

# On the Tradeoff Between Correctness and Completeness in Argumentative Explainable AI

Nico Potyka<sup>1</sup>, Xiang Yin<sup>1</sup> and Francesca Toni<sup>1</sup>

<sup>1</sup>Imperial College London, UK

## Abstract

Explainable AI aims at making the decisions of autonomous systems human-understandable. Argumentation frameworks are a natural tool for this purpose. Among them, bipolar abstract argumentation frameworks seem well suited to explain the effect of features on a classification decision and their formal properties can potentially be used to derive formal guarantees for explanations. Two particular interesting properties are correctness (if the explanation says that  $X$  affects  $Y$ , then  $X$  affects  $Y$ ) and completeness (if  $X$  affects  $Y$ , then the explanation says that  $X$  affects  $Y$ ). The reinforcement property of bipolar argumentation frameworks has been used as a natural correctness counterpart in previous work. Applied to the classification context, it basically states that attacking features should decrease and supporting features should increase the confidence of a classifier. In this short discussion paper, we revisit this idea, discuss potential limitations when considering reinforcement without a corresponding completeness property and how these limitations can potentially be overcome.

## Keywords

Argumentation, Abstract Bipolar Argumentation, Explainable AI

## 1. Introduction

Automatic decision making is increasingly driven by black-box machine learning models. However, their opaqueness raises questions about fairness, reliability and safety. Explanation methods aim at making the decision process transparent [1]. In recent years, various explanation methods have been proposed in the argumentation literature, we refer to [2, 3] for an overview.

One easily comprehensible argumentation model are bipolar abstract argumentation frameworks. They represent arguments in a graph, where nodes correspond to abstract arguments (entities that can be accepted or rejected) and edges to attack or support relationships between them. This representation seems well suited to represent the influence of features in a classification problem on the class decision. For example, in a credit approval setting, the income may have a positive effect (support), while existing debts may have a negative effect (attack) on the decision. This is in line with the idea of the *reinforcement property* in bipolar argumentation [4], which roughly states that attacks should decrease and supports should increase the strength of the addressed argument. More precisely, in a quantitative setting, the effect should be relative to the strength of the attacker/supporter and our a priori belief in the addressed argument.


---

1st International Workshop on Argumentation for eXplainable AI (ArgXAI, co-located with COMMA '22), September 12, 2022, Cardiff, UK

✉ n.potyka@imperial.ac.uk (N. Potyka); x.yin20@imperial.ac.uk (X. Yin); f.toni@imperial.ac.uk (F. Toni)



© 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The *faithfulness* of an argumentative explanation to the classifier can then naturally be evaluated by checking to which extent the reinforcement property is satisfied. In recent work, authors actually guaranteed perfect faithfulness by showing only those connections that respect reinforcement [5]. While this is a highly desirable correctness property of the explanation, it leaves the question to which extent the explanation is complete. That is, while it is guaranteed that all shown relationships have indeed the intended effect, it remains unclear if all effects on the decision have been captured by the explanation.

We argue that, while reinforcement is a desirable property, it can be too strong when the arguments in the argumentation framework are not carefully selected. In particular, introducing a single argument per feature is often not sufficient. One reason is that classifiers are often non-monotonic and a feature may have a positive effect in one and a negative effect in another region of the input domain. Another reason is that the effect of features often cannot be captured independently of the other features. Of course, this does not mean that we should give up reinforcement. Rather, we should refine the bipolar abstract argumentation frameworks based on the application domain to capture more complicated effects of features and to improve the completeness of the explanation.

