Everyday Argumentative Explanations for Classification

Jowan van Lente[1], AnneMarie Borg[1], and Floris Bex[1,2]

[1]Department of Information and Computing Sciences, Utrecht University
[2]Tilburg Institute for Law, Technology, and Society, Tilburg University

Abstract

In this paper we study *everyday explanations* for classification tasks with formal argumentation. Everyday explanations describe how humans explain in day-to-day life, which is important when explaining decisions of AI systems to lay users. We introduce *EVAX*, a model-agnostic explanation method for classifiers with which contrastive, selected and social explanations can be generated. The resulting explanations can be adjusted in their size and retain high fidelity scores (an average of 0.95).

1 Introduction

A recent trend in explainable artificial intelligence (XAI) is hybrid (or neuro-symbolic) approaches, where the performance of learning-based systems is combined with the transparency of knowledge-based AI [6]. One such knowledge-based approach that seems suitable for this purpose is *formal argumentation* [12], see e.g., [10]. Formal argumentation is designed to model the argumentative nature and defeasible character of human reasoning, by means of argumentation frameworks: a set of arguments and an attack relation between these arguments. Although *argumentative XAI* is relatively new, several methods have been proposed, see [10] for a recent overview.

In this paper we are interested how formal argumentation can contribute to the modeling of explanations, as described in [16]: explanations of a specific event or decision for human (non-expert) end users. Specifically, we study:

- *everyday explanations*: explanations as used by humans in day-to-day life. Unlike scientific explanations, these need not be based on general laws. We will focus on local explanations (i.e., explanations for a specific outcome) and assume a receiver who benefits from a smaller explanation.
- *contrastive, selected and social explanations*: among the main findings in [16] is that explanations are: *contrastive*, when explaining P, humans often expect the explanation to highlight the difference between P and something else (e.g., Q), the explanation answers *why P rather than Q?*; *selected*, not all possible explanations are returned, but rather just one or two are selected based on a cognitive bias, such as abnormality or responsibility; and *social*, explanations (e.g., their size or content) are adjusted to the receiver.
- *faithfulness*: when explaining the outcome of a black box, learning-based system, it should be faithful to the system and its mechanisms.

In order to model the above properties in an argumentative explanation method, we introduce *EVAX*, an argumentative explanation method for everyday explanations of decisions derived with a classifier. *EVAX* is a model-agnostic method, which only requires the input and output of a classifier and can then compute, faithfully, explanations which are contrastive, selected and social.

Published in the Proceedings of the 1st International Workshop on Argumentation for eXplainable AI.

The paper is structured as follows. Section 2 contains the preliminaries after which *EVAX* is introduced (Section 3). We present a quantitative (Section 4) and qualitative (Section 5) evaluation. Related work is discussed in Section 6 and we conclude in Section 7.

2 Preliminaries

In this section we recall the necessary preliminaries on formal argumentation and classification tasks and present our definition of arguments and defeats.

2.1 Formal argumentation

An *abstract argumentation framework* (AF) [12] is a pair $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Def} \rangle$, where Args is a set of *arguments* and $\mathsf{Def} \subseteq \mathsf{Args} \times \mathsf{Args}$ is an *defeat relation* on these arguments. Given an argumentation framework $\mathcal{AF}$, Dung-style semantics [12] can be applied to it, to determine what combinations of arguments (called *extensions*) can collectively be accepted.

Definition 1. *Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Def} \rangle$ be an AF, $\mathcal{S} \subseteq \mathsf{Args}$ be a set of arguments and $A \in \mathsf{Args}$ an argument. Then $\mathcal{S}$* defeats *A if there is an $A' \in \mathcal{S}$ such that $(A', A) \in \mathsf{Def}$; $\mathcal{S}$* defends *A if $\mathcal{S}$ defeats every defeater of A; $\mathcal{S}$ is* conflict-free *if there are no $A_1, A_2 \in \mathcal{S}$ such that $(A_1, A_2) \in \mathsf{Def}$. $\mathcal{S}$ is* admissible *if it is conflict-free and it defends all of its elements, $\mathcal{S}$ is* complete *if it is admissible and it contains all the arguments it defends. The grounded extension of $\mathcal{AF}$ is the minimal (w.r.t. $\subseteq$) complete extension, denoted by $\mathsf{Grd}(\mathcal{AF})$.*

In abstract argumentation, arguments are abstract entities and the attack relation is predetermined. In contrast, in structured argumentation [3], arguments are constructed from a knowledge base and a set of rules and the attacks are based on the structure of the resulting arguments. In both cases, the strength of the arguments determines whether an attack is successful (e.g., an attack by a stronger argument is successful and therefore also a defeat, but an attack by a weaker argument is not successful). While there is a variety of approaches to structured argumentation, we will use a simple notion of an argument: a triple of a premise, a conclusion and the strength of the argument.

