




An A-Star Algorithm for Argumentative Rule Extraction

Benoît Alcaraz¹^a, Adam Kaliski¹^b and Christopher Leturc²^c

¹University of Luxembourg, Luxembourg

²Université Côte d'Azur, France

{firstname.lastname}@uni.lu, christopher.leturc@univ-cotedazur.fr

Keywords: Rule Induction, Formal Argumentation, Explainability.

Abstract: In this paper, we present an approach for inferring logical rules in the form of formal argumentation frameworks using the A^* algorithm. We show that contextual argumentation frameworks — in which arguments are activated and deactivated based on the values of the boolean variables that the arguments represent — allow for a concise, graphical, and hence *explainable* representation of logical rules. We define the proposed approach as a tool to understand the behaviour of already deployed black-box agents. Additionally, we show several applications where having an argumentation framework representing an agent decision's model is required or could be beneficial. We then apply our algorithm to several datasets in order to evaluate its performances. The algorithm reaches high accuracy scores on discrete datasets, indicating that our approach could be a promising avenue for alternative data-driven AI learning techniques, especially in the context of explainable AI.

1 INTRODUCTION


Machine Learning (ML) approaches have been widely used in many applications, such as autonomous vehicles and medical analysis. These applications, as they may injure humans or cause severe damages, are subject to regulations, demand that the decisions made by ML models and other algorithms must be *explainable* to aid human understanding and validation (Goodman and Flaxman, 2017). Some methods are considered as more interpretable, such as rule-based learning or decision trees (Catlett, 1991). However, they usually imply a trade-off between explainability and performance (*i.e.*, accuracy or time) compared to machine learning methods such as deep learning or deep reinforcement learning. Thus, in some scenarios, it is preferable to continue using machine learning approaches and to try to explain their behaviour, following the distill-and-compare paradigm (Tan et al., 2018).


Additionally, while the form IF *premises* THEN *conclusion* can be used in order to make these explanations human-readable, they often fail to convey the notion of causality (Lewis, 2013). Consider the scenario: 'If it is not raining or I have an umbrella, then I won't get wet.' When expressed in propositional


logic or any other logic made of classical boolean connectors, it fails to convey the crucial point that the presence of an umbrella becomes irrelevant when it is not raining. Such loss of contextual information can lead to misinterpretation and misunderstanding of the rule's logic. This is practically problematic, for example for ethics and compliance reasons. Hence, we claim that propositional logic might not be an ideal way of presenting an explanation in an intelligible way.

Formal argumentation — a collection of approaches to automated reasoning, in which logical statements are considered arguments and their relations are graphically modelled — is widely considered as being a facilitator of the explainability of logic-based reasoning (Fan and Toni, 2014; Rizzo and Longo, 2018). Argumentation can better express relations between attributes instead of only showing the final reasons of a decision. Such capability could enhance the trust users have in a system. Also, because argumentation graphs are relatively easily understandable and manipulable by humans (Rizzo and Longo, 2018), argumentation-based approaches to rule learning could allow expert modifications over the produced result to refine the rules or to learn more about a decision in an interactive manner.

In this paper, we introduce a rule induction approach based on formal argumentation and heuristic search, which we call *Argumentative Rule Induction*

^a <https://orcid.org/0000-0002-7507-5328>

^b <https://orcid.org/0000-0003-2756-719X>

^c <https://orcid.org/0000-0003-0821-6095>

A-star (ARIA). It aims at representing the decision model of a learning agent through an argumentation graph, which can be used as a tool to justify the decision of a black-box model. We chose to use an A^* search, as it manages to find a potential solution in a reasonable amount of time.

Furthermore, we want to outline the relevance of this work in regard to the newer works in the field of multi-agent systems combining autonomous agents and reasoning methods. In particular, we believe that enabling the possibility of obtaining an argumentation framework matching an agent's decisions opens a broad field of possibilities in application domains such as symbolic reinforcement learning, or multi-agent negotiation and decision making. Also, as previously mentioned, the use of formal argumentation provides a notion of causality. In some multi-agent communication protocols, such as Fatio's protocol (McBurney and Parsons, 2004)—a formal dialogue protocol focusing onto argumentative dialogues—such causality notion is necessary for agents to identify if they can invoke an argument or not. If we consider again the example of the umbrella, one cannot tell to the other agent that one has an umbrella before he asserted that it was about to rain. If performed in the opposite order, the interaction may generate an unnecessary amount of locutions by trying to anticipate arguments which may not be asserted by any agent.

The remainder of the paper is organised as follow. First, we briefly review relevant rule induction literature in Section 2. Subsequently, Section 3 introduces the basic prerequisite from formal argumentation to understand the mechanisms inside of our proposed algorithm. In Section 4, we present how our model is used to compute a justification which may be inferred from a black-box model. Then, in Section 5, we test our approach over a selection of benchmark datasets from the literature, and compare to a well-known rule induction algorithm, as well as a custom test case fitting our application context. Those results are discussed in Section 6. Finally, we conclude in Section 7, recalling the main elements of our contribution and outlining potential future work.

2 RELATED WORK

In recent years, deep neural networks, have been shown to be particularly successful at solving classification problems. However these algorithms suffer from a lack of explainability (Szegedy et al., 2013).

A solution to make a system understandable is to apply decision tree classification approaches.

The algorithm C4.5 (Quinlan, 2014), based on ID3 (Quinlan, 1986), has been developed following this paradigm. However, it has been shown that those approaches tend to be outperformed in many domains by rule induction algorithms (Bagallo and Haussler, 1990; Quinlan, 1987; Weiss and Indurkha, 1991).

