Visual Explanation of Machine Learning Models in Shifted Paired Coordinates in 3D

Boris Kovalerchuk
Department of Computer Science
Central Washington University
Ellensburg, WA, USA
Boris.Kovalerchuk@cwu.edu

Joshua Martinez
Department of Computer Science
Central Washington University
Ellensburg, WA, USA
Joshua.Martinez@cwu.edu

Michael Fleagle
Department of Computer Science
Central Washington University
Ellensburg, WA, USA
Michael.Fleagle@cwu.edu

***Abstract*—Machine learning (ML) methods achieved remarkable success recently. However, the trust by domain experts for many new black-box models is quite low. Visualization is natural way to involve domain experts to the process of development of explainable models, which can mitigate the deficiencies of black-box models. It is typically easier for humans to understand and reason within visualizations of data and models. Recent Sequential Rule Generation (SRG) algorithms for categorical qualitative data, and a lossless visualization system in 3-D based on the Shifted Paired Coordinates (SPC-3D) allow producing ML models as (1) interpretable approximators of black-boxes or as (2) independent interpretable models with accuracy comparable with black boxes on the same data. However, SRG can generate a large set of rules that are difficult to analyze and visualize. This paper proposes a new algorithm to Join and Modify Rules (JMR), which creates a smaller set of rules with the same precision and coverage for a given set of rules. It is explored in the case studies, which show its efficiency along with SPC-3D visualization system for getting trustable models.**

Keywords— machine learning, shifted paired coordinates, visual knowledge discovery, general line coordinates, lossless 3D visualization.

I. Introduction

Recently, machine learning (ML) techniques have shown tremendous success. Nonetheless, domain experts (DE) have little faith in many black-box models. Involving domain experts in the process of developing interpretable models supported by visualization is a promising way to resolve this issue [1,14]. It works in two ways:

(1) by discovering interpretable models as *approximators* of black-box models and

(2) by discovering rules as *independent* interpretable models with accuracy comparable to it for black boxes on the same data. For (1) the source of the data is an output of the black-box model on data generated in simulation runs.

Logic rules form a major class of interpretable models, and this paper focuses on them. Multiple methods have been proposed for years to discover logic rules with several reviews available, e.g., [7,8]. Logic rules can be represented in multiple ways as logic formulas, as decision tables [4] or hyperblocks [5] with decision trees representing a special situation, when rules can be organized and visualized hierarchically [4].

These representations can be linked with different lossless visualizations of rules and cases covered by these rules [6,14]. Logic rules fit well to *categorical data*, which do not have intrinsic distances or orders resulting in visualization challenges [13]. Sequential Rule Generation (SRG) algorithms for *categorical qualitative data* [2], and a lossless visualization system in 3-D based on the Shifted Paired Coordinates (SPC-3D) [3], are used to support (1) and (2). However, a set of rules from [2] can be large and difficult to analyze and visualize.

To resolve this issue this work offers a new algorithm to **Join and Modify Rules** (**JMR**). It creates a smaller set of rules with the same precision and coverage from a given set of rules. The JMR algorithm is explored in this paper. Its efficiency is shown along with SRG algorithms and SPC-3D visualization system on real data as tools to support a domain expert in discovering interpretable and trustable ML models as logic rules.

The combination of rule extraction and visualization is becoming a key emerging approach in the domain of interpretable ML (ILM) [1,14]. The combination that we propose has several advantages over presented in [14]. In [14] rules are discovered by *approximation* of black-box models to explain them. In contrast algorithms presented in this paper discover rules as *independent* interpretable models from original interpretable training data. In avoids many deficiencies of model approximation noted in the literature [16]. Our algorithms can discover rules as *approximators* of black-box models too. Next [14] does not propose new rule learning algorithms but uses Scalable Bayesian Rule Lists (SBRL) algorithm from [15]. This algorithm produced rules on Mushroom data with the probability of correct classification from 95.8 to 99.9 [15]. Our algorithms produced rules with 100% precision and coverage on the same data. See section III A. These rules are simpler, they use only 5 attributes but rules in [15] use 8 attributes. The rule visualization process in [14] uses only traditional visualization methods without lossless visualization of high-dimensional data. In contrast SPC-3D is a lossless visualization method for high-dimensional data allowing to observe rules along with full underling n-D data.

This paper is organized as follows. Section 1 introduces the main concepts. Sections 2 and 3 provide the information on the case studies with Mushroom and Wisconsin breast cancer data respectively. Section 4 highlights the features of SPC-3D

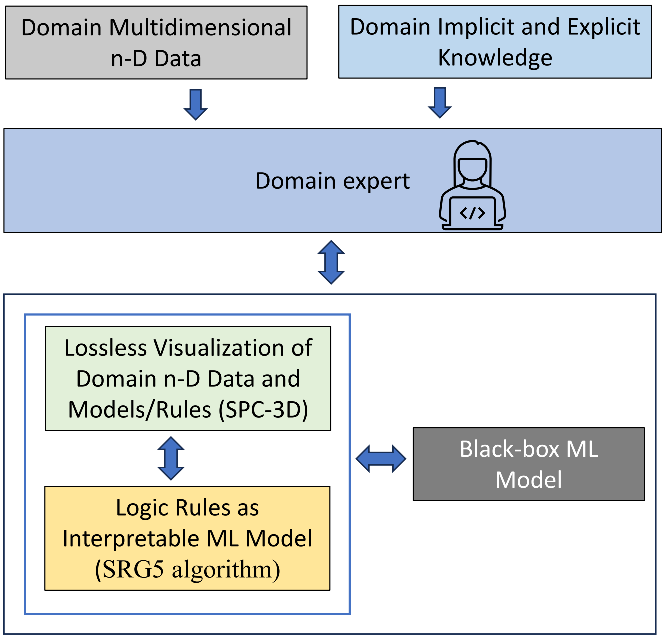


Fig. 1. Paradigm of the domain expert in the driver seat of the interpretable model development and evaluation.

software tool and Section 5 summarizes the paper and outlines future work strategy.

II. MAIN CONCEPTS

A. Domain Expert in the driver seat of modeling

The key idea to facilitate acceptance of ML models by domain expert to **trust** and **deploy** ML models in the domain is putting domain experts in the **driver seat** of the model development and evaluation. This paradigm is much more than just putting **human-in-the-loop** in an auxiliary role.

The case studies presented below demonstrate that domain experts can be provided with interpretable rules, which are easily understandable by them in analytical and visual forms. It expands an opportunity for domain experts to explore the rules, and cases covered by these rules, visually *without any loss of multidimensional information.* Another important advantage of both logical rules and SPC-3D visualization is abilities for the domain expert to understand them without learning mathematics of ML algorithms and models, which are expressed in the terms that are foreign for the domain of the expert. Hidden layers, activations, softmax and many others are among these terms. This is a core property for the domain experts to be in the driver seat of the model development and evaluation utilizing the domain knowledge in this process. This paradigm was successfully demonstrated in [6] with lossless Shifted Paired Coordinates visualized in 2-D space. This work expands this paradigm to any logical rules and ML models.

