining Black-box Classification Models with Arguments. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 791–795, Washington, DC, USA, Nov. 2021a. IEEE. ISBN 978-1-66540-898-1. doi: 10.1109/ICTAI52525.2021.00126. URL <https://ieeexplore.ieee.org/document/9643355/>. Cited on pages 15, 61, and 68.
- L. Amgoud. Non-monotonic Explanation Functions. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 16th European Conference, ECSQARU 2021*,

- Prague, Czech Republic, September 21–24, 2021, Proceedings*, pages 19–31, Berlin, Heidelberg, Sept. 2021b. Springer-Verlag. ISBN 978-3-030-86771-3. doi: 10.1007/978-3-030-86772-0_2. URL https://doi.org/10.1007/978-3-030-86772-0_2. Cited on pages 16, 61, 62, 77, and 78.
- L. Amgoud and J. Ben-Naim. Ranking-Based Semantics for Argumentation Frameworks. In W. Liu, V. S. Subrahmanian, and J. Wijsen, editors, *Scalable Uncertainty Management*, Lecture Notes in Computer Science, pages 134–147, Berlin, Heidelberg, 2013. Springer. ISBN 978-3-642-40381-1. doi: 10.1007/978-3-642-40381-1_11. Cited on page 49.
- L. Amgoud and J. Ben-Naim. Axiomatic Foundations of Explainability. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 636–642, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3. doi: 10.24963/ijcai.2022/90. URL <https://www.ijcai.org/proceedings/2022/90>. Cited on page 67.
- L. Amgoud and C. Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. *International Journal of Automated Reasoning*, 29 (2):125–169, 2002a. Cited on page 59.
- L. Amgoud and C. Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Mathematics and Artificial Intelligence*, 34:197–216, 2002b. Cited on page 53.
- L. Amgoud and S. Kaci. An argumentation framework for merging conflicting knowledge bases. *International Journal of Approximate Reasoning*, 45:321–340, 2007. Cited on page 59.
- L. Amgoud and S. Parsons. An argumentation framework for merging conflicting knowledge bases. In *Proceedings of the 8th European Conference on Logics in Artificial Intelligence (JELIA'02)*, pages 27–37, 2002. Cited on page 59.
- L. Amgoud and H. Prade. Reaching agreement through argumentation: A possibilistic approach. In *Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning (KR'04)*, pages 175–182. AAAI Press, 2004. Cited on page 60.
- L. Amgoud and H. Prade. Explaining qualitative decision under uncertainty by argumentation. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06)*, pages 219–224. AAAI Press, 2006. Cited on page 59.
- L. Amgoud and M. Serrurier. Agents that argue and explain classifications. *Autonomous Agents and Multi-Agent Systems*, 16(2):187–209, Apr. 2008. ISSN 1573-7454. doi: 10.

- 1007/s10458-007-9025-6. URL <https://doi.org/10.1007/s10458-007-9025-6>. Cited on page 60.
- L. Amgoud, N. Maudet, and S. Parsons. Modelling dialogues using argumentation. In *Proceedings of the 4th International Conference on MultiAgent Systems (ICMAS'00)*, pages 31–38. ACM Press, 2000a. Cited on page 60.
- L. Amgoud, S. Parsons, and N. Maudet. Arguments, dialogue, and negotiation. In *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI'00)*, pages 338–342. IOS Press, 2000b. Cited on page 60.
- L. Amgoud, P. Muller, and H. Trenquier. Argument-based Explanation Functions. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '23*, pages 2373–2375, Richland, SC, May 2023a. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-9432-1. Cited on page 16.
- L. Amgoud, P. Muller, and H. Trenquier. Leveraging Argumentation for Generating Robust Sample-based Explanations. page 3104. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023b. doi: 10.24963/ijcai.2023/346. URL <https://laas.hal.science/hal-04096982>. Cited on page 17.
- K. Atkinson, T. Bench-Capon, and P. McBurney. Justifying practical reasoning. In *Proceedings of the Fourth Workshop on Computational Models of Natural Argument (CMNA 2004)*, pages 87–90, 2004. Cited on page 59.
- G. Audemard, S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, and P. Marquis. On Preferred Abductive Explanations for Decision Trees and Random Forests. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 643–650, Vienna, Austria, July 2022. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3. doi: 10.24963/ijcai.2022/91. URL <https://www.ijcai.org/proceedings/2022/91>. Cited on pages 15, 16, and 61.
- S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7):e0130140, July 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0130140. URL <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140>. Publisher: Public Library of Science. Cited on pages 26 and 27.
- D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to Explain Individual Classification Decisions. *The Journal of Machine Learning Research*, 11:1803–1831, Aug. 2010. ISSN 1532-4435. Cited on pages 25 and 26.