Definition 2. *An* argument *A is a triple (ψ, ϕ, p), where ψ is the* premise *(e.g., a feature), denoted by $\mathtt{prem}(A) = \psi$, ϕ is the* conclusion *inferred from ψ (e.g., a class), denoted by $\mathtt{conc}(A) = \phi$ and p is the strength value of the argument, denoted by $\mathtt{str}(A) = p$ where $0 \leq p \leq 1$.*

Based on the structure of the arguments, the defeat relation is determined:

Definition 3. *Let $\mathcal{AF} = \langle \mathsf{Args}, \mathsf{Def} \rangle$ be an AF and $A, B \in \mathsf{Args}$. Then $(A, B) \in \mathsf{Def}$ iff $\mathtt{conc}(A) \neq \mathtt{conc}(B)$ and $\mathtt{str}(A) \geq \mathtt{str}(B)$.*

2.2 Classification

As mentioned in the introduction, in this paper we are interested in explaining the outcome of a classification task with argumentation. Intuitively, classification is an inference task in which it is checked whether an object (e.g., an image, sound or text file) belongs to a category [19].

Definition 4. *A feature is an attribute-value pair $(a, v) \in \mathcal{F}$, where a is the* label *of the feature and v is its corresponding* value*. Let $\mathcal{F}$ be a set of features, $\mathcal{X} = \{x_1, \ldots, x_n\}$ be the input space, consisting of n input points such that $x_i \subseteq \mathcal{F}$ for all $i \in \{1, \ldots, n\}$ and $\mathcal{C} = \{c_1, \ldots, c_m\}$. A* classification task *is a function which assigns to an input point x_k a class $c_i \in \mathcal{C}$ based on the input space $\mathcal{X}$.*

Example 1. *Let $x_1, \ldots, x_6 \in \mathcal{X}$, where every $x \in \mathcal{X}$ is a student, and let $\mathcal{C} = \{0, 1\}$, where 1 represents a student being accepted to university, and 0 a rejection. The set of features consists of $\{g, t, m\} \in \mathcal{F}$, where g corresponds to the (rounded) average grade of the student, t to whether or not the student passed the entry test and m to whether or not they are motivated. Suppose that we are given the following input space:*

$\mathcal{X}$	g	t	m	c
x_1	*8*	*1*	*0*	*1*
x_2	*7*	*0*	*0*	*0*
x_3	*6*	*1*	*1*	*1*
x_4	*8*	*1*	*1*	*1*
x_5	*7*	*0*	*1*	*0*
x_6	*6*	*1*	*1*	*?*

A classification task is then to determine whether student x_6 is accepted or not.

3 *EVAX*: everyday argumentative explanations

In this paper, we are interested in *everyday explanations* as described in [16], i.e., explanations of why a specific event/property/decision occurred for end users in a day-to-day setting. To ensure that our explanations fulfill these requirements, we follow the major findings in [16]:

- *contrastive* explanations provide reasons *pro* and *con* the outcome [10, 8]. In an argumentative setting, explanations are contrastive when arguments and counterarguments for the outcome are present in the explanation.
- *selected* explanations have a fixed maximum size, the elements of which are selected based on at least one cognitive bias. In an argumentative setting, explanations contain a maximum number of arguments.
- *social* explanations can be adjusted to the receiver, by varying the complexity or size of an explanation. Since explanations based on argumentation frameworks can be represented in a variety of ways [10], argumentative explanations are social by definition.

Additionally, the explanations should remain faithful to the model (i.e., they explain the behavior of the model accurately).

Our method *EVAX* takes as input a trained black box model and constructs a global set of arguments: for each feature f and each class c it determines the probability that input containing f will be assigned c. For a specific input point, consisting of a set of features, a local argumentation framework is created (i.e., only containing the arguments corresponding to features from that input point and the defeats between them), from which the conclusion is predicted. This local argumentation framework can then be used to derive explanations, the size of which can be set by the user. We have implemented two ways to present explanations: based on abnormality and in a dialogue form.

We start by describing the method of *EVAX* (Section 3.1), we will illustrate it with a toy example in Section 3.2.

