City Research Online

City St George's, University of London

Citation: Adilova, L., Kamp, M., Andrienko, G. & Andrienko, N. (2025). Re-interpreting rules interpretability. International Journal of Data Science and Analytics, 20, pp. 25-45. doi: 10.1007/s41060-023-00398-5

This is the published version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/31082/

Link to published version: https://doi.org/10.1007/s41060-023-00398-5

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

International Journal of Data Science and Analytics
https://doi.org/10.1007/s41060-023-00398-5

REGULAR PAPER

Re-interpreting rules interpretability

Linara Adilova[1,2] · **Michael Kamp**[1,3,4] · **Gennady Andrienko**[2,5] · **Natalia Andrienko**[2,5]

Received: 5 April 2022 / Accepted: 24 May 2023
© The Author(s) 2023

Abstract
Trustworthy machine learning requires a high level of interpretability of machine learning models, yet many models are inherently black-boxes. Training interpretable models instead—or using them to mimic the black-box model—seems like a viable solution. In practice, however, these interpretable models are still unintelligible due to their size and complexity. In this paper, we present an approach to explain the logic of large interpretable models that can be represented as sets of logical rules by a simple, and thus intelligible, descriptive model. The coarseness of this descriptive model and its fidelity to the original model can be controlled, so that a user can understand the original model in varying levels of depth. We showcase and discuss this approach on three real-world problems from healthcare, material science, and finance.

Keywords Interpretability · Descriptive model · Global explanation · Generalization

1 Introduction

One of the key challenges for machine learning (ML) models to be adopted in critical applications, such as autonomous driving and healthcare, is that the model must be explainable [1]. The explainability is not only demanded by practitioners, but is in fact required by law in the EU with the European Parliament's General Data Protection Regulation (GDPR) introducing the right to receive explanations of decisions made by AI systems. There are two different types of explanations: (i) local explanations, i.e., a justification for an individual decision, also termed post hoc explanation [1], and (ii) a global explanation of the overall logic and behaviour of a model. The latter one is often a generalization of the former, since from such an understanding of the model individual decisions can be justified as well.

The usual way to have a global explanation is to use a model that inherently allows such an understanding. Typical examples are decision trees, or rule ensembles. Studying the rules, or equivalently the paths in the decision tree, allows a user to understand the logic of the model, as well as to justify individual predictions. If instead an existing model that is not inherently explainable (i.e., a deep neural network) needs a global explanation, then it can be obtained by training an explainable mimic-model [2] that approximates the black-box model's behaviour.

For explainable models to achieve high predictive quality often requires them to be very large in terms of their number of rules. This also holds for mimic models that aspire to achieve high fidelity to the original black-box model. For example, a tree may have hundreds of nodes and tens of levels, and a rule ensemble may consist of hundreds of rules with complex conditions. Therefore, models that are interpretable in principle often remain beyond human perceptual and cognitive capabilities due to their size [3].

The level of understanding of an interpretable model can be enhanced by reducing its size. To achieve this goal, techniques such as special tuning or post-processing have been investigated in research [4]. However, the achievable degree of reduction is significantly constrained by the striving to preserve the prediction accuracy. Even more importantly, when a mimic model is used for explaining, it has a different logic than the original model, and its relationship to the original model may be unclear. Hence, instead of explaining how the original model comes to its predictions, the mimic model demonstrates alternative ways to come to the same or similar predictions. While this is, perhaps, the only viable possibility when the original model is a true black box, it may

✉ Linara Adilova
linara.adilova@ruhr-uni-bochum.de

1 Ruhr University Bochum, 44801 Bochum, Germany

2 Fraunhofer Institute IAIS, 53757 Sankt Augustin, Germany

3 IKIM, University Medicine Essen, 45131 Essen, Germany

4 Monash University, Melbourne, VIC 3800, Australia

5 City, University of London, London EC1V 0HB, UK

Published online: 05 July 2023

be less desirable when the model logic can, in principle, be understood by a human. In the latter case, a preferable approach would be to facilitate the comprehension of the original model logic rather than to substitute it by another logic.

In this paper, we present an approach to facilitating comprehension of an existing model, representable as a set of conjunction rules (e.g., rule ensembles themselves, decision trees, random forests, tree ensembles), that is explainable in principle but not in practice due to its size. Note that we treat decision trees as equivalent to rule sets since they can always be transformed into one [5, 6]. The idea is to extract the general logic from the model by uniting its rules based on their similarity. A union rule not only substitutes multiple original rules, but also typically consists of fewer logical conditions than each of the original rules. Thus, the resulting set of union rules, even with additionally possible exceptions from them, becomes more comprehensible.

Unlike the existing methods that aim at reducing the size of a model while preserving its accuracy on the data, our method creates a new model that *describes* the original model at hand. The original model serves as an input for the algorithm and the output is *a descriptive model of the original model*, having also the form of the set of conjunctive rules. The purpose of this descriptive model is not to make predictions for data instances but to tell how the original model works. Therefore, the descriptive model is not evaluated in terms of the accuracy of its predictions but in terms of its correspondence to the original model. For this purpose, we introduce a novel measure called *Coherence Coefficient* showing how consistent the descriptive rules are with the rules they are intended to describe. This measure allows for a user to regulate the degree of inconsistency of the descriptive model with the model at hand.

Hence, the very idea of our approach is principally different from the ideas behind the existing methods for model simplification that strive to preserve and improve the performance. Our contributions thus are:

- Introduction of an approach that produces a descriptive model of a model that is explainable in principle but too large for comprehension for the purpose of facilitating the understanding of the model logic.
- The achievable degree of simplification is not restricted by the requirement to preserve the prediction quality of the original model, different from the multitude of known approaches for training more compact rule-based models.
- Union rules of the descriptive model can be explored in detail by tracing the hierarchy of more specific rules that were involved in the derivation of the union rules.
- The construction of the descriptive model is fully transparent, and its relationship to the original model is absolutely clear.

The remainder of this paper is organized as follows: We first discuss the relation of the proposed approach to training compact (mimic) models in Sect. 2. We then present our approach in Sect. 3, followed by exploring the algorithm via visualizations on a typical application in Sect. 4. We empirically evaluate the approach in Sect. 5 and conclude by a discussion of the contribution and its limitations, as well as future work in Sect. 6.

2 Related work discussion

In ML, certain types of models, namely decision trees, decision tables, and rules, are considered to be inherently interpretable [2], as they can be represented in a human-readable form. However, the actual comprehensibility of such a model greatly depends on its complexity [7, 8], which is typically roughly estimated in terms of the model size [2]. Therefore, the existing ML algorithms that generate decision trees or rules usually strive to reduce the model size by pruning the tree or compressing the set of rules (e.g., RuleFit [9]) so that the smaller model is still as accurate as the big one. Making models more compact is a vast area of research mainly due to the expected improvements in generalization and stability properties of the obtained solutions. Al-Akhras et al. [10] discuss a popular approach to avoiding overfitting in decision trees in which a more compact tree is produced via reducing the amount of instances used to build the model. The approaches directed to reducing the amount of instances were surveyed by Wilson and Martinez [11]. Pruning of decision trees is another popular way to achieve higher stability [12]. Helmbold and Schapire [13] propose an alternative algorithm that avoids pruning. Compactness of sets of rules is also a matter of concern. Dash et al. [14] propose an algorithm for creating compact whilst sufficiently accurate sets of rules using integer programming. In general, enforcing sparseness (i.e., excluding features from consideration in the rules) of the learned rules is a popular problem addressed by, e.g., Su et al. [15]. Alternative approaches propose a different interpretable class of models that is trained in a way to be sparse [16, 17].

The research on compressing intelligible models is mostly based on the regularization techniques applied while training. Thus, Joly et al. [18] propose to use L1 compression for random forests in order to decrease the prohibitively long computation time for the big forests. Alternatively, Painsky and Rosset [19] propose to encode a random forest in a lossless or lossy with guarantees way, which allows not to store the full models—motivated by the limitations of the storage space. Sometimes an interpretable by design model is even compressed into a black-box model, like a neural network [20], in order to sustain the small storage space and high performance. In general, aforementioned works aim at

achieving more stable, smaller and better generalizing intelligible models while training. A big restriction to the degree of the compactness is always final performance of the model [21].

On the other hand, while the creation of rule-based mimic models is a typical approach to explaining the behaviour of black-box models, such as neural networks [2], the research on improving the thereby obtained explanations is ongoing. It is clear that applying pruning or other compression training techniques when trying to mimic a complex black-box model will lead to a loss in fidelity. To achieve both goals, Qiao et al. [22] recently proposed a novel approach in which a set of decision rules is generated by a neural network with a special two-layer architecture. The authors also proposed a sparsity-based regularization approach to balance between classification accuracy and the simplicity of the derived rules. For now this is a limited approach, that does not allow to work with any black-box model at hand.

Freitas [23] discusses that decision trees and rules have different properties in terms of interpretability and that decision trees are usually perceived better when transformed to rule sets. This is also confirmed by Quinlan [24], who considers multiple approaches to pruning decision trees and finalizes with the transformation to rules as a help for understanding. A random forest model consisting of multiple trees can also be transformed to a set of rules, for example, using a novel approach from Bénard et al. [25], which is close to the RuleFit [9]. Furthermore, it is argued that a representation in the form of rules can be more compact than a decision tree, because rules can include only significant clauses and have no repeated occurrences of the same variable [2]. Another work [26] discusses high redundancy in decision trees and proposes a method for extracting non-redundant rule-like explanations from a decision tree. The arguments about advantages of rules over trees substantiate the focus of our research on sets of rules.

Since our approach involves unions of rules, it is partly related to the works where rules or decision trees are merged for various purposes. Hierarchical merging of several trees was addressed in the context of the problem of learning decision trees from multiple sources of the data—so the challenge is to produce one tree that will cover the decisions of others [27]. Another problem that is addressed is construction of consensus trees from different ones with the goal of producing a more stable model [28]. A framework for combining multiple rule-based models that have been created for different subproblems is proposed by Strecht et al. [29]. Rules from different models are combined by computing their intersections. After resolving conflicts, the resulting rule set is minimized by uniting nearly identical rules. A similar approach to joining rules is taken by Andrzejak et al. [30]. Our approach also involves an operation of rule union, but, unlike others, it allows controlled decrease of rule accuracy for achieving a higher degree of simplification.

Our research involves not only the development of an algorithmic method to obtain a descriptive model, but also the creation of interactive visual techniques for exploring sets of rules and investigating the behaviour and results of the algorithm. Combining computational methods with interactive visual interfaces is at the core of Visual Analytics (VA) [31]. In particular, VA techniques allow human experts to be involved in the creation of ML models [32]. This way, humans can contribute not only their background knowledge, but also new knowledge gained in the process of interactive data analysis [33] through discovery and abstraction of patterns existing in data [34]. Currently, the problem of explaining ML/AI models is receiving much attention in VA [35]; however, the techniques proposed so far address mostly the need of model developers rather than domain experts. As an exception, RuleMatrix [36] visualizes rule sets for users with little machine learning experience, but it does not address the problem of model simplification and is severely limited by the size of the rule set.

In the area of visualization research, a comparative evaluation of four basic techniques for visual representation of rules sets, namely symbolic and graphical encoding of conditions with and without vertical alignment of conditions referring to the same features, has been conducted recently [37]. The experiments showed the superiority of the representations that use feature alignment, which is valid for our table view. Graphical display is advantageous to textual representation, although the effect is less pronounced compared to that of feature alignment. However, the experiments were conducted using small sets of rules, whereas effective visualization of large models is still a challenging task.

So, the current research, on the one hand, acknowledges the problem of comprehending large rule sets or decision trees, on the other hand, does not consider the possibility of creating approximate simpler descriptive models instead of directly training more compact ones. Note that creating a descriptive model that helps to interpret the original model is fundamentally different from training a more compact model with similar accuracy. The descriptive model seeks to explain a given model at hand, while training a different, more compact model seeks to replace it and makes it much harder to connect functionality of the initial black-box with the interpretable and compact mimic model.

The techniques proposed in this paper differ from existing methods in their goals. Instead of replacing a model with a more compact one, we aim to create concise descriptions. Therefore, comparing our approach to others in terms of accuracy or compression would be inappropriate. In the following, we proceed with describing our approach and evaluate it on various tasks without explicit comparison to other techniques.

3 Rule set simplification

3.1 Main concepts

In the following, we define the terms that we will be using throughout the paper. We assume that one wants to interpret a predictive model $h : \mathcal{X} \rightarrow \mathcal{Y}$ at hand with input and output spaces $\mathcal{X}$, resp. $\mathcal{Y}$ that can be rewritten as a collection of rules, i.e., $h = \mathcal{R}$, where each rule $R \in \mathcal{R}$ consists of an antecedent which is a *conjunction of conditions* and consequent which is a *prediction r of the rule*. The input space $\mathcal{X}$ consists of instances with d features f_i, $i = [d]$, which can be numerical or categorical. The output space $\mathcal{Y}$ can be either categorical for classification or numerical for regression tasks.