Rule induction algorithms are a category of approaches that usually tries to generate a set of rules for each decision or class. Then, if an input triggers one of the rules for a given class, it is considered as being part of this class. There exist a variant of C4.5, called *C4.5-rules*, on which many approaches were based. For instance, IREP (Fürnkranz and Widmer, 1994) has been introduced to accommodate the issues of C4.5 relative to its computation time by making the pruning more efficient. However, it was usually producing more errors than C4.5 on domain specific datasets. For this reason, Cohen developed an improved version called RIPPER k (Cohen, 1995) which is at the same time more efficient, more accurate, and more resilient to the noise in the data. An algorithm of rule induction based on a genetic algorithm, SIA (Venturini, 1993), generates a population of rules and compares the predictions performed with it to an actual dataset. The algorithm has to maximise the number of correct predictions. The algorithm ESIA (Liu and Kwok, 2000) (Extended SIA) is an extension of SIA. While the base principle remains similar in both approaches, ESIA contains several modifications to the operators of SIA, such as the specialization operator, and introducing a separate-and-conquer search method.

In parallel, much work has been done in explainability, especially with the new Deep Reinforcement Learning (DRL) algorithms presenting high performance but poor interpretability (Adadi and Berrada, 2018). In the next years, such capability could become an obligation more than an option with incoming legislation in the right to explanation (Selbst and Powles, 2018). We divide these works in two categories. The first one concerns the algorithms which have an intrinsic intelligibility, meaning that the algorithm itself provides information for its understanding. The second one concerns algorithms producing post-hoc explanations (Puiutta and Veith, 2020), meaning that we can apply these algorithms to various AI models to extract information.

Verma *et al.* (Verma et al., 2018) propose an intrinsic explainability method by presenting a reinforcement learning agent producing interpretable policy rules. By representing the states in a high-level manner, it can express the rules determining the action to perform, *i.e.*, the policy to follow. Even though this work is competitive with DRL algorithms when work-

ing with symbolic inputs, it cannot handle perceptual inputs, such as pixels of an image, or stochastic policies, useful in game environment. Moreover, policies generated in this work remain hard to grasp especially for a non-expert user, due to the large amount of numerical values present in the rule which greatly decreases intelligibility. Additionally, the replacement of black-box models by newer intrinsically explainable methods might either be something not applicable, or too expensive for a company. This might limit the spreading of such a method. Last, it is often the case that those explainable methods show slightly lower performance than black-box models. Thus, it forces to a trade-off between performance and explainability.

On the other hand, post-hoc methods for explainability (Puiutta and Veith, 2020; Hein et al., 2018; Adadi and Berrada, 2018) aim at generating from reinforcement learning policies a set of equations describing trajectories of the analysed policy. These outputs are presented with a certain amount of complexity in terms of explainability. Yet, authors admit that, even if pretty low, complexity equation sets allow good performance in addition to showing some explainability relief when compared to Neural Network approaches, they still under-perform it in terms of pure performance. Moreover, the equation system may start to become hard to understand due to the abstract nature of some thresholds. Also, because this algorithm is computing rules of trajectory, it may struggle in highly discretized environments such as the ones with categorical inputs. Furthermore, this work and others presented in Puiutta and Veith (Puiutta and Veith, 2020) such as Liu *et al.* (Liu et al., 2018) or Zahavy *et al.* (Zahavy et al., 2016) are not agnostic to the learning algorithm for which they provide explanations and need access to the policy of the agent. Another post-hoc approach is the counterfactual explanation which consists in giving bits to the end-user to help him understand what the machine is doing by presenting slight input variations to obtain different outputs (Wachter et al., 2017). The problem of such a method is that it leaves the responsibility to the end user to make suppositions on what impacts the model's decision and what does not. Additionally, it is not at all a scalable methodology. Tan *et al.* (Tan et al., 2018) presented a model distillation. This method transfers knowledge from a large, complex model (teacher) to a faster, simpler model (student). Yet, even if it can successfully extract data from black-box algorithms, the problem of the explainability of the extracted data remains.

Last, the algorithm PSyKE (Sabbatini et al., 2021) is the closest to our approach in term of application.

In this work, authors use a rule induction generating some Prolog-like rules based on the classifications of a given model, such as CART, GridEx, or k-nn. While it tends to under-performs the initial model, this is not a major issue as it only aims at providing a justification for the decisions taken by the classification model by giving the set of rules leading to this aforementioned decision. As such, they define a value called black-box fidelity, and which shows how accurately the generated rules mimic the classification of the black-box model.

3 ABSTRACT ARGUMENTATION

In this section, we provide the necessary formal preliminaries of abstract argumentation, based on Dung's seminal paper (Dung, 1995).

Definition 1 (Argumentation framework). *An argumentation framework is a tuple $F = (\mathcal{A}rgs, \mathcal{R})$, where $\mathcal{A}rgs$ is a set of elements called arguments and $\mathcal{R} \subseteq \mathcal{A}rgs \times \mathcal{A}rgs$ is a relation over the arguments referred to as attack.*

Given $S \subseteq \mathcal{A}rgs$, we recall the notions of conflict-free and acceptability. A conflict-free set is one in which no argument in the set attacks another. Acceptability represents the constraint that an argument is only acceptable if all its attackers are themselves attacked by an argument in the set.

- S is a *conflict-free* set of arguments w.r.t. \mathcal{R} iff $\nexists a, b \in S$ s.t. $a\mathcal{R}b$,
- For all $a \in \mathcal{A}rgs$, a is acceptable w.r.t. S iff $\forall b \in \mathcal{A}rgs$, if $b\mathcal{R}a$, then $\exists c \in S$, such that $c\mathcal{R}b$.

We recall the standard definitions for extensions :

- S is an *admissible extension* iff S is conflict-free and all arguments $a \in S$ are acceptable wrt S .
- S is a *complete extension* iff S is admissible and contains all acceptable arguments wrt S .
- S is a *grounded extension* iff S is a minimal complete extension with respect to strict set inclusion i.e. $\nexists S' \subseteq \mathcal{A}rgs$ s.t. $S' \subset S$, and S' is a complete extension.