## 2. Probabilistic Classifiers and Correct Argumentative Explanations

The abstract goal of classification is to map inputs  $\mathbf{x}$  to outputs  $y$ . We think of the inputs as vectors  $\mathbf{x} = (x_1, \dots, x_k)$ , where the  $i$ -th value is taken from some domain  $D_i$ . We let  $\mathcal{D} = \times_{i=1}^k D_i$  denote the cartesian product of the individual domains. Given an input  $\mathbf{x} = (x_1, \dots, x_k)$ , with a slight abuse of notation, we let  $(\mathbf{x}_{-i}, x) = (x_1, \dots, x_{i-1}, x, x_{i+1}, \dots, x_k)$  denote the input where the  $i$ -th component has been replaced with  $x$ . The output  $y$  is taken from a finite set  $L$  of class labels. A *classification problem*  $P = ((D_1, \dots, D_k), L, E)$  consists of  $k$  feature domains  $D_i$ , class labels  $L$  and a set of training examples  $E = \{(\mathbf{x}_i, y_i) \mid 1 \leq i \leq N, \mathbf{x}_i \in \mathcal{D}, y_i \in L\}$ . For example, in a credit approval scenario, the first feature could be *Age* with corresponding domain  $D_1 = \mathbb{N}$ , the second feature *Income* with domain  $D_2 = \mathbb{R}$  and the third feature *Debts* with domain  $D_3 = \{0, 1\}$ . In this case, the label set could be  $L = \{0, 1\}$  indicating whether the credit application is accepted or rejected.  $E$  then consists of previous cases, that are used to train a classifier. We will not be concerned with training here and just assume that a classifier is given and is supposed to be explained.

A *probabilistic classifier* is a function  $C : \mathcal{D} \times L \rightarrow [0, 1]$  that assigns a probability  $C(\mathbf{x}, y)$  to every pair  $(\mathbf{x}, y)$  such that  $\sum_{l \in L} C(\mathbf{x}, l) = 1$ .  $C(\mathbf{x}, y) \in [0, 1]$  can be understood as the confidence of the classifier that an example with features  $\mathbf{x}$  belongs to the class  $y$ . A deterministic classifier is the special case, where the probabilistic classifier always assigns probability 1 to one label and 0 to all others. In general, a classification decision can be made by picking the label with the highest probability or by defining a threshold value for the probability. To simplify notation, we will often write  $C_y(\mathbf{x})$  instead of  $C(\mathbf{x}, y)$  in the remainder.

A *bipolar argumentation graph (BAG)* is a tuple  $(\mathcal{A}, \text{Att}, \text{Sup})$  consisting of a set of abstract arguments and attack and support relationships between them [6]. Here, BAGs are mainly used as a tool to represent the effects of features on the class label and we will not discuss

their various semantics and extensions in more detail. For our purposes, the most important semantical property is reinforcement [4], which basically demands that attackers should weaken and supporters should strengthen an argument.

In order to explain a classifier, we can associate the underlying classification problem with a set of abstract arguments that allow us to argue about the classification decision. The most straightforward way to do this is to introduce one argument per feature and one argument per class label.

**Definition 1** (Naive Classification Arguments). Given a classification problem  $P = ((D_1, \dots, D_k), L, E)$ , the *naive classification arguments* associated with  $P$  are the

- $k$  feature arguments  $A_i^F$ ,  $1 \leq i \leq k$  for the  $k$  features and
- $|L|$  class arguments  $A_i^C$ ,  $1 \leq i \leq k$  for the class labels

We can then build an *explanation BAG* from a probabilistic classifier by taking the naive classification arguments and adding support and attack edges from feature arguments to class arguments.

Roughly speaking, we say that an *explanation BAG satisfies reinforcement* if the following two conditions are satisfied:

1. If there is an attack from  $A_i^F$  to  $A_j^C$ , then increasing (decreasing) the value of the  $i$ -th feature decreases (increases) the probability of the  $j$ -th class and
2. If there is an support from  $A_i^F$  to  $A_j^C$ , then increasing (decreasing) the value of the  $i$ -th feature increases (decreases) the probability of the  $j$ -th class.

While there is a natural order for boolean and ordinal features, the definition has to be made more precise for categorical features. However, it is sufficient for our purposes as we mainly want to create awareness for potential limitations of this idea and how they can be addressed.

As discussed in the introduction, reinforcement is an interesting correctness/faithfulness property of an explanation. However, we should not only consider correctness, but also completeness of the explanation. As an extreme example, the empty graph defined over the naive classification arguments satisfies reinforcement. However, it is not a very interesting explanation because it does not explain anything. In general, a corresponding completeness property is desirable that explains to which extent the explanation graph captures the existing relationships. As a first step in this direction, we explain reasons for why an explanation BAG can result in incomplete explanations.