A *condition c* is a logical expression of the form $f_i \in V$, where V can be a set of values (for a categorical f_i) or an interval (for a numeric f_i) that is restricting the values that f_i can get. Such c can be a splitting condition from a decision tree node or a part of a conjunctive logical rule.

For the following definitions, we will use a running example of a rule set defining whether one goes for a walk or not:

- R_1^W "If precipitation $\in$ {rain, snow} then walk = false"
- R_2^W "If precipitation = rain and wind speed $\geq$ 20 km/h then walk = false"
- R_1^{WT} "If precipitation = none then walk time = 40 minutes"
- R_2^{WT} "If precipitation =mist then walk time = 30 minutes"
- R_3^{WT} "If wind speed $\leq$ 5 km/h then walk time $\in$ [35, 40] minutes"
- R_4^{WT} "If wind speed $\in$ [5, 10] km/h then walk time $\in$ [30, 35] minutes"
- R_5^{WT} "If wind speed $\leq$ 10 km/h then walk time $\in$ [30, 40] minutes"
- R_6^{WT} "If wind speed $\in$ [8, 10] km/h then walk time $\in$ [25, 30] minutes"

Definition 1 A condition $c_1 = (f_i \in V_1)$ **subsumes** another condition $c_2 = (f_j \in V_2)$ iff $i = j$ and $V_2 \subseteq V_1$. A rule R_1 **subsumes** or **covers** rule R_2 $R_1 \supseteq R_2$, if every condition of rule R_1 subsumes some condition of rule R_2.

By definition, any rule covers itself. When $R_1 \supseteq R_2$ and $R_1 \neq R_2$, then R_1 is more general and R_2 is more specific.

Rule R_1^W covers rule R_2^W because every condition of R_1^W covers some condition of R_2^W (namely, the first condition).

R_2 may include conditions involving features that do not appear in conditions of R_1, i.e., R_1 may have fewer conditions than R_2.

For each rule $R \in \mathcal{R}$, where $\mathcal{R}$ is the set of all rules in the model, we can identify set of rules $\mathcal{R}^{\supseteq}$ that are covered by it. When the set of rules $\mathcal{R}$ is optimal in the sense of our approach, such sets are trivial $\mathcal{R}^{\supseteq} = \{R\}$.

Definition 2 Predictions of two rules R_1 and R_2, denoted by r_1 and r_2, are **congruent** $r_1 \cong r_2$ if one of the following conditions holds:

- $r_1 = r_2$;
- $|r_1 - r_2| \leq \epsilon$ when r_1 and r_2 are numbers;
- $\max(r_1^{\text{up}}, r_2^{\text{up}}) - \min(r_1^{\text{low}}, r_2^{\text{low}}) \leq \epsilon$ when r_1 and r_2 are numeric intervals, $r_1 = [r_1^{low}, r_1^{up}]$, $r_2 = [r_2^{low}, r_2^{up}]$,

where ϵ is a **tolerance threshold** used during the run of the algorithm.

The rules R_1^W and R_2^W are congruent because they have equal predictions walk = false. Rule with numeric predictions R_1^{WT} and R_2^{WT} are congruent when $\epsilon = 10$ min. Rule R_3^{WT} and R_4^{WT} are also congruent given the same ϵ.

Note that the case of interval predictions is needed for the work of the algorithm with union rules (defined in the following).

Definition 3 We say that $R_1 \supseteq R_2$ correctly, if $r_1 \cong r_2$. In this case, the coverage of R_2 by R_1 is **correct**; otherwise, the coverage is **wrong**. If $R_1 \supseteq R_2$ wrongly, then R_2 is an **exception** of the covering rule R_1.

R_1^W correctly covers R_2^W. Rule R_5^{WT} correctly covers both rules R_3^{WT} and R_4^{WT}, whereas rule R_6^{WT} is wrongly covered by R_5^{WT}, as well as by R_4^{WT}; hence, R_6^{WT} is an exception of both R_5^{WT} and R_4^{WT}.

Definition 4 The **coherence coefficient (CC)** of a rule is the ratio of the number of correctly covered rules to the total number of covered rules:

$$\text{CC}(R) = \frac{|\mathcal{R}^{\supseteq correct}|}{|\mathcal{R}^{\supseteq}|}$$

Definition 5 A rule whose CC < 1, i.e., a rule having at least one exception, is called a **rough rule**.

In our example rule base, rule R_5^{WT} covers three rules (including itself) correctly and one rule wrongly; so, this is a rough rule with CC = 3/4 = 0.75. Rule R_4^{WT} covering one rule wrongly and one (itself) correctly has CC = 1/2 = 0.5.

Definition 6 A **roughness threshold** $\rho \in [0, 1]$ defines the minimal acceptable value of CC of a rule included in a descriptive model during the run of the algorithm.

So, specifying $\rho = 1$ means that no rough rules are allowed, and the smaller ρ gets, the more exceptions rough rules are allowed to have.

For a better understanding of the concept of rule coverage, imagine the multi-dimensional space of the features (assuming, for simplicity, that all features are numeric). Conditions of a rule antecedent define a multi-dimensional shape (namely, a rectangular hyper-parallelepiped) in this space. When some feature f_i is not used in a rule explicitly, it can be treated as being involved in an implicit condition $f_i \in V$ where V is the whole range of possible feature values. A rule R_1 covers rule R_2 (Definition 1) when the shape p_1 defined by R_1 includes the shape p_2 defined by R_2. Please note that *any* rectangular parallelepiped p in this space corresponds to some conjunction of conditions, even if there is no rule with such an antecedent. For two or more shapes, it is possible to create a rectangular parallelepiped that encloses all these shapes. The smallest parallelepiped $p^{\cup}$ enclosing the shapes p_1 and p_2 defined by the conditions of rules R_1 and R_2 represents the *union* of the antecedents of R_1 and R_2.

When we apply the union operation also to the predictions r_1 and r_2 of the rules R_1 and R_2, we obtain a new rule $R^{\cup}$, which is the *union of the rules* R_1 and R_2. The rule $R^{\cup}$ is meaningful only when the predictions r_1 and r_2 are congruent (Definition 2), so our algorithm makes unions only from rules with congruent predictions.

$R_5^{WT} = R_3^{WT} \cup R_4^{WT}$

Accidentally, $p^{\cup}$, apart from p_1 and p_2, may also include parallelepipeds corresponding to antecedents of some other rules; hence, a union $R^{\cup}$ of two rules R_1 and R_2 may additionally cover other rules. Some of those other rules may have predictions incongruent to the prediction of $R^{\cup}$. In such a case, $R^{\cup}$ is a rough rule (Definition 4), and the rules with incongruent predictions are its exceptions (Definition 3).

An example is rule R_5^{WT}: being the union of rules R_3^{WT} and R_4^{WT}, it wrongly covers rule R_6^{WT}, which makes R_5^{WT} a rough rule.

Let us now define the union of two rules more formally.

Definition 7 A **union of two conditions** $c_1 = (f_i \in V_1)$ and $c_2 = (f_i \in V_2)$ involving the same feature f_i is the condition $c^{\cup} = (f_i \in V^{\cup})$, where

- $V^{\cup} = (V_1 \cup V_2)$ if V_1 and V_2 are sets of discrete values;
- $V^{\cup} = [min(v_1^{low}, v_2^{low}), max(v_1^{up}, v_2^{up})]$ if $V_1 = [v_1^{low}, v_1^{up}]$ and $V_2 = [v_2^{low}, v_2^{up}]$ are intervals.

The union of conditions $precipitation = none$ and $precipitation = mist$ is condition $precipitation \in \{none, mist\}$. The union of conditions $windspeed \leq 5$ km/h and $windspeed \in [8, 10]$ km/h is $windspeed \leq 10$ km/h.

Definition 8 A **union of two predictions** r_1 and r_2, denoted $r^{\cup}$, is defined as

- $r^{\cup} = r_1 = r_2$ when $r_1 = r_2$,
- $r^{\cup} = r_1 \cup r_2$ when r_1 and r_2 are distinct sets of discrete values,
- $r^{\cup} = [min(r_1^{low}, r_2^{low}), max(r_1^{up}, r_2^{up})]$ when r_1 and r_2 are numeric intervals, $r_1 = [r_1^{low}, r_1^{up}]$, $r_2 = [r_2^{low}, r_2^{up}]$.

Definition 9 A **union of two rules** R_1 and R_2 with *congruent predictions* r_1 and r_2 is a rule $R^{\cup}$ where each condition is a union of conditions from R_1 and R_2 according to Definition 7, and the prediction $r^{\cup}$ is the union of r_1 and r_2 according to Definition 8. Since union is defined for congruent rules, it follows that $R^{\cup} \supseteq R_1$ and $R^{\cup} \supseteq R_2$ correctly.

In the union rule $R_5^{\mathrm{WT}} = R_3^{\mathrm{WT}} \cup R_4^{\mathrm{WT}}$, the condition wind speed $\leq$ 10 km/h is the union of the conditions $windspeed \leq 5$ km/h from R_3^{WT} and $windspeed \in [5, 10]$ km/h from R_4^{WT}, and the prediction walk time $\in [30, 40]$ min is the union of the respective predictions $walktime \in [35, 40]$ and walk time $\in [30, 35]$. The union of rule R_2^W and rule R_3^W is rule R_1^W, where the condition precipitation $\in \{rain, snow\}$ is the union of precipitation = rain from R_2^W and precipitation = snow from R_3^W, and the prediction coincides with the predictions of R_2^W and R_3^W. Please note that the union rule R_1^W does not include any condition involving feature wind speed because R_3^W has no condition for wind speed.

As mentioned earlier, an absence of a condition for some feature in a rule means that the feature may take any value from its full range of values.

3.2 Distance function

In order to perform the hierarchical merging of rules, we define a distance function on the space of rule antecedents. We set the distance between two rule antecedents to be the sum of the distances between the value intervals V of the same feature f_i in the conditions of the rules. So if $c_1 = f_i \in [v_1^{low}, v_1^{up}]$ and $c_2 = f_i \in [v_2^{low}, v_2^{up}]$, then distance between c_1 and c_2 is

$$d_{f_i} = \frac{|v_1^{\mathrm{low}} - v_2^{\mathrm{low}}| + |v_1^{\mathrm{up}} - v_2^{\mathrm{up}}|}{2\,(v_{\max} - v_{\min})}$$

where $v_{\max}$ and $v_{\min}$ are the absolute maximal and minimal, respectively, values of the feature f_i that may occur in practice. This distance metric is, in fact, a specific formulation of the Hausdorff distance [38] for numeric intervals. The division by $(v_{\max} - v_{\min})$ is done for normalization of all distances

between conditions to the interval [0, 1]. For instance, the distance between intervals [0, 5] and [5, 10] of feature with the value range [0, 200] is $(|0-5|+|5-10|)/2(200-0) = (5+5)/400 = 0.025$, whereas the distance between [5, 10] and [8, 10] is $(|5-8|+|10-10|)/400 = (3+0)/400 = 0.0075$.

For categorical features, rule conditions contain discrete sets of categorical values instead of numeric intervals. In this case, the distance between two conditions can be defined as the Jaccard similarity index [39] subtracted from 1, i.e., if $c_1 = f_i \in A$ and $c_1 = f_i \in B$, where A and B are sets, then $d_{f_i} = 1 - |A \cap B|/|A \cup B|$. The distance equals 0 when A and B are identical and 1 when the sets have no common elements. Thus, the distance between the sets $\{rain, snow\}$ and $\{rain, mist\}$ is $1 - 1/3 = 0.667$, where 1 is the size of the set intersection $\{rain\}$ and 3 is the size of the set union $\{rain, snow, mist\}$.

Based on the distances between corresponding conditions, the distance between the rules R_1 and R_2 is $\sum_{f_i} d_{f_i}$, where $f_i \in$ {features used in R_1 and R_2}. It corresponds to the definition of the Manhattan distance. The interval endpoints are normalized to values between 0 and 1: When some feature is absent in the conditions of one of the rules, it is assumed to have an interval from 0 to 1. Note that since we are not aiming at creating a new compact model that will be used on novel data, this assumption makes sense.

Note that the distance metric is defined solely for the rule antecedents and does not take into account the rule predictions. Since merging is applied only to the rules with congruent predictions, there is no need to include the predictions in the calculation of the rule similarity. Besides, the distance metric defined in this way can be used for detection of similar rules with incongruent predictions, which may be useful in examining the quality of a rule set.

3.3 Basic algorithm for rule set generalization

Input

- A classification or regression model in the form of a set of rules or a decision tree. In the latter case, the tree is transformed to an equivalent set of rules by one of existing methods (e.g., [6]).
- A roughness threshold ρ (Definition 6).
- Optional: For a regression model, a tolerance threshold ϵ (Definition 2).

