Multivariate Data Explanation by Jumping Emerging Patterns Visualization

Mário Popolin Neto and Fernando V. Paulovich, Member, IEEE

Abstract—Multivariate or multidimensional visualization plays an essential role in exploratory data analysis by allowing users to derive insights and formulate hypotheses. Despite their popularity, it is usually users’ responsibility to (visually) discover the data patterns, which can be cumbersome and time-consuming. Visual Analytics (VA) and machine learning techniques can be instrumental in mitigating this problem by automatically discovering and representing such patterns. One example is the integration of classification models with (visual) interpretability strategies, where models are used as surrogates for data patterns so that understanding a model enables understanding the phenomenon represented by the data. Although useful and inspiring, the few proposed solutions are based on visual representations of so-called black-box models, so the interpretation of the patterns captured by the models is not straightforward, requiring mechanisms to transform them into human-understandable pieces of information. This paper presents multivariate data explanation (VAX), a new VA method to support identifying and visual interpreting patterns in multivariate datasets. Unlike the existing similar approaches, VAX uses the concept of Jumping Emerging Patterns, inherent interpretable logic statements representing class-variable relationships (patterns) derived from random Decision Trees. The potential of VAX is shown through use cases employing two real-world datasets covering different scenarios where intricate patterns are discovered and represented, something challenging to be done using usual exploratory approaches.

1 INTRODUCTION

Multidimensional or multivariate visualization plays an essential role in exploratory data analysis [1], [2], with the literature presenting several methods leveraging visual representations to capture and represent data patterns [3]. Despite their popularity, allowing users to derive insights and formulate hypotheses, multidimensional visualization methods typically put on users’ shoulders the burden of discovering (visually) the existing patterns, which can be cumbersome and time-consuming, especially for complex datasets. There is a high chance of users being unable to uncover intricate relationships in the data as the number of data variables grows [4], and essential information can be lost due to human cognition limitations.

To help mitigate this issue, data mining and machine learning techniques can be instrumental, supporting users in discovering patterns in the visualizations or discovering patterns in the data that are then visualized [5]. One interesting example of the former category is the use of classification models to capture and represent patterns, acting as descriptive instead of predictive tools [6]. The core idea is that if a model is transparent and understandable [7], for instance, through interpretability strategies [8], it can be used as a proxy to discover patterns in the data enabling tasks that involve the analysis of attribute-class relationships [4], [9].

Although, at first glance, any solution combining classification models and interpretability strategies, such as the recent Visual Analytics (VA) approaches for model interpretation [10], [11], [12] could be used to employ classification models as descriptive strategies, there is an important bottleneck. Such approaches are model-centered, with generalization in mind, so inter-class separation is usually well represented while intra-class information is typically lost, limiting their application to explain data. To our knowledge, a few solutions have been proposed in the visualization literature to use classification models for descriptive purposes. Such as based on Support Vector Machines [9] and Artificial Neural Networks [4]. Although inspiring approaches, both are based on visual representations of so-called black-box models [10], [13], [14], so the interpretation of the patterns captured by the models is not straightforward, requiring mechanisms to transform them into human-understandable pieces of information.

This paper presents multivariate data explanation (VAX), a new VA approach to leverage prediction models’ descriptive power for multivariate data analysis. VAX was designed to capture inter, and intra-class data patterns through the so-called Jumping Emerging Patterns (JEPs) [15]. JEPs are logic statements representing class relationships (patterns) that may be extracted from random Decision Trees (DTs) inferred using the notion of Random Forests (RFs) [16], [17]. JEPs are high (class) discriminative, diversified, and expressive patterns [18], [19] depicting multi-class multi-attribute
relationships in an inherent interpretable format. VAX aggregates JEPs to reduce their number and displays them using a compact matrix metaphor and Dimensionality Reduction (DR) layouts, supporting different analytical tasks involving pattern and data content (clusters and outliers) analysis, revealing intricate and complex information that may be challenging to discover using usual exploratory approaches.

In summary, the main contributions of this paper are:

- A new strategy for JEPs selection and aggregation from random DTs that helps to summarize large sets of patterns while representing the entire data set;
- A new method for JEPs visualization, where a matrix metaphor is used to display patterns as rows, variables as columns, and data information through histograms in the cells; and
- A DR similarity map for analyzing data instances from the perspective of the discovered patterns, composing an analytical cycle that goes from data to patterns and from patterns to data.