3.1 Method outline

EVAX takes as input a labeled dataset, a trained black box model `BB` and a threshold value τ_{select} that controls the size of the output. *EVAX* returns a set of predictions $\mathcal{Y}_{\text{pred}}$ and a set of local explanations $\mathcal{E}$. The explanations $e \in \mathcal{E}$ answer the question: "Why did black box `BB` assign class c to input instance x?" These explanations are deployments of an argumentation framework that represent the behavior of `BB` around a single datapoint in argumentative terms. This AF thus forms the basis for the explanations, and will, for every classified instance, be referred to as $\mathcal{AF}_l$. The size of $\mathcal{AF}_l$ can be manually altered by τ_{select}.

The procedure of *EVAX* is shown in Algorithm 1.[1] First, *EVAX* divides the labeled dataset into a set of unlabeled datapoints $\mathcal{X}$ (the input space) and a set of labels $\mathcal{Y}$ (the target space), which are then split up into a train set and a test set, respectively $\mathcal{X}_{\text{train}}, \mathcal{X}_{\text{test}}$ and $\mathcal{Y}_{\text{train}}, \mathcal{Y}_{\text{test}}$. The default size of the test set is 0.2, and the default τ_{select} value is 20. Afterward, the method can

[1] See `https://github.com/jowanvanlente/EVAX` for the implementation.

Algorithm 1 EVAX

procedure EVAX(BB, labeled_dataset, $\tau_{\text{select}} = 20$)
  $\mathcal{X}_{train}, \mathcal{X}_{test}, \mathcal{Y}_{train}, \mathcal{Y}_{test} \leftarrow$ split_dataset(labeled_dataset, test_size $= 0.2$)
  global_arguments $\leftarrow$ get_global_arguments(BB, $\mathcal{X}_{\text{train}}$) ▷ step 1
  for x_i in $\mathcal{X}_{\text{test}}$ **do**
    $\mathcal{AF}_l \leftarrow$ create_local_AF(x_i, global_arguments, τ_{select}) ▷ step 2
    predict($\mathcal{AF}_l$) ▷ step 3
    explain($\mathcal{AF}_l$) ▷ step 4
    results()
  get_results(BB, predictions, $\mathcal{Y}_{\text{test}}$)

be divided into four main steps, which are described below. The first step handles all datapoints and is executed just once, whereas the other three steps handle a single datapoint and may be repeated multiple times, up to a maximum of the size of the test set.

- **Step 1: Extract a global list of arguments**, to represent the global behavior of BB.
  - *EVAX* first iterates over all features $(a_k, v_k) \in \mathcal{F}$ of all (unlabeled) datapoints $x_j \in \mathcal{X}_{\text{train}}$ and all output classes $c_i \in \mathcal{C}$, and computes for every feature-class pair a decision rule. These rules are accompanied by a precision score $p_{(k,i)}$ that articulates the probability that BB will assign a datapoint with that particular feature to that particular class. It then saves all $p_{(k,i)}$ scores in a triple $((a_k, v_k), c_i, p_{(k,i)})$, which is added to a list of triples.
  - Arguments are constructed based on the list of triples. For every triple an argument is constructed in which the feature (a_k, v_k) is the premise `prem`, the output-class c_i is the conclusion `conc` and the precision score $p_{(k,i)}$ is set as the argument strength `str`. Together these arguments form the global list of arguments.
- **Step 2: Create a local AF, $\mathcal{AF}_l$.**
  - *EVAX* creates a local AF: $\mathcal{AF}_l$ in every iteration of this step. This is an argumentation framework $\mathcal{AF}_l = \langle \mathsf{Args}_l, \mathsf{Def}_l \rangle$ that represents the classifier's behavior around one particular datapoint. Based on the values of that datapoint, it selects a set of relevant arguments ($\mathsf{Args}_l \subseteq \mathsf{Args}$) from the global list of arguments (Args) and determines the defeats (Def_l).
  - Argument selection is done by matching the features of the datapoint from $\mathcal{X}_{\text{test}}$ with the premises of the arguments: given a datapoint $x_i \in \mathcal{X}_{\text{test}}$ and an argument $A \in \mathsf{Args}$, if $\texttt{prem}(A)$ is one of the features in x_i then A is added to the local AF (meaning $A \in \mathsf{Args}_l$). As a result, all arguments with a premise corresponding to one of the features of the datapoint are selected. To gain computational efficiency and maintain selectedness, a threshold τ_{select} can be defined, which ensures only the top τ_{select} strongest arguments are included in the list.
  - The defeats are determined as in Definition 3.
- **Step 3: Predict** the output class based on $\mathcal{AF}_l$.
  - First, the grounded extension of the local AF ($\mathsf{Grd}(\mathcal{AF}_l)$) is computed, after which the conclusion of the arguments in $\mathsf{Grd}(\mathcal{AF}_l)$ is picked as the prediction. Formally, this means that prediction $y_i \in \mathcal{Y}$ is equal to $\texttt{conc}(A)$ such that $A \in \mathsf{Grd}(\mathcal{AF}_l)$. Since arguments in the grounded extension are non-conflicting, they always have the same conclusion. Therefore it does not matter what argument in $\mathsf{Grd}(\mathcal{AF}_l)$ is picked. When $\mathsf{Grd}(\mathcal{AF}_l)$ is empty, *EVAX* will predict the majority class.
- **Step 4: Explain**