The modeling process with a domain expert in the driver seat is demonstrated in Fig. 1. First a domain expert explores data in the SPC-3D visualization system before creating any rules interactively by observing data classes colored by distinct colors to discover any visible pattern separation classes. This is the initial data exploration stage.

Next a user can run a rule discovery process. In our case it is the algorithm SRG5 [2] and the JMR algorithm. Then the user can visualize these rules in the SPC-3D system to explore if they can be trusted from the viewpoint of explicit and implicit domain knowledge. At this stage the user can analyze rules together or each rule individually, which is demonstrated later.

The confidence for the user in this exploration can come from domain knowledge and data with attributes that are directly interpretable in the domain. The logical relations between interpretable attributes captured in rules are also interpretable. Visualization of cases and rules simplifies this analysis. In addition, a user can see contributions of each individual attribute to a linear discrimination function (LDF) as it is illustrated later.

B. Join and Modify rules (JMR) algorithm

The steps of **Join and Modify rules** (**JMR**) **algorithm** are:

1. Order given rules in descending order by their coverage and simplify them if possible keeping their precision and coverage.
2. Check that rules are combinable.
3. Order combinable rules in descending order by coverage.
4. Combine rules with highest coverage first.
5. Generalize combined rules, if possible:

- Simplify the rule by excluding one attribute clause like $x_1 \in [1,5]$ or $x_2=1$.
- Compute coverage of the simplified rule. If the rule coverage goes up and precision of the rule does not go below low precision limit then keep this simplified rule.
- Do this for each attribute clause of each combined rule.

For example, rules R_a: $x_1 \in [1,2] \lor (x_2=3 \ \& \ x_3=1)$ and R_b: $x_1 \in [2,4] \lor (x_2=3) \lor (x_3=1)$, can produce a new rule R_c: $x_1 \in [1,4] \lor (x_2=3) \lor (x_3=1)$. Examples of use of JMR algorithm are in the Section IV.

Step 5 is iterated for possible removal of each attribute. Removing two or more attributes in one step is possible, but shortening he process will depend on the extra time needed to find statistically correlated attributes likely removable together.

C. Shifted Paired Coordinates with LDF in 3-D

Below we describe the idea of **Shifted Paired Coordinates** (**SPC**) in 3D with 4-D Iris Data based on [3]. Fig. 2 shows each 4-D point $\mathbf{x}=(x_1,x_2,x_3,x_4)$ as a line that connects pair (x_1,x_2) with pair (x_3,x_4). Each pair is shown in its Cartesian Coordinates: (x_1,x_2) in (X_1,X_2) and (x_3,x_4) in (X_3,X_4). The colors of lines indicate the class of Iris flowers.

Each line is a **lossless** representation of a respective 4-D point $\mathbf{x}$. While we can discriminate classes in this visualization, it does not show the discrimination function. Also, some lines of different classes overlap especially in (X_3,X_4). To represent losslessly the data with more than 4 dimensions, SPC used more cubes with more lines connecting points in cubes, which we illustrate later.

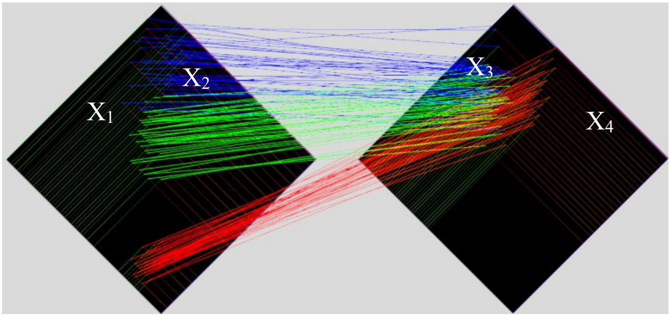


Fig. 2. All Iris data in lossless SPC-3D (top 2-D projection).

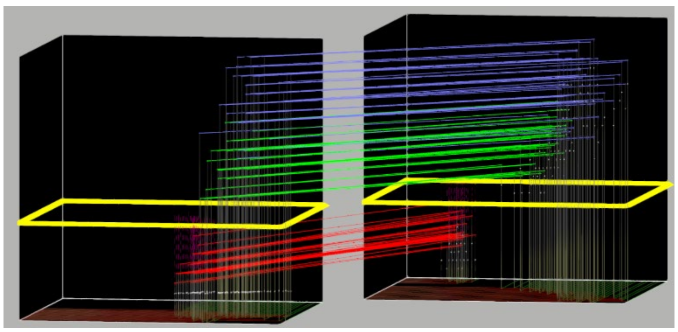


Fig. 3. All Iris cases losslessly vsuailized in SPC-3D (front proejction) with a linear function/plane discriminating blue and green cases from red cases.

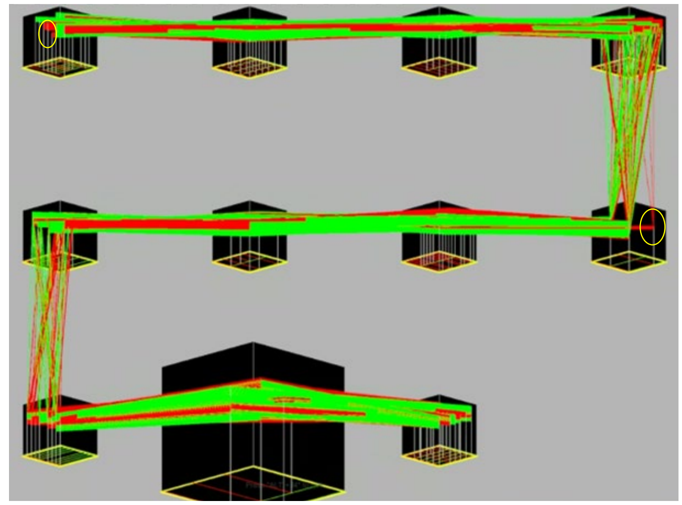
To make separation classes better the current SPC-3D system implementation uses a vertical dimension Z with conversion of each pair to a triple: (x_1,x_2,z) and (x_3,x_4,z). Here $z= f(x_1,x_2,x_3,x_4)$. A user can assign this function interactively or it can be found by machine learning algorithms, which discriminate classes. In the current implementation this function is a linear discriminant function (LDF). Two cubes in Fig. 3 illustrate it, where a yellow plane is an LDF that separates blue and green cases from red cases. Only red cases are below the yellow discrimination plane. This plane was found interactively in [3].

More generally, each pair can be converted into a triple: (x_1,x_2,z_{12}) and (x_3,x_4,z_{34}) with functions $z_{12}= f_{12}(x_1,x_2,x_3,x_4)$ and $z_{34}=f_{34}(x_1,x_2,x_3,x_4))=z_{34}$. These functions can represent different characteristics of the classification model. For example, we can visualize with them the activations at the different levels, or contributions of a set of attributes to the total classifier value.

