Generation and Evaluation of Factual and Counterfactual Explanations for Decision Trees and Fuzzy Rule-based Classifiers

Ilia Stepin, Jose M. Alonso, Alejandro Catala
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)
Universidade de Santiago de Compostela
Santiago de Compostela, Spain
{ilia.stepin,josemaria.alonso.moral,alejandro.catala}@usc.es

Martin Pereira-Fariña
Departamento de Filosofía e Antropoloxía
Universidade de Santiago de Compostela
Santiago de Compostela, Spain
martin.pereira@usc.es

***Abstract*—Data-driven classification algorithms have proven highly effective in a range of complex tasks. However, their output is sometimes questioned, as the reasoning behind it may remain unclear due to a high number of poorly interpretable parameters used during training. Evidence-based (factual) explanations for single classifications answer the question why a particular class is selected in terms of the given observations. On the contrary, counterfactual explanations pay attention to why the rest of classes are not selected. Accordingly, we hypothesize that providing classifiers with a combination of both factual and counterfactual explanations is likely to make them more trustworthy. In order to investigate how such explanations can be produced, we introduce a new method to generate factual and counterfactual explanations for the output of pretrained decision trees and fuzzy rule-based classifiers. Experimental results show that unification of factual and counterfactual explanations under the paradigm of fuzzy inference systems proves promising for explaining the reasoning of classification algorithms.**

***Index Terms*—Explainable Artificial Intelligence, Counterfactuals, Decision Trees, Fuzzy Inference Systems, Natural Language Generation**

I. Introduction

Intelligent systems tend to make a wide use of data-driven classification algorithms to make automatic decisions. While a number of such algorithms achieve outstanding results in various complex tasks, their underlying reasoning is often left unclear to end-users [1]. Despite a high degree of accuracy, individual decisions may be questioned and therefore require a satisfactory explanation, particularly in case of high-stakes decisions. This is the main reason why the European Commission pushes for increasing both public and private resources to develop research on Artificial Intelligence (AI) in agreement with the European values and fundamental rights [2]. Accordingly, the "right to explanation", which is included in the General Data Protection Regulation (GDPR) issued by the European Parliament [3], affects humans but also AI techniques and systems.

Interpretable models can explain the output of a black-box algorithm in terms of the feature values that led to a given classification [4]. When the features are known and their values can be observed, such "observation-based" (or *factual*) explanations reflect the most critical characteristics of the data instance that influenced the output of the algorithm. For instance, a factual explanation retrieved from a decision tree (DT) can be regarded as a set of feature-value pairs that justifies the corresponding root-to-leaf decision path. However, factual explanations do not necessarily offer an insight into why alternative classification options were discarded. Instead of really explaining a classifier's outcome, a piece of factual information on decisive feature values is argued to merely summarize the statistics on an automatic prediction, which may not be sufficient to ensure trust in the given classification [5].

Indeed, findings from cognitive science testify that human explanations are intrinsically contrastive [6]. Thus, to explain a certain fact P means to answer the template question "Why P rather than Q?" where Q is a set of hypothetical non-occurring situations (or *counterfactuals*) that would have led to a different state of affairs. A counterfactual explanation complements its factual counterpart specifying minimal conditions for a given data instance to be classified differently. In the context of machine learning (ML)-based classification, a counterfactual must satisfy two conditions: (1) It describes a set of feature-value pairs that are minimally different from those inherent to the original data point requiring explanation; and (2) changing the feature values in accordance with the given explanation makes the same model produce a different classification for the same the data instance.

Being accurate and relatively easy to interpret [7], DTs have deserved a wide use in industry [8]. However, the explainability of structurally complex DTs turns out to be far from a trivial issue. As the amount of nodes and branches in a tree increases, its explanatory capacity reduces accordingly [9]. This is in agreement with the "Principle of Incompatibility" postulated by Zadeh [10]: "... as the complexity of a system increases our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics".

In addition, ML-based explanations are expected to be

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

multi-modal (i.e., a mix of text and graphs), natural, and comparable to those given by humans [11]. Unfortunately, explanations generated by crisp DTs are typically restricted to representing the local semantics of the tree and lack the ability to derive immediate explanations in natural language. Instead, fuzzy rule-based systems are endowed with global semantics thanks to the use of strong fuzzy partitions [12]. They are therefore capable of explaining the reasoning of black-box ML algorithms under uncertainty in natural language [13]–[15].

The contribution of this paper is twofold. First, we introduce a novel method for factual and counterfactual explanation generation associated to both crisp and fuzzy DTs. This method is validated in a real-world multi-class beer-style classification problem where we consider three crisp DTs, and two fuzzy rule-based classifiers (i.e., a fuzzy DT before and after running a linguistic simplification procedure). Second, we develop a Natural Language Generation (NLG) module to verbalize the generated explanations. In addition, we compare the expressiveness of textual explanations derived from both crisp and fuzzy DTs.

The rest of the manuscript is structured as follows. In Section II, we review existing approaches tackling various issues of explanation generation for decision trees. Section III introduces our rule-based approach to generate comprehensive factual and counterfactual explanations for tree-based classifiers and provides the reader with implementation details of the algorithm proposed. Section IV describes the experimental setting and presents the results of the experiments carried out. Finally, we outline directions for future work and conclude in Section V.

II. Related work

Recent years have witnessed a substantial rise in attention to the problem of counterfactual explanation generation from AI researchers. Counterfactuals are becoming increasingly commonplace in cognitive engineering [16], automatic planning [17], and text mining [18], among other applications. Counterfactual explanations have been shown to be user-friendly in a range of applications [19]. Nevertheless, generating counterfactual explanations (in particular, for tree-based classifiers) still poses a number of challenges for researchers and developers.