It is well-known result that for all argumentation frameworks $F = (\mathcal{A}rgs, \mathcal{R})$, there exists a unique grounded extension (Baumann, 2017). Thus, it is of interest to apply the grounded semantics. Furthermore, the grounded semantics is also interesting for its computational aspect since it can be computed in time $O(|\mathcal{A}rgs| + |\mathcal{R}|)$ by efficient algorithms like those proposed by Nofal *et al.* (Nofal et al., 2021). Thus, in this approach we will consider only grounded

semantics although the theoretical framework can apply other extension semantics.

4 MODEL

This section describes how each component of our model works. First, we present how our model would bring intelligibility to a black-box algorithm when deployed. Second, we detail how it generates an explainable model with the help of some data.

4.1 Computing a Justification

As said previously, the goal of our approach is to justify the decision of a black-box algorithm (which will be referred in the rest of the section as a BB or agent). More specifically, we want to observe one action and why it was, or was not, chosen. For instance, why the autonomous car applied the brakes or the accelerator. To do so, we introduce two core elements, as well as an example to better understand how our approach should work.

4.1.1 Universal Graph

In order to be used, our approach requires a dataset denoting the behaviour of an agent (or any BB algorithm) in various situations. Such a dataset should provide as input the perceptions of the BB algorithm (which can be raw or pre-processed information), and as a label, if the tracked action has been performed or not. Once this dataset is available, the search process, detailed later in Section 4.3, can start.

Once finished, this search process will return a graph, inspired by the structure of an argumentation framework, representing the overall relationship between the perceptions, and how they can influence the final decision. We call this graph the Universal Graph, as it tends to represent the overall behaviour of the BB algorithm in any situation. In our implementation, the nodes of the graph are having as a value a tuple attribute-value from the dataset inputs. However, it is possible to use expert knowledge to provide more sophisticated nodes. For example, instead of having a node with a very arbitrary value “size=3.5cm”, an expert could design values such as “size=small”, which would include anything with a size less than 5 cm.

On the other hand, there are two additional nodes. The first one is called the target, and is written as τ . This argument represents the decision performed by the agent. As will be detailed in the next section, it will denote the fact that the agent performed the tracked action if it is part of the grounded extension,

or otherwise that the agent did not. The second additional node is written as λ . While this argument is not bound to any couple attribute-value, it denotes a support relation to the target from the other nodes, as explained by Boella *et al.* (Boella et al., 2009). As such, this node can only attack the target. Any other node attacking it will then be considered as supporting the target due to the defense relationship created between them.

4.1.2 Contextual Graph

Once in possession of the Universal Graph, one can then compute what is called the Contextual Graph. This graph is a projection of the Universal Graph given a set of facts *i.e.*, a set of perceptions from the agent in a specific situation¹. As a consequence, it filters out all the nodes whose value is not part of the facts, except for τ and λ which are always in the Contextual Graph. The resulting graph is then a proper argumentation framework.

It is now possible to compute an extension based on this Contextual Graph. While in theory any extension would be usable with some adjustments, we chose to use the grounded extensions as it features two advantages over the others. First, it is unique, and as such less ambiguous than a set of extensions when it comes to providing an explanation to the end user. Second, it computes in polynomial time² which is an interesting feature to shorten the search phase.

Then, if the target τ is part of the extension, and if this matches the decision that the BB made (*i.e.*, doing the tracked action or not), it can serve to justify this decision by potentially providing an explanation based on this graph. This includes the cases where the BB took the wrong decisions, as long as the BB output matches the graph output. There exist several ways to provide an explanation from an argumentation framework (Liao et al., 2021; Fan and Toni, 2014; Doutre et al., 2023), which may, for instance, consist of extracting the smallest set of arguments such that the target is defended or the smallest chain such that it is defeated. As this is not the focus of this paper, we leave this choice to the designer.

4.1.3 Example

In order to make it clearer, we provide the following running example inspired from the Car dataset (Bohanec, 1997). Fig. 1 represents a graph which has

¹This may also correspond to the input of one entry of the BB’s behavioural dataset.

²While the preferred extension also share a polynomial complexity, it still requires a greater time to be computed.

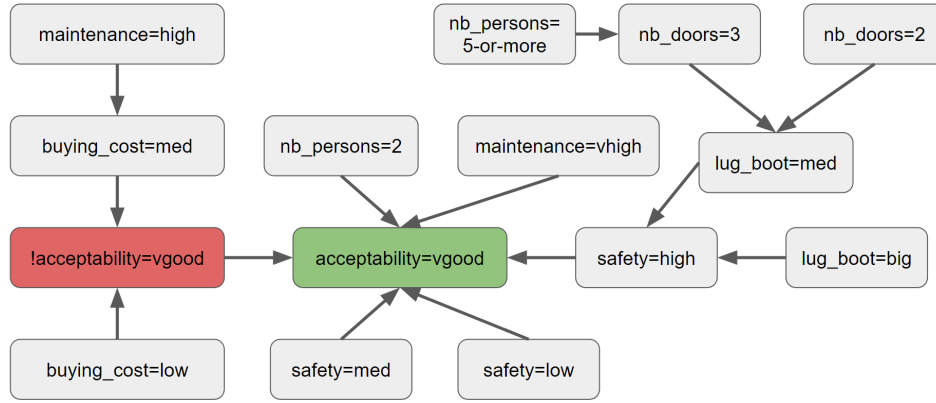


Figure 1: Universal graph for the Car dataset. The τ argument is in green. The λ argument is in red.

Table 1: Summary of the facts, *i.e.*, the values for the different attributes, for the presented example.

