Bag-of-Words Similarity in eXplainable AI

Sara Narteni[1,2], Melissa Ferretti[1], Vittorio Rampa[*1], and Maurizio Mongelli[1]

[1] *Institute of Electronics, Computer and Telecommunication Engineering (IEIIT)*
National Research Council of Italy (CNR)
Genoa, *Milan, Italy
name.surname@ieiit.cnr.it
[2] *Department of Control and Computer Engineering (DAUIN)*
Politecnico di Torino

Abstract. eXplainable AI (XAI) does not only lie in the interpretation of the rules generated by AI systems, but also in the evaluation and selection, among many rules automatically generated by large datasets, of those that may be more relevant and meaningful for domain experts. With this work, we propose a method for similarity evaluation between rules, which identifies similar rules, or very different ones, by exploiting techniques developed for Natural Language Processing (NLP) scenarios. We evaluate the similarity of *if-then* rules by interpreting them as sentences and generating a similarity matrix acting as an enabler for domain experts to analyse the generated rules and thus discover new knowledge. Rule similarity may be applied to rule analysis and manipulation in different scenarios: the first one deals with rule analysis and interpretation, while the second scenario refers to pruning unnecessary rules within a single ruleset. Rule similarity allows also the automatic comparison and evaluation of rulesets. Two different examples are provided to evaluate the effectiveness of the proposed method for rules analysis for knowledge extraction and rule pruning.

Keywords: eXplainable AI, Rule similarity, Cosine similarity, Bag-of-Words, Physical fatigue detection, Vehicle platooning.

1 Introduction

Artificial intelligence (AI) is increasingly relevant and persistent in all areas of our lives: it is already widely used, for instance, for diagnostic purposes, to predict natural phenomena such as floods and earthquakes, to anticipate and effectively combat fraud in transactions, in product recommendations and in many other areas. The issues raised by the use of AI are increasingly related to legal, ethical and security aspects: think, for instance, of all difficulties involved in bringing self-driving vehicles to the market. The use of AI in even more critical areas (e.g., health) raises legitimate questions about why and how a system reaches a particular decision. The transparency of the models used in the decision-making process and to obtain results as much comprehensible as possible are becoming fundamental issues [1]. Both scientific and political communities

are grappling with these issues and there is a strong effort trying to regulate this matter. An example is the European General Data Protection Regulation (GDPR, see https://gdpr.eu/tag/gdpr/), put into effect in 2018, that introduces the need for a *right to an explanation* when dealing with automated systems.

1.1 eXplainable AI

The problem is that AI algorithms, while achieving considerable results in terms of performance and accuracy, are, in practice, black boxes that fail to meet the emerging need for transparency and trustworthiness. The current Trustworthy AI framework of scientific community is therefore working on creating methods to make such systems understandable by humans. This area of investigation, known as eXplainable AI (XAI) [2], is one of the main open challenges in the AI field [3]. XAI goal is to build trust in AI solutions and in all those systems that incorporate decisions made by them [4]. XAI describes the logic behind a decision by providing transparency regarding its decision-making process, and by presenting information able to explain the strength or weakness of such a decision. By incorporating interpretability and explainability at different levels of AI systems, the models become white-box (i.e., intelligible or explainable) [5] as they are associated to a set of Boolean rules, of the *if-then* type, that fully describe the systems [6].

1.2 Knowledge extraction from XAI

Achieving explainability is however a more subtle issue than just having a set of rules at hand. For the sake of simplicity, we focus here on canonical supervised learning via classification. Depending on the problem under analysis, the resulting rules may be not so simple and intuitive, for example due to intricate distribution of the points in the classes. Geometrically speaking, rules define hyper-rectangles in the feature space. However, their shape may be too simple to follow the boundaries between the classes, that may be far more complex: trying to approximate this complex geometrical shape, the model may end up containing large amounts of rules with low covering (i.e., the percentage of points for which a rule is true). Hence, understanding any logic from rule extraction may be a hard task. However, explainable-by-design models still play an important role because of their low computational cost (rules are provided in a single step), with respect to other post-hoc explainability techniques [7], and allow error control (due to the white-box nature). On the other hand, downgrading a rule-based model by just cutting rules (e.g., those with small covering) may lead to loose information gain. In [8], the problem of ruleset complexity after post-hoc rule extraction from black-box models on high-dimensional sparse datasets is addressed through the introduction of meta-features. Pursuing a similar goal, our approach, based on cosine rule similarity, will enable ruleset simplifications and an easier-to-understand information extraction.

1.3 Rule similarity

A possible approach to explore XAI models relies on the automatic selection of rules. This key technique consists in pruning rules with similar information. Pruning may be triggered either when the syntactic of two rules is very similar (i.e., when they focus on similar conditions over the same features subspace) or when both rules cover the same set of points in the feature space (i.e., in case of semantic similarity).

Automatic computation of rule similarity can be also useful to compare XAI models, e.g., coming from different viewpoints of the same dataset (e.g., forecasts over seasonal time periods) or from different datasets on the same problem (e.g., disease prediction over different populations geographically or in time). That means quantifying the similarity (or the distance) between two sets of rules, namely the *rulesets*, and by focusing the attention to those rules affecting the similarity with the largest contribution.