– The current implementation allows for two variations: adding selectedness based on abnormality and presenting the explanation in a conversational form.
– Abnormality is one of the methods humans use when selecting an explanation and describes how people tend to choose a cause that is unusual [22]. We have defined the abnormality of an argument as $1 - \textit{coverage}$. The coverage value refers to the fraction of datapoints that the decision rule, out of which the argument is constructed, 'rules over'. In other words, the coverage of argument A refers to the fraction of input instances that have a feature equal to $\texttt{prem}(A)$. Since the coverage describes how often a feature is present in a dataset, it essentially describes how 'normal' a feature is. Therefore, a lower coverage means that a feature becomes less normal, thus becomes increasingly abnormal. The deployment of $\mathcal{AF}_l$ then amounts to selecting the argument with the highest abnormality score that argues for the predicted class. An example of the output is given in Figure 1.

a130: odor = 6 → (precision = 1.0, abnormality = 0.989)
'x_1 is poisonous because of its unusual pungent odor'

Figure 1: Example output of the most abnormal argument of $\mathcal{AF}_l$ that explains why BB assigned x_1 (a mushroom) to class c_1 (poisonous). On the right, we see the same explanation, but in natural language.

– *EVAX* can also provide a dialectical representation of $\mathcal{AF}_l$, similar to a dispute tree [13]. This representation has the form of a discussion between a proponent (P) and opponent (O) about what class to assign to the datapoint in question. A threshold τ_{explain} allows the user to choose the number of arguments to include in the explanation. Arguments are divided into *pro* and *con* arguments and are put forward by P and O, who take turns. If the value of threshold τ_{explain} is even, O starts the dispute, and if it is odd, P starts. After the first argument is put forward, the strongest counterargument is replied.[2] Note that the threshold is different from τ_{select}, because it does not affect the size of $\mathcal{AF}_l$, but merely the size of the dialectical representation of $\mathcal{AF}_l$. See Figure 2 for an example with $\tau_{\text{explain}} = 4$.

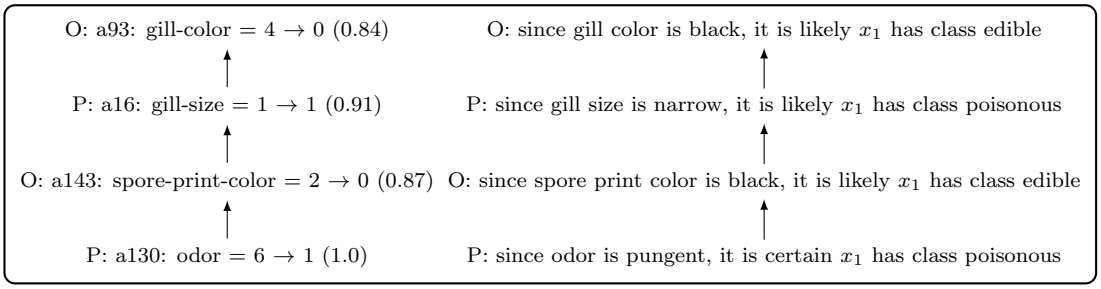

Figure 2: The dialectical explanation of the assignment of x_1 to c_1 by BB, as in Figure 1. The values between brackets refer to the precision score. One must read from top to bottom; the arrows solely indicate the conflicts, not necessarily defeats.

[2]The only requirement of the counterargument is a conflicting conclusion, and not necessarily a higher strength value, i.e., it does not have to defeat the argument. This is to ensure that counterarguments are included in this explanation form.

3.2 Toy example

Recall the classification task described in Example 1, on students being accepted into university. In Figure 3 we present a similar case to illustrate *EVAX*. It represents one iteration of Steps 2, 3, and 4. It thus assumes that the global list of arguments has already been computed.