Output

- A set of rules such that:
  1. Each original rule is correctly covered by some resulting rule (Definitions 1, 3);
  2. The resulting set of rules has smaller cardinality than the original one. In case when the resulting set of rules has the same cardinality as the original one, we say that the algorithm failed;
  3. The coherence coefficient (Definition 4) of any union rule in the resulting set is not less than the roughness threshold, i.e., $CC \geq \rho$ (Definition 6).

The pseudocode of the rule set generalization algorithm is given below (Algorithm 1).

The algorithm repeatedly finds the closest (according to the defined distance metric) pair of rules whose predictions are congruent by Definition 2 and applies the operation of rule union (Definition 9). If the united rule has $CC \geq \rho$ (Definitions 4, 6), it substitutes the two rules it was produced from; otherwise, it is discarded. After accepting a new rule, the algorithm searches for the other rules that are correctly covered by this rule (Definitions 1–3) and, if found, removes them from the resulting set. The algorithm terminates when no new union rule was accepted during an iteration.

3.4 Checking fidelity in terms of data predictions

Since a union rule is more general than the rules it has been derived from, it may be applicable to additional data instances not described by the original rules. For some of these additional instances, the prediction of the union rule may be incongruent with the predictions of corresponding rules from the original model. If we consider some reference dataset, that we have at hand (not necessarily the training dataset used for the original black-box model), we can define the following notion of fidelity with respect to the original rule set.

Definition 10 The **fidelity of a union rule** is the ratio of the number of data instances in some reference dataset for which the union rule gives predictions congruent to the predictions of the original model to the total number of data instances this rule is applicable to.

A reference dataset can be, for example, the set from which the model was derived. In the following, we define the overall fidelity of a descriptive model, i.e., a generalized rule set, with respect to the original model.

Definition 11 The **fidelity of a descriptive model** is the ratio of the number of data instances in the reference dataset for which the descriptive model gives predictions congruent to the predictions of the original model to the total number of data instances both models are applicable to.

When some set of data instances described by the original rule set is available, the fidelity of the derived union rules to the original predictions can be additionally checked. A reasonable requirement is that the fidelity must not be less than

Algorithm 1 generalization of the set of rules

Input $\mathcal{R}$
Output $\mathcal{R}^{\mathcal{G}}$

1: $\mathcal{R}^{\mathcal{G}} \leftarrow \mathcal{R}$
2: changed $\leftarrow True$
3: **while** changed **do**
4: $PD \leftarrow \varnothing$ ▷ find distances between all congruent rules in the set
5: **for each** $(R_i, R_j) : R_i \in \mathcal{R}^{\mathcal{G}}, R_j \in \mathcal{R}^{\mathcal{G}}, i \neq j$ **do** ▷ apply Definition 2
6: **if** $congruent(r_i, r_j)$ **then**
7: $d_{ij} \leftarrow distance(R_i, R_j)$
8: $PD \leftarrow PD \cup \{(R_i, R_j, d_{ij})\}$
9: **end if**
10: **end for**
11: changed $\leftarrow False$
12: **while** $PD \neq \varnothing \wedge \neg$changed **do**
13: $(i, j) \leftarrow argmin_{d_{i,j}} PD$ ▷ find the minimal distance pair
14: $PD \leftarrow PD \setminus \{(R_i, R_j, d_{ij})\}$
15: $R^{\cup} \leftarrow R_i \cup R_j$ ▷ unite the closest rules according to Definition 9
16: **if** $CC(R^{\cup}) \geq \rho$ **then** ▷ check if the union is acceptable according to Definition 6
17: $\mathcal{R}^{\mathcal{G}} \leftarrow \mathcal{R}^{\mathcal{G}} \setminus \{R_i, R_j\}$
18: **for each** $R_k \in \mathcal{R}^{\mathcal{G}}$ **do** ▷ remove all correctly covered rules (Definitions 1–3)
19: **if** $congruent(r_k, r^{\cup}) \wedge R^{\cup} \supseteq R_k$ **then**
20: $\mathcal{R}^{\mathcal{G}} \leftarrow \mathcal{R}^{\mathcal{G}} \setminus \{R_k\}$
21: **end if**
22: **end for**
23: $\mathcal{R}^{\mathcal{G}} \leftarrow \mathcal{R}^{\mathcal{G}} \cup \{R^{\cup}\}$
24: changed $\leftarrow True$
25: **end if**
26: **end while**
27: **end while**

ρ. A condition for checking the fidelity should be added in the "if" statement on line 16 of Algorithm 1, i.e., the extended condition is $\mathrm{CC}(R^{\cup}) \geq \rho \wedge \mathrm{fidelity}(R^{\cup}) \geq \rho$.

3.5 Iterative lowering of the roughness threshold

There is a possibility to apply Algorithm 1 in an iterative manner. For this purpose, the user specifies an interval $[\rho^{\mathrm{low}}, \rho^{\mathrm{up}}]$ and a step $\Delta(\rho)$, where $\Delta(\rho) < \rho^{\mathrm{up}} - \rho^{\mathrm{low}}$. Algorithm 1 is executed several times with consecutively setting the roughness threshold ρ to ρ^{up}, $\rho^{\mathrm{up}} - \Delta(\rho)$,..., ρ^{low}, i.e., starting from ρ^{up} and decreasing the threshold in each following run by $\Delta(\rho)$. The output of run i is used as the input of run $i + 1$.

This extension of the method prioritizes more coherent rules, i.e., it will strive to produce united rules with higher CC before attempting to achieve higher compression at the cost of reducing the coherence.

To demonstrate possible differences between the results of the basic algorithm and its multi-step variant, Fig. 1 shows two projection displays where rules are represented by dots. The dots are arranged on a plane based on the distances between the rules. The projections have been obtained using the method t-SNE [40]. The dot colours encode the predictions, and the sizes are proportional to the number of the data instances the rule applies to. The lines connect dots representing rules that were united by the generalization algorithm. The display on the left corresponds to the base algorithm and the one on the right to the multi-step variant. The dots marked in black represent a group of original rules that were united in a single rule by the multi-step variant and included in three different unions by the base variant.

The illustrations refer to an example classification model consisting of 109 rules including in total 818 conditions. With the roughness threshold of 0.6, the base variant reduces the original set to 54 rules with 342 conditions. Out of which 33 rules are the same as in the original set (i.e., the algorithm cannot generalize them) and 21 rules are unions obtained from 76 original rules. The coherence coefficient of the union rules ranges from 0.6 to 1. However, only one union has CC = 1, three rules have CC from 0.71 to 0.78, and the remaining 17 rules have CC $\leq$ 0.67.

The multi-step variant that iteratively lowers the roughness threshold from 1 to 0.6 in steps of 0.05 reduces the original set to 62 rules with 399 conditions. 43 rules remain the same as in the original set and 19 rules are unions. The coherence coefficient of the union rules ranges from 0.67 to 1. 5 rules have CC = 1, 6 rules have CC from 0.7 to 0.8, and the remaining 8 rules have CC = 0.67. Hence, the multi-step variant produces more accurate union rules but achieves a lower degree of generalization and compression than the base algorithm.

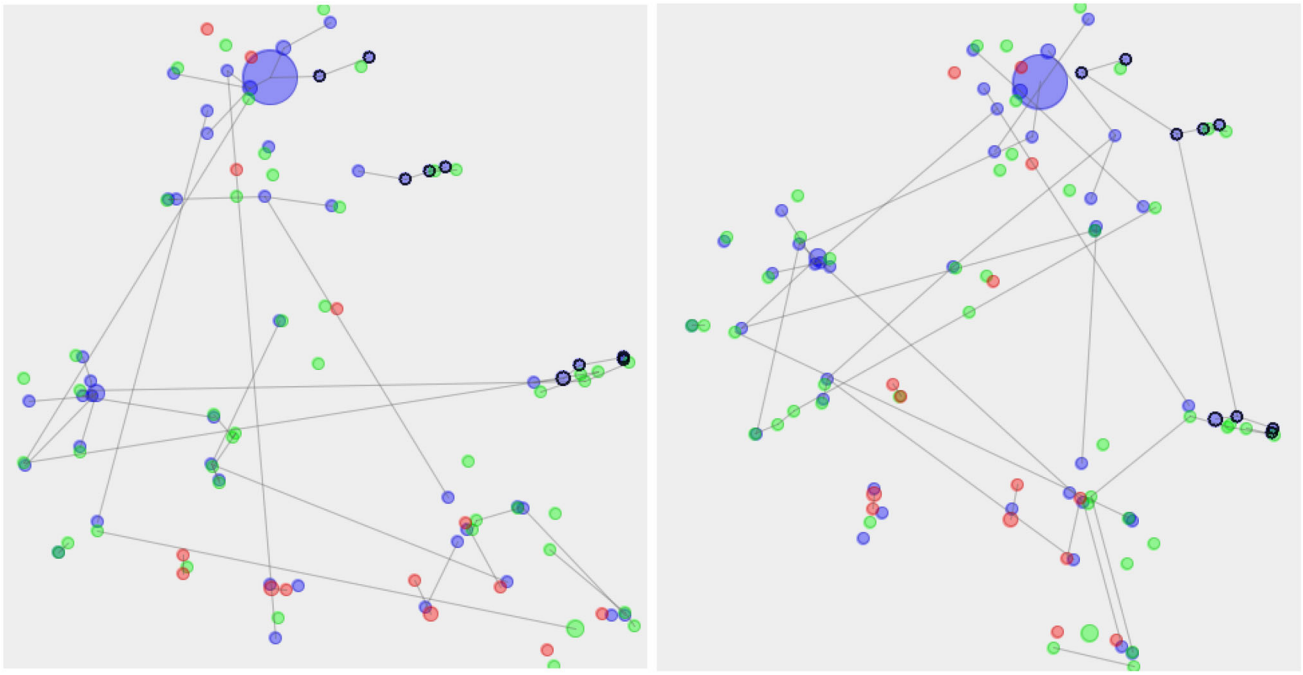

Fig. 1 Two displays demonstrate differences between results of the base algorithm (left) and the multi-step variant (right). The lines connect dots corresponding to rules that were united by the respective variants of the algorithm. The dots marked in black in both displays represent the same selected group of original rules that were united into a single rule by the multi-step variant and included in three different unions by the base algorithm (colour figure online)

4 Visualizations

We have designed and implemented several visualizations[1] that enable researchers to explore rule sets and gain insights into the functionality of our algorithm. It is important to note that these visualizations are not an integral part of the rule generalization method. Instead, they serve as supportive tools to demonstrate how the method works, rather than being the central focus of the paper. Furthermore, our software implementation is a proof of concept, demonstrating the feasibility and potential of our approach. In its current state, it is limited to handling rules with numeric feature conditions; however, it is important to emphasize that this limitation is specific to our implementation and does not restrict the broader applicability of the approach itself.

We display a set of rules in the form of a table, as shown in Fig. 2. Each table row corresponds to one rule; each table column corresponds to one feature. Value intervals of the conditions are represented by horizontal bars, which show the relative position of the interval between the minimal and maximal feature values. If a feature is not used in a rule, the corresponding cell is empty. Besides, there in a column entitled "Rule", where each rule is represented as a whole by a glyph with vertical axes corresponding to all available features and vertical bars corresponding to the features used in the rule.

A table showing the results of the rule generalization algorithm (Fig. 3) includes additional columns containing (1) counts of correct and wrong applications of the rule to data instances, (2) counts of correctly and wrongly covered original rules, (3) fidelity, (4) coherence coefficient, (5) number of rules in the derivation hierarchy, and (6) depth of the hierarchy.

Detailed information about a rule is provided in a popup window (Fig. 4) appearing when the user points on a table row. The window shows all conditions of the rule as well as the minimal and maximal values of all features involved in the rule. Besides, it includes a graphical representation of the rule in a form of a glyph. A glyph representing a rule includes as many vertical axes as there are features used in the whole set of rules (as well as in the "Rule" column of the table). For each feature that is used in the given rule, there is a vertical bar drawn on top of the corresponding axis. The bar represents the interval of the feature values that is specified in the rule condition involving this feature. While the whole length of the axis represents the full range of the feature values (i.e., from the minimal to the maximal values that occur in the

[1] While visualizations like these can be used for explaining models to users, there is plenty of room for improvement. User-centred design and user evaluation are necessary for ensuring that visualizations are effective, well understood, and easy to use. This kind of work is beyond the scope of this paper.