More specifically, attention should be posed on the evaluation of domain change, bias and discovery of anomalies between two datasets through the related rulesets. To mention quite a few, the fields of applications may be: image analysis, disease prediction and cybersecurity, respectively. Moreover, data augmentation may be empowered by rule similarity as new data should be validated in terms of potential discovery of new knowledge, not available from real data directly [9,10].

Another key application concerns the domain experts that want to find out, in an automatic way, the rulesets with both good performance and low similarities, thus allowing them to extract knowledge from the data.

1.4 Related works

Despite its importance, the literature about rule similarity methods is still surprisingly poor. Quite a few works are available for the automatic computation of rule similarity: reference [9] poses the foundation of the problem, by focusing the attention on differences between syntactic (i.e., features and thresholds of conditions) and semantic (i.e., covering and accuracy). In addition, it defines some metrics in this perspective, as well as visual analytics based on the Principal Component Analysis (PCA) of the similarity results. Reference [11] presents the problem in fuzzy logic systems and outlines the dependencies between linguistic approximation and similarity. To this aim, it makes use of the Jaccard metric [12], as it is widely used and also relatively easy to compute.

Reference [13] builds stable rules stratifying neuroblastoma patients through cross validation, in which each run exploits rule pruning on the basis of a rule distance criterion. In [14] an innovative approach of Jaccard similarity-based rule simplification is performed by applying graph theory to fuzzy rules. Cosine similarity is also adopted in [15] for text similarity in NLP applications and in [16] for association rules clustering from gene expression data, while in [17] it is used for image recognition where it is adopted along with other similarity metrics namely Dice and Tanimoto similarity.

To the best of our knowledge, there is no previous study investigating cosine similarity for knowledge extraction and rule pruning of rules derived from classification of multivariate data.

1.5 Paper contributions

In this paper, we show how the cosine similarity function can be applied to the rule similarity problem. This function measures the similarity between two vectors by looking at the angle between them and determines whether two vectors follow (almost) the same direction or not. We borrowed this approach from the literature about NLP applications where cosine similarity is often used to measure document similarity in text analysis [1]. To apply cosine similarity, rules are transformed into vectors by exploiting the Bag-of-Words (BOW) approach [2]. Then, the cosine similarity function returns a similarity measure based on the number of matching features of the rules and their respective closeness of the thresholds. Rule similarity may drive different kinds of analysis: in this paper, we are interested in rule analysis and, eventually, pruning of unnecessary rules within a single ruleset. This usually happens when several rules are extracted e.g., when a Logic Learning Machine (LLM) is forced to replicate several clusters from the data, that are later mapped into rules. However, this procedure generates a redundant ruleset structure, where the semantic structure of two or more rules may be almost the same. An example of rule analysis is shown in Sect. 5, while another example of rule pruning (with negligible effects on the ruleset performances) is presented in Sect. 6.

1.6 Paper organization

The rest of the paper is organised as follows. In Sect. 2, we introduce the XAI solver, the feature and value ranking used later in the paper. In Sect. 3, we describe how to transform a set of rules into a set of vectors by constructing a Bag-of-Words matrix. In Sect. 4, we define the methodology to measure the similarity between rules by using the corresponding vectors previously built while, in Sect. 5, we show a preliminary simplified example of application of the presented methodology for physical fatigue detection in different group ages. In Sect. 6, we present another practical example of the proposed technique, applied to a set of rules, to predict collision in vehicle platooning. Finally, in Sect. 7, we draw some preliminary conclusions.

2 XAI tools

2.1 Logic Learning Machine

The Logic Learning Machine [4, 18] is a global supervised method that constructs classifiers described by a set of intelligible *if-then* rules (see Sect. 3). It is used here as a XAI solver. This study is however not polarized on the

LLM architecture as any other solver may be considered for ruleset generation, such as canonical decision trees [19], or more complex tree structures such as Skope-rules [20], that use predictive rule generation over an ensemble of decision trees [21]. The rationale behind the choice of the LLM relies on its flexibility in rule generation and its higher accuracy than decision trees [18].

2.2 Feature and value ranking

Feature and value rankings help rule interpretation and knowledge discovery. An example is reported later in Sect. 6 for a smart mobility scenario. Feature ranking indicates the importance of each feature in inferring the right classification (e.g., distance and speed of vehicles are determinant for collision prediction for the example of Sect. 6). It is typically used for feature reduction in order to synthesize the model that uses only the most relevant features. Value ranking gives a similar rank for each feature with respect to the most relevant intervals for classification (e.g., the most relevant intervals of speed for collision are [80, 90] Km/h and [90, 110] Km/h for the example of Sect. 6). The candidate intervals to be ranked are derived from the thresholds available in the conditions of the rules.

Whatever the XAI solver is, both feature and value rankings may be easily derived from the ruleset, by applying sensitivity analysis on model accuracy, with and without the feature to be ranked, or, in case of value ranking, with respect to the interval of a feature to be ranked. The interested reader is referred to [13] for further details on that subject. These two kinds of ranking are later used to introduce the problem, before the application of rule similarity. This may help understanding the intuition behind the rulesets comparison and the need of further investigation via rule similarity. It is finally worth noting that the proposed methodology is independent of the XAI solver at hand.