In this example, a black box predicts that an input instance 'John' will be accepted into university. The same input instance is used as input for *EVAX*. Based on that input, *EVAX* creates a local argumentation framework $\mathcal{AF}_l$ by selecting three relevant arguments, based on the three different features, and defines defeats over them. It then calculates the grounded extension and predicts that John will be accepted into university. In addition, it computes an AF-based explanation, which in this case is a dialectical representation of $\mathcal{AF}_l$, as described in Step 4. The threshold τ_{explain} has a value of 3. The arrows in the representation are the defeats. Note that the arrows between the different components of *EVAX* *do not* represent defeats, but indicate the information flow.

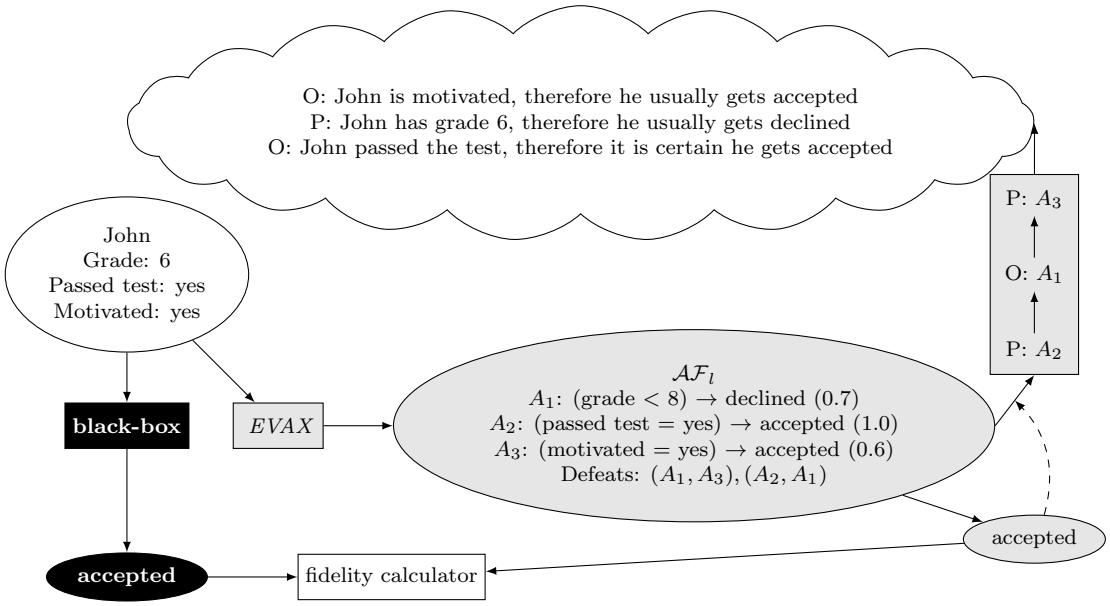

Figure 3: Illustation of *EVAX* applied to Example 1.

4 Quantitative evaluation

We have tested *EVAX* on four datasets and used five quantitative metrics for the evaluation. The (labeled) datasets are from the UCI Machine Learning Repository [11]:

- With the **Adult** dataset one tries to predict whether or not a person makes more than 50.000 dollars a year. We removed all datapoints with unknown values and discretized the continuous features.
- The **Mushroom** dataset includes instances of 23 different species of mushrooms. The task is to identify whether a mushroom is poisonous or edible. We did not perform any alterations on this dataset.
- The task of the **Iris** dataset is to predict the type of iris plant. We discretized the continuous values.
- With the **Wine** dataset one wants to predict the type of wine of an input instance. Again we discretized the continuous values.

The discretization of continuous variables is necessary to constrain the number of arguments that are added to the global list of arguments. We have used the *cut* method by pandas [15] with a bin value of 10. Since higher bin values tend to give better performance but reduce the computational efficiency, we have tuned this value by incrementally increasing the value from 3 up to 20. We found that from a bin value of 10 and upwards, the fidelity did not significantly increase (sometimes it even decreased), while the computational efficiency consistently decreased with higher bin values.

For each of these datasets we have chosen four different machine learning models with different complexity to test the performance and range of *EVAX*: logistic regression, support vector machines (SVM), random forest, and neural networks. All four models are initialized from the scikit-learn library [17].