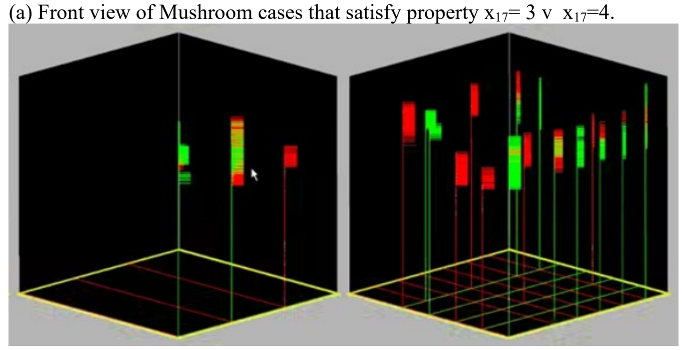
D. Interactions to build rules

The SPC-3D visualization allows a user selecting a subset of data by describing their property and visualizing that subset. It aids user's comprehension of data. A subset can be described by any propositional logic formula P, e.g., $P = (x_{17}= 3 \vee x_{17}=4)$. This subset is visualized in Fig. 4 with and without connecting lines for the 22-D Mushroom data from UCI ML repository [9]. The property $x_{17}=3 \vee x_{17}=4$ can be considered as a rule candidate.

It allows to visually explore how pure this property covers the red or the green class. A quick observation shows a heavy mix of cases of both classes. It clearly shows that it is not a good candidate for a high-quality rule that should contain only cases of a single class. It also shows that the green class likely dominates under this property. A user can find additional properties to increase data purity.



(a) Front view of Mushroom cases that satisfy property $x_{17}= 3 \vee x_{17}=4$.



(b) Another mushroom data subset with a mixture of cases of red and green classes with connecting lines ommited.

Fig. 4. Two muchroom data subsets in in SPC-3D.

In Fig 4a the yellow ovals on the first and second rows show pure red areas in the two cubes. Selecting these areas will create a rule A for the red class. Also adding negation of rule A to property P, P & (notA), will make this rule purer for the green class. This example shows how a user can interactively build rules observing all multidimensional data in SPC-3D.

In Fig. 4a individual cases are much better visible for pairs of cubes that are one on the top of another in different rows than when they are on the same row. It shows advantages of locating cubes differently relative to each other. Trial and error interactive rule discovery can produce very impure rules as Fig. 4 shows. Therefore, in this paper interactive exploration is accompanied by analytical discovering rules using SRG5 algorithm [2] combined with by lossless visualization in SPC-3D [3].

III. MUSHROOM DATA CASE STUDY

A. Rules for Muchroom data

Below we describe Mushroom data rules discovered in [2, 29, 32]. Total mushroom data have 8124 cases with 22 attributes. Four rules R_1-R_4 discovered in [2] with SRG5 algorithm are slightly less complex than CR_1-CR_3 rules discovered in [29,32]. Both sets of rules have 100 % precision and coverage of all poisonous mushroom cases (class C_1). Rules

R_1-R_4 contain 11 clauses and CR_1-CR_3 rules contain 13 clauses. Both sets of rules are relatively small and can be visualized directly without the need to use JMR algorithm to generalize them further. These rules are listed below in the form of propositional logic formulas and in the form of the decision tables. This full coverage and 100% precision for the poisonous class means that cases that violate these rules belong to the class C_2 (eatable mushroom) with 100% precision.

R₁: $[(x_5{=}3) \vee (x_5{=}4) \vee (x_5{=}5) \vee (x_5{=}6) \vee (x_5{=}8) \vee (x_5{=}9)] \Rightarrow \mathbf{x} \in C_1$

R₂: $[(x_{20}{=}5)] \Rightarrow \mathbf{x} \in C_1$

R₃: $[(x_{12}{=}3)\ \&\ (x_{21}{=}5)] \Rightarrow \mathbf{x} \in C_1$

R₄: $[(x_8 \neq 1)\ \&\ (x_{21}{=}2)] \Rightarrow \mathbf{x} \in C_1$

TABLE 1. DECISION TABLE FOR RULES R_1-R_4.

R_{11}	R_{12}	R_2	R_3	R_4
$3 \le x_5 \le 6$	$8 \le x_5 \le 9$	$x_{20}=5$	$x_{12}=3$	$x_8 \neq 1$
C_1	C_1	C_1	$x_{21}=5$	$x_{21}=2$
			C_1	C_1

CR Rules [29,32]

CR₁: $[(x_5{=}3) \vee (x_5{=}4) \vee (x_5{=}5) \vee (x_5{=}6) \vee (x_5{=}8) \vee (x_5{=}9)] \Rightarrow \mathbf{x} \in C_1$,

CR₂: $[(x_{20}{=}5)] \Rightarrow \mathbf{x} \in C_1$

CR₃₁: $(x_8{=}2)\ \&\ (x_{12}{=}3) \Rightarrow \mathbf{x} \in C_1$

CR₃₂: $(x_8{=}2)\ \&\ (x_{12}{=}2) \Rightarrow \mathbf{x} \in C_1$

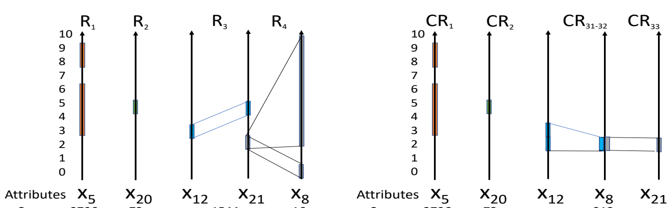
CR₃₃: $(x_8{=}2)\ \&\ (x_{21}{=}2) \Rightarrow \mathbf{x} \in C_1$

TABLE 2. DECISION TABLE FOR RULES CR_1-CR_3.

CR_{11}	CR_{12}	CR_2	R_{31}	R_{32}	R_{33}
$3 \le x5 \le 6$	$8 \le x5 \le 9$	x20=5	x8=2		
C_1	C_1	C_1	$x_{12}=3$	$x_{12}=2$	$x_{21}=2$
			C_1	C_1	C_1

The rules above use only a subset of all 22 attributes, where x_5 has 9 values, x_8 has 2 values, x_{12} has 4 values, x_{20} has 9 values and x_{21} has 6 values. Rules R_1-R_4 have been discovered in [2] using extensive analytical computations by machine learning SRG5 algorithm. Similarly, CR rules also have been discovered using extensive computations by other machine learning algorithms.

The CR computations are hidden for the user to be able to trust the produced rules. Visualization is an efficient method to help users to understand and trust rules. See Fig. 5. Rules R_1 and R_2 are the same as the rules CR_1 and CR_2. Fig. 5 visualizes the *mathematical description* of the rules without showing the actual cases. In the next sections we visualize the rules R_1-R_4, with *cases* covered by them, using SPC-3D system.



(a) Rules R_1-R_4. b) Rules CR_1-CR_3.