- P. Baroni, M. Giacomin, and G. Guida. Scc-recursiveness: a general schema for argumentation semantics. *Artificial Intelligence Journal*, 168:162–210, 2005. Cited on page 55.
- A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, June 2020. ISSN 15662535. doi: 10.1016/j.inffus.2019.12.012. URL <https://linkinghub.elsevier.com/retrieve/pii/S1566253519308103>. Cited on pages 23 and 39.
- D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network Dissection: Quantifying Interpretability of Deep Visual Representations. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 3319–3327. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.354. Cited on page 27.
- V. Bellotti and K. Edwards. Intelligibility and Accountability: Human Considerations in Context-Aware Systems. *Human-Computer Interaction*, 16(2-4):193–212, Dec. 2001. ISSN 0737-0024, 1532-7051. doi: 10.1207/S15327051HCI16234_05. URL https://www.tandfonline.com/doi/full/10.1207/S15327051HCI16234_05. Cited on page 39.
- T. J. M. Bench-Capon and P. E. Dunne. Argumentation in artificial intelligence. *Artificial Intelligence*, 171(10):619–641, July 2007. ISSN 0004-3702. doi: 10.1016/j.artint.2007.05.001. URL <https://www.sciencedirect.com/science/article/pii/S0004370207000793>. Cited on page 45.
- P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artificial Intelligence Journal*, 128 (1-2):203–235, 2001. Cited on page 59.
- P. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008a. Cited on page 59.
- P. Besnard and A. Hunter. *Elements of Argumentation*, 2008b. URL <https://mitpress.mit.edu/9780262026437/elements-of-argumentation/>. Cited on page 49.
- K. Bhavsar, A. Abugabah, J. Singla, A. AlZubi, A. Bashir, and Nikita. A Comprehensive Review on Medical Diagnosis Using Machine Learning. *Computers, Materials & Continua*, 67(2):1997–2014, 2021. ISSN 1546-2218, 1546-2226. doi: 10.32604/cmc.2021.014943. URL <https://www.techscience.com/cmc/v67n2/41354>. Publisher: Tech Science Press. Cited on page 14.
- O. Biran and C. Cotton. Explanation and justification in machine learning: A survey. In *IJCAI-17 workshop on explainable AI (XAI)*, volume 8, pages 8–13, 2017. URL

- http://www.cs.columbia.edu/~orb/papers/xai_survey_paper_2017.pdf. Issue: 1.
Cited on page 22.
- O. Biran and K. McKeown. Human-Centric Justification of Machine Learning Predictions. pages 1461–1467, 2017. URL <https://www.ijcai.org/proceedings/2017/202>. Cited on page 16.
- E. Black and A. Hunter. A generative inquiry dialogue system. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multi-Agents systems (AAMAS'07)*, 2007. Cited on page 60.
- B. Bonet and H. Geffner. Arguing for decisions: A qualitative model of decision making. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI'96)*, pages 98–105, 1996. Cited on page 59.
- R. Brena, C. Chesñevar, and J. Aguirre. Argumentation-supported information distribution in a multiagent system for knowledge management. In *2nd International Workshop on Argumentation in Multiagent Systems (ArgMAS 2005)*, pages 26–33, 2005. Cited on page 59.
- C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, Sept. 1973. ISSN 0001-0782. doi: 10.1145/362342.362367. URL <https://dl.acm.org/doi/10.1145/362342.362367>. Cited on page 97.
- N. Burkart and M. F. Huber. A Survey on the Explainability of Supervised Machine Learning. *Journal of Artificial Intelligence Research*, 70:245–317, May 2021. ISSN 1076-9757. doi: 10.1613/jair.1.12228. URL <https://dl.acm.org/doi/10.1613/jair.1.12228>. Cited on page 14.
- M. Caminada. Semi-stable semantics. In *Proceedings of the 1st International Conference on Computational Models of Argument, COMMA '06*, pages 121–130, 2006. Cited on page 55.
- M. W. A. Caminada, W. A. Carnielli, and P. E. Dunne. Semi-stable semantics. *Journal of Logic and Computation*, 22(5):1207–1254, Oct. 2012. ISSN 0955-792X, 1465-363X. doi: 10.1093/logcom/exr033. URL <https://academic.oup.com/logcom/article-lookup/doi/10.1093/logcom/exr033>. Cited on page 58.
- C. Cayrol and M. C. Lagasquie-Schiex. Graduality in Argumentation. *Journal of Artificial Intelligence Research*, 23:245–297, Mar. 2005. ISSN 1076-9757. doi: 10.1613/jair.1411. URL <http://arxiv.org/abs/1107.0045>. arXiv:1107.0045 [cs]. Cited on page 49.

- F. Cerutti, M. Giacomin, and M. Vallati. ArgSemSAT: Solving argumentation problems using SAT. *Proceedings of the 5th International Conference on Computational Models of Argument*, 266:455–456, Jan. 2014. Cited on page 59.
- C. Chesnevar and G. Simari. Computational models for argumentation in mas. *Tutorial in EASSS'2005*, 2005. Cited on page 48.
- D. Cian, J. van Gemert, and A. Lengyel. Evaluating the performance of the LIME and Grad-CAM explanation methods on a LEGO multi-label image classification task, Aug. 2020. URL <https://arxiv.org/abs/2008.01584v1>. Cited on pages 34 and 43.
- O. Cocarascu and A. Stylianou. Data-Empowered Argumentation for Dialectically Explainable Predictions. *Santiago de Compostela*, 2020. Cited on page 60.
- M. Cooper and L. Amgoud. Abductive Explanations of Classifiers Under Constraints: Complexity and Properties. Sept. 2023. ISBN 978-1-64368-436-9. doi: 10.3233/FAIA230305. Cited on page 67.
- M. C. Cooper and J. Marques-Silva. On the Tractability of Explaining Decisions of Classifiers. In L. D. Michel, editor, *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*, volume 210 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-211-2. doi: 10.4230/LIPIcs.CP.2021.21. URL <https://drops.dagstuhl.de/opus/volltexte/2021/15312>. ISSN: 1868-8969. Cited on pages 16, 61, 64, and 65.
- S. Coste-Marquis, C. Devred, and P. Marquis. Symmetric argumentation frameworks. In *Proceedings of the European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU*, pages 317–328, 2005. Cited on pages 72, 75, and 76.
- M. Craven and J. Shavlik. Extracting Tree-Structured Representations of Trained Networks. In *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1995. URL https://proceedings.neurips.cc/paper_files/paper/1995/hash/45f31d16b1058d586fc3be7207b58053-Abstract.html. Cited on page 27.
- W. Cukierski. Titanic - Machine Learning from Disaster, 2012. URL <https://kaggle.com/competitions/titanic>. Cited on page 82.
- A. Darwiche and A. Hirth. On the Reasons Behind Decisions. In G. D. Giacomo, A. Catalá, B. Dilkina, M. Milano, S. Barro, A. Bugarín, and J. Lang, editors, *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th*

- Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 712–720. IOS Press, 2020. doi: 10.3233/FAIA200158. Cited on pages 15, 16, 61, and 64.
- A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das. Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives. 2018. Cited on pages 15 and 16.
- Y. Dimopoulos, S. Dzeroski, and A. Kakas. Integrating Explanatory and Descriptive Learning in ILP. 1997. Cited on page 64.
- F. Doshi-Velez and B. Kim. Towards A Rigorous Science of Interpretable Machine Learning, Mar. 2017. URL <http://arxiv.org/abs/1702.08608>. arXiv:1702.08608 [cs, stat]. Cited on pages 15, 38, 39, and 40.
- P. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence Journal*, 171:642–674, 2007a. Cited on page 55.
- P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–357, Sept. 1995. ISSN 0004-3702. doi: 10.1016/0004-3702(94)00041-X. URL <https://www.sciencedirect.com/science/article/pii/000437029400041X>. Cited on pages 17, 49, 50, 51, 52, 53, 55, 60, 62, 72, and 94.
- P. M. Dung, P. Mancarella, and F. Toni. Computing ideal sceptical argumentation. *Artificial Intelligence*, 171(10):642–674, July 2007b. ISSN 0004-3702. doi: 10.1016/j.artint.2007.05.003. URL <https://www.sciencedirect.com/science/article/pii/S000437020700080X>. Cited on page 58.
- P. E. Dunne. Computational properties of argument systems satisfying graph-theoretic constraints. *Artificial Intelligence*, 171(10):701–729, July 2007. ISSN 0004-3702. doi: 10.1016/j.artint.2007.03.006. URL <https://www.sciencedirect.com/science/article/pii/S0004370207000537>. Cited on page 58.
- P. E. Dunne and T. J. M. Bench-Capon. Coherence in finite argument systems. *Artificial Intelligence*, 141(1–2):187–203, 2002. Cited on page 55.
- W. Dvořák, M. Järvisalo, J. P. Wallner, and S. Woltran. Complexity-sensitive decision procedures for abstract argumentation. *Artificial Intelligence*, 206:53–78, Jan. 2014. ISSN 0004-3702. doi: 10.1016/j.artint.2013.10.001. URL <https://www.sciencedirect.com/science/article/pii/S0004370213001069>. Cited on page 59.

- W. Dvořák, A. Rapberger, J. P. Wallner, and S. Woltran. ASPARTIX-V19 - An Answer-Set Programming Based System for Abstract Argumentation. In A. Herzig and J. Kontinen, editors, *Foundations of Information and Knowledge Systems*, Lecture Notes in Computer Science, pages 79–89, Cham, 2020. Springer International Publishing. ISBN 978-3-030-39951-1. doi: 10.1007/978-3-030-39951-1_5. Cited on page 59.
- D. Eppstein. Small Maximal Independent Sets and Faster Exact Graph Coloring, Nov. 2000. URL <http://arxiv.org/abs/cs/0011009>. arXiv:cs/0011009. Cited on page 96.
- D. Eppstein and D. Strash. Listing All Maximal Cliques in Large Sparse Real-World Graphs, Mar. 2011. URL <http://arxiv.org/abs/1103.0318>. arXiv:1103.0318 [cs]. Cited on page 97.
- B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline. Machine Learning for Medical Imaging. *RadioGraphics*, 37(2):505–515, Mar. 2017. ISSN 0271-5333. doi: 10.1148/rg.2017160130. URL <https://pubs.rsna.org/doi/abs/10.1148/rg.2017160130>. Publisher: Radiological Society of North America. Cited on page 14.
- T. Fel and D. Vigouroux. Representativity and Consistency Measures for Deep Neural Network Explanations. Sept. 2020. URL <https://hal.science/hal-02930949>. Cited on pages 39 and 40.
- J. Ferreira, M. d. S. Ribeiro, R. Gonçalves, and J. Leite. Looking Inside the Black-Box: Logic-based Explanations for Neural Networks. *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, 19(1):432–442, July 2022. ISSN 2334-1033. doi: 10.24963/kr.2022/45. URL <https://proceedings.kr.org/2022/45/>. Conference Name: Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning. Cited on page 15.
- R. Fong and A. Vedaldi. Net2Vec: Quantifying and Explaining how Concepts are Encoded by Filters in Deep Neural Networks, Mar. 2018. URL <http://arxiv.org/abs/1801.03454>. arXiv:1801.03454 [cs, stat]. Cited on page 27.
- J. Fox and P. McBurney. Decision making by intelligent agents: logical argument, probabilistic inference and the maintenance of beliefs and acts. In *Proceedings 9th International Workshop on Non-Monotonic Reasoning (NMR'2002)*, 2002. Cited on page 59.
- J. Fox and S. Parsons. On using arguments for reasoning about actions and values. In *Proceedings of the AAAI Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning, Stanford*, 1997. Cited on page 59.