The remainder of the paper is organized as follows. Section 2 covers the literature in classification model visualization for descriptive analysis, discussing the current limitations and positioning our solution. Section 3 details our proposed approach, showing how JEPs are extracted, aggregated, and visualized. Section 4 presents two different use cases explaining how to use our solution for data explanation. Finally, Section 5 lists our limitations and Section 6 outlines our conclusions and future work.

2 RELATED WORK

There is extensive literature on the visual analysis of multivariate data (see [3] for a survey), granting a myriad of methods and systems, including Dimension Reduction (DR) strategies [20], [21], [22], [23], [24], [25], [26], clustering approaches [27], [28], [29], statistical correlation methods [30], [31], [32], [33], [34], to mention a few. Typically, these methods and systems focus on representing data to allow users to visually capture and interpret data patterns. In contrast, VAX uses classification models to capture data patterns that are then displayed through a visual representation, leveraging the capability of classifiers to help users discover and understand such patterns.

Visual representations of classification models have been typically used to support building [35], [36], [37], comparing [38], [39], and explaining/predicting prediction models [10], [11], [12], [14], [40], [41], [42]. However, in the visualization literature, the idea of using classification models specifically as descriptive tools instead of predictive engines, using them as proxies to understand or describe multivariate data, is relatively new. Here, we divide the existing approaches into two groups: (1) model specific [4], [9], [43], [44]; and (2) logic rules and emerging patterns [18], [19], [45], [46].

2.1 Model Specific

Classification models combined with visual representations have been used for descriptive purposes. Explainers [9] is one example. It employs DR layouts created using linear functions from Support Vector Machines (SVM) models. Explainers allows data analysis using such linear functions to interpret DR layout patterns. It is an inspiring and pioneering approach but is limited to present patterns resulting from linear combinations of up to three variables, missing patterns in more complex non-linear associations. RankAxis [44] also employs linear SVM models, ranking data instances and obtaining attribute weights that serve as input to build DR layouts. Despite being a promising technique, RankAxis may present disparities in ranking and DR results and is also limited in providing linear approximations for non-linear data relationships. In our approach, the patterns extracted can involve more than three variables and represent non-linear relationships among instances and their classes. To allow that, we use the concept of Jumping Emerging Patterns (JEPs) [18], [45], [46] to extract patterns with different variable combinations. Furthermore, patterns are also used to create DR layouts, enabling analysis involving patterns and data instances.

Another example is Visual Neural Decomposition (VND) [4]. VND enables multivariate data explanation by visually presenting Artificial Neural Network hidden node weights through stacked bars to depict the relations between variable ranges and classes (e.g., class A with a threshold probability). The nodes are organized in cards containing variables ordered by importance to a node. Although VND can show non-linear relationships among instances and a particular class, the captured patterns’ complexity is bounded by the simple neural network architecture (one hidden layer) employed to allow interpretability. Also, their class-specific visualizations can make the analysis of multiple classes difficult. A single hidden layer network has also been used as a customized classification loss function for creating DR layouts [43]. Such enhanced layouts can reveal clusters and outliers, where data instances selections are explained using SHAP [41], so lacking the ability to convey information involving multiple variables combinations. In our approach, since we use inherent interpretable logic statements from JEPs and a matrix visual representation, complex multiple class relations can be discovered and concisely displayed. Other positive aspects of our approach are capturing patterns with maximum confidence (and statistical significance), so that a specific class pattern does not support instances from another class (not limited to a class probability threshold), and extracting patterns that involve multiple variable combinations.

The above-mentioned approaches are inspiring but lack the explanatory power of JEPs [18], [19], [45], [46], where one of the main goals is to obtain patterns for data explanation.

2.2 Logic Rules and Emerging Patterns

JEPs consist of class-associated relational statements among variables, providing class differentiation and emerging trends [19], [45], [46], [47]. VAX extracts JEPs using random DTs inducted based on the concept of Random Forests (RFs) [18], [48], [49], obtaining many diversified and expressive patterns [18], [19]. These patterns are then post-processed by a selection and aggregation strategy to reduce their number before visualization. JEPs are a particular case of a
Emerging Patterns (EPs) 

Instances group, like a set of points relative closed to each other on a scatter plot. 

The strengths of model explanations are explained through visual metaphors, representing a meaningful composition of two or more insights types.

Method

Variables values distribution, such as histogram plots.

I2 Distribution. Variables values distribution, such as histogram plots.

I3 Cluster. Instances group, like a set of points relative closed to each other on a scatter plot.

I4 Outlier. Particular instance with distinct variables values to the distribution, such as an instance relative apart from other instances in a scatter plot.

I5 Compound fact. Meaningful composition of two or more insights types.