Several works argue that test points are best explained counterfactually via the data instances of opposing classes generated in the region closest to the test point to be explained. For instance, the model-agnostic *FoilTrees* framework [20] trains a decision tree on a subset of random points from the original dataset centered around the test point in question to ensure that the generated counterfactual is local. Indeed, the data point of a contrast-class that is minimally distant from the test instance may not be sufficiently explanatory, as the corresponding branch in the tree may cover only a small subset of data instances. A counterfactual explanation is then argued to base on the nearest tree node where the paths to the factual and counterfactual data points diverge. Despite several heuristics proposed to identify the closest counterfactual data point, the study is restricted to generating an explanation based on a minimal difference in terms of the number of decision nodes.

A similar approach is employed in the *LOcal Rule-based Explanations* (LORE) framework [21]. An evolutionary algorithm is used to generate a synthetic local neighbourhood driven by the assumption that a black-box model's behaviour is best explained by means of a set of data points (including those generated artificially) that share a maximal amount of characteristics with the test data instance. Subsequently, a factual explanation is complemented with a counterfactual one based on a set of falsified conditions extracted from a decision tree trained on the newly generated data.

The above works show a number of limitations. First, the frameworks are designed to address only binary classification problems. Second, no relevance measures for multiple counterfactuals of the same opposed class are indicated. Thus, two or more counterfactuals may be claimed equivalent despite their different degrees of proximity to the test instance prediction leaf. This might result in ignoring significant semantic and structural differences between several counterfactuals and therefore presenting less relevant explanation to the end-user.

As multiple counterfactuals are widely encountered in DTs, the matter of evaluating the relevance of different counterfactuals arises naturally. Thus, counterfactuals can be evaluated in terms of the meta-features of the leaf node corresponding to the test instance and the leaves leading to counterfactual outcomes. In this case, a variant of the L_1 distance metric is proposed to determine such a counterfactual that the resulting explanation is the shortest [22].

Conversely, counterfactual explanation generators are suggested to be evaluated with respect to the diversity of the generated explanations [23]. Thus, diversity constraints can be imposed in the form of rule-based heuristics applied to the newly generated counterfactuals sequentially to offer a set of multiple coherent explanations. However, no empirical evaluation of comprehension of such explanations is provided. Alternatively, a quantitative measure of counterfactual diversity is found to be defined as "the mean of the distances between each pair of examples" [24].

Despite a rising interest towards counterfactual explanation generation in explainable AI, as far as we know, there are no works on counterfactual generation either for fuzzy decision trees in particular or for fuzzy rule-based systems in a more general sense.

III. Method

Our approach for generating factual and counterfactual explanations is generalized to work with both crisp and fuzzy tree-based classifiers. We focus on a multi-class classification task setting, that is learning a mapping function $f : X \longrightarrow Y$ from a training dataset of n labeled instances $X = \{x_i \mid 1 \leq i \leq n\}$ to a discrete output variable (class) $Y = \{y_j \mid 1 \leq j \leq m\}$ where m is the number of classes. Every data instance $\{x_i = (F_i, cl_i) \in X \mid (1 \leq i \leq n)\}$ is characterized by an output class label $cl_i \in Y$ and k features $F_i = \{(f_j, v_j)$

$| \ 1 \leq j \leq k\}$, where f_j is the name of the j-th feature, v_j being the corresponding numeric value.

A. Preliminary notation

Let us formally introduce the main notions that we utilize in this work. Thus, we operate on a finite set $T = \{T_c, T_f\}$ of pretrained crisp DTs $T_c = \{t_{c_i} \mid 1 \leq i \leq |T_c|\}$ and fuzzy DTs or fuzzy rule-based classifiers $T_f = \{t_{f_j} \mid 1 \leq j \leq |T_f|\}$.

Following [25], we define a crisp DT $t_c \in T_c$ to be a labeled data structure $t_c = \langle r_c, N_c, E_c, L_c \rangle$ with the root $r_c \in N_c$ where N_c is a set of nodes, $E_c \subseteq N_c \times N_c$ is a set of edges, L_c is a labeling function over all edges in E_c. Thus, every edge $e_c \in E_c$ in a crisp DT $t_c \in T_c$ is assigned a binary-valued label following the mapping:

$$L_c : E_c \rightarrow \{0, 1\}. \tag{1}$$

The set of nodes N_c is composed of terminal (leaf) nodes $Term = \{n_{c_t} \mid n_{c_t} \in N_c, 1 \leq t \leq |N_c|\}$ and non-terminal (non-leaf) nodes $Nonterm = \{n_{c_{nt}} \mid n_{c_{nt}} \in N_c \setminus Term, 1 \leq nt \leq |N_c| - |Term|\}$ so that $N_c = Term \cup Nonterm$. Every non-leaf node contains a condition $c = \{\langle f_i, operator_i, v_i \rangle \mid 1 \leq i \leq k\}$ specifying the label of the outcoming edge where the components of a feature-value pair $\langle f_i, v_i \rangle$ of each condition $c \in C$ are related by means of one of (in-)equality operators $operator_i = \{<, \leqslant, =, >, \geqslant\}$ (e.g., $height \leqslant 180$). Altogether, all the unique conditions across the tree form a set of conditions $C = \{c_1, c_2, ..., c_{|C|}\}$.

For a given crisp DT $t_c \in T_c$, a set of root-to-leaf decision paths $P(t_c) = \{p_1, ..., p_{|P|}\}$ represents a collection of all possible classifications of that tree. Then, a root-to-leaf decision path $p \in P(t_c) : p = \{c_{vc} \mid 1 \leq vc \leq |C|\}$ is a set of verified conditions that justifies a classification of the given data instance x.