- A. Garcia and G. Simari. Defeasible logic programming: an argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004. Cited on page 59.
- M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. *Proceedings of ICLP’88*, MIT Press, 1990. Cited on page 53.
- A. Ghorbani, A. Abid, and J. Zou. Interpretation of Neural Networks Is Fragile. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):3681–3688, July 2019. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v33i01.33013681. URL <https://ojs.aaai.org/index.php/AAAI/article/view/4252>. Cited on page 27.
- L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining Explanations: An Overview of Interpretability of Machine Learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89, Turin, Italy, Oct. 2018. IEEE. ISBN 978-1-5386-5090-5. doi: 10.1109/DSAA.2018.00018. URL <https://ieeexplore.ieee.org/document/8631448/>. Cited on pages 23 and 24.
- B. Goodman and S. Flaxman. European Union regulations on algorithmic decision-making and a ”right to explanation”. *AI Magazine*, 38(3):50–57, Sept. 2017. ISSN 0738-4602, 2371-9621. doi: 10.1609/aimag.v38i3.2741. URL <http://arxiv.org/abs/1606.08813>. arXiv:1606.08813 [cs, stat]. Cited on page 15.
- T. Gordon and N. Karacapilidis. The zeno argumentation framework. In *Proceedings of the sixth international conference on Artificial intelligence and law*, pages 10 – 18. ACM Press, 1997. Cited on page 59.
- G. Governatori, M. Maher, G. Antoniou, and D. Billington. Argumentation semantics for defeasible logic. *Journal of Logic and Computation*, 14(5):675–702, 2004. Cited on page 59.
- R. Gozalo-Brizuela and E. C. Garrido-Merchan. ChatGPT is not all you need. A State of the Art Review of large Generative AI models, Jan. 2023. URL <http://arxiv.org/abs/2301.04655>. arXiv:2301.04655 [cs]. Cited on page 14.
- A. Gramegna and P. Giudici. SHAP and LIME: An Evaluation of Discriminative Power in Credit Risk. *Frontiers in Artificial Intelligence*, 4, 2021. ISSN 2624-8212. URL <https://www.frontiersin.org/articles/10.3389/frai.2021.752558>. Cited on pages 37 and 43.
- R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti. A Survey Of Methods For Explaining Black Box Models, June 2018. URL <http://arxiv.org/abs/1802.01933>. arXiv:1802.01933 [cs]. Cited on page 22.
- C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*,

- pages 1321–1330. PMLR, July 2017. URL <https://proceedings.mlr.press/v70/guo17a.html>. ISSN: 2640-3498. Cited on page 28.
- G. Harman. Practical aspects of theoretical rationality. *The Oxford Handbook of Rationality, Al Mele and Piers Rawling, eds. (Oxford: Oxford University Press)*, pages 45–56, 2004. Cited on page 56.
- P. Hase and M. Bansal. Evaluating Explainable AI: Which Algorithmic Explanations Help Users Predict Model Behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5540–5552, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.491. URL <https://www.aclweb.org/anthology/2020.acl-main.491>. Cited on pages 34, 36, 39, and 43.
- D. Heckerman. A Tutorial on Learning with Bayesian Networks. In D. E. Holmes and L. C. Jain, editors, *Innovations in Bayesian Networks: Theory and Applications*, Studies in Computational Intelligence, pages 33–82. Springer, Berlin, Heidelberg, 2008. ISBN 978-3-540-85066-3. doi: 10.1007/978-3-540-85066-3_3. URL https://doi.org/10.1007/978-3-540-85066-3_3. Cited on page 15.
- M. Henne, A. Schwaiger, K. Roscher, and G. Weiß. Benchmarking Uncertainty Estimation Methods for Deep Learning with Safety-Related Metrics. 2020. URL <https://publica.fraunhofer.de/handle/publica/407174>. Cited on page 28.
- J. Hulstijn and L. van der Torre. Combining goal generation and planning in an argumentation framework. In *Proceedings of the 10th Workshop on Non-Monotonic Reasoning (NMR'04)*, 2004. Cited on page 59.
- A. Ignatiev and J. Marques-Silva. SAT-Based Rigorous Explanations for Decision Lists. In C.-M. Li and F. Manyà, editors, *Theory and Applications of Satisfiability Testing – SAT 2021*, Lecture Notes in Computer Science, pages 251–269, Cham, 2021. Springer International Publishing. ISBN 978-3-030-80223-3. doi: 10.1007/978-3-030-80223-3_18. Cited on pages 15 and 16.
- A. Ignatiev, N. Narodytska, and J. Marques-Silva. Abduction-Based Explanations for Machine Learning Models, Nov. 2018. URL <http://arxiv.org/abs/1811.10656>. arXiv:1811.10656 [cs]. Cited on pages 16, 29, and 64.
- A. Ignatiev, N. Narodytska, and J. Marques-Silva. On Relating Explanations and Adversarial Examples. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/7392ea4ca76ad2fb4c9c3b6a5c6e31e3-Abstract.html. Cited on pages 15, 16, 29, and 61.