Attribute	Value
Buying cost	= Medium
Maintenance cost	= Low
Number of doors	= 3
Number of seats	= 5-or-more
Luggage boot	= Medium
Safety	= High
Acceptability (Label)	= Very-good

been inferred with our approach from the aforementioned dataset. The task is to evaluate a car buying acceptability based on several attributes and their values, such as the number of seats, the buying cost, the maintenance cost, and so on. There are four classes, namely “Unacceptable”, “Acceptable”, “Good”, and “Very good”. In this specific case, the target argument is associated to the class “Very good”, *i.e.*, “The car has a very good buying acceptability”. As such, the target not being in the extension can classify as “Unacceptable”, “Acceptable”, or “Good”. Fig. 1 shows the Universal Graph and then gives an overview of how each argument (*i.e.*, couple attribute-value) can influence the final decision. However, it does not justify any specific decision yet.

Fig. 2 shows the Contextual Graph derived from the Universal Graph shown in Fig. 1. The later corresponds to the set of facts presented in Table 1, *i.e.*, the values assigned to the different attributes. In this example, the Car has been classified as “Very good” in terms of buying acceptability. From the Contextual Graph, we can see the elements which lead to this decision. For instance, because high safety was not sufficient, it was also required that it has a medium luggage boot. On the other hand, if it did not have at least 5 seats, the car would not have been classified as “Very good” since the number of doors were 3. On the other hand, the medium buying cost supports the classification by attacking the λ argument.

4.2 n -Arguments

Let be the set of arguments corresponding to the presence of a couple attribute-value in the facts be called “Positive arguments” (or p -arguments). It is then possible to respectively define “Negative arguments” (or n -arguments). In opposition to p -arguments, which appear in the contextual graph only if the facts match their condition, n -arguments are present in the contextual graph if their condition is not present in the facts. They denote a missing fact. While using only the set of p -arguments makes the representation of some logic formula impossible, the addition of the n -arguments allows to represent a broader range of formulae. However, adding these extra arguments increases the number of neighbours for each node, and as such, increases the total computation time.

More formally, let \mathcal{P} be a non empty set of propositional atoms. In this application, each propositional atom represents a possible valuation for an attribute, *e.g.* $q := \text{”Buying cost = Medium”} \in \mathcal{P}$. We define the language $\mathcal{L}_{\mathcal{P}}$ as the set of well-formed formulas (wff), with the following BNF grammar, for any $p \in \mathcal{P}$:

$$\phi ::= \perp \mid p \mid \neg\phi \mid \phi \vee \psi$$

As usual, we use the following notation shortcuts:

- $\top := \neg\perp$,
- $\phi \wedge \psi := \neg(\neg\phi \vee \neg\psi)$,
- $\phi \Rightarrow \psi := \neg\phi \vee \psi$

An interpretation model I over a valuation $V : \mathcal{P} \rightarrow \{\top, \perp\}$ is given by the function $I : \mathcal{L}_{\mathcal{P}} \rightarrow \{\perp, \top\}$ s.t. $\forall \phi \in \mathcal{L}_{\mathcal{P}}, I \models \phi$ iff $I(\phi) = \top$ where :

1. $\forall p \in \mathcal{P}, I \models p$ iff $V(p) = \top$
2. $\forall \phi, \psi \in \mathcal{L}_{\mathcal{P}}, I \models \phi \vee \psi$ iff $I \models \phi$ or $I \models \psi$
3. $\forall \phi \in \mathcal{L}_{\mathcal{P}}, I \models \neg\phi$ iff $I \not\models \phi$

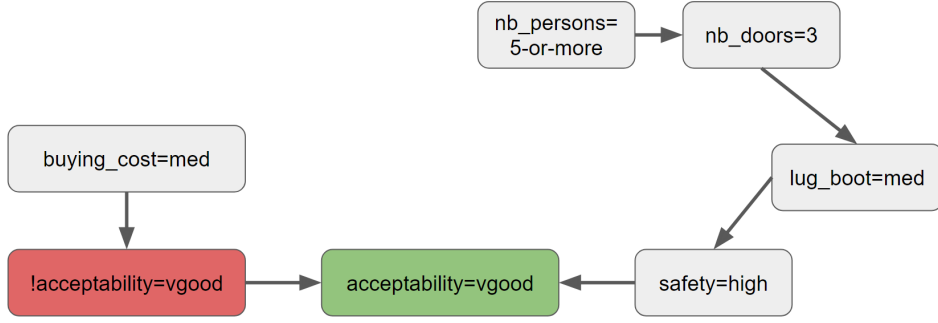


Figure 2: Contextual graph for the Car dataset and a specific input. The target argument is in green. The λ argument is in red.

Additionally, we will denote the set of symbols $\bar{\mathcal{P}} := \{\bar{p} : p \in \mathcal{P}\}$ which corresponds to the negation of propositional atoms. Henceforth, we call a class of interpretation models based on the set of valuations $\Omega \subseteq \{\top, \perp\}^{\mathcal{P}}$ a *propositional dataset*. We note that $\forall \phi \in \mathcal{L}_{\mathcal{P}}, \Omega \models \phi$ iff for all $V \in \Omega$, the interpretation model I_V over V is s.t. $I_V \models \phi$.

Example 1. *The following example aims at showing a situation that cannot be represented by solely using p -arguments, and would require the addition of the n -arguments. We represent an argument as a couple, composed by a name A , and a condition, represented as a propositional formula of $\mathcal{L}_{\mathcal{P}}$. Let $\mathcal{A}rgs = \{(\tau, \top), (\lambda, \top), (A, a), (B, b)\}$ with $a, b \in \mathcal{P}$, be the set of arguments and $\mathcal{F} \in \Omega$ be the set of facts. We represent a situation where τ should be part of the grounded extension, denoted $In(\tau)$, under the following condition:*

$$In(\tau) \rightarrow ((a \in \mathcal{F} \wedge b \notin \mathcal{F}) \vee (a \notin \mathcal{F} \wedge b \in \mathcal{F}))$$

Without duplicating the arguments, it is not possible to represent such condition. Now, let us add to our current set of arguments the n -arguments $\bar{a}, \bar{b} \in \bar{\mathcal{P}}$, such that: $\mathcal{A}rgs = \{(\tau, \top), (\lambda, \top), (A, a), (B, b), (C, \bar{a}), (D, \bar{b})\}$. It is now possible to construct an argumentation framework with the set of arguments $\mathcal{A} = \{\tau, \lambda, A, B, C\}$ and the attack relation $\mathcal{R} = \{(\lambda, \tau), (A, \lambda), (B, A), (C, B), (C, \lambda), (D, C)\}$.