### 3. Completeness Problems for Boolean Data

To begin with, let us focus on boolean data. That is, we assume that we have  $D_i = \{0, 1\}$  for all features. In this case, there is a straightforward (even though computationally expensive) way to create an explanation BAG that satisfies reinforcement.

**Definition 2** (Naive Explanation BAG). The *naive explanation BAG* for a probabilistic classifier  $C : \mathcal{D} \times L \rightarrow [0, 1]$  for a classification problem  $P$  is the BAG  $(\mathcal{A}, \text{Att}, \text{Sup})$ , where

- $\mathcal{A}$  contains all feature and class arguments for  $P$ ,

- there is an attack edge from  $A_i^F$  to  $A_j^C$  iff for every assignment  $\mathbf{x}_{-i}$  to the remaining features, changing the  $i$ -th feature from 0 to 1 decreases the probability of  $l_j$ , that is,  $C((\mathbf{x}_{-i}, 0), l_j) > C((\mathbf{x}_{-i}, 1), l_j)$ ,
- there is a support edge from  $A_i^F$  to  $A_j^C$  iff for every assignment  $\mathbf{x}_{-i}$  to the remaining features, changing the  $i$ -th feature from 0 to 1 increases the probability of  $l_j$ , that is,  $C((\mathbf{x}_{-i}, 0), l_j) < C((\mathbf{x}_{-i}, 1), l_j)$ ,
- there are no other edges.

Let us note that one may also want to introduce attack arguments between the class arguments (only one class argument should be accepted). We refrain from doing so here because, at this point, we consider the explanation BAG merely as a visualization of the relationships between feature and class arguments. However, when applying argumentation semantics to the explanation BAG, these additional edges may be necessary.

While constructing the naive explanation BAG in a straightforward way takes exponential time, it is easy to check that it satisfies reinforcement. However, in many cases, it will only explain a fraction of the actual effects of features because it only looks at the individual effects of features.

To illustrate the problem, consider the XOR function  $XOR(A, B)$  that returns 1 if exactly one of  $A$  and  $B$  is 1. In this case, our explanation BAG would actually not contain any edges. However,  $A$  and  $B$  clearly have an effect on the decision, so that the explanation BAG is not an accurate explanation even though it satisfies reinforcement. The problem in this example is that the effect of features cannot be determined independently of each other. While the XOR function is an extreme example, similar dependencies naturally occur in many datasets. For example, in a credit approval setting, the highest educational degree may be relevant for an application if the applicant has a low income (e.g., a student after graduation), but not if the applicant has a high income. In a medical scenario, a drug may be an effective treatment for most patients, but could be detrimental for patients with particular medical or physical conditions.

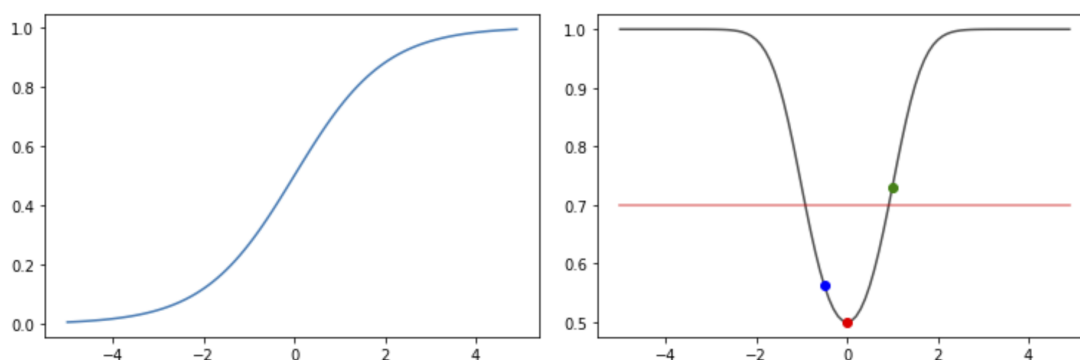
In order to take account of such joint effects, we need to refine our naive explanation BAG. Two natural candidate refinements are the following.