- A. Jacovi and Y. Goldberg. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.386. URL <https://www.aclweb.org/anthology/2020.acl-main.386>. Cited on page 40.
- H. Johnson and P. Johnson. Explanation facilities and interactive systems. In *Proceedings of the 1st international conference on Intelligent user interfaces, IUI '93*, pages 159–166, New York, NY, USA, Feb. 1993. Association for Computing Machinery. ISBN 978-0-89791-556-4. doi: 10.1145/169891.169951. URL <https://dl.acm.org/doi/10.1145/169891.169951>. Cited on page 23.
- I. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2002. ISBN 978-0-387-95442-4. doi: 10.1007/b98835. URL <http://link.springer.com/10.1007/b98835>. Cited on page 28.
- A. Kakas and P. Moraitis. Adaptive agent negotiation via argumentation. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agents systems (AAMAS'06)*, pages 384–391, 2006. Cited on page 60.
- A. C. Kakas and F. Riguzzi. Abductive concept learning. *New Generation Computing*, 18(3):243–294, Sept. 2000. ISSN 1882-7055. doi: 10.1007/BF03037531. URL <https://doi.org/10.1007/BF03037531>. Cited on page 64.
- E. Kaufmann and S. Kalyanakrishnan. Information Complexity in Bandit Subset Selection. 2013. Cited on page 35.
- B. Kim, R. Khanna, and O. O. Koyejo. Examples are not enough, learn to criticize! Criticism for Interpretability. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://papers.nips.cc/paper_files/paper/2016/hash/5680522b8e2bb01943234bce7bf84534-Abstract.html. Cited on page 22.
- B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. Sayres. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *Proceedings of the 35th International Conference on Machine Learning*, pages 2668–2677. PMLR, July 2018. URL <https://proceedings.mlr.press/v80/kim18d.html>. ISSN: 2640-3498. Cited on page 27.
- P. W. Koh and P. Liang. Understanding Black-box Predictions via Influence Functions, Dec. 2020. URL <http://arxiv.org/abs/1703.04730>. arXiv:1703.04730 [cs, stat]. Cited on page 28.

- R. Kohavi. Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 202–207, Portland, Oregon, Aug. 1996. AAAI Press. Cited on page 83.
- S. Kraus, K. Sycara, and A. Evenchik. *Reaching agreements through argumentation: a logical model and implementation*, volume 104. Journal of Artificial Intelligence, 1998. Cited on page 60.
- M. Kröll, R. Pichler, and S. Woltran. On the Complexity of Enumerating the Extensions of Abstract Argumentation Frameworks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 1145–1152, Melbourne, Australia, Aug. 2017. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-0-3. doi: 10.24963/ijcai.2017/159. URL <https://www.ijcai.org/proceedings/2017/159>. Cited on page 57.
- T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W.-K. Wong. Too much, too little, or just right? Ways explanations impact end users' mental models. In *2013 IEEE Symposium on Visual Languages and Human Centric Computing*, pages 3–10, San Jose, CA, USA, Sept. 2013. IEEE. ISBN 978-1-4799-0369-6. doi: 10.1109/VLHCC.2013.6645235. URL <https://ieeexplore.ieee.org/document/6645235>. Cited on page 39.
- T. Kulesza, M. Burnett, W.-K. Wong, and S. Stumpf. Principles of Explanatory Debugging to Personalize Interactive Machine Learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, pages 126–137, Atlanta Georgia USA, Mar. 2015. ACM. ISBN 978-1-4503-3306-1. doi: 10.1145/2678025.2701399. URL <https://dl.acm.org/doi/10.1145/2678025.2701399>. Cited on page 40.
- A. Kumar, P. S. Liang, and T. Ma. Verified Uncertainty Calibration. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL <https://proceedings.neurips.cc/paper/2019/hash/f8c0c968632845cd133308b1a494967f-Abstract.html>. Cited on page 28.
- J.-M. Lagniez, E. Lonca, and J.-G. Maily. CoQuiAAS: A Constraint-based Quick Abstract Argumentation Solver. Nov. 2015. doi: 10.1109/ICTAI.2015.134. Cited on page 59.
- Z. C. Lipton. The Mythos of Model Interpretability, Mar. 2017. URL <http://arxiv.org/abs/1606.03490>. arXiv:1606.03490 [cs, stat]. Cited on page 39.
- T. Lombrozo. Simplicity and probability in causal explanation. *Cognitive Psychology*, 55(3): 232–257, Nov. 2007. ISSN 00100285. doi: 10.1016/j.cogpsych.2006.09.006. URL <https://linkinghub.elsevier.com/retrieve/pii/S0010028506000739>. Cited on page 39.

- Y. Lou, R. Caruana, and J. Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, Beijing China, Aug. 2012. ACM. ISBN 978-1-4503-1462-6. doi: 10.1145/2339530.2339556. URL <https://dl.acm.org/doi/10.1145/2339530.2339556>. Cited on page 22.
- S. Lundberg and S.-I. Lee. A Unified Approach to Interpreting Model Predictions, Nov. 2017. URL <http://arxiv.org/abs/1705.07874>. arXiv:1705.07874 [cs, stat]. Cited on pages 15, 25, 30, 36, and 37.
- S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence*, 2(1):56–67, Jan. 2020. ISSN 2522-5839. doi: 10.1038/s42256-019-0138-9. URL <https://www.nature.com/articles/s42256-019-0138-9>. Number: 1 Publisher: Nature Publishing Group. Cited on page 37.
- R. Luss, P.-Y. Chen, A. Dhurandhar, P. Sattigeri, K. Shanmugam, and C.-C. Tu. Generating Contrastive Explanations with Monotonic Attribute Functions. *ArXiv*, May 2019. URL <https://www.semanticscholar.org/paper/Generating-Contrastive-Explanations-with-Monotonic-Luss-Chen/712d211cc66e1ac00076bf331c9bd9e3ab59e2ad>. Cited on page 16.
- L. v. d. Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. ISSN 1533-7928. URL <http://jmlr.org/papers/v9/vandermaaten08a.html>. Cited on page 28.
- T. Miller. Explanation in Artificial Intelligence: Insights from the Social Sciences, Aug. 2018. URL <http://arxiv.org/abs/1706.07269>. arXiv:1706.07269 [cs]. Cited on pages 14 and 29.
- T. Miller. Contrastive explanation: a structural-model approach. *The Knowledge Engineering Review*, 36:e14, Jan. 2021. ISSN 0269-8889, 1469-8005. doi: 10.1017/S0269888921000102. URL <https://www.cambridge.org/core/journals/knowledge-engineering-review/article/abs/contrastive-explanation-a-structuralmodel-approach/69A2E32B160C2C7FB65BC88670D7AEA7#access-block>. Publisher: Cambridge University Press. Cited on page 27.
- T. Miller, P. Howe, and L. Sonenbergh. Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences. Dec. 2017. Cited on page 22.