This argumentation framework can correctly represent the fact that τ should be part of the grounded extension if a is in the facts but not b , or if b is in the facts but not a .

4.3 Search Method

As previously said in the paper, our algorithm is performing an A^* search to induce the final argumentation framework. This algorithm is described in Algorithm 1. The main principle is to explore a search space by walking from neighbour to neighbour, following a heuristic to maximise prediction accuracy.

In our implementation, given a set of attributes Att and a set of available values Val_i for an attribute i , we define the set of all the arguments which can be represented by the dataset attributes and values as:

$$\mathcal{A} := \{\tau, \lambda\} \cup \{q : i \in Att, j \in Val_i, q := "Att_i = Val_{ij}"\}$$

In order to avoid generating too many meaningless arguments in the event of continuous numerical attributes, we segment them into intervals. As such, an attribute ϕ having a value ranging from 0 to 10 would generate the arguments $\phi = 0 - 2, \phi = 2.1 - 4, \dots$, with an interval size depending on the number of segments we wish to generate. While this is convenient to quickly parse a dataset, we recommend to use expert knowledge to design arguments based on those numerical values, such as $\phi = "Above\ the\ average"$.

We can then encode our attack relation as a matrix of size $\mathcal{R} = |\mathcal{A}| \times |\mathcal{A}|$, where the element \mathcal{R}_{ij} is equal to 1 if there is an attack from \mathcal{A}_i to \mathcal{A}_j , 0 if not, and -1 if the attack is disallowed (*i.e.*, would create a reflexive attack, a symmetrical attack³, or the attacker would be the τ or λ argument). Moreover, we forbid attacks between two arguments instantiated from the same attribute, as they would not be able to be part of the set of facts at the same time. However, if the dataset used is multi-valued, this feature can be removed. As such, we define two nodes (*i.e.*, solutions) as neighbours if they differ by one value in this matrix, or in other words, by whether their associated graphs differ by one attack.

Additionally, we define a heuristic $h(x)$ given a solution x which is equal to the sum of the incorrect predictions over the training data, plus a small fraction corresponding to the number of attacks in the graph, noted $x_{\mathcal{R}}$, divided by $|\mathcal{A}|^2$, where \mathcal{A} is the set of all the

³As the choice of the grounded extension—to guarantee the uniqueness of our explanation—is an undesirable semantics for graphs containing cycles, we try to avoid them. However, it is possible to allow them if more sophisticated extensions are used.

```

Requires: MaxIterations
iteration ← 0
queue ← {}
bestNode ← getStartingNode()
node ← getStartingNode()
queue.add(node)
while ¬ Empty(queue) and
  GetAcc(node) < 100 and
  iteration < MaxIterations do
  iteration ← iteration + 1
  neighboursList ← getNeighbours(node)
  for neighbour in neighboursList do
  if ¬ Visited(neighbour) then
  queue.add(neighbour)
  Visited(neighbour) ← T
  end if
  end for
  node ← getNextPriorityNode(queue)
  queue.remove(node)
  if GetAcc(node) > GetAcc(bestNode) then
  bestNode ← node
  end if
end while
return bestNode

```

Algorithm 1: A* search algorithm.

arguments present in the dataset, or more formally:

$$h(x) = \sum_{i \in \text{data}} \begin{cases} 1, & \text{if } \text{pred}_i \neq \text{label}_i \\ 0, & \text{otherwise} \end{cases} + \frac{|x_{\mathcal{R}}|}{|\mathcal{A}|^2}$$

Solutions should minimise the value of this function. This way, a solution getting fewer errors than another one is systematically preferred, but in case of an equal error count, the one presenting the fewest attacks is preferred.

In order to build the graph, we follow a bottom-up tactic. The starting node has a graph composed of a single argument which is the target argument. An attack can only be added if there exists a path from the attacking argument to the target, using this new attack. This way, when progressing in the search, arguments and attacks are progressively added and remain connected to the target. In order to reduce as much as possible the exploration space, we ensure that a solution is not visited twice by computing an associated hash and saving it. We then compare the candidate neighbours' hash to the ones already saved and add them to the queue only if they have not already been visited. Furthermore, if the addition of an attack had no effect on the correctness of the predictions compared to the solution without this attack, we simply stop exploring this branch. This greatly reduces the time required to explore the solution space.

5 EVALUATION

This section is divided into two parts. The first one focuses on the learning capabilities of the algorithm and presents the obtained accuracy given various datasets from the literature. The second is a test case where we apply our approach to a reinforcement learning agent for playing Blackjack, focusing on the explainability side.

5.1 Quantitative Evaluation

Testing this algorithm amounts to running it over a selection of benchmark datasets. Congressional Voting (mis, 1987) contains the voting patterns of several politicians. The task is to guess if they belong to the Republican party or Democrat party based on their votes. In this dataset, all attributes are categorical and binary. The Breast Cancer Wisconsin (Wolberg et al., 1995) dataset presents diagnostic data based on several attributes, such as the tumour size or position, having their value ranging from 1 to 10. The task is then to classify a tumour as benign or malignant. This dataset is interesting as the numerical values are not already segmented. Lastly, the Heart Disease (Janosi et al., 1988) analysis from Cleveland contains medical data with many numerical and categorical attributes such as blood pressure or the patient's sex. The task is to classify whether the patient has heart disease or not. This dataset has been proven to be difficult for learning algorithms. We should also note that each of the aforementioned datasets presents a binary classification task. The algorithm used as the benchmark for this approach was the C4.5 decision tree algorithm (Quinlan, 2014).