Finally, we applied the following five metrics for the evaluation:

- **Fidelity** indicates how well the explanation approximates the prediction of the black box model. It represents the fraction of datapoints that are assigned to the same output class by *EVAX* and BB.
- **Accuracy** *(BB)* indicates how well our model performs on unseen data. It represents the fraction of correctly classified datapoints. The value between brackets () refers to the original accuracy of BB.
- **Size** measures the average minimum amount of arguments necessary to retain the same prediction. In other words, it is the lowest possible τ_{select} score without affecting the accuracy or fidelity. A consistent low size value indicates the method can guarantee to compute small explanations that are consistently faithful.
- **Empty Grd** specifies the fraction of datapoints for which the grounded extension $\mathsf{Grd}(\mathcal{AF}_l)$ is an empty set. When $\mathsf{Grd}(\mathcal{AF}_l)$ is an empty set, *EVAX* relies on a default prediction. A higher 'Empty Grd' value thus means that accuracy and fidelity scores are increasingly determined by the default prediction, and therefore become less reliable.
- **Time** indicates the number of seconds needed to run the program.

		Fidelity	**Accuracy** *(BB)*	**Size**	**Empty Grd**	**Time (s)**
Adult	*Logistic regression*	0.95	0.73 *(0.72)*	1	0.0	5.06
	SVM	0.93	0.75 *(0.74)*	1	0.0	13.61
	Random forest	0.88	0.77 *(0.78)*	1	0.0	5.06
	Neural network	0.91	0.75 *(0.75)*	1	0.0	5.28
Mushroom	*Logistic regression*	0.98	0.96 *(0.95)*	1	0.0	17.92
	SVM	0.99	0.98 *(0.99)*	1	0.0	13.45
	Random forest	1.0	0.99 *(1.0)*	1	0.0	13.63
	Neural network	1.0	0.95 *(0.95)*	1	0.0	17.87
Iris	*Logistic regression*	0.97	0.97 *(0.9)*	1	0.0	0.24
	SVM	1.0	0.97 *(0.97)*	1	0.0	0.25
	Random forest	1.0	0.97 *(0.97)*	1	0.0	0.26
	Neural network	0.93	0.97 *(0.9)*	1	0.0	0.23
Wine	*Logistic regression*	0.94	1.0 *(0.94)*	1	0.0	2.32
	SVM	0.94	1.0 *(0.94)*	1	0.0	2.60
	Random forest	0.92	1.0 *(0.91)*	1	0.0	2.73
	Neural network	0.86	0.97 *(0.83)*	1	0.0	3.86

Table 1: Quantitative results of *EVAX*.

The results in Table 1 show high fidelity (an average of 0.95) for all four ML models, which indicates a sufficient degree of faithfulness. Only the adult dataset and the neural network of the

wine dataset have relatively low scores. This might be due to the relatively low accuracy of the BB in those cases. Since the argument with the highest argument strength is always in $\mathsf{Grd}(\mathcal{AF}_l)$, the minimum size is always equal to 1. This indicates that the model is capable of computing small explanations without losing faithfulness. Moreover, we see that the method never computes an empty grounded extension $\mathsf{Grd}(\mathcal{AF}_l)$, and hence requires no reliance on a default prediction. These results are obtained on a Windows 64-bit operating system with 16GB RAM and an Intel(R) Core(TM) i5-1145G7 @ 2.60GHz processor.

5 Qualitative evaluation

The purpose of *EVAX* is the modeling of explanations as described in [16]. In this section we discuss how the explanations generated through *EVAX* are contrastive, selected and social, as described at the beginning of Section 3.

Contrastive explanations explain the fact (e.g., P) by highlighting its differences with the foil (e.g., Q), by answering the question *why P rather than Q?* [16]. Argumentative explanations are *contrastive* when they include arguments *pro* and *con* the conclusion. For explanations computed by *EVAX* this is the case when there is at least one argument with a fact conclusion and a counter-argument with a foil conclusion. Such counterarguments make it possible to explain the outcome *relative to* an alternative outcome, by showing what features give reason to believe that foil. As shown in Figures 2 and 3, when $\tau_{\text{explain}} > 1$, explanations contain at least one counterargument and are therefore contrastive.

While an event might have infinitely many causes, humans are able to select one or two as the explanation. To this end a variety of cognitivie biases are employed [16]. *EVAX* incorporates *selectedness* by implementing both minimality and biasedness. Minimality amounts to including just a few arguments as the explanation. This is enabled by guaranteeing that the number of arguments in $\mathcal{AF}_l$ does not exceed threshold τ_{select}. In addition, this restricted size has shown not to affect the fidelity score. *EVAX* allows for biasedness in the form of *abnormality*, which is a common cognitive bias in everyday explanations [22].