- B. Mittelstadt, C. Russell, and S. Wachter. Explaining Explanations in AI. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19*, pages 279–288, New York, NY, USA, Jan. 2019. Association for Computing Machinery. ISBN 978-1-4503-6125-5. doi: 10.1145/3287560.3287574. URL <https://doi.org/10.1145/3287560.3287574>. Cited on page 16.
- C. Molnar. *Interpretable Machine Learning*. 2022. URL <https://christophm.github.io/interpretable-ml-book/>. Cited on pages 23, 40, and 42.
- J. W. Moon and L. Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1): 23–28, Mar. 1965. ISSN 1565-8511. doi: 10.1007/BF02760024. URL <https://doi.org/10.1007/BF02760024>. Cited on page 96.
- N. Narodytska, A. Shrotri, K. Meel, A. Ignatiev, and J. Marques-Silva. Assessing Heuristic Machine Learning Explanations with Model Counting. pages 267–278. June 2019a. ISBN 978-3-030-24257-2. doi: 10.1007/978-3-030-24258-9_19. Cited on page 16.
- N. Narodytska, A. Shrotri, K. S. Meel, A. Ignatiev, and J. Marques-Silva. Assessing Heuristic Machine Learning Explanations with Model Counting. In M. Janota and I. Lynce, editors, *Theory and Applications of Satisfiability Testing – SAT 2019*, volume 11628, pages 267–278. Springer International Publishing, Cham, 2019b. ISBN 978-3-030-24257-2 978-3-030-24258-9. doi: 10.1007/978-3-030-24258-9_19. URL http://link.springer.com/10.1007/978-3-030-24258-9_19. Series Title: Lecture Notes in Computer Science. Cited on page 61.
- S. Parsons and N. R. Jennings. Negotiation through argumentation—a preliminary report. In *Proceedings of the 2nd International Conference on Multi Agent Systems*, pages 267–274, 1996. Cited on page 60.
- S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *Journal of Logic and Computation*, 13 (3):347–376, 2003. Cited on page 60.
- V. Petsiuk, A. Das, and K. Saenko. RISE: Randomized Input Sampling for Explanation of Black-box Models, Sept. 2018. URL <http://arxiv.org/abs/1806.07421>. arXiv:1806.07421 [cs]. Cited on pages 25 and 27.
- N. Potyka, X. Yin, and F. Toni. Explaining Random Forests using Bipolar Argumentation and Markov Networks (Technical Report), Nov. 2022. URL <http://arxiv.org/abs/2211.11699>. arXiv:2211.11699 [cs]. Cited on page 60.
- H. Prakken. Formal systems for persuasion dialogue. *Knowledge Engineering Review*, 21: 163–188, 2006. Cited on page 60.

- M. Proietti and F. Toni. A Roadmap for Neuro-argumentative Learning. 2023. Cited on page 60.
- A. Páez. The Pragmatic Turn in Explainable Artificial Intelligence (XAI). *Minds and Machines*, 29:441–459, Sept. 2019. doi: 10.1007/s11023-019-09502-w. Cited on pages 19 and 22.
- J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, Mar. 1986. ISSN 1573-0565. doi: 10.1007/BF00116251. URL <https://doi.org/10.1007/BF00116251>. Cited on page 15.
- A. Rago, O. Cocarascu, and F. Toni. Argumentation-Based Recommendations: Fantastic Explanations and How to Find Them. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 1949–1955, Stockholm, Sweden, July 2018. International Joint Conferences on Artificial Intelligence Organization. ISBN 978-0-9992411-2-7. doi: 10.24963/ijcai.2018/269. URL <https://www.ijcai.org/proceedings/2018/269>. Cited on page 60.
- A. Rago, H. Li, and F. Toni. Interactive Explanations by Conflict Resolution via Argumentative Exchanges, June 2023. URL <http://arxiv.org/abs/2303.15022>. arXiv:2303.15022 [cs]. Cited on page 60.
- I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS’06)*, pages 347–354. ACM Press, 2006. Cited on page 59.
- S. Ransbotham, D. Kiron, P. Gerbert, and M. Reeves. Reshaping Business With Artificial Intelligence. *MIT Sloan Management Review*, Sept. 2017. URL <https://sloanreview.mit.edu/projects/reshaping-business-with-artificial-intelligence/>. Cited on page 13.
- R. Reiter. A logic of default reasoning. *Artificial Intelligence Journal*, 81–132:203–242, 1980. Cited on page 53.
- X. Renard, N. Woloszko, J. Aigrain, and M. Detyniecki. Concept Tree: High-Level Representation of Variables for More Interpretable Surrogate Decision Trees, June 2019. URL <http://arxiv.org/abs/1906.01297>. arXiv:1906.01297 [cs, stat]. Cited on page 27.
- M. T. Ribeiro, S. Singh, and C. Guestrin. ”Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144, San Francisco California USA, Aug. 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.