We decided to evaluate two versions of our approach, namely ARIA and n -ARIA⁴. The tests took the form of ten runs of one hundred iterations each over the datasets, which were split such that 70% comprised the training data, and the remaining 30% comprised the test data. In the evaluation table, the mean of the final convergence values across the ten runs is reported, along with the standard deviation. The code was written in C++, and executed on a machine with an 11th Gen Intel Core i9-11950H processor running at a 2.60GHz clock speed, with 8 physical cores and 16 logical cores, and with 32GB of memory.

The results of these tests, measured against the benchmark C4.5, are summarised in Table 2.

⁴No value for the Voting dataset as all the arguments are already expressed with their negation in the base dataset.

⁵Breast Cancer Wisconsin.

⁶Heart Disease Cleveland.

Table 2: Accuracy of the baseline (C4.5) and our approach (ARIA) on several datasets. Last column shows the accuracy with the use of the n -arguments.

Dataset	C4.5	ARIA	n -ARIA
Voting	94.3 \pm 2.4	95.7 \pm 1.9	-
BCW ⁵	93.5 \pm 1.5	94.4 \pm 1.2	95.6 \pm 1.4
HDC ⁶	60.9 \pm 3.7	79.3 \pm 3.0	78.7 \pm 2.4

As we can see, our approach proves to be more capable than C4.5 at learning over the various tested dataset, especially onto the Heart Disease dataset where the difference is statistically significant. On the other hand, the n -ARIA variant does not show a clear improvement. We believe that datasets usually feature enough attributes and values to avoid requiring extra expressiveness capabilities.

It is also relevant to note that our approach learned over the data in a reasonable time (*i.e.*, it could realistically be used) as it does not exceed three minutes for each dataset to perform one hundred iterations. We should also note that the final solution is usually found before reaching the iteration limit.

Additionally, an important result to mention is the growth of the graph size, and in particular the attack relation. This value can greatly differ between the dataset and the runs. In our tests, it approximately averages to 10 for the Voting dataset, 10 for the Breast Cancer Wisconsin dataset, and 25 for the Heart Disease dataset. Also, we should note that when computing a contextual graph in order to provide an explanation, unless the dataset is multi-valued, the number of nodes (*i.e.*, arguments) present in the graph cannot exceed the number of attributes of the dataset (plus τ and λ).

Similarly, the above tests were replicated for the n -ARIA algorithm, an extension of ARIA with the inclusion of n -arguments as described in Section 4.2. The datasets chosen to test n -ARIA were Breast Cancer Wisconsin (Wolberg et al., 1995), and Cleveland Heart Disease (Janosi et al., 1988). The exclusion of the Congressional Voting dataset for these tests was due to the arguments already being expressed in a true/false manner, as opposed to the two chosen benchmarks.

Compared to C4.5, n -ARIA performs slightly better on the Breast Cancer Wisconsin dataset, and significantly better on the Heart Disease dataset. However, when compared to ARIA, n -ARIA's performance is practically indistinguishable. On the contrary, n -ARIA shows as expected a longer computation time, usually more than thrice the time required by ARIA.

5.2 Test Case: RL-Blackjack

In order to qualitatively analyse our approach, we trained a Reinforcement Learning (RL) agent to play Blackjack. Then, a dataset was generated such that each entry included the state of play, and the action chosen by the agent, as suggested in Section 4.

5.2.1 Reinforcement Learning Environment

The rules of Blackjack are common knowledge, and can be summarised as follow: the goal is to beat the dealer's hand without exceeding a total card value of 21. Number cards retain their face value, face cards (Jack, Queen, King) are worth 10, and Aces can be worth 1 or 11, depending on the player's preference. Players are dealt two cards initially and can request additional cards (*i.e.*, HIT) to improve their hand. They can HIT as many times as they wish, and then STAND, which ends their turn. Additionally, they can DOUBLE, which is similar to HIT, but also doubles their initial bet (so in our environment, double the potential reward outcome). Using DOUBLE also ends the turn of the player. After all players have finished their turns, the dealer reveals their hand and must HIT until they reach a total of 17 or higher. If a player's hand exceeds 21, they bust and lose the round. If neither player nor dealer busts, the hand with a total closer to 21 wins. A natural blackjack (an Ace and a 10-value card) typically pays out 3:2.

In our environment, the reward is +1 if the agent wins, -1 if it loses, and respectively +2 and -2 if it doubled. Draws are worth 0, and having a blackjack multiply the reward by 1.5 according to the rules. As this is not the focus of this paper, we do not recall the equations related to reinforcement learning, such as Bellman's equation, but we encourage the reader to read the work from Wiering and Van Otterlo (Wiering and Van Otterlo, 2012) to better understand the functionality of this architecture.

After the learning phase, we generate a 600 lines dataset featuring some random state as input, and the action the agent would perform following its optimal policy as the label. We then ran ARIA with the following arguments (booleans having the values T or F for True or False):

Player-Sum (PS). The sum of the player's cards. Ranges from 4 to 21.

Dealer-Sum (DS). The sum of the dealer's cards. Ranges from 4 to 21.

Player-Less-Than-Dealer (P<D). Boolean stating that the player's total is less than the dealer's total.

Player-Above-11 (P11). Boolean stating the player’s total is above 11.

Dealer-Above-11 (D11). Boolean stating the dealer’s total is above 11.

Player-Has-Ace (PA). Boolean stating the player has an Ace.

Dealer-Has-Ace (DA). Boolean stating the dealer has an Ace.

Player-STAND (STAND). The target (τ). Corresponds to the player deciding to STAND (rather than HIT or DOUBLE).