3 Transforming rules into vectors

As already mentioned, the proposed approach is inspired by the analysis of texts similarity found in the algorithms used in NLP applications [15]. Treating our problem as a problem of linguistic nature, our rules can be equated to *special sentences* where we are interested in the meaning of words and not in their arrangement within the sentence since, given the peculiarity of the sentences, the order is irrelevant for the result. Thus, we concentrate only on the semantics of our rules in the *if-then* form:

$$\textbf{if} < premise > \textbf{then} < consequence >$$

where $< premise >$ is a logical product of conditions on the input features, while $< consequence >$ specifies the output class (or classes). The ruleset is composed by R rules arranged in the set:

$$RuleSet = \{R(r)\}_{r=1}^{R}. \tag{1}$$

For the sake of simplicity, but with no loss in generality, each rule $R(r)$ from the previous ruleset is composed by N_r conditions $br_n(r)$ and one class (i.e., binary) assignment $y(r)$:

$$R(r) = (\{br_n(r)\}_{n=1}^{N_r}, y(r)) \tag{2}$$

where the generic n-th condition

$$br_n(r) = (fs_n(r), t_n(r)) \tag{3}$$

is composed by two terms: the first term $fs_n(r)$ indicates the words that belong to the feature name, combined with a comparison operator ($<$, $>$) e.g., $f_1 <$, while the second one $t_n(r)$ is the corresponding threshold after the comparison operator (e.g., the term t_1 in $f_1 < t_1$).

First, we transform our rules into vectors, by constructing the BOW matrix. The Bag-of-Words is a model used in NLP to represent texts [2]. According to this model, a text is seen as an unordered collection of words that make it up, regardless of syntax and word order.

Actually, in the case of text classification, a word in a document is assigned to a weight. This weight can be related to the frequency of the word, or the relative frequency of terms, or a combination of term frequency and inverse document frequency (tf-idf)[1]. The words, together with their weights, form the BOW matrix. The cosine distance is then applied to the terms of the matrix to determine the similarity between the corresponding texts.

Our case deals with the BOW matrix built on the *RuleSet* at hand. In order to define the BOW matrix for a specific set of rules, we must first define what a *word* is in this context and consequently how we assign weights to the different *words* that build the rules.

From the *RuleSet*, we thus extract:

$$FS = \{fs_n(r)\}_{n=1,\ldots,N_r;r=1,\ldots,R} \tag{4}$$

with the corresponding thresholds:

$$T = \{t_n(r)\}_{n=1,\ldots,N_r;r=1,\ldots,R} \tag{5}$$

The latter set can be further divided into two groups: thresholds on ordered features and thresholds on categorical features.

The syntactic differences of the thresholds (under the same $fs_n(r)$) are topical as they determine the level of similarity between each generic couple $fs_n(r)$ and $fs_m(s)$ in comparison (with $n \neq m$ and $r \neq s$). The construction of the BOW matrix must be coherent with this assumption.

Such a matrix is composed by a set of columns defined as follows. A first subset of columns refers to the elements in FS while a second subset of columns refers to the elements in T. Each row r of the matrix corresponds to the r-th rule in the *RuleSet*. The BOW matrix is detailed in the next sub-sections.

[1] The inverse document frequency weight function (tf-idf) is a function used in information retrieval procedures to measure the importance of a term with respect to a document or a collection of documents.

3.1 FS columns

Since, in the conditions, repetitive occurrences of instances of feature plus operator (e.g., $f_1 <$) could exist in FS (i.e., the same conditional comparison may appear on the same feature with the same operator in different rules), we define the set FS' that contains only all single instances of FS without repetitions, denoted with index k, by $f's_k(r)$. For each element k in FS', a column of the BOW matrix, Cfs_k, is derived as follows:

$\forall r : R(r) \in RuleSet$:

$$Cfs_k(r) = \begin{cases} 1, & \text{if } f's_k(r) \in R(r) \\ 0, & \text{if } f's_k(r) \notin R(r) \end{cases} \tag{6}$$

The index r denotes the row of the matrix and the corresponding rule from which the $f's_k(r)$ element has been taken.

3.2 Numerical thresholds columns

The numerical threshold columns Ctn_k of the BOW matrix are derived for each $f's_k(r)$ as follows. First of all, the following normalisation is applied to the threshold values of each rule $t_k(r)$, by taking as a reference the maximum $\max(t_k(r))$ and minimum $\min(t_k(r))$ of the thresholds over k and r in the ruleset:

$\forall r : R(r) \in RuleSet$:

$$Ctn_k(r) = \begin{cases} \sigma(t_k(r)), & \text{if } \max(t_k(r)) \neq \min(t_k(r)) \\ 1, & \text{if } \max(t_k(r)) = \min(t_k(r)) \\ 0, & \text{if } f's_k(r) \notin R(r) \end{cases} \tag{7}$$

where $\sigma(t_k(r))$ is given by:

$$\sigma(t_k(r)) = \frac{t_k(r) - \min(t_k(r))}{\max(t_k(r)) - \min(t_k(r))}. \tag{8}$$

3.3 Categorical thresholds columns

The same approach applies also to the categorical features.