Fig. 5. Rule Visualization for poisonous mushroom [3]

B. *Visualization for rules R_1-R_4 jointly*

Figs. 6 and 7 visualize mushroom data for the rules R_1-R_4, where these rules are shown in the white box as follows:

X5 = 3 | X5 = 4 | X5 = 5 | X5 = 6 |
X5 = 8 | X5 = 9
+
X20 = 5
+
X12 = 3 & X21 = 5
+
X8 != 1 & X21 = 2

The notation like x8 !=1 compactly captures $x_8{=}2 \vee x_8{=}3 \vee x_8{=}4 \vee x_8{=}5 \vee x_8{=}6 \vee x_8{=}7$ v $x_8{=}9$ v $x_8{=}10$. The OR operator "$\vee$" is shown in the white box as "I". **Cubes** which do not have an attribute within the rule(s) are **scaled** down 50% (this can be turned off if desired). To visualize **multiple rules** jointly SPC-3D shows them in the white box separated with a "+" sign, which means the OR operator. The **precision** and **coverage** of each class are displayed, when a rule is entered.

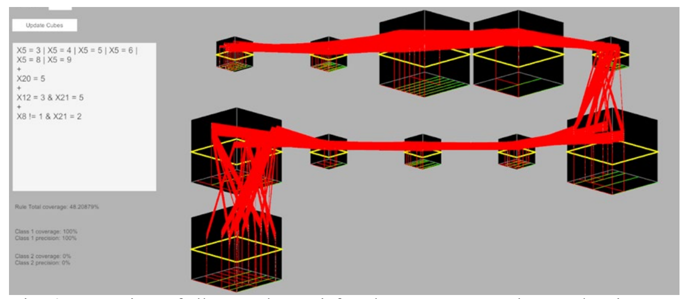


Fig. 6. Front view of all cases that satisfy rules R_1-R_4 on Mushroom data in SPC-3D with large cubes for pairs (**X₅**,X₆), (X₇,**X₈**), (X₁₁,**X₁₂**), (X₁₉,**X₂₀**), (**X₂₁**,X₂₂) where the bold attributes are involved in the rules R_1-R_4.

Figs. 6 and 7 show all cases that satisfy the rules R_1-R_4 in the front and top views, respectively. They visually confirm purity of these rules for class, C_1 of poisonous mushroom (red) without any green (eatable) case present. This allows to explore visually the distribution of these cases in all attributes. If some values of attributes are absent, then they can be present in the opposite class of eatable mushroom. Thus, Figs. 6 and 7 open an opportunity to explore the distribution of both classes.

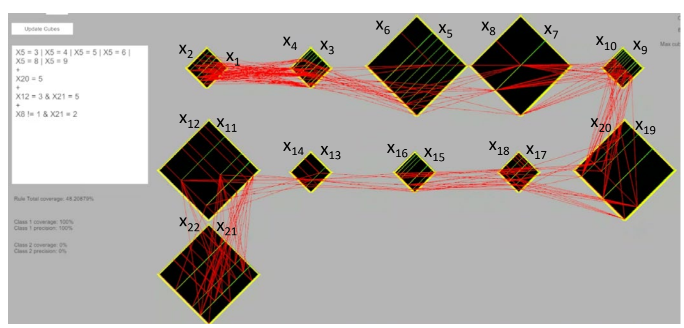


Fig. 7. Top view of Mushroom data that satisfy rules R_1-R_4 from Fig. 5.

A user can switch and group several cubes and attributes to find an equivalent but simpler representation, and to discover interesting and most informative properties of the data. Figs. 8 and 9 present the same data as Figs. 6 and 7, but with the large cubes grouped together. These figures allow a user to focus on the attributes that are involved in the rules R_1-R_4. In Fig. 9, the lines connecting points between cubes are omitted. It decreases the occlusion of cases. The top view in Fig. 7 allows to see better the individual cases, while Fig. 8 and 9 allow to see better the values of LDF on the Z dimension.

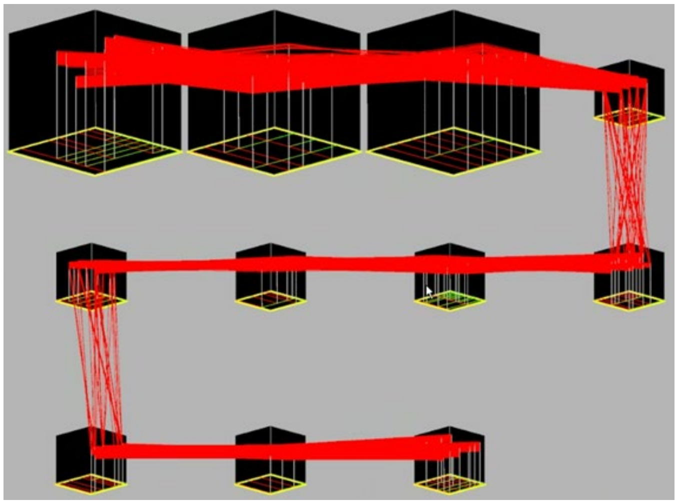


Fig. 8. Front view of cases that satisfy rules R_1-R_4 on Mushroom data in SPC-3D. The large cubes grouped contain attributes that are in rules R_1-R_4.

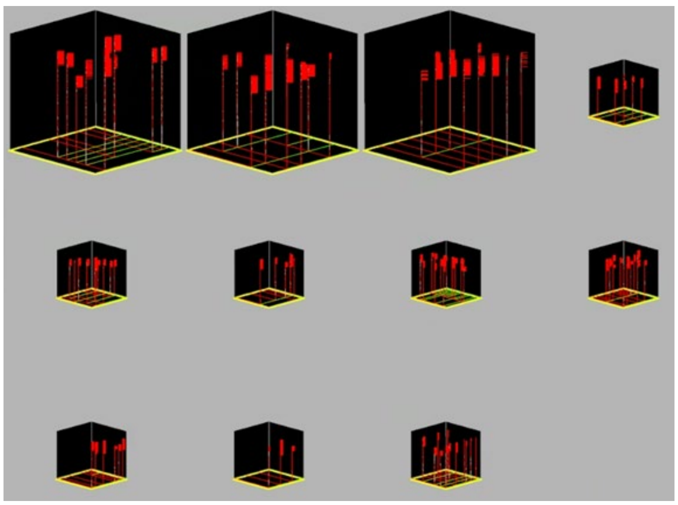


Fig. 9. Front view of cases that satisfy rules R_1-R_4 without connecting lines.

C. Visualization for Rules R_1 - R_4 individually

The SPC-3D system allows a user to observe and explore individual rules and cases covered by each rule. Below we present visualizations for the rules R_1-R_4 individually.