1. **Increase the BAG:** Introduce joint feature arguments that capture the state of multiple arguments simultaneously.
2. **Consider more expressive BAGs:** Introduce joint attacks/ joint supports that capture the joint effect of arguments rather than considering them independently. Joint effects can be represented, for example, by logical formulas similar to ADFs [7] or by considering abstract joint edges as considered in classical [8] and weighted SETAFs [9].

## 4. Completeness Problems for Continuous Data

Let us now look at continuous data. That is, we assume that  $D_i = [l_i, u_i]$ ,  $l_i, u_i \in \mathbb{R}$ ,  $l_i < u_i$  is a real interval for all features. Now, the effect of features may not only depend on the effect of other features, but the effects of features can actually change on the domain.

To illustrate the problem, consider a simple binary classification problem over a single feature with domain  $D = \mathbb{R}$ . Intuitively, an input should be classified as positive if the value of the



**Figure 1:** Function graphs of logistic function  $\phi_l(x)$  (left) and non-monotonic classifier  $C_1(x) = \phi_l(x^2)$  (right).  $x$  is classified as an anomaly if  $C_1(x) \geq 0.7$  (orange line). Increasing  $x$  can have a negative (blue point), neutral (red point) or positive (green point) effect on the probability dependent on the region where it is located.

feature is “sufficiently far away from 0”. A typical example is anomaly detection, where the feature value corresponds to the deviation of an observation from the mean or median. Consider the classifier  $C_1(x) = \phi_l(x^2)$ , where  $\phi_l(z) = \frac{1}{1+\exp(-z)}$  is the logistic function. Figure 1 shows the graphs of the functions  $\phi_l(x)$  and  $C_1(x)$ .

Intuitively,  $\phi_l(x)$  squashes its input between 0 and 1. While the logistic function  $\phi_l$  is monotonically increasing, our classifier  $C_1(x) = \phi_l(x^2)$  is monotonically decreasing for  $x < 0$  and monotonically increasing for  $x > 0$ . Hence, similar to our previous XOR example, we are again in a situation where our feature can neither be characterized as an attacker nor as a supporter even though it clearly has an effect on the classification. Let us note that non-monotonic behaviour naturally occurs in many domains. For example, when predicting health risks, the probability often increases when particular health markers deviate substantially from a default value. For instance, both underweight and overweight and both low and high blood pressure could be seen as red flags. For spatial features, even multiple changes in the behaviour of a classifier can occur. One example are latitude and longitude when predicting property demand in real estate datasets because areas with high and low popularity are often spread throughout cities.

In order to deal with non-monotonicity, we basically have to consider value intervals for features rather than just features. For example,  $C_1(x)$  in Figure 1 is decreasing on the open interval  $(-\infty, 0)$  and increasing on  $(0, \infty)$ . However determining these intervals can be challenging in general. For example, when the classifier is a large neural network, it is hard to tell at which points the model changes its behaviour. Taking many small intervals will increase the chance that the model behaves monotonically on the interval, but can potentially blow up the explanation unnecessarily. Taking too large intervals increase the chance that the model is non-monotonic on the interval, so that the explanation may miss important effects (reducing completeness) or may misrepresent the effect (violating reinforcement). However, let us note that finding good intervals is easier for discrete classifiers like tree ensembles. This is because they internally discretize continuous features, so that the potential critical points can be found by traversing all trees in the ensemble.

To summarize, for continuous features, we do not only have to take account of joint effects of features, but also of their potential non-monotonicity. In addition to the two refinements proposed at the end of the previous section, we may consider the following variations to improve completeness when features are continuous.

1. **Increase the BAG:** Instead of having a single feature argument per feature, we can consider multiple arguments that represent the influence of the feature in a particular region.
2. **Consider more expressive BAGs:** similar to the boolean setting, joint attacks/supports can capture the joint effects of continuous feature arguments that take a value in a particular region or conditional effects based on the state of boolean feature arguments.