Hence, a factual explanation $e_f(x, t_c)$ for the given data instance x and tree $t_c \in T_c$ is defined as a tuple $e_f(x, t_c) = \langle p_f, y_f \rangle$ where $p_f \in P(t_c)$ and $y_f \in Y$ so that it constitutes a decision path from the root node of the tree to a leaf node indicating an output class as predicted by the classifier. Similarly, a (single) counterfactual explanation $e_{cf}(x, t_c, y_{cf}) = \langle p_{cf}, y_{cf} \rangle$ determines a decision path $p_{cf} \in \{P_{cf} = \{P(t_c) \setminus p_f\}\}$ that classifies the data instance x to be of an alternative class $y_{cf} \in \{Y_{cf} = \{Y \setminus y_f\}\}$.

Every data instance x is assumed to have only one factual explanation $e_f(x, t_c)$ for a given crisp DT t_c. Consequently, there exist at most $(m-1)$ counterfactual explanations that form a set $E_{cf}(x, t_c, Y_{cf}) = \{e_{cf_j}(x, t_c, y_{cf}) \mid 1 \leq j \leq (m-1), y_{cf} \in Y_{cf}\}$ of counterfactual explanations, each of them matching one of the alternative classes. As multiple counterfactuals can possibly be found for any predefined class, we further generalize the existence of multiple counterfactual explanations for each output class. Hence, the exhaustive set of all potential counterfactual explanations for the data instance x is further defined as a set of decision paths to all the classifications of each alternative class:

$$E_{cf}(x, t_c, Y_{cf}) = \{\langle P_{cf}, Y_{cf} \rangle\} \tag{2}$$

As follows from the formulation of the classification problem above:

$$\forall (p_f \in P, p_{cf} \in P_{cf}, y_f \in Y, y_{cf} \in Y_{cf}) : p_f \neq p_{cf}, y_f \neq y_{cf} \tag{3}$$

Being a generalization of a crisp DT [26], a fuzzy DT $t_f \in T_f$ preserves the same structure as a crisp tree: $t_f = \langle r_f, N_f, E_f, L_f \rangle$, with the root, nodes, and edges defined analogously to those of a crisp tree. However, a fuzzy tree operates on fuzzy sets in the universe of discourse U so that the labeling is achieved by means of applying a membership function $\mu(e_f \in E_f)$:

$$L_f : \mu(e_f \in E_f) \rightarrow [0, 1] \tag{4}$$

determining how likely the data instance x to be of the class specified in the leaf node. In our case, each feature $\{f_j \in F_i \mid 1 \leq i \leq n, 1 \leq j \leq k\}$ is defined by a set of linguistic terms $L(f_j) = \{L^f_{j_1}, ..., L^f_{j_k}\}$. Each input feature f_j is associated with a uniform strong fuzzy partition (SFP) that is defined in U. It is worthy to note that SFPs satisfy all properties (e.g., coverage, distinguishability, etc.) demanded for interpretable fuzzy partitions [12]. Regarding the output, each class is associated with a weight that is represented by the related branch of the tree. Note that the α-cut is used to keep the verified conditions of a fuzzy DT consistent with those of a crisp DT. The notions of a root-to-leaf path, factual and counterfactual explanation are defined similarly to those for crisp DTs.

Let us conclude the present section with a general remark concerning all the tree-based classifiers considered. A path $p \in P(t)$ in an arbitrary (either crisp or fuzzy) tree $t \in T$ can be unrolled into a chain of conditions and thus transformed into a conditional statement of the form

$$\text{IF } c_1 \text{ AND } c_2 \text{ AND ... AND } c_{|p|} \text{ THEN } x \text{ is of class } cl_j \tag{5}$$

where $cl_j \in Y (1 \leq j \leq m)$. Thus, a path in a crisp tree can be represented following the same formalism as a fuzzy DT (or a fuzzy rule-based classifier) with an activation of 1. Therefore, such a conditional statement is further complemented in a fuzzy tree-based classifier with the corresponding activation function value so that it is transformed to a fuzzy rule. The textual representation of a factual explanation $e_f(x, t)$ and a (minimal) set $E_{cf}(x, t)$ of counterfactual explanations are claimed to fully explain a particular classification. In order to generate a corresponding textual (factual and/or counterfactual) explanation, we further assume a set $\{c_1, ..., c_{|p|}\}$ of conditions in (5) to be an *antecedent* of the output explanation, whereas the conclusive classification $cl_j \in Y$ is referred to as a *consequent*.

B. General framework

We generate a factual explanation for the given test instance $x = (F, cl_x)$ and a set of counterfactual explanations E_{cf} for all the alternative classes $\{cl \mid cl \in \{Y \setminus cl_x\}\}$. Furthermore, we distinguish multiple counterfactual explanations ranking

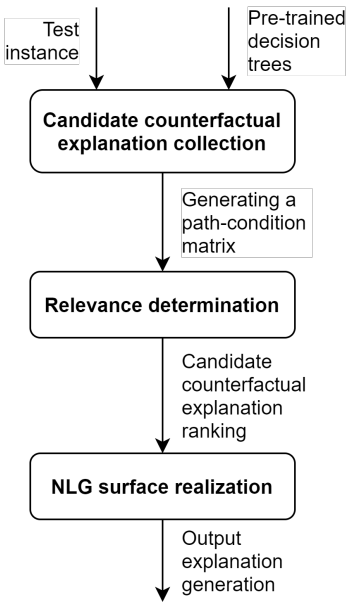

Fig. 1. Counterfactual explanation generation pipeline

them in accordance with their explanatory capacity with respect to the test instance.