Figs. 10-13 show different views of cases that satisfy **rule R_1**:

$$[(x_5=3)\vee(x_5=4)\vee(x_5=5)\vee(x_5=6)\vee(x_5=8)\vee(x_5=9)] \Rightarrow \mathbf{x}\in C_1.$$

Figs. 10, 12 and 13 show not only the cases in each cube, but also their difference in other attributes by their height. Fig. 11 allows to observe and analyze individual cases well separated. Two views of **rule R_2**: $x_{20}=5$ are shown in Fig. 14. A large cube and a square clearly show that only a single value of x_{20} is used in this rule. It is visible that R_2 covers fewer cases than rule R_1, which was hidden in the joint visualization in Figs. 6-9. In these visualizations a user can see alternative rules that differ in the number of attributes used. A user can pick up most important or causal attributes for visual exploration. A common situation in rule discovery is that not all attributes of the dataset are equally important and used in the rules. In the Mushroom rules R_1-R_4 and CR_1-CR_3 only 5 attributes are used out of total 22

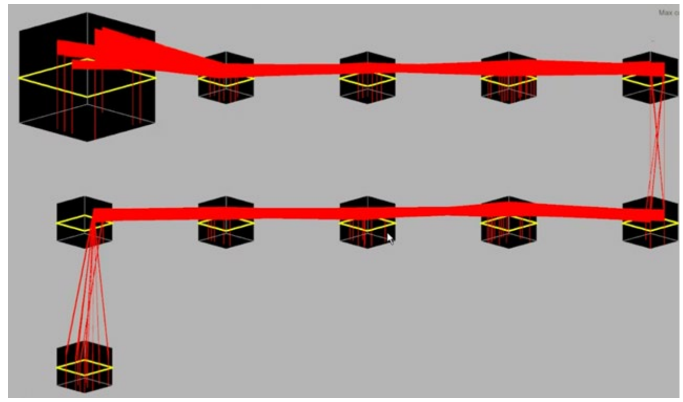


Fig. 10. Top view of cases that satisfy rule R_1 on Mushroom data in SPC-3D. Attributes x_5 and x_{20} are visualized in the large cube.

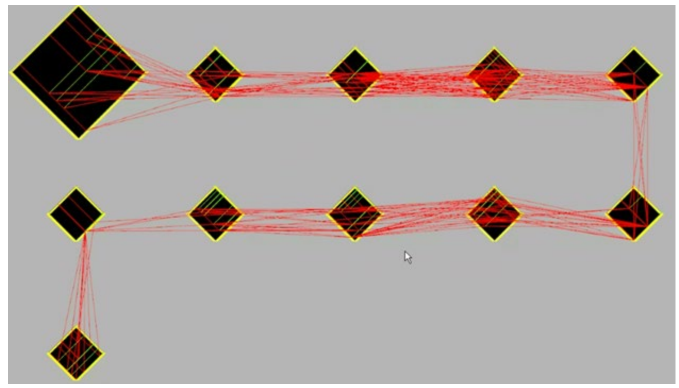


Fig. 11.Top view of cases that satisfy rules R_1 on Mushroom data in SPC-3D.

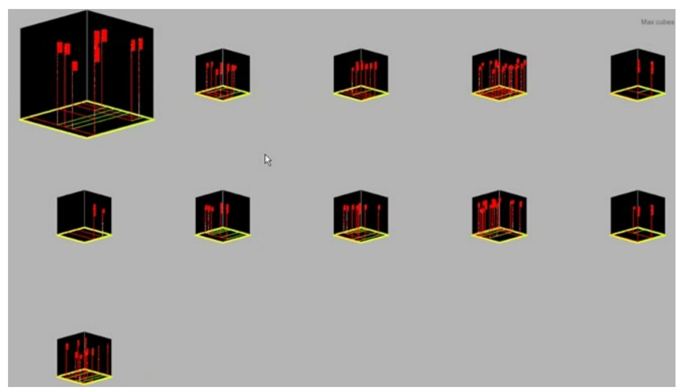


Fig. 12. Front view of the cases of the first cube of rule R_1 on Mushroom data in SPC-3D without showing connections to the next cubes.

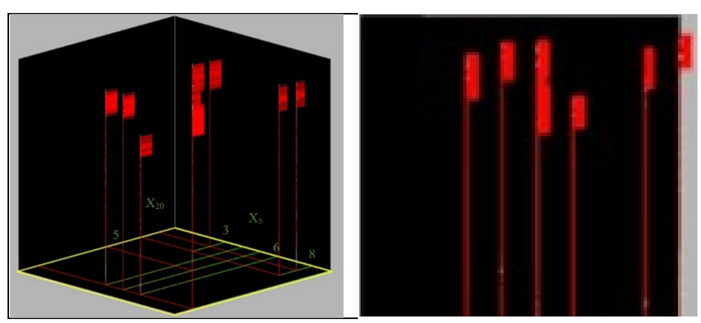
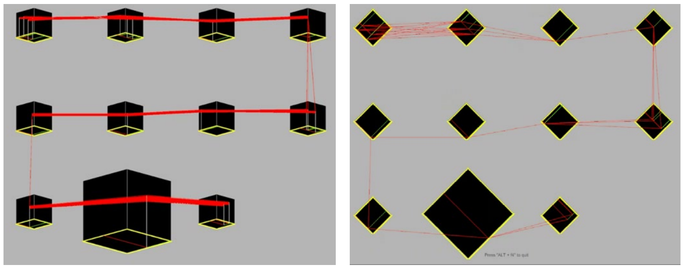


Fig. 13. Zoomed 3-D and 2-D front view of rule R_1 from Fig. 12 on Mushroom data in SPC-3D without showing connections to the next cubes. The cube contains attributes X_5 and X_{20}, where X_5 is constrained by rule R_1, but X_{20} is not constrained by rule R_2, where $x_{20}=5$.



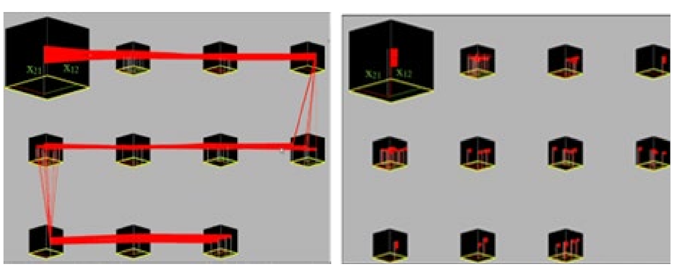
(a) Front view. (b) Top view.

Fig. 14. Mushroom cases covered by rule R_2. The large cube is formed by (X_{19},X_{20}).

In general, it is not the case. Attribute x_k can be absent in the rules and can have low variability between classes and within each class in the given dataset. However, a new case **x** that needs to be classified can have value of x_k that is an outlier relative to the existing values of this attribute in the dataset. It means that similar cases for this attribute are absent in the training dataset. This outlier value can completely change the classification of case **x** if similar cases would be in the training data. However, rules do not include x_k and can misclassify this case.

Lossless SPC-3D visualization of such case **x**, with all 22 attributes, will immediately show the outlier in attribute x_k. It will warn a domain expert, who is in the driver seat of the model, that the prediction needs to be reexamined and likely corrected. It will not be based on the given data or outlier themselves, but on the implicit domain knowledge by the domain expert that this outlier can change prediction. Thus, we cannot rely only on the attributes, which are in the rules. Rules without visualization simply do not show outliers to the users.

The current methodology of AutoML systems tends to simplify the user work, to the extent of providing just the class prediction without such warnings. This happens every day, when recommender systems recommend something to the user without any such warning. This is potentially dangerous especially for the task with high stakes decisions. Two views of **rule R_3**: $[(x_{12}=3)\ \&\ (x_{21}=5)] \Rightarrow \mathbf{x} \in C_1$ are shown in Fig. 15.