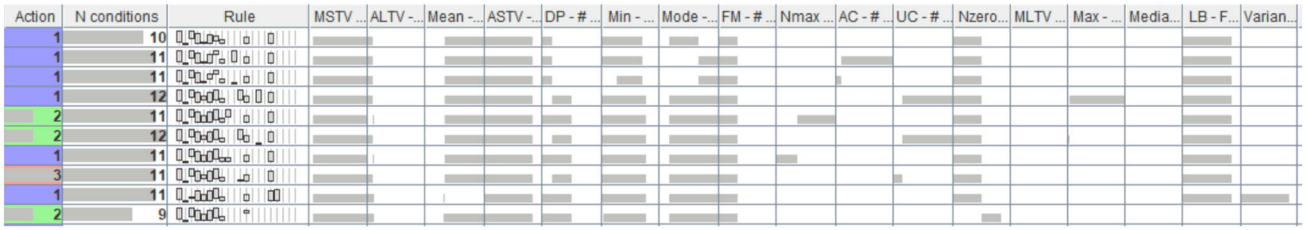

Fig. 2 A fragment of a table representing rules in a graphical form. The rows are ordered by rule similarity using the algorithm OPTICS [41] applying the distance function introduced in Sect. 3

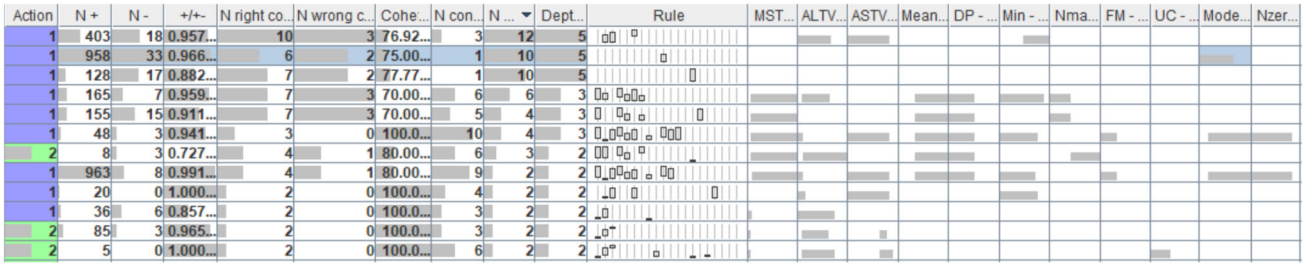

Fig. 3 A fragment of a table representing results of rule generalization

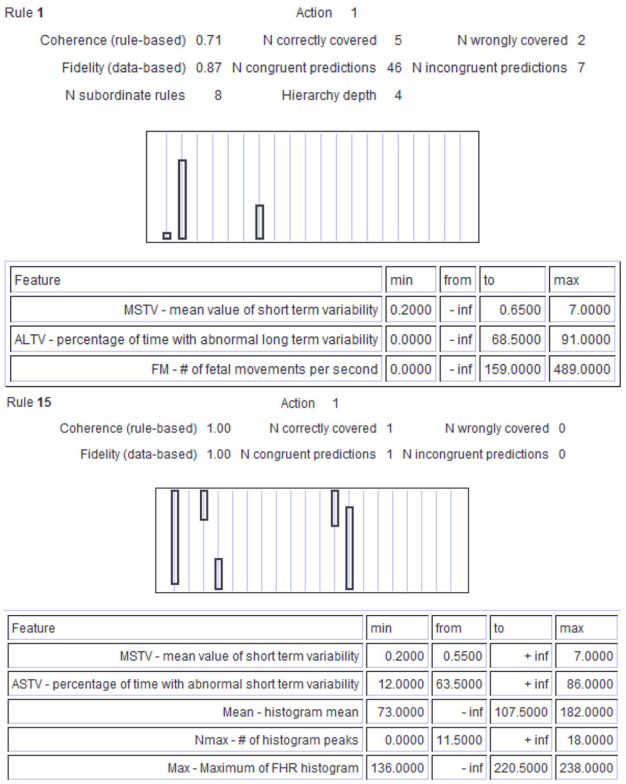

Fig. 4 Examples of popup windows with detailed information about individual rules

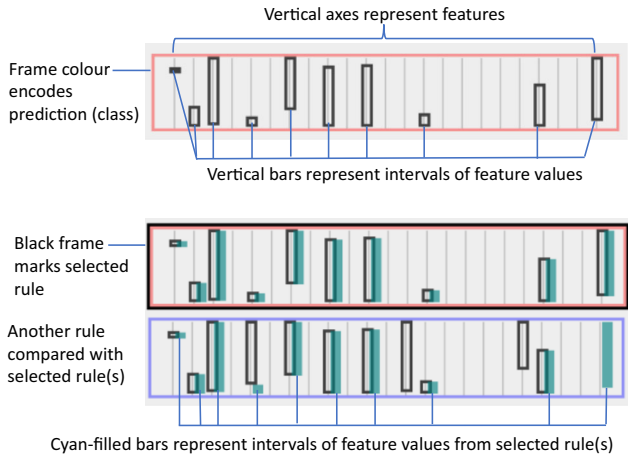

Fig. 5 Explanation of the visual representation of rules by glyphs (colour figure online)

dataset), the vertical positions of the lower and upper sides of the bar correspond to the lower and upper ends of the value interval specified in the rule condition. The representation is explained in the upper part of Fig. 5.

Another type of display we use is a panel with multiple rules represented by glyphs, as shown in Fig. 6. A popup window with rule details, as in Fig. 4, can be obtained by pointing on a glyph. Comparison of the rules is supported by a mode of glyph drawing in which conditions of one rule are represented by hollow bars with black frames and conditions of one or more previously selected rules are represented by cyan-filled semi-transparent bars without frames, as explained in the lower part of Fig. 5. The bar shading is darker where feature value intervals from two or more selected rules overlap. For example, in Fig. 6, the conditions of three selected rules (their glyphs have black frames around them) are represented within all glyphs in the display. Rules are selected by clicking on their glyphs or rows in a table view.

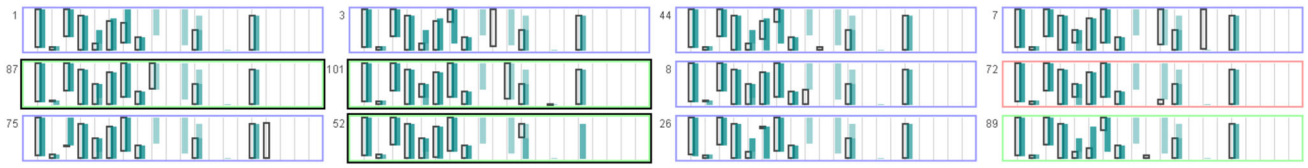

Fig. 6 A display with multiple rules represented by glyphs. The numeric labels are rule identifiers. The colours of the glyph frames encode the predictions made by the rules. Three selected rules are marked by additional black frames. The conditions from the selected rules are represented in all glyphs by semi-transparent vertical bars shaded in cyan (colour figure online)

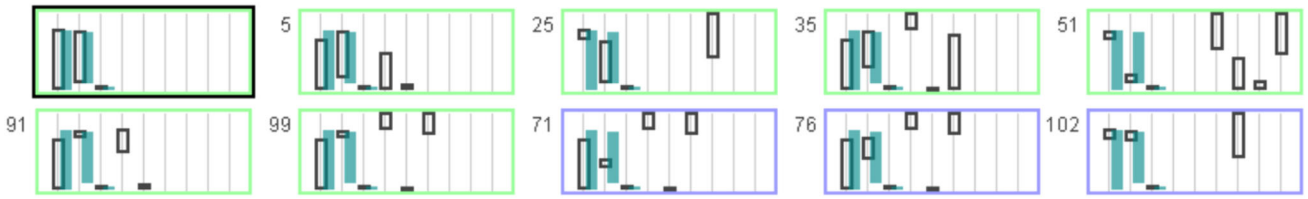

Fig. 7 An illustration of the concept of rule coverage. The colours of the glyph frames represent the predicted classes of the rules. The rule shown on the top left covers the remaining 9 rules, of which 6 are covered correctly (they predict the same class as the first rule, indicated by the green colour of the frames) and 3 wrongly (they predict a different class, indicated by the blue colour of the frames). The first rule is selected for comparison with the others; its conditions are represented by cyan-filled bars in all glyphs (colour figure online)

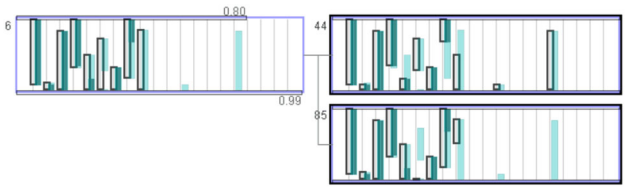

Fig. 8 An illustration of rule union. The glyphs on the right represent two original rules and the glyph on the left their union. The two original rules are selected for comparison, as indicated by the black frames of their glyphs. The conditions from these rules are represented in all three glyphs by bars shaded in cyan (colour figure online)

Figure 7 illustrates the concept of rule coverage. Here, the rule shown on the top left (it is selected, so that the glyph is marked with a black frame) covers the remaining rules represented in the image. The conditions of the covering rule are represented in all glyphs by cyan-shaded bars. The colours of the glyph frames encode the predicted classes of the rules. Five of the nine covered rules have the same prediction as the covering rule and three rules have a different one. Hence, five rules are covered correctly and three rules incorrectly.

Figure 8 illustrates the operation of rule union. Two original rules are shown on the right and their union on the left. The original rules are selected, and their conditions are represented by cyan-shaded bars in all three glyphs. Darker bar shading signifies overlapping conditions. The first four conditions and the seventh condition are identical in the two original rules; so, the same conditions are included in the union rule. In the fifth and eighths conditions, the value interval of one rule includes the value interval of the other rule; so, the union rule includes the larger intervals. In the sixth and ninth conditions, the value intervals do not overlap; so, the union contains the interval from the lower end of the lower interval to the upper end of the higher interval. There are two conditions with features appearing only in one of the original rules. For these features, the union has no conditions.

The numbers 0.80 and 0.99 above and below the glyph of the union rule represent the coherence coefficient and the fidelity of the union rule, respectively. In this example, the union rule covers four original rules correctly and one original rule incorrectly; so, it is a rough rule with CC = 4/(4+1) = 0.8. This union rule gives the same predictions as the original model for 963 data instances and different predictions for 8 data instances; hence, its fidelity is $963/(963+8) = 0.99$.

Figure 9 illustrates the work of the algorithm by example of deriving one generalized rule. Original and derived rules are represented by glyphs. The lines represent inclusions of rules into more general rules covering them. In one of the iteration steps, the algorithm unites original rules labelled 25 and 51 (on the upper right of the image) into a union rule shown in the centre of the upper row of glyphs. In another step, the algorithm unites original rules 91 and 99, which are shown in the lower right corner of the display. In one of the following steps, the algorithm unites the earlier produced union of the rules 25 and 51 (top middle) with an original rule labelled 5 (its glyph is in the middle of the figure). The resulting union rule is shown on the top left. The algorithm finds out that this rule also covers two other rules, an original rule labelled 35 and the earlier obtained union of the rules 91 and 99. The glyphs of these rules are drawn in the central

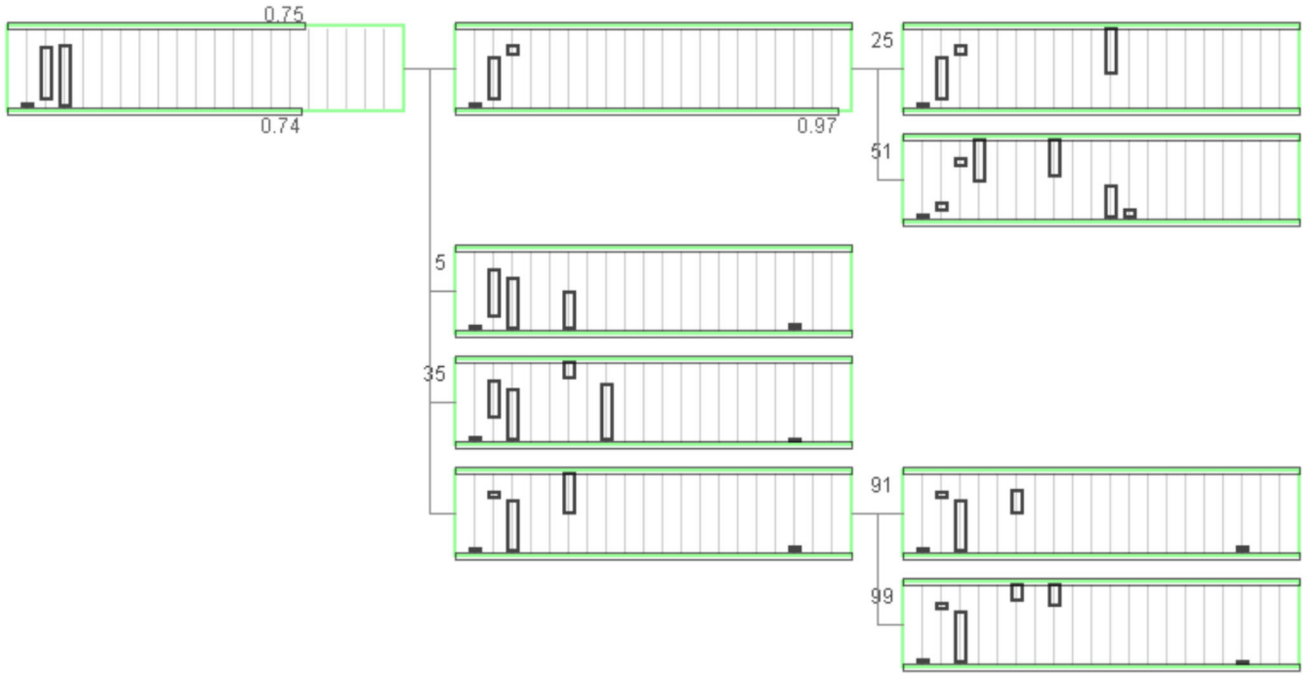

Fig. 9 A visual display of the derivation hierarchy of a rule. Only coherence and fidelity coefficients that are lower than 1 are represented by text labels. The absence of a text label indicates that the coefficient value is 1

part of the display below the glyphs of the two rules that have been united. Hence, the final union rule generalizes 6 original rules. In deriving it, two intermediate unions have been made. The coherence coefficient of the final rule is 0.75. The rules correctly and wrongly covered by this rule are shown in Fig. 7.