Finally, explanations are social, since the explainer will adapt the explanation to the explainee, for example, by adjusting the size, the content or the form of the explanation. Explanations can be adjusted in two ways with *EVAX*. First, the number of arguments that are included in $\mathcal{AF}_l$ can be adjusted with τ_{select} and the size of the explanation can be adjusted with τ_{explain}. In that way, an inexperienced end-user who requires a single argument to explain the prediction can set $\tau_{\text{select}} = \tau_{\text{explain}} = 1$. A more experienced user who wants a completer set of arguments and counterarguments can set higher values. Second, because a computed explanation $e \in \mathcal{E}$ stems from an AF, the explanation can be presented in various ways. In the current paper we have illustrated one of these representations in Figure 2, in the form of a dialogue.

6 Related work

As the survey [10] shows, the field of argumentative XAI goes in several directions. In addition to explaining argumentation-based conclusions with argumentation (e.g., [13, 4, 14, 20]), argumentation can also be employed to explain conclusions derived with other AI approaches. Here a distinction can be made between *intrinsic* methods, which provide explanations for conclusions drawn by argumentation mechanisms (e.g., [7, 9, 18]) and *post-hoc* methods, which provide argumentative explanations for conclusions drawn from non-argumentative methods (e.g., [1, 2, 21]).

Closest related to our work is [23, 24]. There several agents engage in a dialogue, by putting forward arguments in the form of classification association rules. This dialogue results in a tree of arguments and counterarguments. The result is an overview of the agents' point of view, with arguments that might contain several premises. Given the focus on the dialogue and the structure of arguments and counterarguments, [23, 24] provide social and contrastive explanations. Our approach also aims at minimal explanations: our arguments have only one premise, the size of the

explanation can be adjusted and the selection of the arguments in the explanation can be based on argument strength or abnormality. The purpose of our work (i.e., providing small, everyday explanations) is therefore different.

Following our interpretation of the explanations in [16], none of the other available argumentative XAI approaches are contrastive, selected *and* social. As mentioned, explanation methods based on argumentation frameworks are social by definition, since AFs allow for a variety of explanation representations, which can be used to adjust the explanation to a receiver. Additionally, most argumentative explanation methods are contrastive, since they present not only arguments for the conclusion, but also counterarguments. The exception to this is [1], since there is no clear relation between arguments and counterarguments in this approach. Selectedness, in the form of minimality and biasedness is most difficult to establish, it seems, since none of the mentioned approaches is selected in our sense. Selection based on a cognitive bias is not integrated at all. Reducing the size of the explanation is discussed, however, the reduction might not be sufficient. Even when a small explanation is provided, it might still contain many causes or result in large dispute trees [7, 9, 1].

In contrast, *EVAX* provides a method which is contrastive (arguments and counterarguments are part of the explanation), selected (the maximum size can be reduced to one argument and the selection of this argument can be based on abnormality) *and* social (the size of the explanation can be adjusted and the explanation can be represented as a dialogue). Since the explanations provided by *EVAX* rely on an argumentation framework, in future work we can look at representing them as dispute trees (as in [13]), apply selectedness based on necessity and sufficiency [5], derive explanations in terms of labeling [14] or strongly rejecting subframeworks [20].

7 Conclusion

We have introduced *EVAX* an argumentative explanation method for everyday explanations for classifiers. It takes as input a trained classifier, calculates a local argumentation framework $\mathcal{AF}_l$ and can present explanations in a variety of ways (recall Figure 3). In particular, we have shown how explanations can be selected, based on abnormality and how explanations can be represented as a dialogue between proponent and opponent. Although the method might seem somewhat naive, our results show that it is a fast explanation method, which satisfies our requirements. Based on the quantitative results (recall Table 1) we have shown that our method is faithful. Moreover, in the qualitative evaluation (Section 5), we have discussed how *EVAX* produces everyday explanations that satisfy the findings from [16] (i.e., contrastive, selected and social explanations). The result is a model-agnostic explanation method with which local explanations can be provided in a faithful manner, based on findings from the social sciences.

The results in this paper show that *EVAX* is a promising explanation method, which can be explored further. First, the properties of everyday explanations can be further worked out by including counterfactual statements, incorporating more cognitive biases, and testing more explanation deployments. Second, experimental evaluations with human users would more closely assess the quality of argumentative explanations.

References

[1] Emanuele Albini, Piyawat Lertvittayakumjorn, Antonio Rago, and Francesca Toni. DAX: deep argumentative explanation for neural networks. *arXiv*, 2012.05766, 2020.

[2] Leila Amgoud. Non-monotonic explanation functions. In Jirina Vejnarová and Nic Wilson, editors, *Proceedings of the 16th European Conference of Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU, 2021*, volume 12897 of *Lecture Notes in Computer Science*, pages 19–31. Springer, 2021.