$\forall r : R(r) \in \ RuleSet$:

$$Ctc_k\,(r) = \begin{cases} 1 \text{ if } f's_k\,(r) \in R\,(r) \wedge t_k\,(r) \in R\,(r) \\ 0 \text{ if } f's_k\,(r) \notin R\,(r) \\ \quad \vee\,(f's_k\,(r) \in R\,(r) \wedge t_k\,(r) \notin R\,(r)) \end{cases} \tag{9}$$

4 Similarity measure

The similarity score s is defined to operate on the cross product of each couples of rules under comparison. As already said, every row of the matrix corresponds to a rule. The similarity score is a mapping

$$s : RuleSet \times RuleSet \rightarrow \Re$$

that returns a value s in the interval [0,1]. Moreover, given two generic rules $R(u)$ and $R(v)$ with $u \neq v$, it is $s(R(u), R(v)) = 1$ iff $R(u) = R(v)$. Unlike distance functions, the greater the value of the similarity function, the closer the two rules are. Furthermore, we have to consider that, under the BOW construction procedure of Sect. 3, we obtain a relatively large sparse matrix.

Traditional distance measures do not work well for sparse matrices. In fact, two rules may indeed have many 0-values in common, but this does not make them similar. We therefore need a measure that focuses on the terms that the two rules have in common and the similarity of their thresholds, thus ignoring null terms with 0-values occurrences.

Referring to the experience in the field of text analysis, we know that the cosine similarity is a measure that works very well with very large sparse matrices and provides a measure of similarity just as described above. For instance, by using the values collected in the matrices according to the procedure described above in (6)-(9) for the generic couple of rules $R(u)$ and $R(v)$, we construct the corresponding column vectors $\mathbf{W}(u)$ and $\mathbf{W}(v)$. Then, we can define the cosine similarity as:

$$s(\mathbf{W}(u), \mathbf{W}(v)) = \frac{\mathbf{W}(u)^T \mathbf{W}(v)}{\|\mathbf{W}(u)\| \, \|\mathbf{W}(v)\|}, \tag{10}$$

where $\|\cdot\|$ is the Euclidean norm. From this, we can see that when the cosine of the angle formed by the two vectors is 0 the two vectors are perpendicular and therefore maximally different, while as the two vectors get closer, the angle between them tends to zero and the cosine tends to 1. When both vectors coincide, and their cosine is therefore 1, the two vectors are maximally similar. If we apply the cosine similarity to the cross product of our *RuleSet*, we get a symmetrical matrix where the value in row u and column v represents the cosine similarity measure between rules $R(u)$ and $R(v)$. The closer this number is to 1, the more similar the rules are.

5 Introductory application example

In order to show how the presented methodology works, we report in this section a first applicative example in the context of Physical Fatigue Prediction (PFP). In this scenario, based on an open-source dataset [22], subjects are asked to perform a simulation of an industrial task while some wearable IMU (Inertial Movement Units) register their activity: the collected measures are then elaborated to extract 38 features. During the performance, their fatigue level is

self-evaluated by the participants themselves by using the Borg scale [23]. According to it, a Rate of Perceived Exertion (RPE) above or equal to 13 indicates a fatigued state, otherwise non-fatigued: hence, the data are labeled according to this criterion. The fatigue-related dataset is divided in two age groups, one formed by subjects up to 40 years old (namely, under 40 in the following) and the other made up of over 40 years old subjects (namely, over 40 in the following). Both groups are then compared according to the rule similarity approach.

The first step is performed by applying the Logic Learning Machine [4] algorithm, which is a global supervised explainable algorithm that builds classifiers described by a set of intelligible *if-then* rules in the form previously shown in eqs. (1)-(3). Due to the high number of involved features, the computed similarities were initially too low and not significant (no results are shown here). Therefore, we performed the rules extraction for each group by applying the LLM algorithm to the initial ruleset taking into account only the first 5 most important features, chosen through the LLM feature ranking procedure [24]. In particular, for both datasets, the LLM algorithm provided a total number of 16 rules, grouped according to the age:

- over 40 group: 4 rules for the non-fatigued class (namely, $R(1), \ldots, R(4)$) and 3 rules for the fatigued class ($R(5), R(6), R(7)$);
- under 40 group: 5 rules for the non-fatigued class ($R(8), \ldots, R(12)$) and 4 rules for the fatigued class ($R(13), \ldots, R(16)$.

The most interesting comparisons are between rules of the same output class for different age groups. We computed the rule similarities in this case, for both fatigued and non-fatigued classes, and we found out that the rules $R(5)$ and $R(14)$ were the ones with the highest similarity value equal to 0.69:

- $R(5)$ (Over 40 - Fatigued): $hip.ACC.Mean <= 3.96$
- $R(14)$ (Under 40 - Fatigued): $hip.ACC.Mean <= 3.73 \wedge$ $leg.rotational.velocity.sag.plane <= 598.11$

For the sake of simplicity, by focusing only on the rules above and disregarding all the other FS and T terms referring to features that do not appear in the rules $R(5)$ and $R(14)$, we can write the simplified BOW matrix as in Tab. 1 (where $hip.ACC.Mean$ is denoted as f_1 and $leg.rotational.velocity.sag.plane$ as f_2). The values reported in the second and fourth columns of the table are obtained as described in Section 3.1, while the values in the third and fifth columns of the same table are computed as explained in Section 3.2. Based on such values, the rule similarity is computed through the formula in (10), where $\mathbf{W}(5) = [1, 0.16, 0, 0]^T$ and $\mathbf{W}(14) = [1, 0, 1, 0]^T$.