In addition to producing a logical representation for computed counterfactuals, the output explanations are designed to be user-friendly pieces of text that not only specify the classifier's prediction but also offer a comprehensive explanation justifying the classifier's behaviour.

A factual explanation is composed of a set of conditions constituting the decision path to the classifier's prediction. As it is trivially reconstructed by pruning the tree from root to leaf following the verified conditions for the test instance, let us instead focus on the method for obtaining counterfactual explanations. The process of generating counterfactuals is graphically depicted in Fig. 1.

Given an arbitrary tree-based classifier $t \in T$ and a test data instance in question $x \in X$, we subdivide the task of counterfactual explanation generation into the following three main phases which are applied iteratively:

- **Collecting candidate counterfactual explanations**. At first, all the paths to the alternative class leaves in the tree are collected. The preorder depth-first traversal algorithm [27] is recursively applied to the input tree in order to extract a set of paths to counterfactuals $P_{cf} \in P(t) : P_{cf} = \{p_{cf_i} \mid 1 \leq i < |P|)\}$ that lead to all the predictions counterfactual to the factual classification. As the retrieved paths P_{cf} are not guaranteed to have the same number of conditions (nodes), we set a *path-condition matrix*, that is an equivalent form of their representation. In order to avoid inconsistency, the set of paths P_{cf} are represented in form of a binary matrix $R_{|P_{cf}|\times|C|}$ where the i-th row corresponds to the counterfactual path $\{p_{cf_i} \in P_{cf} \mid 1 \leq i \leq |P_{cf}|\}$ and the j-th column is a unique condition $c_j \in C$ $(1 \leq j \leq |C|)$ in the set of all the unique conditions in the tree. Each cell $R_{ij}(1 \leq i \leq |P_{cf}|, 1 \leq j \leq |C|)$ of the path-condition matrix is populated with binary values so that:

$$R_{ij} = \begin{cases} 1, & \text{if } c_j \in p_{cf_i}, \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Similarly, the path $p_f \in P(t)$ leading to the factual prediction is encoded in form of a binary vector $test_{1\times|C|} = [test_1, test_2, ..., test_{|C|}]$. In order to ensure the flexibility of the approach and the consistency of further calculations, we distinguish two ways of populating the test vector $test_{1\times|C|}$ depending on the nature of the tree-based classifier. In the case of a crisp tree-based classifier, populating the factual prediction vector $test_j(1 \leq j \leq |C|)$ is considered a special case of populating a path-condition matrix $R_{1\times|C|}$. Hence, its values are calculated as in (6) for a single row of R. In the case of a fuzzy tree-based classifier, the factual prediction vector contains binarized membership function values as a result of the α-cut,

$$test_j = \begin{cases} 1, & \text{if } \mu(c_j) \geq \alpha \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

- **Counterfactual relevance determination**. If there exists only one counterfactual prediction ($|P_{cf}| = 1$), the only identified counterfactual constitutes the set of counterfactual explanations $E_{cf} = \{p_{cf_1}\}$. Otherwise, candidate counterfactual explanations $E_{cf} = \{e_{cf_1}, e_{cf_2}, ..., e_{cf_{|P_{cf}|}}\}$ are ranked to determine the most relevant one in accordance with their distance to the factual classification. Ranking counterfactuals allows us to ensure that the test instance and the best counterfactual data point are minimally different. To do so, we calculate the bitwise XOR-based distance for each pair of vectors $\langle test_{1\times|C|} : r_{1\times|C|}\rangle$ for $\forall r_i \in R$ $(1 \leq i \leq |P_{cf}|)$, normalized over the number of conditions and thus transformed into a scalar:

$$dist(test, r) = \frac{\sum_{i=1}^{|r|} 1[test_i = r_i]}{|r|}. \tag{8}$$

All the obtained distance values are sorted to enable us to find the minimally distant counterfactual $e_{cf}(x, t)$, so that it is claimed to be the most relevant to complement the factual classification. If multiple counterfactuals have the same minimal distance, they are claimed equivalently relevant, so the most optimal counterfactual is picked randomly.

Given the consecutive root-to-leaf paths $p_f(t)$ to the factual explanation and $p_{cf}(t)$ to the most optimal counterfactual, they are browsed in the top-down fashion to identify a *critical condition*, i.e., the node where the corresponding branches of the tree first diverge. Finally, the remainders of the paths are inspected for *differing* feature

values constituting the antecedents of the counterfactual explanations.

- **NLG surface realization**. All explanations are designed to be two-sentence pieces of text where the first sentence is a linguistic realization of a factual explanation $e_f(x,t)$ whereas the second offers the best ranked counterfactual explanation $e_{cf}(x,t,y_{cf}) \in E_{cf}$. It is important to note that crisp tree-based explanations are supported by linguistic approximations. More precisely, each numerical condition c_j associated to feature f_j is verbalized as "f_j is L^f_{ja}", being L^f_{ja} the most similar linguistic term in $L(f_j)$. Namely, we compute the similarity degree $S(A,B_l)$ between each pair of numerical intervals A and B_l as follows:

$$S(A,B_l) = \frac{A \cap B_l}{A \cup B_l} \in [0,1], l \in [1,|L(f_j)|] \quad (9)$$