[3] Philippe Besnard, Alejandro Javier García, Anthony Hunter, Sanjay Modgil, Henry Prakken, Guillermo Ricardo Simari, and Francesca Toni. Introduction to structured argumentation. *Argument & Computation*, 5(1):1–4, 2014.

[4] AnneMarie Borg and Floris Bex. A basic framework for explanations in argumentation. *IEEE Intelligent Systems*, 36(2):25–35, 2021.

[5] AnneMarie Borg and Floris Bex. Necessary and sufficient explanations for argumentation-based conclusions. In Jiřina Vejnarová and Nic Wilson, editors, *Proceedings of the 16th European Conference of Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU, 2021*, volume 12897 of *Lecture Notes in Computer Science*, pages 19–31, 2021.

[6] Roberta Calegari, Giovanni Ciatto, and Andrea Omicini. On the integration of symbolic and sub-symbolic techniques for XAI: A survey. *Intelligenza Artificiale*, 14(1):7–32, 2020.

[7] Oana Cocarascu, Andria Stylianou, Kristijonas Čyras, and Francesca Toni. Data-empowered argumentation for dialectically explainable predictions. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *Proceedings of the 24th European Conference on Artificial Intelligence, ECAI, 2020*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2449–2456. IOS Press, 2020.

[8] Kristijonas Čyras, Ramamurthy Badrinath, Swarup Kumar Mohalik, Anusha Mujumdar, Alexandros Nikou, Alessandro Previti, Vaishnavi Sundararajan, and Aneta Vulgarakis Feljan. Machine reasoning explainability. *arXiv*, 2009.00418, 2020.

[9] Kristijonas Čyras, David Birch, Yike Guo, Francesca Toni, Rajvinder Dulay, Sally Turvey, Daniel Greenberg, and Tharindi Hapuarachchi. Explanations by arbitrated argumentative dispute. *Expert Systems with Applications*, 127:141–156, 2019.

[10] Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and Francesca Toni. Argumentative XAI: A survey. In Zhi-Hua Zhou, editor, *Proceedings of the 30th International Joint Conference on Artificial Intelligence, IJCAI, 2021*, pages 4392–4399, 2021.

[11] Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017.

[12] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[13] Xiuyi Fan and Francesca Toni. On computing explanations in argumentation. In Blai Bonet and Sven Koenig, editors, *Proceedings of the 29th Conference on Artificial Intelligence, AAAI, 2015*, pages 1496–1502. AAAI Press, 2015.

[14] Beishui Liao and Leendert van der Torre. Explanation semantics for abstract argumentation. In Henry Prakken, Stefano Bistarelli, Francesco Santini, and Carlo Taticchi, editors, *Proceedings of the 8th International Conference on Computational Models of Argument, COMMA, 2020*, volume 326 of *Frontiers in Artificial Intelligence and Applications*, pages 271–282. IOS Press, 2020.

[15] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

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

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[18] Henry Prakken and Rosa Ratsma. A top-level model of case-based argumentation for explanation: Formalisation and experiments. *Argument & Computation*, pages 1–36, 2021.

[19] Stuart J. Russell and Peter Norvig. *Artificial Intelligence - A Modern Approach, Third International Edition*. Pearson Education, 2010.

[20] Zeynep Saribatur, Johannes Wallner, and Stefan Woltran. Explaining non-acceptability in abstract argumentation. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *Proceedings of the 24th European Conference on Artificial Intelligence, ECAI, 2020*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 881–888. IOS Press, 2020.

[21] Naziha Sendi, Nadia Abchiche-Mimouni, and Farida Zehraoui. A new transparent ensemble method based on deep learning. In Imre J. Rudas, János Csirik, Carlos Toro, János Botzheim, Robert J. Howlett, and Lakhmi C. Jain, editors, *Proceedings of the 23rd International Conference of Knowledge-Based and Intelligent Information & Engineering Systems, KES, 2019*, volume 159 of *Procedia Computer Science*, pages 271–280. Elsevier, 2019.

[22] Paul Thagard. Explanatory coherence. *Behavioral and brain sciences*, 12(3):435–467, 1989.

[23] Maya Wardeh, Trevor Bench-Capon, and Frans Coenen. PADUA: a protocol for argumentation dialogue using association rules. *Artificial Intelligence and Law*, 17(3):183–215, 2009.

[24] Maya Wardeh, Frans Coenen, and Trevor Bench-Capon. Pisa: A framework for multiagent classification using argumentation. *Data & Knowledge Engineering*, 75:34–57, 2012.