2939778. URL <https://dl.acm.org/doi/10.1145/2939672.2939778>. Cited on pages 15, 16, 22, 25, 26, 27, 30, 32, 39, 40, 41, 42, 43, and 61.
- M. T. Ribeiro, S. Singh, and C. Guestrin. Anchors: High-Precision Model-Agnostic Explanations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v32i1.11491. URL <https://ojs.aaai.org/index.php/AAAI/article/view/11491>. Cited on pages 15, 16, 25, 26, 34, 36, 41, 42, 61, 82, 83, and 109.
- N. Rotstein, M. Moguillansky, M. Falappa, A. Garcia, and G. Simari. Argument theory change: Revision upon warrant. In *Proceedings of the 2nd International conference on Computational Models of Argument (COMMA'08)*, pages 336–347, 2008. Cited on page 59.
- G. Sartor. A Formal Model of Legal Argumentation. *Ratio Juris*, 7(2):177–211, July 1994. ISSN 0952-1917, 1467-9337. doi: 10.1111/j.1467-9337.1994.tb00175.x. URL <https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9337.1994.tb00175.x>. Cited on page 45.
- P. Schmidt and A. D. Witte. Predicting Recidivism in North Carolina, 1978 and 1980: Archival Version, 1989. URL <https://www.icpsr.umich.edu/web/NACJD/studies/8987>. Cited on page 84.
- G. Schwalbe and B. Finzel. A Comprehensive Taxonomy for Explainable Artificial Intelligence: A Systematic Survey of Surveys on Methods and Concepts. *Data Mining and Knowledge Discovery*, Jan. 2023. ISSN 1384-5810, 1573-756X. doi: 10.1007/s10618-022-00867-8. URL <http://arxiv.org/abs/2105.07190>. arXiv:2105.07190 [cs]. Cited on pages 22, 23, 24, 39, and 40.
- S. M. Shankaranarayana and D. Runje. ALIME: Autoencoder Based Approach for Local Interpretability, Sept. 2019. URL <http://arxiv.org/abs/1909.02437>. arXiv:1909.02437 [cs, stat]. Cited on page 42.
- L. S. Shapley. A value for n-person games. 1953. URL <https://books.google.com/books?hl=en&lr=&id=Pd3TCwAAQBAJ&oi=fnd&pg=PA307&dq=Lloyd+Shapley+in+1953+values&ots=gtwZC9bht-&sig=oxwa8TXppRZvnT6oG1X2Dd1CxCO>. Publisher: Princeton University Press Princeton. Cited on page 36.
- A. Shih, A. Choi, and A. Darwiche. A Symbolic Approach to Explaining Bayesian Network Classifiers, May 2018. URL <http://arxiv.org/abs/1805.03364>. arXiv:1805.03364 [cs]. Cited on pages 15 and 64.
- G. Simari and R. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence Journal*, 53:125–157, 1992. Cited on page 59.

- K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, Apr. 2014. URL <http://arxiv.org/abs/1312.6034>. arXiv:1312.6034 [cs]. Cited on page 26.
- J. W. Smith, J. Everhart, W. Dickson, W. Knowler, and R. Johannes. Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 261–265, Nov. 1988. ISSN 0195-4210. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2245318/>. Cited on page 85.
- K. Sokol and P. Flach. Explainability fact sheets: a framework for systematic assessment of explainable approaches. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 56–67, Barcelona Spain, Jan. 2020. ACM. ISBN 978-1-4503-6936-7. doi: 10.1145/3351095.3372870. URL <https://dl.acm.org/doi/10.1145/3351095.3372870>. Cited on pages 34 and 40.
- S. Thrun. Extracting Rules from Artificial Neural Networks with Distributed Representations. In *Advances in Neural Information Processing Systems*, volume 7. MIT Press, 1994. URL <https://proceedings.neurips.cc/paper/1994/hash/beat5955b308361a1b07bc55042e25e54-Abstract.html>. Cited on page 25.
- F. van Eemeren, R. Grootendorst, and F. Snoeck Henkemans. *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Applications*. Lawrence Erlbaum Associates, Hillsdale NJ, USA, 1996. Cited on pages 46 and 47.
- M. Vega García and J. L. Aznarte. Shapley additive explanations for NO2 forecasting. *Ecological Informatics*, 56:101039, Mar. 2020. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2019.101039. URL <https://www.sciencedirect.com/science/article/pii/S1574954119303498>. Cited on page 37.
- B. Verheij. Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages. 1996. Cited on page 58.
- G. Vilone and L. Longo. Explainable Artificial Intelligence: a Systematic Review, Oct. 2020. URL <http://arxiv.org/abs/2006.00093>. arXiv:2006.00093 [cs]. Cited on pages 15, 22, 39, and 40.
- S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GDPR, Nov. 2017. URL <https://arxiv.org/abs/1711.00399v3>. Cited on pages 16 and 29.