It is worth noticing that the *hip* feature determines the similarity for the fatigued class of the two age groups. The corresponding thresholds in the two rules are very similar as well. On the other hand, the *leg* feature for the under 40 group suggests that the *hip* status is not sufficient to move the under 40 group into the fatigued class. In turn, this result may suggest that feature $leg.rotational.velocity.sag.plane$ is able to discriminate between the age groups.

Table 1. Simplified BOW matrix for the fatigued class

Rules	$f_1 <=$	$V_{f_1<=}$	$f2 <=$	$V_{f_2<=}$
$R(5)$	1	0.16	0	0
$R(14)$	1	0	1	0

In this example, the dataset split between age groups was chosen *a-priori*, according to the original data distribution. However, the presented rule similarity approach may be iteratively applied to different age splits and the obtained rule similarity values could be useful to individuate which is the most adequate age stratification for the fatigue problem. Analogies and differences could be automatically extracted for each group and a selection of the most significant ones could be finally presented to a clinician expert in this field.

6 Collision prediction in vehicle platooning

For a more structured and detailed example, we used a classification problem of collision detection in vehicle platooning [4]. In smart mobility scenarios, vehicle platooning is a challenging problem aiming at getting a group of vehicles to travel autonomously and safely by finding a compromise between performance (e.g., speed) and safety. The basic idea of the platoon is to have one primary vehicle that controls the speed by exchanging messages (i.e., data packets) through a wireless communication channel. All vehicles, that follow the platoon leader, communicate with each other and respond to the movements of the leading vehicle.

In the example considered here, a platoon of vehicles (consisting of a variable number of vehicles) was assumed to be traveling at a constant speed and at a constant reciprocal inter-vehicular distance according to the Cooperative Adaptive Cruise Control (CACC) scenario described in detail in [4]. The event that can lead to a collision is a braking force applied by the platoon leader. As a result of this braking, a collision is assumed to occur as shown in [4] if two vehicles are less than 2 meters apart.

The used dataset, generated by the simulator *Plexe 3* [25], consists of 20000 samples and includes 5 features, namely: the number of vehicles in the platoon (N), the braking force ($F0$), the Packet Error Rate (PER) during communication, the initial distance between vehicles ($d0$), and the initial speed ($v0$). Here, we apply the rule similarity method according to the same scenario of [4], where PER is constant for each simulation run but uniformly randomly selected in the interval [0.1, 0.9]. As seen in the previous work [4], PER is a parameter that largely influences collision events. As shown in the PER histogram of the original dataset of Fig. 1, there is no a clear evidence of a PER threshold separating collision and no collision samples.

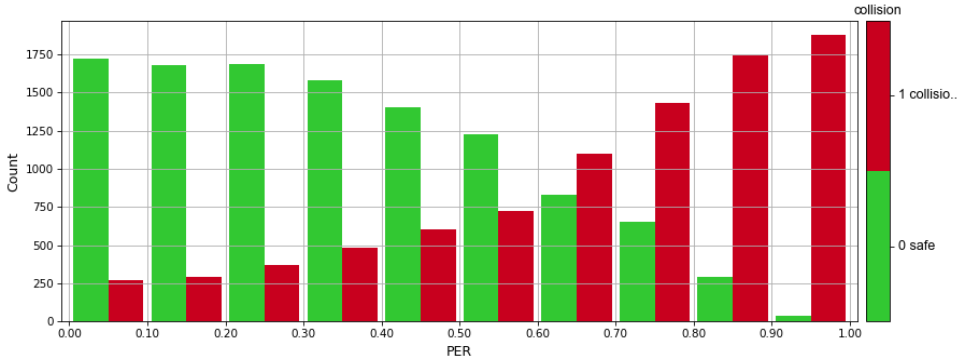

Fig. 1. Packet Error Rate (PER) histogram of the original dataset.

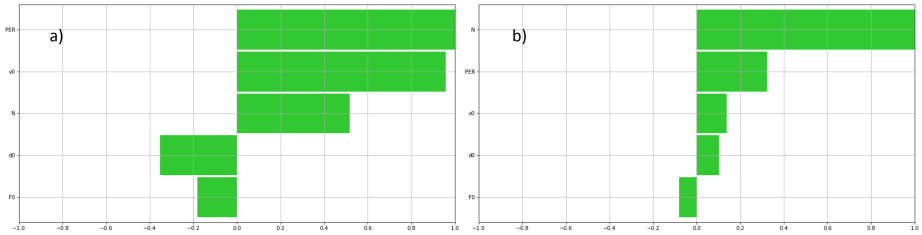

Fig. 2. From left to right: a) feature ranking for collision (PERH); b) feature ranking for collision (PERL).