5 Experiments

We describe our investigation of 4 models from three real-world tasks. For each rule set, we ran the basic rule generalization algorithm 9 times setting the parameter ρ to 1.00, 0.95, 0.90, 0.85, ..., 0.60. Each run was applied to the original rule set. For each run we are analysing the statistics that describe the comprehensibility of the compressed model (the number of the resulting rules, the total number of conditions in all the rules, the mean number of conditions per rule, the number and percentage of the rules including more than 5 conditions, considered as complex), the roughness of the descriptive model (the minimal CC that was actually achieved, the minimal fidelity of a rule, the total fidelity of the whole rule set, and the number of rough rules), and the characteristics of the algorithm work (the number of generated union rules and the maximal depth of a rule derivation hierarchy). While the first group of the results allows to access the interpretability, the second and third ones give a deeper understanding of the mechanics that allows to achieve such compressed descriptive model. It is important to note once again, that our aim is to interpret the global logic of a model at hand, not to understand the data. So we achieve our goal if we can explain the main rules learned by the model, the main features that affect its decisions, the possible outliers that require highly specific rules, etc.

5.1 Cardiocartography dataset

Since medical domain is of high interest for interpretability opportunities, as a primal experiment we looked at a medical dataset. It is an UCI [42] dataset of cardiocartography records [43]. It contains 2126 foetal cardiotocograms for which various diagnostic features were measured. They were classified with respect to a morphologic pattern into 10 classes and to a foetal state into 3 classes. For both cases, we directly learned a decision tree and analysed them using our algorithm.

Three classes task

The 3 classes model consists of 109 rules describing 1700 data instances of the training dataset. The statistical characteristics of the generalized rule sets obtained for different settings of the parameter ρ are presented in Fig. 10.

It can be noticed that decreasing the roughness threshold ρ from 1 to 0.85 does not lead to generation of any rough rule, i.e., none of the resulting rules has exceptions. However, the union rules, even when their CC = 1, are more general than the original rules and applicable to larger subsets of the

rho	N rules	Total N conditions	Mean N conditions	N rules >5 conditions	% rules >5 conditions	min *CC*	min fidelity	Total fidelity	N rough rules	N union rules	Max depth
	109	818	7.50	75	68.81	1.00	1.00	1.00	0	0	1
1.00	103	762	7.40	68	66.02	1.00	1.00	1.00	0	6	2
0.95	98	708	7.22	63	64.29	1.00	0.95	0.99	0	10	2
0.90	95	678	7.14	61	64.21	1.00	0.91	0.98	0	10	3
0.85	94	678	7.21	62	65.96	1.00	0.85	0.98	0	10	3
0.80	87	609	7.00	54	62.07	0.80	0.81	0.98	3	12	5
0.75	84	594	7.07	52	61.90	0.75	0.75	0.97	5	12	5
0.70	78	542	6.95	47	60.26	0.75	0.71	0.97	7	13	6
0.65	64	415	6.48	35	54.69	0.67	0.67	0.96	17	20	5
0.60	54	342	6.33	28	51.85	0.60	0.62	0.90	20	21	6

Fig. 10 Results of experimenting with the 3-classes classification model trained on the cardiocartography dataset

data instances, which may include instances with incongruent predictions. Hence, union rules may be fully coherent with regard to the covered original rules, but at the same moment their fidelity may be less than 1.

Another observation is that there are many original rules that cannot be united with others and remain standalone even when the roughness threshold is low. Thus, for $\rho = 0.60$, only 21 out of 54 rules in the resulting model are union rules. Nevertheless, the achievable degree of simplification can be judged as quite high, especially in terms of the number of conditions and the proportion of complex rules with more than 5 conditions. Moreover, such rules help to identify outlier instances that require different logic than most of the other ones. For example, the rule at the bottom of Fig. 4 describes only one data instance, and it could not be united with any other rule.

An important property of the generalized rule set is that simpler rules (i.e., including fewer conditions) describe a much larger proportion of the data instances than in the original model. So, the minimal number of conditions in one rule is 3 in the original model and 1 in the simplified versions obtained with $\rho = 0.65$ and $\rho = 0.60$ (the maximal number of conditions per rule is 12 in all models). The original model contains 4 rules with 3 conditions describing 47 data instances, 7 rules with 4 conditions describing 62 instances, and 23 rules with 5 conditions describing 163 instances. Taken together, the 34 simpler rules describe 272 data instances out of 1700, i.e., only 16%. In the model obtained with $\rho = 0.65$, the numbers of the rules including from 1 to 5 conditions are, respectively, 2, 1, 6, 9, and 11, and these 29 rules describe 1009, 36, 201, 200, and 48 data instances, respectively, i.e., 1494 instances in total. As two or more rules from a generalized model may be applicable to the same data instances, the cumulative number of the data instances correctly (i.e., in congruence with the original model) described by the model with $\rho = 0.65$ is 2709, and thus the simplest rules make 55% $(1494/2709 * 100)$ of the correct descriptions. However, these rules describe 96 data instances incorrectly, i.e., their joint fidelity is $0.94 = 1494/(1494 + 96)$.

Hence, there are multiple aspects of simplification: the number of rules, the number of conditions, the proportion of simple rules, and the proportion of the data described by these simple rules. Moreover, the conditions of the simplest rules applicable to large number of instances indicate which features have higher importance than others. For example, the model with $\rho = 0.65$ contains a rule with a single condition "**If** *histogram mode* < 148.5 **then** class $= 1$" correctly describing 881 data instances and having fidelity 0.97. This rule reveals the importance of the feature *"histogram mode"*. Another example is that *percentage of time with abnormal short term variability* is rather low for class 1 (healthy), but gets higher for 2 and 3 (suspect and pathology), at the same moment the *histogram mean* is lower for the pathology class, compared to other two.

Additionally we investigated the effects of model pruning on the performance of our algorithm and presented the results in Sect. 5.4.

Ten classes task

This dataset allows us to see the difference between interpretability for simpler and more complex task on the same data features. Decision tree for 10 classes consists of 202 rules describing the same 1700 data instances of the training dataset. The descriptive statistics of the results of the experiments are shown in Fig. 11. As it could be expected, the potential for compression and generalization is lower when the number of classes is higher due to the congruence requirement. Compared to the 3-class model, the 10-class model also consists of more complex rules, i.e., ones that have more conditions. The generalization increases the proportion of simpler rules having up to 5 conditions, which contributes to better comprehensibility, along with the decrease of the number of the rules. The fact that with $\rho = 0.60$ we see many more union rules that are not rough, compared to the

rho	N rules	Total N conditions	Mean N conditions	N rules >5 conditions	% rules >5 conditions	min *CC*	min fidelity	Total fidelity	N rough rules	N union rules	Max depth
	202	1739	8.61	185	91.58	1.00	1.00	1.00	0	0	1
1.00	197	1682	8.54	177	89.85	1.00	1.00	1.00	0	5	2
0.95	188	1567	8.34	166	88.30	1.00	0.95	0.99	0	14	2
0.90	185	1536	8.30	163	88.11	1.00	0.90	0.98	0	17	2
0.85	183	1518	8.30	161	87.98	1.00	0.88	0.98	0	19	2
0.80	177	1472	8.32	157	88.70	0.83	0.80	0.98	3	19	3
0.75	167	1357	8.13	145	86.83	0.75	0.77	0.95	9	25	4
0.70	163	1326	8.13	139	85.28	0.75	0.70	0.92	10	23	5
0.65	149	1172	7.87	121	81.21	0.67	0.67	0.90	26	36	5
0.60	139	1062	7.64	106	76.26	0.60	0.63	0.85	25	34	5

Fig. 11 Results of experimenting with the 10-classes classification model derived from the cardio dataset

3-class model also confirms that the global logic of the model is more complex.

5.2 Home equity line of credit (HELOC), 2 classes

This example application is based on the Explainable Machine Learning Challenge organized by a group of commercial and academic organisations.[2] Based on an anonymized dataset of applications made by homeowners, the challenge requires creation of a readily explainable model predicting the value of the variable Risk Performance, which may be either "bad" or "good". In order to allow a correct decision tree creation, we excluded records with special values and two categorical features. We first created an obviously incomprehensible random forest model with 50 trees without depth restriction, that achieves perfect accuracy, and then generated a mimic model approximating the behaviour of the random forest model. The mimic model consists of 384 rules containing in total 3019 conditions which involve 21 features with numeric value domains. The statistics describing the results of the generalization are presented in Fig. 12.

It can be seen that the mimic model can be slightly simplified even with $\rho = 1$. It means that the model has some redundancies. While increasing the degree of simplification, the total fidelity of the simplified model to the original one decreases gradually but more substantially than it was in other experiments. A probable reason is high similarities between rules giving opposite predictions: when a rule gets more general, it may become applicable to additional data instances that are described by other rules, even if it does not cover those other rules (i.e., the conditions of the rules partly overlap). The projection plot on the left of Fig. 13 supports this guess: blue and red dots representing rules with negative and positive outcomes, respectively, tend to be very close in the plot. An interesting side effect of the simplification is that it increases the separation, i.e., the dissimilarity between the rules with the positive and negative outcomes. This can be seen from comparing the projection of the original rule set on the left of Fig. 13 to the projections of the simplified rule sets obtained with $\rho = 0.85$ (Fig. 13, centre) and with $\rho = 0.60$ (Fig. 13, right).

The similarities between rules are demonstrated in Fig. 14, where a table displays a group of rules represented by a cluster of closely positioned dots in the projection plot shown in Fig. 13, left. The cluster has been interactively selected by dragging a frame around it. The table shows that the rules with negative results (Action $= 0$) differ from the closest rules with positive results (Action $= 1$) by just one condition.

Using this example, we can demonstrate how our techniques can be used to answer the question of the challenge organizers: if an applicant who has got a negative result ("bad"), can the model easily explain what should be changed to turn the result to positive ("good")? For this purpose, the rule R^0 that gave the negative result needs to be identified in the projection plot (the localization of rules is supported by highlighting) and the rules with positive results having close positions in the plot need to be selected, for example, as shown in Fig. 13, left. The rule R^0 can be conveniently compared with the other selected rules using the table display shown in Fig. 14 or a glyph representation, as in Fig. 15. In this figure, the rule no. 41 is selected as a reference for comparison. Its conditions are represented by cyan-filled bars in all glyphs. It is easy to see that a small increase of the value of the third feature (Percent Installment Trades) will make rule no. 271 with a positive outcome applicable to this case instead of the rule no. 41. Other possibilities are to make rule no. 241 applicable by increasing the value of the 7th feature (Net Fraction Installment Burden), or rule no. 340 by increasing the value of the 8th feature (Percent Trades Never Delinquent), or rule no. 254 by increasing the value of the second feature (Consolidated Version of Risk Markers), or rule no. 79 by decreasing the value of the 5th feature (Number Trades 60+ ever).