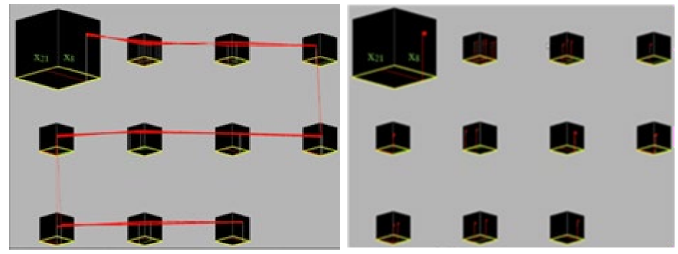


(a) (b)

Fig. 15. Front view of mushroom cases covered by Rule R_3 with connecting lines (a) and without them (b). Attributes x_{12} and x_{21} are in the first cube,

We can clearly see a single point in the first cube on the third row. This is a property of the data that satisfy R_3 in this dataset, but it is not a part of the rule. The rule is wider not requiring a single point there.

It shows that the rule R_3 generalizes data beyond given points. It also allows to put that cube next to the first cube to simplify visualization for further exploration. The front views of **rule R_4:** $[(x_8 \neq 1)\ \&\ (x_{21}=2)] \Rightarrow \mathbf{x} \in C_1$ are shown in Fig. 16. It is visible that the number of cases covered by this rule in smaller than for rules R_1 and R_3.



(a) (b)

Fig.16. Front view of mushroom cases covered by Rule R_4 with connecting lines (a) and without them (b).

IV. WBC Data Case Study

A. *WBC rules from SRG algorithm*

This section presents a case study with discovering rules for Wisconsin Breast Cancer (WBC) data [9]. These data can be treated as qualitative discrete data with 10 values on each of 9 attributes like the Mushroom data. Discovering rules on these data is a significant computational challenge because this space has 10^9 9-D points. However, the Sequential Rule Generation (SRG) algorithms proposed in [2] with the enhancement proposed here, in the form of Join and Modify Rule (JMR) algorithm, allowed not only to successfully run the program for such a big space, but was able to find a simple set of rules presented below. These rules are interpretable, and also simpler than some other models for these data [12]. In our experiments the SRG and JMR algorithms implemented in the VisCanvas 2.0 system [10] produced multiple rules for the WBC data. First we generated 16 rules using the SRG algorithm [2] presented in Table 3, where class C_1 is a class of benign cases.

TABLE 3. RULES WITH 100% PRECISION OF BENIGN CLASS PREDICTION PRODUCED BY SRG ALGORITHM.

R1: $[(x_5=2)\ \&\ (x_6=1)] \Rightarrow \mathbf{x} \in C_1$	**R9**:$[(x_1=4)\ \&\ (x_2=4)] \Rightarrow \mathbf{x} \in C_1$
R2:$[(x_5=1)\ \&\ (x_6\neq5)] \Rightarrow \mathbf{x} \in C_1$	**R10**:$[(x_1=2)\ \&\ (x_3=1)] \Rightarrow \mathbf{x} \in C_1$
R3:$[(x_5=3)\ \&\ (x_6=1)] \Rightarrow \mathbf{x} \in C_1$	**R11**:$[(x_1=2)\ \&\ (x_3=2)] \Rightarrow \mathbf{x} \in C_1$
R4:$[(x_1=1)\ \&\ (x_2=1)] \Rightarrow \mathbf{x} \in C_1$	**R12**:$[(x_5=7)\ \&\ (x_6=4)] \Rightarrow \mathbf{x} \in C_1$
R5:$[(x_1=3)\ \&\ (x_2=1)] \Rightarrow \mathbf{x} \in C_1$	**R13**:$[(x_5=4)\ \&\ (x_6=7)] \Rightarrow \mathbf{x} \in C_1$
R6:$[(x_1=5)\ \&\ (x_2=1)] \Rightarrow \mathbf{x} \in C_1$	**R14**:$[(x_4=9)\ \&\ (x_6=4)] \Rightarrow \mathbf{x} \in C_1$
R7:$[(x_2\neq1)\ \&\ (x_3=1)] \Rightarrow \mathbf{x} \in C_1$	**R15**:$[(x_1=6)\ \&\ (x_3=8)] \Rightarrow \mathbf{x} \in C_1$
R8:$[(x_1=3)\ \&\ (x_3=2)] \Rightarrow \mathbf{x} \in C_1$	**R16**:$[(x_1=6)\ \&\ (x_2=9)] \Rightarrow \mathbf{x} \in C_1$

TABLE 4. CHARACTERISTICS OF RULES R_1-R_{16} WITH 100% PRECISION.

Rules	**R_1**	**R_2**	**R_3**	**R_4**
Coverage %	69.65	10.04	4.15	28.60
Cases	319	46	19	131
Rules	***R_5***	***R_6***	***R_7***	***R_8***
Coverage %	17.47	13.76	4.80	3.28
Cases	80	63	22	15
Rules	***R_9***	***R_{10}***	***R_{11}***	***R_{12}***
Coverage %	0.66	9.17	0.66	0.22
Cases	3	42	3	1
Rules	***R_{13}***	***R_{14}***	***R_{15}***	***R_{16}***
Coverage %	0.22	0.22	0.22	0.22
Cases	1	1	1	1

All these rules have 100% precision with different coverage of benign cases from 319 cases (rule R_1) to a single case (rule R_{12}). See Table 4. Table 5 shows most significant overlaps among rules R_1-R_{16}. For instance, rules R_1 and R_5 cover together 287 cases with 57 cases in their overlap (19.51%), which satisfy both rules.

TABLE 5. MOST SIGNIFICANT OVERLAPS AMONG RULES R_1-R_{16}.

		R_4	R_5	R_6	R_7	R_8	R_{10}
R_1	Overlap %	32.72	19.51	17.73	4.47	2.85	11.19
	Total cases	272	287	282	313	316	295
	Overlap	89	56	50	14	9	33
R_2	Overlap %	18.60	6.25	2.91	3.13	0	0
	Total cases	129	112	103	64	61	49
	Overlap	24	7	3	2	0	0
R_3	Overlap %	0.68	3.23	2.56	3.56	6.67	7.55
	Total cases	148	93	78	39	30	53
	Overlap	1	3	2	1	2	4

In Table 5 rows "Total cases" present all cases of two rules, rows "Overlap" present the number of cases in the rule overlap and rows "Overlap %" present the percent of overlap cases in the set of all cases of two classes.

Total 442 cases are covered by rules R_1-R_{16}, while the total number of cases of the benign class is 458 cases. Respectively, these rules cover 96.51% of all benign cases with only 16 cases not covered by them. Note that these rules use only 6 out of 9 attributes of WBC data.

B. Joining and modifying WBC rules with JMR algorithm

In this section we apply the JMR algorithm presented in section II B, to join and modify these rules to get a smaller set of rules. The goal is to simplify the user's work with these rules. Table 6 presents the results, where 16 rules have been transformed to just two rules JM1 and JM2 keeping 100% precision.