5.2.2 Extracted Graph

After running ARIA over the data, we obtained the graph shown in Fig. 3. Examining the graph, it is relatively easy to interpret it—assuming sufficient knowledge of the game—and figure out what the strategy of the agent is. For the cases where it is above 17 (included) it prefers to STAND to avoid being busted. However, if it has 16 and that the dealer has 7, 8, or 10 (note that here, the value 9 does not appear, maybe because this case was not covered by the dataset), it prefers to HIT, as the chances that the dealer has a 10-value card are high, and this would result in a bigger value than the agent. Also, if the player has an ace, he can safely HIT as it cannot exceed the limit of 21. For the reason the agent systematically chooses to HIT if there is an ace.

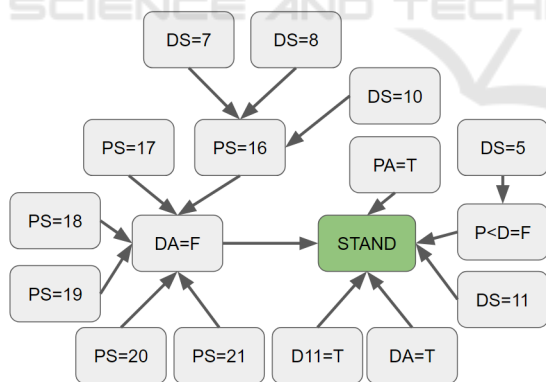


Figure 3: Universal Graph obtained after running ARIA on the RL-Blackjack dataset.

While we can see some other patterns appear in the graph, we want to outline the simplicity of extracting them by examining the visualisation (at least for someone familiar with the domain). While some domains may require an expert to look at the graph in order to understand the agent’s decision model (*e.g.*, medicine), some other domains are easier to analyse by laypeople (*e.g.*, autonomous driving).

6 DISCUSSIONS

As explained during the introduction, the aim of our approach is not to replace black-box algorithms such as neural networks, but rather to provide insight on what factors determine a certain decision.

This can serve either as a post-hoc explanation of an incident involving the black-box model, or as a way to verify before the deployment that the agent will behave correctly in specific situations. While this latter can be done by looking at the generated graph, it is recommended to compute the action of the black-box in parallel and compare it to the graph prediction, as in the event they differ, the explanation provided by the graph might not be accurate.

In Section 5, we compared our approach to a benchmark on several tabular datasets presenting a binary classification task. We focus on binary classification since the aim was not to explain the entire agent’s decision model, but rather solely to justify the choice of a particular action. For this reason, we should also note that even though the datasets we chose are noisy and currently no state of the art approach is able to achieve a perfect score, improving noise resiliency is not very relevant for our approach as in our case, those data would be obtained from the black-box itself, which usually would not have any stochastic behaviour in its decisions.

Furthermore, we evaluated the *n*-ARIA variant which should provide more possibilities to express the underlying patterns of the dataset. However, as shown by our experiments, it does not clearly enhance the results obtained by ARIA, while at the same time greatly increasing computing time. As such, we do not recommend its usage.

While we acknowledge the fact that another rule induction approach could be used to achieve something similar to our application case, we want to outline an advantage of using ARIA which resides in its readability. Indeed, among the various rule induction approaches mentioned in Section 2, most of them end up with a large set of rules (usually more than a hundred on datasets such as Heart Disease Cleveland). We believe that such a number is beyond the capacity for a layperson to detect patterns and recognise key elements leading to the decision.

Last, as mentioned in introduction and even if this was not our original focus, we believe that such an argumentation rule induction model could be particularly relevant for several application domains related to multi-agent systems. First, some approaches from the field of Machine Ethics are relevant. In the Jiminy architecture from Liao *et al.* (Liao *et al.*, 2019; Liao *et al.*, 2023), agents are modelled by an argumenta-

tion framework which is then used to solve normative conflicts among them. While those graphs could in principle be handcrafted, we believe that using ARIA is more scalable and less prone to human bias. In some other approaches, such as Alcaraz *et al.* (Alcaraz et al., 2023), argumentation graphs are used to model judging agents building a reward for some reinforcement learning agent. Using our algorithm could enable the possibility of automatically building those judging agents over a population of agents already interacting in a normative environment. Lastly, our approach can be used to allow some agents to reason and negotiate about norms, as it creates a way to organise the arguments and attacks, as well as computing them, as proposed by Yu *et al.* (Yu et al., 2021), which introduces the notion of individual and collective defence.

7 CONCLUSION AND FUTURE WORKS

In this paper, we presented ARIA, a novel approach to rule induction which combines heuristic search with formal argumentation to represent the decision model of a black-box algorithm. We first outlined the advantages of using formal argumentation over classical propositional logic, as well as the potential applications. Then, we have shown that our approach obtains satisfactory results over selected tabular datasets containing both categorical and numerical information, and can be ran in an acceptable amount of real time. Finally, we presented a short use-case showcasing our approach when used on an RL agent.

Perspectives and Research Directions

To conclude this paper, we would like to share potential directions for future work.

A natural future work is to try our approach over some real scenarios, as for now we only try it over benchmark datasets which are not reflective of the way an agent would make decisions.

Also, we currently stick to the argumentation as defined by Dung. Nevertheless, we believe that using select modern frameworks, such as bipolar argumentation, could greatly improve the expressivity of the generated graphs. As such, we recommend exploring various possibilities in the field of formal argumentation.

Additionally, we acknowledge that the current way we generate the arguments could be too basic, and that having a human expertly designing them is not easily scalable. For this reason, we would like to seize the opportunity of the recent advances

in the field of ChatBots and Large Language Models (LLMs), and use their generative capabilities to automatically produce more sophisticated arguments. While we do not provide any implementation yet, we believe that a good starting point would be to provide an Application Programming Interface (API) to the LLM so it could design more complex arguments based on pre-processed elements. This would especially improve the handling of the numerical attributes.