[2] See https://community.fico.com/s/explainable-machine-learning-challenge.

rho	N rules	Total N conditions	Mean N conditions	N rules >5 conditions	% rules >5 conditions	min *CC*	min fidelity	Total fidelity	N rough rules	N union rules	Max depth
	384	3019	7.86	343	89.32	1.00	1.00	1.00	0	0	1
1.00	351	2603	7.42	287	81.77	1.00	1.00	1.00	0	33	2
0.95	339	2463	7.27	269	79.35	1.00	0.95	0.99	0	42	3
0.90	323	2291	7.09	242	74.92	1.00	0.90	0.97	0	57	3
0.85	313	2195	7.01	231	73.80	1.00	0.86	0.95	0	62	4
0.80	293	2008	6.85	206	70.31	0.80	0.80	0.91	9	65	4
0.75	257	1693	6.59	171	66.54	0.75	0.75	0.85	37	71	5
0.70	234	1497	6.40	146	62.39	0.70	0.70	0.83	38	75	5
0.65	201	1232	6.13	113	56.22	0.67	0.65	0.78	70	86	6
0.60	158	941	5.96	84	53.16	0.60	0.60	0.73	53	67	7

Fig. 12 Results of experimenting with the 2-classes classification model derived from the HELOC dataset

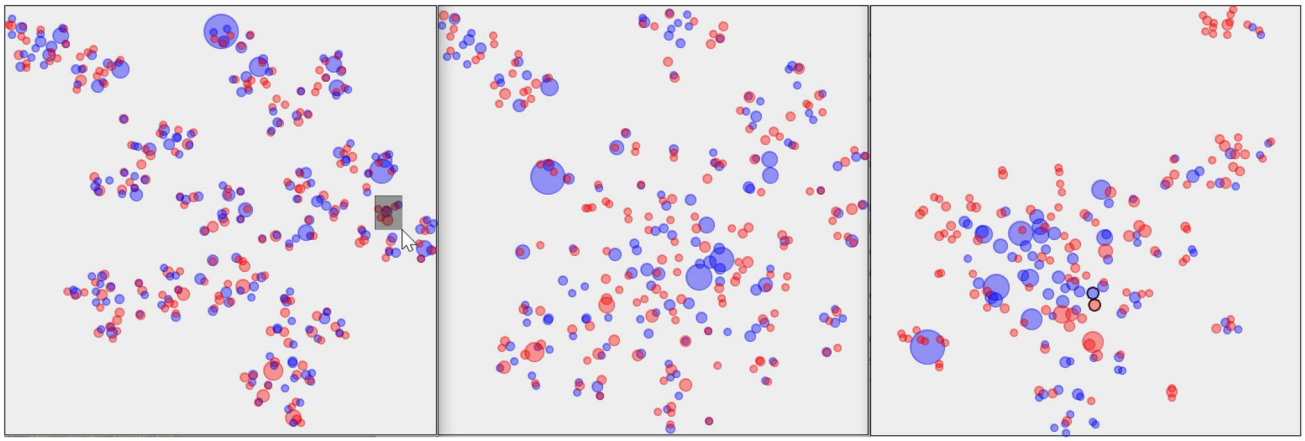

Fig. 13 Similarity-based t-SNE projections of the original HELOC rule set (left) and the simplified versions obtained with $\rho = 0.85$ (centre) and with $\rho = 0.60$ (right). Blue dots correspond to rules predicting negative outcome and red to rules with positive predictions (colour figure online)

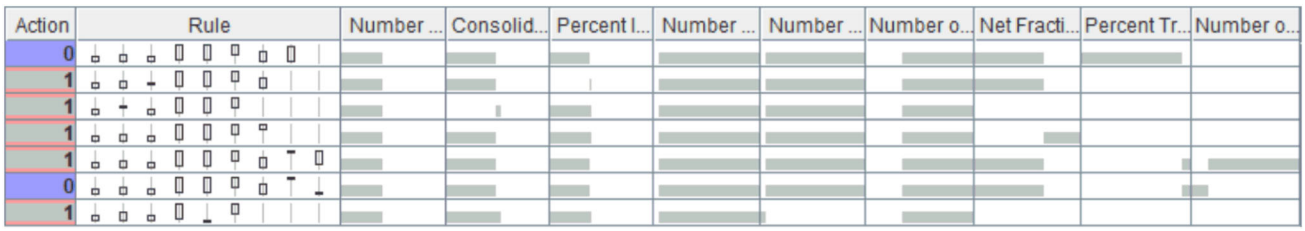

Fig. 14 A group of rules represented by closely positioned dots in the projection plot in Fig. 13, left (enclosed in a dark grey rectangle) is shown in a table view (colour figure online)

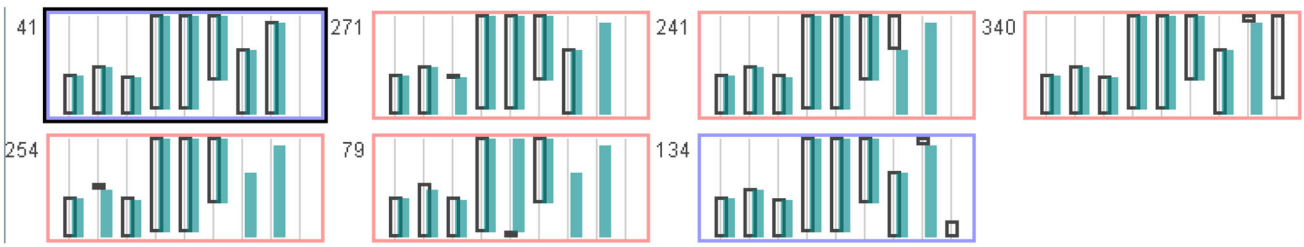

Fig. 15 The same rules as in Fig. 14 are represented by glyphs. The first rule (no. 41) is selected for comparison with the others. One can see how the negative result may be turned to positive by changing the value of just one feature

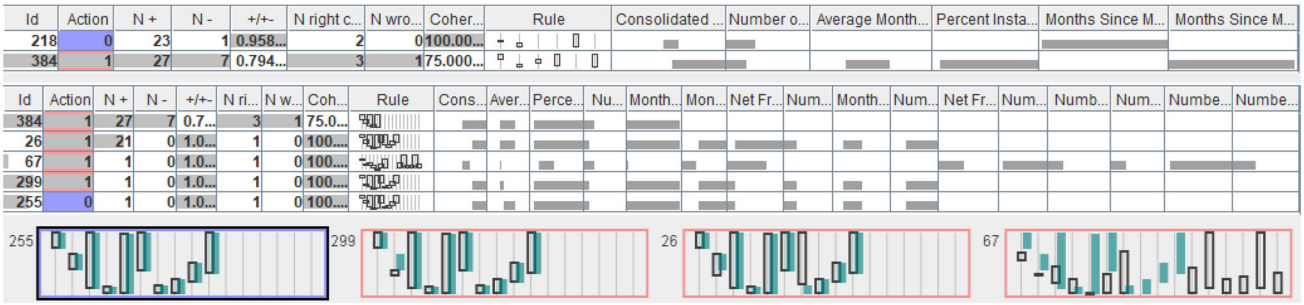

Fig. 16 Finding possibilities of transforming a negative outcome to a positive one using a generalized model obtained with $\rho = 0.60$. Top: The table view displays two selected rules, which are shown as circles with black outline in the projection on the right in Fig. 13. Middle: The table shows the original rules covered by rule 384, which predicts a positive outcome. Rule 384 itself is presented in the top row of the table. It has one exception (Action = 0) presented in the last row. Bottom: The glyphs represent the original rules covered by the generalized rule 384. The exception rule is selected for comparison with the rules predicting positive results (colour figure online)

In a similar way, one can use more general rules of a simplified model version, in which critical features like these can be easily identified. An example is demonstrated in Fig. 16, where the descriptive model is obtained with $\rho = 0.6$. Let a specific case with a negative outcome be described by the generalized rule 218. The closest generalized rule with a positive outcome is 384. These two rules are represented by the circles with black outline in the projection on the right of Fig. 13 and shown in the table view at the top of Fig. 16. According to rule 384, achieving a positive result requires increasing the value of the feature "Consolidated Version of Risk Markers" (CVRM). However, rule 384 has one exception in addition to correctly covering three original rules. Using interactive visualization, all covered rules can be extracted from the original model for examination and comparison. These rules, along with rule 384, are displayed in the table view in the middle of Fig. 16. It becomes evident that simply increasing the value of CVRM will not be sufficient, as the exception rule (255) shown in the last row of the table has a condition with a high value of this feature. To understand the difference between all conditions in the exception rule and the positive rules, the glyph representation in the lower part of Fig. 16 can be utilized. The user can observe that, in addition to increasing CVRM, achieving a positive outcome is possible by having a lower value of feature 2 (Average Months in File) according to rule 299 (row 5 in the table), or a higher value of feature 7 (Net Fraction Installment Burden) according to rule 26 (row 2). However, rule 67 (row 3) is more challenging to fulfil as it requires meeting multiple specific conditions.

This example highlights an important aspect. With interactive visual tools, it is possible to extract, examine, and compare all the original rules that are correctly and wrongly covered by a generalized rule. This capability enables the user to identify any conditions that might have been omitted during the generalization process. As a result, the user can gain comprehensive insights into the various possibilities for achieving the desired outcome. By considering the complete information, users can make more informed decisions and better understand the potential trade-offs.

5.3 Material science, regression

This example is taken from the NOMAD 2018 Kaggle challenge to predict the formation energies and bandgap energies of alloys from transparent conductors.[3] In contrast to the previous examples, this is a regression task.

In material science, a state-of-the-art prediction method is RuleFit [9], a rule ensemble for regression that learns a sparse linear model over a large number of candidate rules together with the original features. Note that the winning methods of the Kaggle challenge (n-gram [44], SOAG [45], MBTR [46]) do not substantially outperform RuleFit on the entirety of the dataset ([47] have shown that they do, however, perform well on well-defined subsets of the data). A prediction is obtained using a weighted sum of rule outputs and feature values. We used RuleFit to train a rule ensemble on the 402 data instances using formation energy as target. The resulting rule ensemble consists of 396 rules. Analysing these rules, we find that most of them describe single data instances which is not surprising given that the number of rules is nearly equal to the number of data instances. This indicates some potential for compression.

The target values range from 0 to 0.7676. We have set the tolerance threshold ϵ to 0.010, 0.020 and 0.050, corresponding to the 2%, 5%, and 10% percentiles of target values. The collected statistics are presented in Fig. 17.

The second rows in all three tables demonstrate that the largest part of the simplification is achieved at the cost of

[3] See https://www.kaggle.com/c/nomad2018-predict-transparent-conductors.

rho	N rules	Total N conditions	Mean N conditions	N rules >5 conditions	% rules >5 conditions	min *CC*	min fidelity	Total fidelity	N rough rules	N union rules	Max depth
	396	3099	7.83	359	90.66	1.00	1.00	1.00	0	0	1
1.00	240	1760	7.33	204	85.00	1.00	1.00	1.00	0	96	6
0.95	240	1760	7.33	204	85.00	1.00	1.00	1.00	0	96	6
0.90	240	1760	7.33	204	85.00	1.00	1.00	1.00	0	96	6
0.85	238	1738	7.30	202	84.87	0.86	0.86	0.99	2	96	6
0.80	237	1726	7.28	201	84.81	0.80	0.80	0.99	3	96	6
0.75	234	1701	7.27	198	84.62	0.75	0.75	0.99	5	96	6
0.70	231	1669	7.23	195	84.42	0.71	0.71	0.97	8	95	6
0.65	230	1659	7.21	194	84.35	0.67	0.67	0.97	9	95	6
0.60	225	1603	7.12	188	83.56	0.60	0.60	0.94	12	93	6

rho	N rules	Total N conditions	Mean N conditions	N rules >5 conditions	% rules >5 conditions	min *CC*	min fidelity	Total fidelity	N rough rules	N union rules	Max depth
	396	3099	7.83	359	90.66	1.00	1.00	1.00	0	0	1
1.00	185	1309	7.08	149	80.54	1.00	1.00	1.00	0	97	8
0.95	185	1309	7.08	149	80.54	1.00	1.00	1.00	0	97	8
0.90	185	1309	7.08	149	80.54	1.00	1.00	1.00	0	97	8
0.85	184	1301	7.07	148	80.43	0.86	0.86	1.00	1	96	8
0.80	182	1282	7.04	146	80.22	0.80	0.80	0.99	3	96	9
0.75	180	1266	7.03	144	80.00	0.80	0.75	0.98	3	96	9
0.70	177	1247	7.05	143	80.79	0.71	0.71	0.96	6	92	9
0.65	175	1224	6.99	140	80.00	0.71	0.65	0.94	7	91	9
0.60	166	1138	6.86	129	77.71	0.60	0.60	0.88	14	87	9

rho	N rules	Total N conditions	Mean N conditions	N rules >5 conditions	% rules >5 conditions	min *CC*	min fidelity	Total fidelity	N rough rules	N union rules	Max depth
	396	3099	7.83	359	90.66	1.00	1.00	1.00	0	0	1
1.00	104	677	6.51	78	75.00	1.00	1.00	1.00	0	70	10
0.95	104	677	6.51	78	75.00	1.00	1.00	1.00	0	70	10
0.90	104	677	6.51	78	75.00	1.00	1.00	1.00	0	70	10
0.85	102	654	6.41	75	73.53	0.86	0.88	1.00	0	70	10
0.80	99	631	6.37	73	73.74	0.80	0.80	0.97	3	67	10
0.75	98	622	6.35	71	72.45	0.80	0.75	0.97	3	67	10
0.70	97	612	6.31	70	72.16	0.74	0.71	0.95	4	66	10
0.65	96	603	6.28	68	70.83	0.68	0.68	0.92	6	65	10
0.60	89	542	6.09	60	67.42	0.60	0.60	0.86	9	61	10

Fig. 17 Results of experimenting with the regression model derived from the material science dataset. The tables, from top to bottom, correspond to $\epsilon = 0.010$, $\epsilon = 0.020$, and $\epsilon = 0.050$, respectively

decreasing the precision of the predictions from specific numbers to intervals. For example, a union rule predicts that the result will be from 0.2179 to 0.2214 instead of predicting a fixed number like 0.22. Hence, the chosen value of the tolerance threshold ϵ has the highest impact on the resulting degree of simplification and generalization, whereas the impact of the roughness threshold ρ is quite small: the decrease in the number of rules due to decreasing ρ from 1 to 0.6 ranges from only 6% for $\epsilon = 0.010$ to 14% for $\epsilon = 0.050$, and the decrease in the number of conditions ranges from 9% for $\epsilon = 0.010$ to 20% for $\epsilon = 0.050$.