TABLE 6. RULES OF J VERSION OF CLASS 1 WITH 100% PRECISION.

#	Benign class Rules	Cases	Class , %
JM1	$(x_1 \neq 1) \vee (x_2 \neq 2) \vee (x_3 = 3 \ \&\ x_4 = 4) \vee (x_5 = 5 \ \&\ x_6 = 6)$	402	87.77
JM2	$(x_1 = 1 \ \&\ x_2 = 2) \vee (x_3 = 3)$	316	68.99

Rule JM1 covers 402 cases and rule JM2 covers 316 cases. These two rules cover the same 442 benign cases as the 16 rules, with 276 overlap cases between these rules (62.44%). Respectively, rules JM1 and JM2 cover the same 96.51% of all benign cases, with only 16 cases not covered by them too. These rules also use the same 6 out of 9 attributes of WBC data.

C. Alternative sets of WBC rules

The SRG algorithm has multiple parameters that can be modified when the rules are discovered. Below we present an alternative set of rules which was discovered with the SRG algorithm denoted as M version. It produced 13 rules with the largest rule covering 206 cases (44.98% of all benign cases). See Table 7. Similarly, another alternative denoted as version E produced 10 rules presented in Tables 8 and 9. Here the two largest rules E_1 and E_2 covered respectively 338 and 211 cases. Together they covered 400 cases with 149 cases in their overlap. Multiplicity of accurate sets of rules we generated shows the importance of involvement of the domain expert to select them. Note that all rules presented in this paper cover a single class. The full classification of cases of both classes is possible by using the else property, i.e., cases that violate rules of class 1 can be classified to class 2, especially if rules for class 1 cover all cases of this class.

TABLE 7. RULES OF M VERSION OF CLASS 1 WITH 100% PRECISION.

#	Benign class Rules (all clauses are connected by &)	Rule cases	Class coverage, %
M1	$x_1\in[1,5]$, $x_2 = x_3 = 1$, $x_4\in[1,2]$, $x_5 = 2$, $x_6 = 1$, $x_7\in[1,3]$, $x_8 = x_9 = 1$	206	44.98
M2	$x_1\in[2,5]$, $x_2\in[1,2]$, $x_3\in[1,3]$, $x_4 = 1$, $x_5\in[1,2]$, $x_6 = 1$, $x_7 = 2$, $x_8 = x_9 = 1$	71	15.5
M3	$x_1\in[2,5]$, $x_2\in[1,2]$, $x_3\in[1,2]$, $x_4 = 1$, $x_5\in[1,2]$, $x_6 = 1$, $x_7 = 3$, $x_8 = 1$, $x_9 = 1$	44	9.61
M4	$x_1 = 3$, $x_2 = 1$, $x_3\in[1,2]$, $x_4 = 1$, $x_5 = 2$ $x_6\in[1,4]$, $x_7\in[1,2]$, $x_8 = 1$, $x_9\in[1,2]$	37	8.08
M5	$x_1\in[1,2]$, $x_2 = 1$, $x_3 = 1$, $x_4\in[1,2]$, $x_5 = 2$, $x_6 = 1$, $x_7 = 3$, $x_8 = x_9 = 1$	35	7.64
M6	$x_1\in[2,3]$, $x_2 = x_3 = x_4 = 1$, $x_5\in[2,3]$, $x_6 = 1$, $x_7 = 2$, $x_8 = x_9 = 1$	32	6.99
M7	$x_1 = 1$, $x_2\in[1,2]$, $x_3\in[1,3]$, $x_4 = 1$ $x_5 = 2$, $x_6 = 1$, $x_7 = 2$, $x_8 = x_9 = 1$	27	5.90
M8	$x_1\in[1,3]$, $x_2 = x_3 = x_4 = x_5 = x_6 = 1$, $x_7\in[1,3]$, $x_8 = x_9 = 1$	25	5.46
M9	$x_1 = 5$, $x_2\in[1,2]$, $x_3 = x_4 = 1$, $x_5 = 2$, $x_6 = 1$, $x_7\in[1,2]$, $x_8 = x_9 = 1$	20	5.37
M10	$x_1 = 2$, $x_2 = 1$, $x_3 = 1$, $x_4\in[1,2]$, $x_5 = 2$, $x_6 = x_7 = x_8 = x_9 = 1$	11	2.40
M11	$x1 = 4$, $x2 = 1$, $x3\in[1,2]$, $x4 = 1$, $x5 = 2$, $x6 = 1$, $x7 = 3$, $x8 = x9 = 1$	10	2.18
M12	$x_1\in[3,6]$, $x_2 = x_3 = 1$, $x_4 = 3$, $x_5 = 2$, $x_6 = x_7 = x_8 = x_9 = 1$	8	1.75
M13	$x_1 = x_2 = x_3 = x_4 = 1$, $x_5\in[1,2]$, $x_6 = 4$, $x_7\in[1,2]$, $x_8 = x_9 = 1$	5	1.09

Total WBC data include 699 cases but only 683 cases have full records. The E version discovered rules on all 699 cases, because found rules do not use all attributes simultaneously.

TABLE 8. RULES OF E VERSION OF CLASS 1 WITH 100% PRECISION.

#	Benign class Rules	Rule cases	Class coverage, %
E_1	$x_6 = 1 \ \&\ (2 \leq x_5 \leq 3$	338	73.80
E_2	$x_2 = 1 \ \&\ (x_1 = 1 \vee x_1 = 3)$	211	46.07
E_3	$x_5 = 1 \ \&\ x_2 = 2$	63	13.78
E_4	$x_3 = 1 \ \&\ (x_1 = 2 \vee x_2 \neq 1)$	61	13.32
E_5	$x_5 = 1 \ \&\ x_6 \neq 5$	46	10.04
E_6	$x_3 = 2 \ \&\ (2 \leq x_1 \leq 3)$	18	3.93
E_7	$x_6 = 4 \ \&\ (x_4 = 9 \vee x_5 = 7)$	2	0.44
E_8	$x_1 = 6 \ \&\ (x_2 = 9 \vee x_3 = 8)$	2	0.44
E_9	$x_1 = 4 \ \&\ x_2 = 4$	3	0.66
E_{10}	$x_5 = 4 \ \&\ x_6 = 7$	1	0.22

TABLE 9. SIGNIFICANT OVERLAPS OF RULES E_1-E_{10}.

Rule pair	Total cases	Overlap cases	Overlap %
E_1, E_2	400	149	37.25
E_1, E_3	349	50	14,33
E_1, E_4	344	12	3.49
E_1, E_8	349	52	14.09
E_2, E_4	224	5	2.32
E_2, E_7	226	31	13.72
E_3, E_7	102	5	4.09
E_4, E_7	63	1	1.59
E_7, E_8	106	3	2.83

V. FEATURE OF SOFTWARE TOOLS

Fig. 17 shows the user interface of the SPC-3D visualization system. It also illustrates the first cube from Fig. 9 with attributes x_5 and x_{20} visualized in it with polylines that go up made visible by activation in SPC-3D. These polylines show cases in GLC-L visualization defined in [3] for SPC-3D. It allows analyzing contribution of all attributes for each n-D point. Polyline segments with larger vertical projections indicate their larger contribution to the LDF of the respective attribute.