- D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Series in Logic and Language. State University of New York Press, Albany, NY, USA, 1995. Cited on page 60.
- L. Wolf, T. Galanti, and T. Hazan. A Formal Approach to Explainability, Jan. 2020. URL <http://arxiv.org/abs/2001.05207>. arXiv:2001.05207 [cs, stat]. Cited on page 40.
- M. R. Zafar and N. M. Khan. DLIME: A Deterministic Local Interpretable Model-Agnostic Explanations Approach for Computer-Aided Diagnosis Systems, June 2019. URL <http://arxiv.org/abs/1906.10263>. arXiv:1906.10263 [cs, stat]. Cited on page 42.
- M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks, Nov. 2013. URL <http://arxiv.org/abs/1311.2901>. arXiv:1311.2901 [cs]. Cited on pages 25 and 26.
- Q. Zhang, R. Cao, F. Shi, Y. N. Wu, and S.-C. Zhu. Interpreting CNN Knowledge via an Explanatory Graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. ISSN 2374-3468. doi: 10.1609/aaai.v32i1.11819. URL <https://ojs.aaai.org/index.php/AAAI/article/view/11819>. Number: 1. Cited on page 30.
- Q. Zhang, J. Lu, and Y. Jin. Artificial intelligence in recommender systems. *Complex & Intelligent Systems*, 7(1):439–457, Feb. 2021. ISSN 2198-6053. doi: 10.1007/s40747-020-00212-w. URL <https://doi.org/10.1007/s40747-020-00212-w>. Cited on page 13.
- Y. Zhang, F. Abu-Khzam, N. Baldwin, E. Chesler, M. Langston, and N. Samatova. Genome-Scale Computational Approaches to Memory-Intensive Applications in Systems Biology. In *SC '05: Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*, pages 12–12, Nov. 2005. doi: 10.1109/SC.2005.29. URL <https://ieeexplore.ieee.org/document/1559964>. Cited on page 97.
- Y. Zhang, K. Song, Y. Sun, S. Tan, and M. Udell. "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations, Apr. 2019. URL <https://arxiv.org/abs/1904.12991v2>. Cited on page 34.
- J. R. Zilke, E. Loza Mencía, and F. Janssen. DeepRED – Rule Extraction from Deep Neural Networks. In T. Calders, M. Ceci, and D. Malerba, editors, *Discovery Science*, Lecture Notes in Computer Science, pages 457–473, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46307-0. doi: 10.1007/978-3-319-46307-0_29. Cited on page 25.
- K. Čyras, D. Birch, Y. Guo, F. Toni, R. Dulay, S. Turvey, D. Greenberg, and T. Hapuarachchi. Explanations by arbitrated argumentative dispute. *Expert Systems with*

Applications, 127:141–156, Aug. 2019a. ISSN 0957-4174. doi: 10.1016/j.eswa.2019.03.012.
URL <https://www.sciencedirect.com/science/article/pii/S0957417419301654>.
Cited on page 60.

K. Čyras, D. Letsios, R. Misener, and F. Toni. Argumentation for Explainable Scheduling (Full Paper with Proofs), Feb. 2019b. URL <http://arxiv.org/abs/1811.05437>.
arXiv:1811.05437 [cs]. Cited on page 60.

Titre : Analyse et explication par des techniques d'argumentation de modèles d'intelligence artificielle basés sur des données

Mots clés : Explicabilité, Apprentissage, Argumentation

Résumé : La classification est une tâche très courante dans le domaine de l'apprentissage automatique et les modèles d'apprentissage automatique créés pour accomplir cette tâche tendent à atteindre une précision comparable à celle des humains, au détriment de leur transparence. L'apparition de ces systèmes intelligents dans le quotidien du public a créé un besoin d'explicabilité. Les explications abductives sont l'un des types d'explications les plus populaires qui sont fournies dans le but d'expliquer le comportement de modèles d'apprentissage complexes, parfois considérés comme des boîtes noires. Elles mettent en évidence les caractéristiques qui sont suffisantes pour que le modèle prédise une certaine classe. Dans la littérature, elles sont générées en explorant l'ensemble de l'espace des caractéristiques, ce qui n'est pas raisonnable en pratique. Cette thèse aborde ce problème en introduisant des fonctions d'explication qui génèrent des explications abductives à partir d'un échantillon arbitraire d'instances. Elle montre que de telles fonctions doivent être définies avec beaucoup de soin car elles ne peuvent pas satisfaire simultanément deux propriétés souhaitables, à savoir l'existence d'explications pour chaque décision individuelle et l'existence d'explications abductives. décision individuelle (success) et l'exactitude des explications (coherence). Cette thèse fournit une paramétrée de fonctions d'explication basées sur l'argumentation, chacune satisfaisant l'une des ces deux propriétés. De plus, elle étudie leurs propriétés formelles ainsi que leur comportement expérimental sur différents ensembles de données.

Title: Analyzing and explaining data-driven Artificial Intelligence Models by argumentation

Key words: Machine Learning, Argumentation, Explainability

Abstract: Classification is a very common task in Machine Learning (ML) and the ML models created to perform this task tend to reach human comparable accuracy, at the cost of transparency. The surge of such AI-based systems in the public's daily life has created a need for explainability. Abductive explanations are one of the most popular types of explanations that are provided for the purpose of explaining the behavior of complex ML models sometimes considered as black-boxes. They highlight feature-values that are sufficient for the model to make a prediction. In the literature, they are generated by exploring the whole feature space, which is unreasonable in practice. This thesis tackles this problem by introducing explanation functions that generate abductive explanations from a sample of instances. It shows that such functions should be defined with great care since they cannot satisfy two desirable properties at the same time, namely existence of explanations for every individual decision (success) and correctness of explanations (coherence). This thesis provides a parameterized family of argumentation-based explanation functions, each of which satisfies one of the two properties. It studies their formal properties and their experimental behaviour on different datasets.