Nevertheless, the potential of our method for simplifying the explanation of regression models can be considered as high. It is quite reasonable to posit that a user rarely needs an exact explanation for each individual numeric value that can be predicted by a model. Rather, the user can be satisfied with a model description telling what combinations of conditions lead to model results fitting in different ranges of values (e.g., high and low). The user-controlled value of ϵ determines how narrow or wide these intervals will be. Thus, by choosing a larger value, a user can obtain a compact description of model behaviour even without decreasing the coherence coefficient of the rules. For example, the same material science model

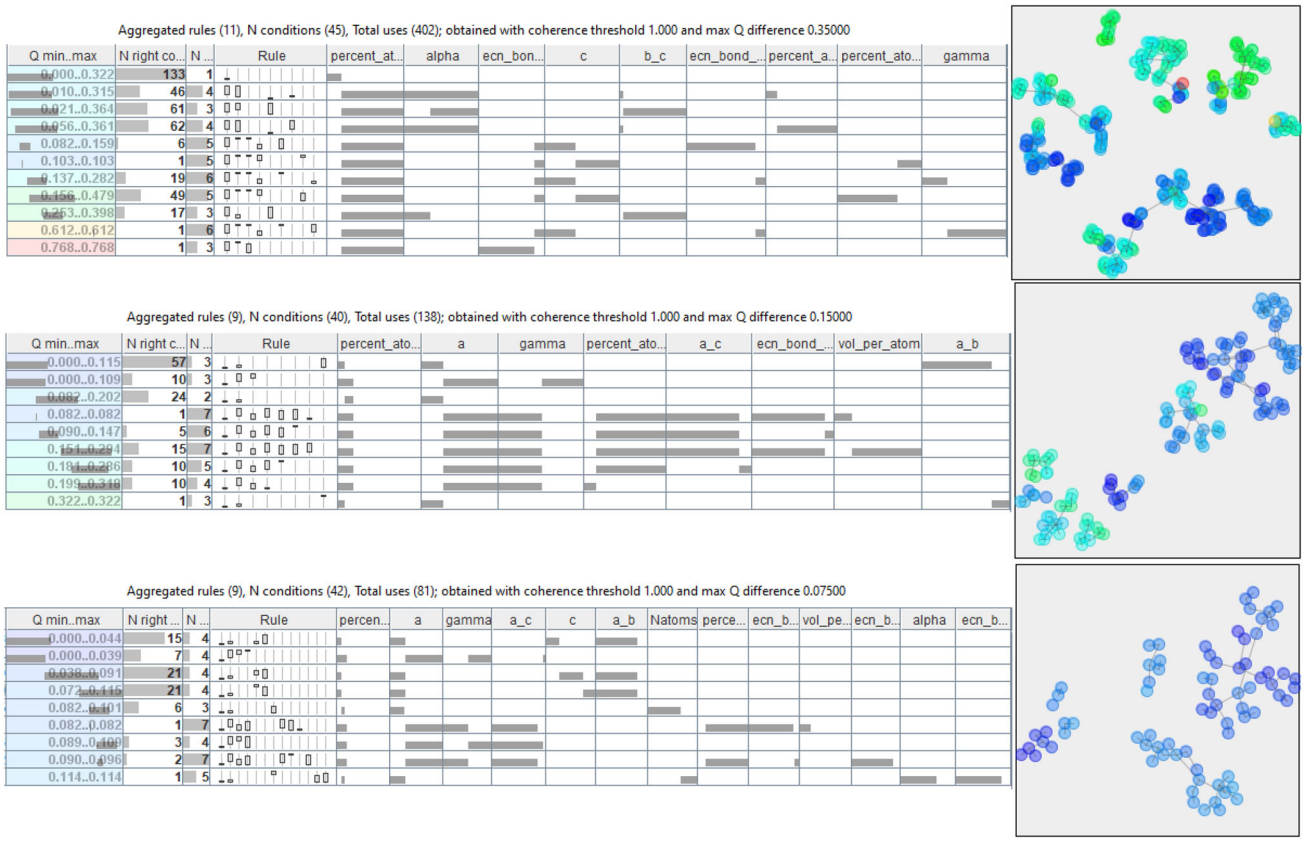

Fig. 18 Hierarchical exploration of the material science rule set. Top: The original set of 396 regression rules has been reduced to 11 generalized rules with $\rho = 1$ and $\epsilon = 0.35$. Middle: The subset of 133 original rules covered by the first rule from the upper table has been generalized to 9 rules with $\rho = 1$ and $\epsilon = 0.15$. Bottom: The subset of 77 original rules predicting values less than 0.115 has been generalized to 9 rules with $\rho = 1$ and $\epsilon = 0.07$. The t-SNE projections on the right show groups of original rules covered by common generalized rules. The colours from deep blue through green and yellow to red represent the rule outcomes (i.e., predicted values) from the lowest to the highest (colour figure online)

generalized with $\epsilon = 0.075$ is described by 80 rules with 494 conditions, and $\epsilon = 0.25$ gives only 22 rules with 94 conditions in total and from 2 to maximum 6 conditions in each individual rule. Like in the other cases, such combinations of conditions, as well as the features they involve, can be considered the most influential for the model result.

This example suggests an interesting possible way of using the generalization method for regression tasks. First, a high-level overview of the behaviour of a model is gained by obtaining a very rough (large ϵ) generalized representation of it. Then, subsets of the original rules that have been unified in the result of the generalization are investigated in more detail by applying the generalization method with a smaller ϵ separately to these subsets.

The idea is demonstrated in Fig. 18. Here, the user has generalized the rule set with $\rho = 1$ and $\epsilon = 0.35$. The resulting descriptive model consists of 11 rules (Fig. 18, top). The table displays the rules, with the first column titled 'Q min.max' indicating the intervals of predicted values. The rule in the top row of the table, which has only one condition, states that when 'percent_atom_in' has a low value (less than 0.1771), the predicted result is below 0.322. It is worth noting that 'percent_atom_in' has values above 0.1771 in all other generalized rules. This upper rule covers 133 original rules. Next, the user selects these rules and applies generalization with $\rho = 1$ and $\epsilon = 0.15$, resulting in 9 generalized rules (Fig. 18, middle). By examining these rules, the user can identify additional features contributing to lower or higher predicted values within the interval from 0 to 0.322. Having noticed that the upper two generalized rules have very similar intervals of predicted values, the user selects the subset of the original rules whose predictions fall within the union of these intervals. To obtain a refined description of this subset, the user applies generalization with $\rho = 1$ and $\epsilon = 0.075$, resulting in 9 descriptive rules (Fig. 18, bottom). The number of features involved in the rules has increased from 8 to 13. This refined description allows the user to investigate the impact of different combinations of feature values on the

Pruning degree	rho	N rules	Total N conditions	Mean N conditions	N rules >5 conditions	% rules >5 conditions	min *CC*	min fidelity	Total fidelity	N rough rules	N union rules	Max depth
1		107	807	7.54	78	72.90	1.00	1.00	1.00	0	0	1
1	1.00	104	779	7.49	74	71.15	1.00	1.00	1.00	0	3	2
1	0.90	93	666	7.16	63	67.74	1.00	0.92	0.98	0	10	3
1	0.80	79	533	6.75	48	60.76	0.86	0.80	0.96	4	12	4
1	0.70	77	528	6.86	48	62.34	0.75	0.71	0.96	7	11	5
1	0.60	58	380	6.55	31	53.45	0.60	0.60	0.94	15	17	5
2		48	285	5.94	22	45.83	1.00	1.00	1.00	0	0	1
2	0.90	47	274	5.83	20	42.55	1.00	0.92	0.99	0	1	2
2	0.80	47	274	5.83	20	42.55	1.00	0.92	0.99	0	1	2
2	0.70	43	240	5.58	17	39.53	0.75	0.82	0.99	3	3	2
2	0.60	32	161	5.03	11	34.38	0.60	0.60	0.95	11	12	3
3		26	131	5.04	9	34.62	1.00	1.00	1.00	0	0	1
3	1.00	24	113	4.71	7	29.17	1.00	1.00	1.00	0	2	2
3	0.90	23	103	4.48	5	21.74	1.00	0.99	0.99	0	3	2
3	0.80	23	103	4.48	5	21.74	1.00	0.99	0.99	0	3	2
3	0.70	23	103	4.48	5	21.74	1.00	0.99	0.99	0	3	2
3	0.60	21	96	4.57	5	23.81	0.60	0.80	0.97	2	3	3

Fig. 19 Results of experimenting with the pruned versions of the 3-classes classification model trained on the cardiocartography dataset

predicted outcome in more detail. It's important to note that all generalizations were performed with $\rho = 1$, indicating that none of the generalized rules have exceptions.

5.4 Experiments with pruned models

In order to investigate the effect of pruning of the original model on the compression that can be achieved with our algorithm, we created decision trees with different levels of cost complexity pruning. In Fig. 19, pruning degree 1 denotes the least compressed model and pruning degree 3—most compressed model. An immediate observation that can be made is that the result of our algorithm is highly dependent on the pruning performed. Thus, the strongly pruned model is highly resistant to generalization. This means that our method can be useful also for practitioners in order to understand if the model is compact enough and does not contain redundancies. Another interesting observation is that the model 2 can be compressed only with significant roughness: the models obtained with $\rho = 0.90$ and $\rho = 0.80$ are identical, while the simplification attempt with $\rho = 1$ fails (no union rules could be produced). This is even more pronounced for the strongest pruning.

It should be noted that each of these pruned models has progressively declining accuracy when trained, which showcases the difference of our approach compared to pruning: while keeping a required degree of coherence and fidelity to the original model, our method gives a simplified description without any effect on the accuracy of the original model.

6 Discussion

We proposed an approach to facilitating comprehension of models that are interpretable by design, but too large to be actually intelligible by a human due to cognitive limitations. For this, we explain the logic of a large (in principle) interpretable model by a simplified descriptive model that suits human cognitive properties: while averse to large volumes of information, humans are good in dealing with vague concepts, approximate statements, and fuzzy reasoning. One can think of a data mart[4] as an example of widely used descriptive models in real world: instead of giving a human full data from business, a special high level view is formed in order to understand the processes happening in it.

Our approach differs from the approaches of regularization or compression techniques for obtaining simpler yet accurate enough predictive models, since our goal is not to retrain or improve a model, but to explain it. That is, our aim is to represent the logic of a complicated pseudo-interpretable model at hand. This is independent of whether the model was derived from data or is an interpretable model mimicking the behaviour of some black-box model. A descriptive model is not meant to be used as a substitute for the model at hand (i.e., it is not used for making predictions), but its purpose is to provide an explanation for the global logic of that model. The cost of high simplification is loss of predictive accuracy. The more complex the global logic of a model, the harder it is to generalize and represent it by a simple descriptive model of sufficient fidelity. Our algorithm for rule generalization

[4] https://en.wikipedia.org/wiki/Data_mart

allows users to control how similar to the original model the descriptive model must be in terms of predictions. Besides, the possibility to see the exceptions and the hierarchy of rule generalization allows a human to increase the fidelity (and, hence, the complexity) of the description as desired. In addition to model description, our approach also supports model exploration in terms of important features, their impact on predictions, and which feature combinations would create outliers.

Similar to a mimic model [2], there is a trade-off between interpretability, i.e., size of the descriptive model, and accuracy of the description, i.e., similarity of the descriptive model to the original one. The goal typically is to have the most concise descriptive model that is still sufficiently similar. The similarity, however, is in general hard to assess: A meaningful similarity measure not only depends on the functional similarity, e.g., as measured by a suitable norm on the function space, but also on the expected difference given the data distribution. We use two measures as a surrogate: fidelity to measure difference in predictions and the coherence coefficient to measure structural similarity.