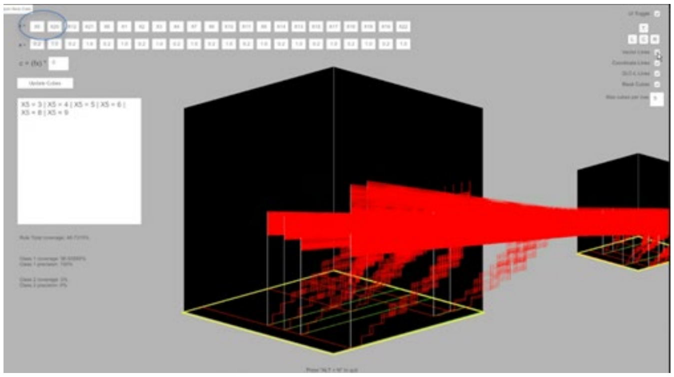


Fig. 17. Zoomed front view of the first cube from Fig. 9 with attributes x_5 and x_{20} visualized with polylines lines that go up. The polylines allow analyzing contribution of each attribute to each case.

Fig. 18 presents samples of SPC-3D user interface. This interface allows a user to set up the order of attributes and their coefficients in the linear discriminant function (Fig. 18a). A user can set up a height of the discrimination line relative to the given value of the linear discrimination function (Fig.18b). The description of the visualized rule is also presented to the user (Fig. 18c).

The first pair of attributes is visualized in the first cube and consecutive pairs are visualized in consecutive cubes. When user changes coefficients the "Update cubes" interface allows to change visualization (Fig. 18d). The rule accuracy analysis is presented in the form of rule coverage per class (Fig. 18e). The 3D controls for manipulating cubes are available to the users too (Fig. 18f). The implementation of SRG and JMR algorithms are available at GitHub [10], and implementation of SPC-3-D visualization system is available at GitHub too [11].

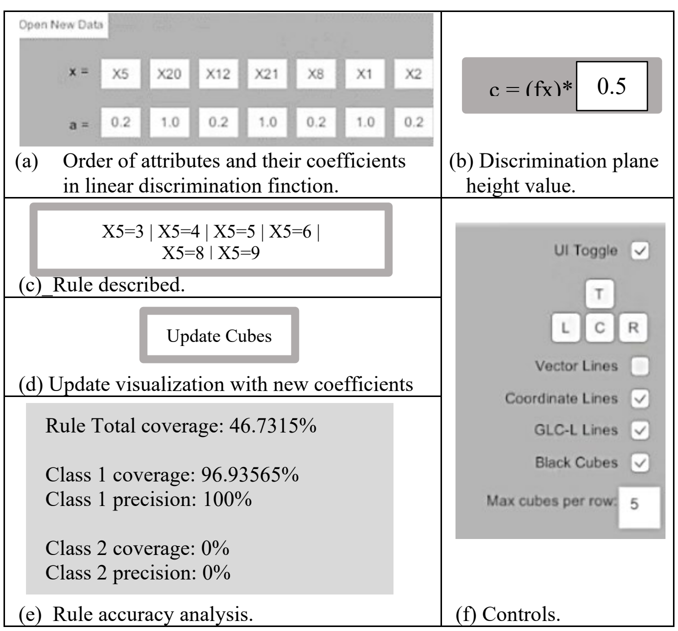


Fig. 18. Samples of SPC-3D user interface.

VI. Conclusion and Future Work

Discovering interpretable rules and visualizations facilitates involvement of domain experts into the process of development of interpretable models, which can mitigate the deficiencies of black-box models. Humans easier understand and reason within visualizations of data and models than in an abstract form. This paper has shown that the Sequential Rule Generation algorithm, Joint and Modify Rule algorithm and Shifted Paired Coordinate visualization system (SPC-3D) allow assisting the domain expert in the development and analysis of interpretable rules as trustable ML models. Multiplicity of sets of rules generated by SRG and JMR algorithms indicates a fundamental problem of selecting an appropriate set of rules among competing accurate rules. There are no obvious ways to resolve it in a pure ML setting. It fundamentally requires a domain expert, who can select trustable rules. Therefore, the intent of this paper, to give a domain expert the ability to be in a driver seat of the model development, is a promising approach, which requires further developments.

VII. References

[1] Kovalerchuk B., Visual Knowledge Discovery and Machine Learning, Springer, 2018.

[2] Kovalerchuk B, McCoy E. Explainable Machine Learning for Categorical and Mixed Data with Lossless Visualization. In: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery 2024, 73-123, Springer. https://arxiv.org/pdf/2305.18437

[3] Martinez J, Kovalerchuk B. General Line Coordinates in 3D. In: 2023 27th International Conference Information Visualisation (IV) 2023, 308-315. IEEE. https://arxiv.org/pdf/2403.13014

[4] 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. Decision Support Systems. 2011;51(1), 141-54.

[5] Huber L, Kovalerchuk B, Recaido C. Visual knowledge discovery with general line coordinates. In: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery 2024, 159-202. Springer. https://arxiv.org/pdf/2305.18429

[6] Kovalerchuk B, Dunn A, Worland A, Wagle S. Interactive decision tree creation and enhancement with complete visualization for explainable modeling. In: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery 2024, 3-40. Springer. https://arxiv.org/pdf/2305.18432

[7] Bodria F, Giannotti F, Guidotti R, Naretto F, Pedreschi D, Rinzivillo S. Benchmarking and survey of explanation methods for black box models. Data Mining and Knowledge Discovery. 2023 Sep;37(5):1719-78.

[8] Fürnkranz J, Kliegr T. A brief overview of rule learning. In:International symposium on rules and rule markup languages for the semantic web 2015, 54-69. Springer.

[9] UC Irvine Machine Learning Repository, https://archive.ics.uci.edu/

[10] VisCanvas 2.0 Rule-Enhancment https://github.com/CWU-VKD-LAB/VisCanvas2.0-Rule-Enhancment

[11] SPC-3D, https://github.com/CWU-VKD-LAB/SPC-3D

[12] Neuhaus N., Kovalerchuk B., Interpretable Machine Learning with Boosting by Boolean Algorithm, Joint 2019 8th Intern. Conf. on Informatics, Electronics & Vision (ICIEV) & 3rd Intern. Conf. on Imaging, Vision & Pattern Recognition (IVPR), 2019, 307-311.

[13] Dennig FL, Joos L, Paetzold P, Blumberg D, Deussen O, Keim DA, Fischer MT. Toward the Categorical Data Map. https://arxiv.org/pdf/2404.16044

[14] Ming Y, Qu H, Bertini E. Rulematrix: Visualizing and understanding classifiers with rules. IEEE transactions on visualization and computer graphics. 2018 Aug 19;25(1):342-52.

[15] Yang H, Rudin C, Seltzer M. Scalable Bayesian rule lists. In: International conference on machine learning 2017, 3921-3930.

[16] Rudin C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence. 2019 May;1(5):206-15.