## 5. Towards Characterizing Correct and Complete Explanations

Before closing the paper, let us briefly discuss a sufficient condition under which the naive explanation BAG can accurately represent a probabilistic classifier. The condition that we consider here is that the classifier behaves monotonically with respect to every individual feature (independent of the remaining features). We call this property *Strong Monotonicity*. To talk about monotonicity, we have to assume that all domains are ordered. Note that this applies to boolean features ( $0 < 1$ ) as well as discrete ( $n < n + 1$ ) and continuous features. Formally, we call a classifier *monotonically increasing (resp. decreasing) wrt. the label  $y$  and the  $i$ -th feature* iff for all inputs  $\mathbf{x} \in \mathcal{D}$ , and  $x'_i \in D_i$ ,  $x_i < x'_i$  implies that  $C_y(\mathbf{x}) \leq C_y((\mathbf{x}_{-i}, x'_i))$  (resp.  $C_y(\mathbf{x}) \geq C_y((\mathbf{x}_{-i}, x'_i))$ ) and  $x_i > x'_i$  implies that  $C_y(\mathbf{x}) \geq C_y((\mathbf{x}_{-i}, x'_i))$  (resp.  $C_y(\mathbf{x}) \leq C_y((\mathbf{x}_{-i}, x'_i))$ ). If these conditions hold with strict inequality, we call the classifier *strictly monotonically increasing (resp. decreasing) wrt.  $y$  and the  $i$ -th feature*. We call a classifier *strongly monotonic* if for every label and for every feature, the classifier is strictly monotonically increasing or strictly monotonically decreasing.

Let us note that if our data is boolean and the classifier is strongly monotonic, then the naive explanation BAG accurately captures all effects by definition. If we have continuous features, then building up the explanation BAG can be difficult because an edge can only be added when the reinforcement condition can be verified for all domain values. As the domain is infinite, this is not always possible. However, if the classifier is composed of differentiable functions like in the case of many neural networks, then it is theoretically possible to apply symbolic differentiation techniques to compute the partial derivatives with respect to all features. Note that the classifier is strongly monotonic if and only if all partial derivatives depend only on the feature itself (independence) and are guaranteed to be always negative (decreasing) or positive (increasing). The conditions for the naive explanation BAG can be checked analogously.  $A_i^F$  is an attacker (supporter) of  $A_j^C$  iff the partial derivative of  $C_{l_j}(\mathbf{x})$  with respect to  $\mathbf{x}_i$  depends only on  $\mathbf{x}_i$  and is always negative (positive). In this setting (continuous data, differentiable classifier), the gradient does indeed have several advantages over other feature attribution methods that evaluate features by numerical scores [10].

Formally, it would be interesting to characterize exactly under which conditions certain explanation BAGs can correctly and completely capture the behaviour of classification models.

Intuitively, it seems that the naive explanation BAG can satisfy both reinforcement and completeness if and only if the classifier satisfies a notion (probably a refinement) of strong monotonicity. However, a formal investigation requires a more rigorous definition of completeness that is out of scope here.

## 6. Conclusions

We revisited the notion of reinforcement of bipolar abstract argumentation classifiers. While the property is important, we believe that it should be considered in combination with a completeness property that guarantees that reinforcement is not satisfied in a trivial way. As we illustrated with some examples, reinforcement and completeness cannot be satisfied simultaneously when the explanation BAG is not sufficiently expressive. We identified two main reasons for this:

1. The effect of features cannot always be determined independently of the state of other features.
2. The effect of features can be non-monotonic and may be positive in one region and negative in another.

For these reasons, a naive explanation BAG that only considers individual features and their effects on class labels, is necessarily incomplete or incorrect. Possible workarounds include adding joint-features or joint-attacks and -supports, and discretizing continuous features into bins on which the classifier behaves monotonically. However, the latter task can be challenging for some models, and one may have to sacrifice correctness to improve completeness. One notable exception may be decision tree ensembles where the critical points can directly be extracted from the trees in the ensemble.