being $S(A,B_l) = 1$ in case A perfectly matches B_l, and $S(A,B_l) = 0$ if both intervals are disjoint. A is the numerical interval representing c_j. B_l is the numerical interval associated to each linguistic term in $L(f_j)$, i.e., it is the numerical interval associated to the j-th fuzzy set in the SFP associated to feature f_j, when such fuzzy set is truncated in accordance with the selected α-cut. For instance, given a feature $f_j \in U = [U_{min}, U_{max}]$, c_j = "$f_j \leq a$", $a \in U$, $A = [U_{min}, a]$. In addition, if f_j were defined by a uniform SFP with two linguistic terms (*Low*, *High*), α=0.5, $B_{Low} = [U_{min}, \frac{U_{min}+U_{max}}{2}]$ and $B_{High} = [\frac{U_{min}+U_{max}}{2}, U_{max}]$. Let's suppose $U_{min} = 10$, $U_{max} = 50$, and $a = 20$. Then, $A = [10,20]$, $B_{Low} = [10,30]$, $B_{High} = [30,50]$, $S(A,B_{Low}) = 0.5$, $S(A,B_{High}) = 0$. Accordingly, we conclude that "f_j is *Low*".
Then, the linguistic realization module would construct an explanation from the antecedent $\{c_i = (f_i, v_i)\} : (1 \leq i \leq k)$ and the consequent of the corresponding explanations ($cl_x \in Y$ for a factual explanation and $y_{cf} \in \{Y \setminus cl_x\}$ for the best counterfactual explanation). The following template is used for all combinations of such explanations:
The test instance is of class $\langle cl_x \rangle$ *because* f_1 *is* v_1 *(and* f_2 *is* v_2 *(and ... (and* f_k *is* v_k*))). It is not of class* $\langle y_{cf} \rangle$ *because* $e_{cf}(x,t,y_{cf})$. It is worth noting that the counterfactual explanation can be paraphrased to point to how the output decision can be changed as follows: *"The test instance would have been of class* $\langle y_{cf} \rangle$ *if it were the case that* $e_{cf}(x,t,y_{cf})$*"*.

C. Illustrative example

In order to illustrate the use of the proposed method, let us consider a set of two input DTs $T = \{T_c = \{t_1\}, T_f = \{t_2\}\}$ where tree t_1 is generated by an arbitrary crisp tree-based classifier and tree t_2 – by an arbitrary fuzzy inference system. The trees are known to have been trained on the *BEER* dataset [28] where 8 beer styles (Blanche, Lager, Pilsner, IPA, Stout, Barleywine, Porter, and Belgian Strong Ale) are

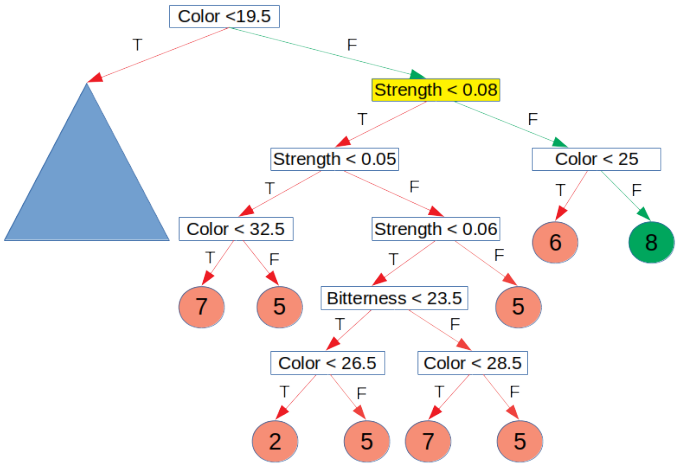

Fig. 2. A sample crisp tree t_1. The decision path corresponding to the factual explanation is colored in green (i.e., Color $<$ 19.5 is False, Strength $<$ 0.08 is False, Color $<$ 25 is False, then output is 8). The decision paths to the leaf nodes colored in red are potential counterfactuals. The node in yellow (i.e., Strength $<$ 0.08) is the critical condition

classified in terms of the following 3 features (see further details in Table II):

Color $\in [0,45]$: *Pale*, *Straw*, *Amber*, *Brown*, *Black*.
Bitterness $\in [8,250]$: *Low*, *Low-medium*, *Medium-high*, *High*.
Strength $\in [0.039, 0.136]$: *Session*, *Standard*, *High*, *Very high*.

Let us sample a test instance x with the following feature-value pairs: $\langle$(Color,28),(Bitterness,39),(Strength,0.089)$\rangle$.

This test instance is assumed to be of class 8 (i.e., Belgian Strong Ale). The given set of feature-value pairs corresponds to the following set of linguistic terms: $\langle$(Color,Brown),(Bitterness,Medium-high),(Strength,High)$\rangle$. A fragment of the crisp DT t_1 is shown in Fig. 2.

The fuzzy DT t_2 can be reconstructed directly from the inferred rule-base. Due to space limitations, let us restrict ourselves to presenting the rules activated during the inference process only (the corresponding firing degrees are indicated in brackets):

RULE 15: (0.6) IF Color is Brown
AND Strength is High THEN Class is BelgianStrongAle

RULE 16: (0.4) IF Color is Brown
AND Strength is Very high THEN Class is BelgianStrongAle

For the sake of simplicity, assume that we apply the MIN-MAX inference mechanism, then the rule with the maximal firing degree (RULE 15) determines the fuzzy classification.

Then, the factual explanation $e_f(x,t_1)$ retrieved from the crisp tree corresponds to the path $p_f(t_1) = \{(\text{Color} \geq 19.5), (\text{Strength} \geq 0.08), (\text{Color} \geq 25)\}$. In addition, the linguistically approximated explanation associated to the crisp tree is represented in the form of the following linguistic terms: $p_{f_{approx}}(t_1) = \{(\text{Color} = \text{Black}), (\text{Strength} = \text{High})\}$. Similarly, the factual explanation $e_f(x,t_2)$ obtained from

the fuzzy DT constitutes the path $p_f(t_2) = \{(\text{Color} = \text{Brown}), (\text{Strength} = \text{High})\}$.