Last, we believe that our architecture could be improved in order to reach higher accuracy. In particular, we would like to treat the multi-class tasks by having all the actions appearing in the same graph. This would allow using the information that the agent will prefer an action to another in certain situations, resulting in more modelling possibilities. Furthermore, we would like to implement a method for a better segmentation of the continuous numerical attributes, both reducing the number of generated arguments, and making them more meaningful.

ACKNOWLEDGMENTS

This research is supported by the Luxembourg National Research Fund (FNR): IPBG2020/IS/14839977/C21.

REFERENCES

- (1987). Congressional Voting Records. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5C01P>.
- Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160.
- Alcaraz, B., Boissier, O., Chaput, R., and Leturc, C. (2023). Ajar: An argumentation-based judging agents framework for ethical reinforcement learning. In *AA-MAS'23: International Conference on Autonomous Agents and Multiagent Systems*.
- Bagallo, G. and Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine learning*, 5:71–99.
- Baumann, R. (2017). On the nature of argumentation semantics: Existence and uniqueness, expressibility, and replaceability. *Journal of Applied Logics*, 4(8):2779–2886.
- Boella, G., Gabbay, D. M., van der Torre, L., and Villata, S. (2009). Meta-argumentation modelling i: Methodology and techniques. *Studia Logica*, 93:297–355.
- Bohanec, M. (1997). Car Evaluation. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5JP48>.

- Catlett, J. (1991). Mega induction: A test flight. In *Machine Learning Proceedings 1991*, pages 596–599. Elsevier.
- Cohen, W. W. (1995). Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier.
- Doutre, S., Duchatelle, T., and Lagasquie-Schiex, M.-C. (2023). Visual explanations for defence in abstract argumentation. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 2346–2348. ACM.
- Dung, P. M. (1995). On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357.
- Fan, X. and Toni, F. (2014). On computing explanations in abstract argumentation. In *ECAI 2014*, pages 1005–1006. IOS Press.
- Fürnkranz, J. and Widmer, G. (1994). Incremental reduced error pruning. In *Machine learning proceedings 1994*, pages 70–77. Elsevier.
- Goodman, B. and Flaxman, S. (2017). European union regulations on algorithmic decision-making and a “right to explanation”. *AI magazine*, 38(3):50–57.
- Hein, D., Udluft, S., and Runkler, T. A. (2018). Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence*, 76:158–169.
- Janosi, A., Steinbrunn, W., Pfisterer, M., and Detrano, R. (1988). Heart Disease. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C52P4X>.
- Lewis, D. (2013). *Counterfactuals*. John Wiley & Sons.
- Liao, B., Anderson, M., and Anderson, S. L. (2021). Representation, justification, and explanation in a value-driven agent: an argumentation-based approach. *AI and Ethics*, 1(1):5–19.
- Liao, B., Pardo, P., Slavkovik, M., and van der Torre, L. (2023). The jiminy advisor: Moral agreements among stakeholders based on norms and argumentation. *Journal of Artificial Intelligence Research*, 77:737–792.
- Liao, B., Slavkovik, M., and van der Torre, L. (2019). Building jiminy cricket: An architecture for moral agreements among stakeholders. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 147–153.
- Liu, G., Schulte, O., Zhu, W., and Li, Q. (2018). Toward interpretable deep reinforcement learning with linear model u-trees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 414–429. Springer.
- Liu, J. J. and Kwok, J. T.-Y. (2000). An extended genetic rule induction algorithm. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*, volume 1, pages 458–463. IEEE.
- McBurney, P. and Parsons, S. (2004). Locutions for argumentation in agent interaction protocols. In *International Workshop on Agent Communication*, pages 209–225. Springer.
- Nofal, S., Atkinson, K., and Dunne, P. E. (2021). Computing grounded extensions of abstract argumentation frameworks. *The Computer Journal*, 64(1):54–63.
- Puiutta, E. and Veith, E. M. (2020). Explainable reinforcement learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*, pages 77–95. Springer.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1:81–106.
- Quinlan, J. R. (1987). Generating production rules from decision trees. In *ijcai*, volume 87, pages 304–307. Citeseer.
- Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.
- Rizzo, L. and Longo, L. (2018). A qualitative investigation of the degree of explainability of defeasible argumentation and non-monotonic fuzzy reasoning.
- Sabbatini, F., Ciatto, G., Calegari, R., Omicini, A., et al. (2021). On the design of psyke: a platform for symbolic knowledge extraction. In *CEUR WORKSHOP PROCEEDINGS*, volume 2963, pages 29–48. Sun SITE Central Europe, RWTH Aachen University.
- Selbst, A. and Powles, J. (2018). “meaningful information” and the right to explanation. In *conference on fairness, accountability and transparency*, pages 48–48. PMLR.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
- Tan, S., Caruana, R., Hooker, G., and Lou, Y. (2018). Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 303–310.
- Venturini, G. (1993). Sia: a supervised inductive algorithm with genetic search for learning attributes based concepts. In *European conference on machine learning*, pages 280–296. Springer.
- Verma, A., Murali, V., Singh, R., Kohli, P., and Chaudhuri, S. (2018). Programmatically interpretable reinforcement learning. In *International Conference on Machine Learning*, pages 5045–5054. PMLR.
- Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841.
- Weiss, S. M. and Indurkha, N. (1991). Reduced complexity rule induction. In *IJCAI*, pages 678–684.
- Wiering, M. A. and Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, 12(3):729.
- Wolberg, W., Mangasarian, O., Street, N., and Street, W. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5DW2B>.
- Yu, L., Chen, D., Qiao, L., Shen, Y., and van der Torre, L. (2021). A principle-based analysis of abstract agent argumentation semantics. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 18, pages 629–639.
- Zahavy, T., Ben-Zrihem, N., and Mannor, S. (2016). Gray-ing the black box: Understanding dqns. In *International Conference on Machine Learning*, pages 1899–1908. PMLR.