Fidelity is widely used in explainable AI [2] and measures the number of data instances in a reference dataset for which the predictions of the original and descriptive differ by more than ϵ. Fidelity can be inaccurate for two reasons: (i) using a reference dataset as an empirical sample of the data distribution is only an approximation, and (ii) the difference of the two models on those data samples where they do not agree is unbounded.

We introduced the coherence coefficient to measure the number of rules subsumed coherently by a generalization in the description. Rules not covered coherently are kept in the description as exception rules so that no structural parts of the original model are discarded. This guarantees that for all data instances where at least one rule in the original model is satisfied, at least one rule in the descriptive model will be satisfied and their predictions do not differ by more than ϵ. Therefore, the set of points for which at least one rule in the descriptive model is satisfied is a superset of the points for which at least one rule in the original model is satisfied. This measure can still be inaccurate on the points for which a rule in the descriptive model is satisfied but no rule in the original model is. For those points, the difference in prediction is in general unbounded.

For exploring the properties of the algorithm, we created a visualization interface and performed a series of experiments applying the rules generalization to four different models. Our case studies showed that the human interaction for setting the acceptable level of description roughness is very helpful—while significant roughness makes the result easier to comprehend, obtaining several descriptive models with different degrees of roughness can help to refine the understanding of the predictions logic. Interesting enough, the experiment with a regression model showcased that in this case imprecision of predictions allows to achieve higher simplification than rules roughness control. Based on this observation, we propose a method for focused exploration of selected parts of a regression model at hand: starting from a very simple but very imprecise descriptive model, a user selects one of the generalized rules, extracts the subset of the original rules it covers, and obtains a more precise descriptive model for this subset. In this way, the understanding of the model logic can be gradually refined and deepened.

We also found out that the distance metric we introduced can be used for answering the prediction justification questions, i.e., determining what features should be changed and how to make a model change its prediction. Knowing the rule by which the current prediction was made, one uses the distance metric to select the closest rules giving the desired outcome and inspects how their conditions differ from the conditions of the rule that was applied.

Our algorithm allows two variants of use (see Sect. 3) and is open to further extensions. For example, it can take into account possible overlaps (partial coverage) between a generalized rule and the original rules. Currently, the coherence coefficient of a general rule is calculated only from the rules fully covered by it. This definition can be extended in an obvious way to including also partial coverage by an appropriate change in the computation of CC.

An interesting direction for future work is to combine generalization of the rules with merging and generalization of the features involved in the rule conditions, which is expected to enable much higher degrees of model logic simplification. Examples of feature merging can be seen in the award-winning solution of the HELOC Challenge[5] [48], where 6 groups of semantically related original features were integrated into composite features thereby reducing the original 23 features to 10 features. Such feature merging is usually hard to perform in the interpretable manner without domain knowledge and human reasoning. However, it may be possible to detect automatically (by analysing a rule set) which features are likely to be related and propose groups of such features to a human expert for considering and controlling integration. This can significantly strengthen the comprehensibility of a descriptive model.

Acknowledgements This work was supported by Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the *Lamarr Institute for Machine Learning and Artificial Intelligence* (Lamarr22B), and by EU in projects *SoBigData++* and *CrexData* (Grant Agreement No. 101092749).

Author Contributions AL: text, experiments; KM: text, experiments; AG: idea, tool implementation, analysis; AN: idea, tool implementation, analysis, text.

[5] https://community.fico.com/s/blog-post/a5Q2E0000001czyUAA/fico1670

Funding Open Access funding enabled and organized by Projekt DEAL. This work was supported by the Cancer Research Center Cologne Essen (CCCE). The authors have no relevant financial or non-financial interest to disclose.

Code availability The code used in this manuscript is available at https://github.com/gandrienko/TapasExplTreeViewer.

Declarations

Conflict of interest The authors declare that they have no conflicts of interest.

Ethical approval This research does not require ethics approval and uses only publicly available data sources that are appropriately referenced.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Ribeiro, M., Singh, S., Guestrin, C.: "Why Should I Trust You?": explaining the predictions of any classifier. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, pp. 97–101. Association for Computational Linguistics, San Diego (2016)
2. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 1–42 (2018). https://doi.org/10.1145/3236009
3. Kovalerchuk, B., Ahmad, M.A., Teredesai, A.: Survey of explainable machine learning with visual and granular methods beyond quasi-explanations. In: Pedrycz, W., Chen, S.M. (eds.) Interpretable artificial intelligence: a perspective of granular computing, pp. 217–267. Springer (2021)
4. Letham, B., Rudin, C., McCormick, T.H., Madigan, D.: Interpretable classifiers using rules and Bayesian analysis: building a better stroke prediction model. Ann. Appl. Stat. **9**(3), 1350–1371 (2015)
5. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 1–42 (2018)
6. Quinlan, J.R.: Generating production rules from decision trees. In: Proceedings of the 10th International Joint Conference on Artificial Intelligence—Volume 1. IJCAI'87, pp. 304–307. Morgan Kaufmann Publishers Inc. (1987)
7. Arya, V., Bellamy, R.K., Chen, P.Y., Dhurandhar, A., Hind, M., Hoffman, S.C., et al.: One explanation does not fit all: a toolkit and taxonomy of ai explainability techniques. arXiv preprint arXiv:1909.03012 (2019)
8. Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., Baesens, B.: An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. Decis. Support Syst. **51**(1), 141–154 (2011)
9. Friedman, J.H., Popescu, B.E.: Predictive learning via rule ensembles. Ann. Appl. Stat. **2**(3), 916–954 (2008)
10. Al-Akhras, M., El Hindi, K., Habib, M., Shawar, B.A., et al.: Instance reduction for avoiding overfitting in decision trees. J. Intell. Syst. **30**(1), 438–459 (2021)
11. Wilson, D.R., Martinez, T.R.: Reduction techniques for instance-based learning algorithms. Mach. Learn. **38**(3), 257–286 (2000)
12. Esposito, F., Malerba, D., Semeraro, G., Kay, J.: A comparative analysis of methods for pruning decision trees. IEEE Trans. Pattern Anal. Mach. Intell. **19**(5), 476–491 (1997). https://doi.org/10.1109/34.589207
13. Helmbold, D.P., Schapire, R.E.: Predicting nearly as well as the best pruning of a decision tree. Mach. Learn. **27**(1), 51–68 (1997)
14. Dash, S., Gunluk, O., Wei, D.: Boolean decision rules via column generation. Adv. Neural Inf. Process. Syst. **31**, 4655–4665 (2018)
15. Su, G., Wei, D., Varshney, K.R., Malioutov, D.M.: Learning sparse two-level boolean rules. In: IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6 IEEE (2016)
16. Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable decision sets: a joint framework for description and prediction. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1675–1684 (2016)
17. Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., MacNeille, P.: A Bayesian framework for learning rule sets for interpretable classification. J. Mach. Learn. Res. **18**(1), 2357–2393 (2017)
18. Joly, A., Schnitzler, F., Geurts, P., Wehenkel, L.: L1-based compression of random forest models. In: 20th European Symposium on Artificial Neural Networks (2012)
19. Painsky, A., Rosset, S.: Lossless compression of random forests. J. Comput. Sci. Technol. **34**(2), 494–506 (2019)
20. BuciluǍC, Caruana, R., Niculescu-Mizil, A.: Model compression. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 535–541 (2006)
21. Bohanec, M., Bratko, I.: Trading accuracy for simplicity in decision trees. Mach. Learn. **15**(3), 223–250 (1994)
22. Qiao, L., Wang, W., Lin, B.: Learning accurate and interpretable decision rule sets from neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 4303–4311 (2021)
23. Freitas, A.A.: Comprehensible classification models: a position paper. ACM SIGKDD explorations newsletter, pp. 1–10 (2014)
24. Quinlan, J.R.: Simplifying decision trees. Int. J. Man Mach. Stud. **27**(3), 221–234 (1987)
25. Bénard, C., Biau, G., Veiga, S., Scornet, E.: Interpretable random forests via rule extraction. In: International Conference on Artificial Intelligence and Statistics. pp. 937–945. PMLR (2021)
26. Izza, Y., Ignatiev, A., Marques-Silva, J.: On explaining decision trees. arXiv preprint arXiv:2010.11034 (2020)
27. Hulot, A., Chiquet, J., Jaffrezic, F., Rigaill, G.: Fast tree aggregation for consensus hierarchical clustering: application to multi-omics data analysis. In: Statistical Methods for Post-Genomic Data (SMPGD) (2019)
28. Kavšek, B., Lavrač, N., Ferligoj, A.: Consensus decision trees: using consensus hierarchical clustering for data relabelling and reduction. In: European Conference on Machine Learning, pp. 251–262. Springer (2001)
29. Strecht, P., Mendes-Moreira, J., Soares, C.: Inmplode: a framework to interpret multiple related rule-based models. Expert Syst. **38**(6), e12702 (2021)

30. Andrzejak, A., Langner, F., Zabala, S.: Interpretable models from distributed data via merging of decision trees. In: IEEE Symposium on Computational Intelligence and Data Mining (CIDM), pp. 1–9. IEEE (2013)
31. Andrienko, N., Andrienko, G., Fuchs, G., Slingsby, A., Turkay, C., Wrobel, S.: Visual analytics for data scientists. Springer (2020)
32. Sacha, D., Kraus, M., Keim, D.A., Chen, M.: VIS4ML: an ontology for visual analytics assisted machine learning. IEEE Trans. Vis. Comput. Graph. **25**(1), 385–395 (2019). https://doi.org/10.1109/TVCG.2018.2864838
33. Andrienko, N., Lammarsch, T., Andrienko, G., Fuchs, G., Keim, D., Miksch, S., et al.: Viewing visual analytics as model building. Comput. Graph. Forum **37**(6), 275–299 (2018). https://doi.org/10.1111/cgf.13324
34. Andrienko, N., Andrienko, G., Miksch, S., Schumann, H., Wrobel, S.: A theoretical model for pattern discovery in visual analytics. Vis. Inf. **5**(1), 23–42 (2021). https://doi.org/10.1016/j.visinf.2020.12.002
35. Spinner, T., Schlegel, U., Schäfer, H., El-Assady, M.: explAIner: a visual analytics framework for interactive and explainable machine learning. IEEE Trans. Vis. Comput. Graph. **26**(1), 1064–1074 (2020). https://doi.org/10.1109/TVCG.2019.2934629
36. Ming, Y., Qu, H., Bertini, E.: RuleMatrix: visualizing and understanding classifiers with rules. IEEE Trans. Vis. Comput. Graph. **25**(1), 342–352 (2019). https://doi.org/10.1109/TVCG.2018.2864812
37. Yuan, J., Nov, O., Bertini, E.: Visualizing rule sets: exploration and validation of a design space. arXiv preprint arXiv:2103.01022 (2021)
38. Rote, G.: Computing the minimum Hausdorff distance between two point sets on a line under translation. Inf. Process. Lett. **38**(3), 123–127 (1991)
39. Jaccard, P.: The distribution of the flora in the alpine zone. New Phytol. **11**(2), 37–50 (1912). https://doi.org/10.1111/j.1469-8137.1912.tb05611.x
40. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **11**(9), 2579–2605 (2008)
41. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: OPTICS: ordering points to identify the clustering structure. SIGMOD Rec. **28**(2), 49–60 (1999). https://doi.org/10.1145/304181.304187
42. Dua, D., Graff, C.: UCI machine learning repository. http://archive.ics.uci.edu/ml
43. Ayres-de Campos, D., Bernardes, J., Garrido, A., Marques-de Sa, J., Pereira-Leite, L.: SisPorto 2.0: a program for automated analysis of cardiotocograms. J. Matern. Fetal Med. **9**(5), 311–318 (2000)
44. Sutton, C., Ghiringhelli, L.M., Yamamoto, T., Lysogorskiy, Y., Blumenthal, L., Hammerschmidt, T., et al.: Crowd-sourcing materials-science challenges with the NOMAD 2018 Kaggle competition. NPJ Comput. Mater. **5**(1), 1–11 (2019)
45. Bartók, A.P., Payne, M.C., Kondor, R., Csányi, G.: Gaussian approximation potentials: the accuracy of quantum mechanics, without the electrons. Phys. Rev. Lett. **104**(13), 136403 (2010)
46. Huo, H., Rupp, M.: Unified representation of molecules and crystals for machine learning. arXiv preprint arXiv:1704.06439 (2017)
47. Sutton, C., Boley, M., Ghiringhelli, L.M., Rupp, M., Vreeken, J., Scheffler, M.: Identifying domains of applicability of machine learning models for materials science. Nat. Commun. **11**(1), 1–9 (2020)
48. Chen, C., Lin, K., Rudin, C., Shaposhnik, Y., Wang, S., Wang, T.: A holistic approach to interpretability in financial lending: models, visualizations, and summary-explanations. Decis. Support Syst. **152**, 113647 (2022). https://doi.org/10.1016/j.dss.2021.113647

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.