Due to space limitations, let us consider the process of counterfactual explanation generation only for the crisp tree t_1. For illustrative purposes, let us indicate the set of potential counterfactual explanations for the crisp tree for class 5 (i.e., Stout). As can be inferred from Fig. 2, it contains (at least) four candidate explanations $E_{cf}(x, t_1, Y_{cf} = \{5\}) = \{e_{cf_1}, ..., e_{cf_4}, ...\}$. Given a set of conditions $C = \{c_1 = \text{“}Color \geq 19.5\text{”}, c_2 = \text{“}Strength < 0.08\text{”}, c_3 = \text{“}Strength < 0.05\text{”}, c_4 = \text{“}Color \geq 32.5\text{”}\ c_5 = \text{“}Strength \geq 0.05\text{”}, c_6 = \text{“}Strength < 0.06\text{”}, c_7 = \text{“}Strength \geq 0.06\text{”}, c_8 = \text{“}Bitterness < 23.5\text{”}, c_9 = \text{“}Bitterness \geq 23.5\text{”}, c_{10} = \text{“}Color \geq 26.5\text{”}, c_{11} = \text{“}Color \geq 28.5\text{”}\}$, where $\{c_i \mid 1 \leq i \leq |C|\} \in C$, the four candidate counterfactual explanations are as follows:

$$e_{cf_1} = \{p_{cf_1} = \{c_1, c_2, c_3, c_4\}, y_{cf_1} = \{5\}\}; \quad (10)$$

$$e_{cf_2} = \{P_{cf_2} = \{c_1, c_2, c_5, c_6, c_8, c_{10}\}, y_{cf_2} = \{5\}\}; \quad (11)$$

$$e_{cf_3} = \{P_{cf_3} = \{c_1, c_2, c_5, c_6, c_9, c_{11}\}, y_{cf_3} = \{5\}\}; \quad (12)$$

$$e_{cf_4} = \{P_{cf_4} = \{c_1, c_2, c_7\}, y_{cf_4} = \{5\}\}. \quad (13)$$

Traversing the tree in the top-down manner, we find it that the critical condition where the paths to the factual prediction and all the counterfactual predictions (including the minimally different) is “Strength < 0.08”. As we construct the path-condition matrix and calculate the distances for all the candidate counterfactuals to the test instance in accordance with (8), we find that the optimal counterfactual explanation for t_1 is e_{cf_4} (13).

Thus, the generated factual explanation associated to the given test instance x and t_1 is therefore as follows: *The test instance is of class Belgian Strong Ale because color is black and strength is high*. The counterfactual explanation for the class “Stout” is: *It is not of class Stout because strength is not standard*. Alternatively, the counterfactual can be paraphrased as follows: “It would have been Stout if strength were standard”.

D. Implementation details

First, we used two ML open source tools for building the AI-based systems under consideration:

- The crisp DTs were built with the ML tool WEKA [29]. We considered the following algorithms: J48, REPTree, and RandomTree. All of them are variants of the well-known C4.5 algorithm first introduced by Quinlan [30].
- The fuzzy modeling tool GUAJE [31] was used to build the fuzzy rule-based classifiers and export them to the JFML format [32]. We considered the implementation of the fuzzy DT algorithm provided by GUAJE. Two fuzzy rule-based classifiers were generated, i.e., the original pruned fuzzy tree (GUAJE-DT) and its simplified version (GUAJE-DT-S) after running the linguistic simplification procedure provided by GUAJE.

TABLE I
SUMMARY OF CHARACTERISTICS OF THE PRETRAINED CLASSIFIERS

	Leaves/rules	***TRL***[a]	***RCCI***[b]	***F-score***
J48	9.8	23.4	95	0.9481
RandomTree	26.8	74.7	93.25	0.93
REPTree	8	18	95.25	0.9501
GUAJE-DT	23.3	51.6	93.42	0.935
GUAJE-DT-S	14.4	32.2	93.63	0.9381

[a]Total rule length
[b]Ratio of correctly classified instances

Then, we implemented the new factual and counterfactual explanation generation method in Python 3.7. Notice that the package Py4JFML [33] was used to deal with the fuzzy classifiers in the JFML format in Python. It is worth noting that FML stands for Fuzzy Markup Language and it complies with the IEEE Std 1855-2016, i.e., the only standard that is recognized world-wide for fuzzy modeling. In addition, the package SimpleNLG [34] was used to implement the NLG surface realization module. For the sake of reproducibility, all the source code and complementary materials are made available online [35].

IV. EVALUATION

A. Experimental settings

In order to assess the performance of our framework, we performed a series of experiments on several pretrained decision trees and fuzzy rule-based classifiers. In our experiments, we generated textual factual and counterfactual explanations for single test instances for selected pretrained classifiers (as specified in III-D). Table I summarizes the characteristics of the classifiers under consideration. All of them exhibit a good interpretability-accuracy trade-off, i.e., accuracy metrics (RCCI and F-score) get high values while interpretability metrics (Leaves/rules, and TRL) remain with small values.

In order to perform a comparative analysis of the generated explanations, we produce the explanations for fuzzy classifiers as well as for crisp trees. The fuzzy inference systems used in the experiments made use of the Mamdani-style inference with the MIN-MAX method. Default weights (i.e., 1.0) were applied to each rule. The α-cut threshold was set to 0.5 in all cases.

We ran the experiments on the *BEER* dataset [28], which contains 400 data instances, each of them having three numeric features: Color, Bitterness, and Strength. Each feature is characterized by a SFP which was defined by an expert. Table II shows the numerical intervals associated to linguistic terms in the SFPs when considering the 0.5-cut. The classification task consists of selecting one out of the following 8 beer styles: Blanche, Lager, Pilsner, IPA, Stout, Barleywine, Porter, Belgian Strong Ale.