We thus want to analyse how these differences affect the generation of the respective rules through similarity analysis. To this aim, we create two new datasets and obtain the corresponding LLM rulesets: one which contains all samples with $PER >= 0.5$ (PER High, namely PERH in the following) and the other that contains the samples with $PER < 0.5$ (PER Low, namely PERL). Both datasets have a comparable number of instances: 10127 and 9873 for the first and second set, respectively. However, the former has a much higher number of collisions (i.e., 6946) than the latter (i.e., 1941). The generation of these two rulesets (i.e., PERH and PERL) leads to the definition of 39 collision and no collision rules with no repetitions. These rules are summarized in Tab. 2.

It is interesting also to compare, in Fig. 2.a and 2.b, the two feature rankings (PERH versus PERL). These figures suggest similar feature importance for collision prediction, but with remarkable differences[2]. The value ranking figures 3.a and 3.b show the most important intervals of values for the feature $v0$ in the collision class [13]: in our case, the resulting value rankings for PERH and

[2] The order of the features in the ranking of Fig. 2.a is not the same of the one of 2.b, as it is also highlighted, in Fig. 3.a and 3.b, by the difference in the initial speed value rankings.

Table 2. Summary of generated rules

	Total rules	*Collision rules*	*Non collision rules*
PER HIGH	27	14	13
PER LOW	12	4	8

PERL cases present different thresholds and relevance. It is worth noting that the value ranking plot individuates specific values for the features, which are likely to be unknown even by domain experts. Both investigated rulesets outline an ideal situation to discover not trivial rule similarity.

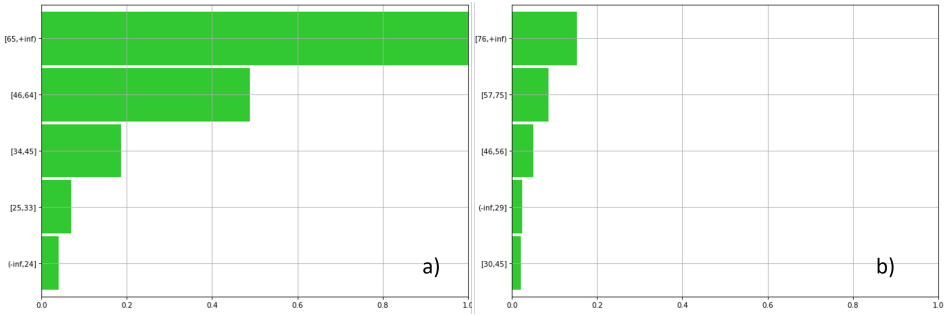

Fig. 3. From left to right: a) value ranking of initial speed for collision (PERH); b) value ranking of initial speed for collision (PERL).

According to the 39 rules in Tab. 2, we setup the BOW matrix and compare the two rulesets PERL and PERH: each rule has a tag identifying the corresponding ruleset and the respective classification output: collision (CO) or no collision (NC). As an example of reduced complexity, we consider at first the following five rules:

- $R(1)$ (PERH - CO): $PER > 0.815 \wedge v0 > 17$
- $R(2)$ (PERH - CO): $N > 3 \wedge PER > 0.765 \wedge v0 > 27$
- $R(3)$ (PERH - CO): $N > 5 \wedge PER > 0.685 \wedge v0 > 28$
- $R(4)$ (PERL - CO): $N > 8 \wedge PER <= 0.365 \wedge v0 > 36$
- $R(5)$ (PERL - CO): $N > 8 \wedge v0 > 42$

These five rules result in the set of terms:

$$FS' = \{N >, PER >, PER <=, v0 >\}.$$

The thresholds in our example are all numerical; thus, we have a set composed of only four numerical values:

$$T = \{V_{N>}, V_{PER>}, V_{PER<=}, V_{v0>}\}.$$

Once created these eight columns, we construct the vectors representing our five rules as sketched in Tab. 3.

Table 3. Representation of the BOW matrix of the reduced complexity example

Rules	$PER >$	$V_{PER>}$	$v0 >$	$V_{v0>}$	$N >$	$V_{N>}$	$PER <=$	$V_{PER<=}$
$R(1)$	1	1	1	0	0	0	0	0
$R(2)$	1	0.6	1	0.4	1	0	0	0
$R(3)$	1	0	1	0.44	1	0.4	0	0
$R(4)$	0	0	1	0.76	1	1	1	0.28
$R(5)$	0	0	1	1	1	1	0	0

After having applied the cosine similarity to each pair of rules in our original rulesets PERL and PERH, we obtain a triangular matrix of size 39 x 39. Each element of this matrix represents the similarity between a given couple of rules and ranges from 0, if the two are entirely different, to 1, when they coincide. If we look at the similarities between the rules of the two different datasets, as we might expect, we find that the ones that have a similarity level above 95% (see those listed below) are only a few. In addition, these rules are very basic and not such informative as shown in Tab. 4.