An interesting research direction corresponding to the correctness-completeness-tradeoff is to characterize for which classification models we can find explanation BAGs that satisfy both reinforcement and completeness. We conjecture that, for the naive explanation BAG, the characterizing property of the classification model is some notion of strong monotonicity, that is, the classifier should be

1. monotonic with respect to every feature,
2. independent of the state of the other features.

Finally, let us note that another way to guarantee an accurate representation of classification models is to establish a one-to-one relationship between classifiers and argumentation frameworks. For example, many multilayer perceptrons can be perfectly represented by gradual BAGs [11]. In this case, we trivially satisfy reinforcement and completeness, but the explanation can be trivial itself in the sense that it is not easier to comprehend than the original classifier. Clustering techniques that have recently been considered for argumentation frameworks [12] could be interesting to obtain more comprehensible explanations. However, it remains to be seen to which extent such compression techniques allow maintaining reinforcement and completeness.

## Acknowledgements

This research was partially funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 101020934). Francesca Toni was partially funded by J.P. Morgan and by the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme. Any views or opinions expressed herein are solely those of the authors listed.

## References

- [1] A. Adadi, M. Berrada, Peeking inside the black-box: a survey on explainable artificial intelligence (xai), *IEEE access* 6 (2018) 52138–52160.
- [2] K. Cyras, A. Rago, E. Albini, P. Baroni, F. Toni, Argumentative XAI: A survey, in: Z. Zhou (Ed.), *International Joint Conference on Artificial Intelligence, IJCAI*, ijcai.org, 2021, pp. 4392–4399.
- [3] A. Vassiliades, N. Bassiliades, T. Patkos, Argumentation and explainable artificial intelligence: a survey, *The Knowledge Engineering Review* 36 (2021).
- [4] L. Amgoud, J. Ben-Naim, Weighted bipolar argumentation graphs: Axioms and semantics, in: J. Lang (Ed.), *International Joint Conference on Artificial Intelligence, IJCAI*, ijcai.org, 2018, pp. 5194–5198.
- [5] A. Rago, P. Baroni, F. Toni, Explaining causal models with argumentation: the case of bi-variate reinforcement, in: *KR, 2022*, p. TBA.
- [6] L. Amgoud, C. Cayrol, M. Lagasquie-Schiex, On the bipolarity in argumentation frameworks, in: J. P. Delgrande, T. Schaub (Eds.), *International Workshop on Non-Monotonic Reasoning (NMR)*, 2004, pp. 1–9.
- [7] G. Brewka, S. Woltran, Abstract dialectical frameworks, in: F. Lin, U. Sattler, M. Truszczynski (Eds.), *12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*, AAAI Press, 2010.
- [8] S. H. Nielsen, S. Parsons, A generalization of dung’s abstract framework for argumentation: Arguing with sets of attacking arguments, in: *International Workshop on Argumentation in Multi-Agent Systems*, Springer, 2006, pp. 54–73.
- [9] B. Yun, S. Vesic, Gradual semantics for weighted bipolar setafs, in: J. Vejnárová, N. Wilson (Eds.), *European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2021)*, volume 12897 of *Lecture Notes in Computer Science*, Springer, 2021, pp. 201–214.
- [10] N. Potyka, X. Yin, F. Toni, Towards a theory of faithfulness: Faithful explanations of differentiable classifiers over continuous data, *CoRR abs/2205.09620* (2022). URL: <https://doi.org/10.48550/arXiv.2205.09620>. doi:10.48550/arXiv.2205.09620. arXiv:2205.09620.
- [11] N. Potyka, Interpreting neural networks as gradual argumentation frameworks, in: *AAAI Conference on Artificial Intelligence (AAAI)*, 2021, pp. 6463–6470.
- [12] Z. G. Saribatur, J. P. Wallner, Existential abstraction on argumentation frameworks via clustering, in: M. Bienvenu, G. Lakemeyer, E. Erdem (Eds.), *International Conference on Principles of Knowledge Representation and Reasoning, KR*, 2021, pp. 549–559.