TABLE II
EXPERT NUMERICAL INTERVALS ASSOCIATED TO THE INPUT FUZZY SETS

Feature	*Linguistic term*	*Range of values*
Color	Pale	0 ... 3
	Straw	3 ... 7.5
	Aber	7.5 ... 19
	Brown	19 ... 29
	Black	29 ... 45
Bitterness	Low	7 ... 21
	Low-medium	21 ... 32.5
	Medium-high	32.5 ... 47.5
	High	47.5 ... 250
Strength	Session	0.035 ... 0.0525
	Standard	0.0525 ... 0.0675
	High	0.0675 ... 0.09
	Very high	0.09 ... 0.136

TABLE III
EVALUATION RESULTS

	NumCF	*BestMinDist*	*FactLength*	*CFLength*
J48	8.8	0.1157	2.315	1.3497
RandomTree	25.8	0.0472	2.64	1.8702
REPTree	7	0.1429	2.25	1.2629
GUAJE-DT	20.3575	0.0769	2.065	2.1139
GUAJE-DT-S	12.6781	0.081	2.0821	2.1446

We applied 10-fold cross-validation to evaluate the generated explanations. The following evaluation criteria were used to assess the quality of the generated factual and counterfactual explanations:

- **Number of counterfactuals (NumCF)**: the average number of candidate counterfactuals;
- **Best minimal distance (BestMinDist)**: the averaged minimal distance to the best counterfactual as defined above;
- **Length of a factual explanation (FactLength)**: the number of conditions in the generated factual explanation;
- **Length of the best counterfactual explanation (CFLength):** the number of conditions in the generated best ranked counterfactual explanation.

B. Analysis of results

Table III summarizes the quality metrics associated to the explanations generated in our experiments. In the light of the reported results, we can make a number of important remarks.

First, the generated factual and counterfactual explanations provide a concise representation of the most critical features of the given test data instance. Altogether, they do not only summarize the known characteristics of the test instance but also complement them with relevant hypothetical scenarios of other most likely outcomes.

Second, the resulting explanations remain short and comprehensive irrespective of the overall number of candidate counterfactuals. Given a greater number of counterfactuals for both fuzzy classifiers, the average length of their factual and counterfactual explanations remains close to each other. This is hypothesized to be due to the effectiveness of the pruning method applied to the set of potential candidate counterfactuals. Furthermore, such compact and balanced explanations are believed to be an effective means of explaining classifier's predictions in the most user-friendly manner.

Third, fuzzy classifiers show better general performance in comparison to crisp tree algorithms with respect to generating minimally distant counterfactuals. Indeed, the fuzzy classifiers enhanced with linguistically approximated explanations appear to infer more relevant conditions. The empirical results show that it is only due to a much more complex tree structure how the RandomTree algorithm generates counterfactuals of a higher degree of proximity to the test instance. On average, the other crisp tree algorithms show worse performance.

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a new method for factual and counterfactual explanation generation. This method is applicable to both crisp and fuzzy decision trees. Moreover, it has also been used with fuzzy rule-based classifiers.

The initial results open broad perspectives on further development of the designed framework and evaluation of subsequent results. From the algorithmic point of view, integrating more complex tree representations could enhance the trustworthiness of the system. For instance, one of algorithmic features to be investigated in future is measuring the impact of weights (weighted edges in crisp decision trees and weighted fuzzy rules in fuzzy inference systems) in terms of explainability.

In addition, it seems promising to make use of combined predictions generated for the same test instances simultaneously. While the classifiers employed in this work present relatively accurate predictions, it seems particularly important to test whether user trust increases in case of ambiguous test points where the data instance to be explained lies near the decision boundary between two or more classes. Providing a user with a fair explanation for an uncertain classification may result in decreased trust to the system, in general. As counterfactual explanations are assumed to be capable of providing a deeper insight in reasoning mechanisms, investigating ways of explaining uncertain classifications in terms of other reliable predictions might soften doubtfulness about system's performance.

Human evaluation of the designed framework is among the next immediate steps reserved for future work. It is a comparative analysis of several types of explanations for various classifiers that could shed some light on further effectiveness of pairing factual and counterfactual explanations. In addition, the practical value of the designed framework is planned to be enhanced by a series of experiments on datasets from domains where high-stakes decisions are widely found, e.g., the healthcare domain.

Finally, the original method could be used among the main modules of a cognitive framework that interacts with the end-user in a dialog to better explain a classifier's decisions. For instance, designing an argumentation-based framework for generating (counter-)factual explanations appears a prospective line of research in human-machine interaction.

ACKNOWLEDGMENT

Jose M. Alonso is a *Ramon y Cajal* Researcher (RYC-2016-19802). Alejandro Catala is a *Juan de la Cierva* Researcher (IJC2018-037522-I). This research is funded by the Spanish Ministry of Science, Innovation and Universities (grants RTI2018-099646-B-I00, TIN2017-84796-C2-1-R, TIN2017-90773-REDT, and RED2018-102641-T) and the Galician Ministry of Education, University and Professional Training (grants ED431F 2018/02, ED431C 2018/29, ED431G/08, ED431G2019/04). Some of the previous grants were co-funded by the European Regional Development Fund (ERDF/FEDER program).

REFERENCES

[1] A. Adadi and M. Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[2] European Commission, "Artificial Intelligence for Europe," European Commission, Brussels, Belgium, Tech. Rep., 2018, Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions (SWD(2018) 137 final).