- $R(6)$ (PERH - CO): $N > 7 \wedge PER > 0.645$
- $R(7)$ (PERH - CO): $N > 8$
- $R(8)$ (PERH - NC): $N <= 8 \wedge PER <= 0.605$
- $R(9)$ (PERL - CO): $N > 9$
- $R(10)$ (PERL - NC): $N <= 8 \wedge PER <= 0.385$
- $R(11)$ (PERL - NC): $N <= 7 \wedge PER <= 0.425$

Table 4. Similarity of rules belonging to different rulesets

	$R(6)$	$R(7)$	$R(8)$	$R(9)$	$R(10)$	$R(11)$
R(6)	1	-	-	-	-	-
R(7)	0.97	1	-	-	-	-
R(8)	0	0	1	-	-	-
R(9)	0.95	0.996	0	1	-	-
R(10)	0	0	0.99	0	1	-
R(11)	0	0	0.99	0	0.99	1

If we focus our attention on a single ruleset, e.g. PERL, whose rules are detailed below, it is immediately apparent that the collision and the no collision rules are, in general, dissimilar. Actually, according to many pairs of these two groups, the similarity is even zero (see Tab. 5).

Table 5. Similarity of CO and NC rules in the PERL ruleset

No Collisions	**Collisions (CO) rules**			
(NC) rules	$R(4)$	$R(5)$	$R(9)$	$R(12)$
$R(10)$	0.33	0	0	0
$R(11)$	0.34	0	0	0
$R(13)$	0	0	0	0.39
$R(14)$	0	0	0	0
$R(15)$	0	0	0	0
$R(16)$	0	0	0	0.36
$R(17)$	0.2	0	0	0.25
$R(18)$	0.53	0.34	0	0

- $R(4)$ (PERL - CO): $N > 8 \wedge PER <= 0.365 \wedge v0 > 36$
- $R(5)$ (PERL - CO): $N > 8 \wedge v0 > 42$
- $R(9)$ (PERL - CO): $N > 9$
- $R(10)$ (PERL - NC): $N <= 8 \wedge PER <= 0.385$
- $R(11)$ (PERL - NC): $N <= 7 \wedge PER <= 0.425$
- $R(12)$ (PERL - CO): $N > 8 \wedge v0 <= 36$
- $R(13)$ (PERL - NC): $N <= 8 \wedge v0 <= 57$
- $R(14)$ (PERL - NC): $N <= 5$
- $R(15)$ (PERL - NC): $N <= 8 \wedge F0 > -5$
- $R(16)$ (PERL - NC): $N <= 9 \wedge v0 <= 32$
- $R(17)$ (PERL - NC): $N <= 9 \wedge F0 > -8 \wedge d0 > 7.485 \wedge PER <= 0.475 \wedge v0 <= 78$
- $R(18)$ (PERL - NC): $N <= 9 \wedge PER <= 0.145 \wedge v0 > 32$

On the contrary, by analysing the similarity between only the collision rules of the PERH ruleset, we can identify several analogies that can help us reduce our significant ruleset. Below are the rules, while the corresponding similarity matrix is sketched in Tab. 6.

- $R(2)$ (PERH - CO): $N > 3 \wedge PER > 0.765 \wedge v0 > 27$
- $R(3)$ (PERH - CO): $N > 5 \wedge PER > 0.685 \wedge v0 > 28$
- $R(6)$ (PERH - CO): $N > 7 \wedge PER > 0.645$
- $R(7)$ (PERH - CO): $N > 8$
- $R(19)$ (PERH - CO): $N > 3 \wedge F0 <= -2 \wedge PER > 0.605 \wedge v0 > 51$
- $R(20)$ (PERH - CO): $N > 3 \wedge d0 > 5.3455 \wedge PER > 0.725 \wedge v0 > 28$
- $R(21)$ (PERH - CO): $N > 3 \wedge F0 <= -2 \wedge$ $PER > 0.575 \wedge v0 > 63$

This syntactic knowledge, together with the covering information, can help the expert to choose between very similar rules (e.g., $R(19)$ and $R(21)$, or $R(6)$ and $R(7)$) and keep the most appropriate, by selecting the appropriate trade-off between knowledge discovery and model accuracy.

Table 6. Similarity of collision rules in PERH ruleset

	$R(2)$	$R(19)$	$R(3)$	$R(20)$	$R(6)$	$R(7)$	$R(21)$
$R(2)$	1	-	-	-	-	-	-
$R(19)$	0.65	1	-	-	-	-	-
$R(3)$	0.96	0.68	1	-	-	-	-
$R(20)$	0.83	0.56	0.81	1	-	-	-
$R(6)$	0.61	0.39	0.71	0.51	1	-	-
$R(7)$	0.47	0.36	0.64	0.41	0.97	1	-
$R(21)$	0.61	0.99	0.65	0.53	0.36	0.34	1

Moreover, the analysis of these similarities may lead the expert to make additional considerations that may result in a growth of knowledge for the specific scenario. For example, $R(2)$ and $R(3)$ have very similar conditions as they all have the same terms and very close thresholds. We can notice, looking at Tab. 6, that rule $R(20)$ is quite similar (with similarity larger than 80%) to these two rules. The main difference is that this rule adds information about the initial distance $d0$. This new information may add a novel link that somehow had not previously emerged and investigated.

The obtained high values of rule similarity may also lead to a ruleset pruning: as an example, by removing rule $R(2)$, which is highly similar to $R(3)$ and $R(20)$, from PERH dataset, the overall classification error increase is below 1%. Hence, rule similarity can be also considered as a way to simplify complex and non-intuitive rulesets.

7 Conclusions and Future Work

The paper deals with the comparison of rule-based machine learning models on the basis of a linguistic approach. The inherent Bag-of-Words matrix organizes the model data structure with feature names, operators and conditions thresholds, while the cosine similarity allows the numerical comparison for each couples of rules, taken from the rulesets at hand. The computational and memory costs for matrix building and the computational cost of all cosine similarity calculations depend on the number of independent words in the rulesets (i.e., the number of feature names plus the operators used in a condition). Two different examples are provided to evaluate the effectiveness of the proposed method for rules analysis for knowledge extraction and rule pruning. The performance evaluation shows how complex interactions between stratifications of the XAI model (age groups in fatigue analysis and performance impact of information loss in smart mobility) may be easily inferred from the proposed method. Future work will deal with management of more complex syntactic of rules, interaction with semantic analysis as well as applications to anomaly detection, Bag-of-Words in computer vision and XAI federated learning.

References

1. L. H. Gilpin, et al., "Explaining Explanations: An Overview of Interpretability of Machine Learning," Proc. of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA'18), pp. 80–89, 2018.
2. A. Adadi, et al., "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," IEEE Access, vol. 6, pp. 52138–52160, 2018.
3. A. B. Arrieta, et al., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," Information Fusion, vol. 58, pp. 82–115, 2020.
4. M. Mongelli, et al., "Performance validation of vehicle platooning through intelligible analytics," IET Cyber-Physical Systems: Theory & Applications, vol. 4, pp. 120–127, 2019.
5. A. Holzinger, "From Machine Learning to Explainable AI," Proc. of the World Symposium on Digital Intelligence for Systems and Machines (DISA'18), pp. 55–66, 2018.
6. R. Guidotti, et al., "A Survey of Methods for Explaining Black Box Models," ACM Comput. Surv. 51, 5, Article 93, 42 pages, Jan. 2019.
7. M. Setzu, et al., "GLocalX - From Local to Global Explanations of Black Box AI Models," Artificial Intelligence, vol. 294, 103457.
8. Y. Ramon, et al., "Metafeatures-based rule-extraction for classifiers on behavioral and textual data," arXiv preprint arXiv:2003.04792, 2020.
9. S. Hirano, et al., "Detection of Differences between Syntactic and Semantic Similarities," Proc. of the International Conference on Rough Sets and Current Trends in Computing (RSCTC'04), pp. 529–538, 2004.
10. I. Vaccari, et al., "A Generative Adversarial Network (GAN) Technique for Internet of Medical Things Data," Sensors, vol. 21, no. 11, art. 3726, 2021.
11. J. M. Mendel, et al., "Critical Thinking About Explainable AI (XAI) for Rule-Based Fuzzy Systems," IEEE Transactions on Fuzzy Systems, vol. 29, no. 12, pp. 3579–3593, 2021.
12. P. N. Tan, et al., "Introduction to data mining," 2nd Edition, Pearson, 2019.
13. D. Cangelosi, et al., "Logic Learning Machine creates explicit and stable rules stratifying neuroblastoma patients," BMC Bioinformatics, vol. 14, suppl. 7, 2013.
14. C. Fuchs, et al., "A graph theory approach to fuzzy rule base simplification," Proc. of the International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems, pp. 387–401, 2020.
15. A. W. Qurashi, et al., "Document processing: Methods for semantic text similarity analysis," Proc. of the 2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA'20), pp. 1–6, Aug. 2020.
16. P. Sethi, et al., "Association rule based similarity measures for the clustering of gene expression data," The Open Medical Informatics Journal, vol. 4, no. 63, 2010.
17. M. Anokhin, M., et al., "Decision-making rule efficiency estimation with applying similarity metrics," ECONTECHMOD: An International Quarterly Journal on Economics of Technology and Modelling Processes, vol. 4, 2015.
18. M. Muselli, et al., "Coupling logical analysis of data and shadow clustering for partially defined positive Boolean function reconstruction," IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 1, pp. 37–50, 2009.
19. A. Gunjan, et al., "A Brief Review of Intelligent Rule Extraction Techniques," Proc. of the International Symposium on Signal and Image Processing, pp. 115–122, Mar. 2020.

20. SKOPE-rules, Github repository, online: https://github.com/scikit-learn-contrib/skope-rules, accessed 2022-03-11.
21. J. H. Friedman, et al., "Predictive learning via rule ensembles," The Annals of Applied Statistics, vol. 2, no. 3, pp. 916–954, 2008.
22. Z. S. Maman, et al., "A data analytic framework for physical fatigue management using wearable sensors," Expert Systems with Applications, vol. 155, 113405, 2020. Github repository, available online: https://github.com/zahrame/FatigueManagement.github.io, accessed 2022-03-11.
23. N. Williams, "The Borg rating of perceived exertion (RPE) scale," Occupational Medicine, Oxford University Press UK, vol. 67, no. 5, pp. 404–405, 2017.
24. S. Narteni, et al., "From Explainable to Reliable Artificial Intelligence," Proc. of the International Cross-Domain Conference for Machine Learning & Knowledge Extraction (MAKE'21), pp. 255–273, Aug. 2021.
25. M. Segata, et al., "Plexe: A Platooning Extension for Veins," Proc. of the 6th IEEE Vehicular Networking Conference (VNC'14), pp. 53–60, Dec. 2014.