[3] Parliament and Council of the European Union, "General data protection regulation (GDPR)," 2016, http://data.europa.eu/eli/reg/2016/679/oj.

[4] C. Molnar, *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable*. Leanpub, 2019.

[5] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, pp. 206–215, 2019.

[6] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[7] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems?" *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.

[8] H. Sharma and S. Kumar, "A Survey on Decision Tree Algorithms of Classification in Data Mining," *International Journal of Science and Research (IJSR)*, vol. 5, no. 4, pp. 2094–2097, 2016.

[9] K. Sokol and P. Flach, "Conversational Explanations of Machine Learning Predictions Through Class-contrastive Counterfactual Statements," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 5785–5786.

[10] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 1, pp. 28–44, 1973.

[11] E. Reiter, "Natural Language Generation Challenge for Explainable AI," in *Proceedings of the 1st Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence (NL4XAI)*. ACL, 2019, pp. 3–7.

[12] J. M. Alonso, C. Castiello, and C. Mencar, "Interpretability of Fuzzy Systems: Current Research Trends and Prospects," in *Springer Handbook of Computational Intelligence*, 2015, pp. 219–237.

[13] J. M. Alonso, A. Ramos-Soto, E. Reiter, and K. van Deemter, "An exploratory study on the benefits of using natural language for explaining fuzzy rule-based systems," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[14] J. M. Alonso, C. Castiello, and C. Mencar, "A Bibliometric Analysis of the Explainable Artificial Intelligence Research Field," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Foundations*. Springer International Publishing, 2018, pp. 3–15.

[15] C. Mencar and J. M. Alonso, "Paving the Way to Explainable Artificial Intelligence with Fuzzy Modeling," in *International Workshop on Fuzzy Logic and Applications*. Springer, 2018, pp. 215–227.

[16] M. Neerincx, J. van der Waa, F. Kaptein, and J. van Diggelen, "Using perceptual and cognitive explanations for enhanced human-agent team performance," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10906, pp. 204–214, 2018.

[17] E. Zhao and R. Sukkerd, "Interactive Explanation for Planning-Based Systems: WIP Abstract," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. ACM, 2019, pp. 322–323.

[18] D. Martens and F. Provost, "Explaining Data-Driven Document Classifications," *Management Information Systems Quarterly*, vol. 38, no. 1, pp. 73–100, 2014.

[19] J. Kunkel, T. Donkers, L. Michael, C.-M. Barbu, and J. Ziegler, "Let Me Explain: Impact of Personal and Impersonal Explanations on Trust in Recommender Systems," in *Conference on Human Factors in Computing Systems (CHI)*. ACM, 2019, pp. 487:1–487:12.

[20] J. van der Waa, M. Robeer, J. van Diggelen, M. Brinkhuis, and M. Neerincx, "Contrastive Explanations with Local Foil Trees," in *Workshop on Human Interpretability in Machine Learning (WHI)*, 2018.

[21] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini, "Factual and Counterfactual Explanations for Black Box Decision Making," *IEEE Intelligent Systems*, vol. 34, no. 6, pp. 14–23, 2019.

[22] K. Sokol and P. Flach, "Glass-Box: Explaining AI Decisions with Counterfactual Statements through Conversation with a Voice-Enabled Virtual Assistant," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 2018, pp. 5868–5870.

[23] C. Russell, "Efficient Search for Diverse Coherent Explanations," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*. ACM, 2019, pp. 20–28.

[24] R. Mothilal, A. Sharma, and C. Tan, "Explaining Machine Learning Classifiers through Diverse Counterfactual Explanations," in *Conference on Fairness, Accountability, and Transparency (FAT*)*. ACM, 2020.

[25] L. Console, C. Picardi, and D. T. Dupré, "Temporal Decision Trees: Model-Based Diagnosis of Dynamic Systems on-Board," *Journal of Artificial Intelligence Research*, vol. 19, no. 1, pp. 469—512, 2003.

[26] X. Wang, B. Chen, G. Qian, and F. Ye, "On the optimization of fuzzy decision trees," *Fuzzy Sets and Systems*, vol. 112, no. 1, pp. 117–125, 2000.

[27] D. E. Knuth, *The Art of Computer Programming, Volume I: Fundamental Algorithms*. Addison-Wesley, 1968.

[28] G. Castellano, C. Castiello, and A. M. Fanelli, "The FISDeT software: Application to beer style classification," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, pp. 1–6.

[29] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[30] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[31] D. P. Pancho, J. M. Alonso, and L. Magdalena, "Quest for interpretability-accuracy trade-off supported by fingrams into the fuzzy modeling tool GUAJE," *International Journal of Computational Intelligence Systems*, vol. 6, pp. 46–60, 2013.

[32] J. M. Soto-Hidalgo, J. M. Alonso, G. Acampora, and J. Alcala-Fdez, "JFML: A Java Library to Design Fuzzy Logic Systems According to the IEEE Std 1855-2016," *IEEE Access*, vol. 6, pp. 54 952–54 964, 2018.

[33] J. Alcala-Fdez, J. M. Alonso, C. Castiello, C. Mencar, and J. Soto-Hidalgo, "Py4JFML: A Python wrapper for using the IEEE Std 1855-2016 through JFML," in *IEEE International Conference on Fuzzy Systems*, 2019, pp. 1–6.

[34] A. Gatt and E. Reiter, "SimpleNLG: A realisation engine for practical applications," *12th European Workshop on Natural Language Generation (ENLG)*, pp. 90–93, 2009.

[35] I. Stepin, "FCFExpGen: Python code for factual and counterfactual textual explanation generation," 2020, https://gitlab.citius.usc.es/jose.alonso/xai.