More general concept called Emerging Patterns (EPs) [19] where the confidence is maximum; the mined patterns are class-exclusive, not supporting instances of different classes.

Visualization is a powerful component when analyzing a phenomenon through data patterns [18], [46]. In this context, two crucial aspects should be supported, the ability to check patterns content (e.g., data distribution) and to support multi-class interpretation (more than two classes) [46]. Some approaches address the visualization of pattern properties (e.g., support) through visual markers [46] but lack patterns’ content representation. Visualizing Subgroup Distribution (VSD) [50] presents patterns as line plots for continuous variables and binary class problems (two classes). Despite being intuitive, VSD [50] is not suitable for multi-class domains, and multiple variables visualization is an issue [46]. Our approach satisfies both content and multi-class requirements, presenting patterns’ content using histograms and classes mapped to categorical colors.

JEPs can be seen as descriptive logic rules [19], [46], and disjoint and consistent logic rules are deemed interpretable [51], [52], [53], therefore popular in VA solutions for model interpretability [10], [11], [40], [52], [54]. Such solutions usually support global and local model interpretation to explain the model itself and its decisions [55]. ExMatrix [11] and RuleMatrix [10] are two examples of such techniques. ExMatrix focuses on supporting the interpretation of RF models, displaying rules extracted from the random Decision Trees (DT) in a matrix visualization. Although ExMatrix has shown to be a good solution for analyzing RF models, no information is conveyed about the data, so limited if the goal is to understand data patterns. RuleMatrix, a matrix visualization to depict logic rules from surrogate models, presents some information about the data. However, it does not support the analysis of data instances in micro and macro contexts. It presents information through global histograms but does not allow the interpretation of the data represented by the logic rules or individual data instances. Despite great solutions, ExMatrix and RuleMatrix are model-centered, where rules are primarily used in explaining models, lacking essential capabilities if the goal is to interpret multivariate data. Our solution also leverages a matrix metaphor, but our goal is to support explanations of multivariate data, not models. It focuses on descriptive rules/patterns, supporting the explanation of groups (e.g., clusters) or distinct data instances (e.g., outliers), offering different visual representations to navigate between patterns and instances.

Finally, another VA approach that could be potentially considered for description tasks is iForest [12]. Similar to ExMatrix, iForest was developed for the interpretation of RF models. However, unlike ExMatrix, it only supports local interpretability, that is, the explanation of individual instances predictions. For descriptive tasks, in which the goal is the interpretation of (complete) datasets, global interpretability needs to be supported so that it is possible to understand data through the average behavior of a model.

3 Method

This section presents multiVariate dAta eXplanation (VAX), a new multidimensional data explanation approach that combines JEPs visualization and DR layouts to support data pattern discovery and interpretation. To reach our general objective, the proposed visual representations implement five automated data insights presented by Law et al. [56] (Table 1).

By implementing these insight types, VAX supports the needed elements for exploratory tasks inside the visual analytics process [1], [2]. We implement I1 - Visual motifs and I2 - Distribution using a matrix-like visual metaphor adopting matrix visualization guidelines [57], [58]. Since descriptive patterns (JEPs) are our objects of analysis, we represent them as rows (I1), data variables as columns, and fill matrix cells with global and local histograms (I2). This arrangement combines the strengths of model explanation approaches [10], [11], improving them for data interpretation instead of model analysis. The insight types I3 - Cluster and I4 - Outlier are implemented using a DR technique, creating a similarity map as viewed through the JEPs lens, where clusters (I3) and outliers (I4) can be observed. Finally, I5 - Compound fact is reached through combining the different proposed visual representations where clusters and outliers (I3 and I4) can be explained by visualizing the patterns along with variables distributions (I1 and I2).

The proposed VAX pipeline is presented in Fig. 1. Initially, random Decision Trees (DTs) are inducted considering the entire dataset (I). JEPs are extracted from the DTs, and the more relevant patterns are selected and aggregated following a well-defined strategy (2). The resulting aggregated patterns are then visualized using a matrix metaphor, and the similarity relationships among data instances are represented using a DR layout with the similarity among instances defined by the patterns (3). This allows users to explore how patterns are related to instances and how instances are connected with patterns. Next, we detail how the JEPs matrix and the similarity map are created.

| Insight Type | Description |
|--------------|-------------|
| I1 | Visual motifs. Unique/special/specific patterns, being but not only custom visual metaphors, representing a particular notion/structure on data. |
| I2 | Distribution. Variables values distribution, such as histogram plots. |
| I3 | Cluster. Instances group, like a set of points relative closed to each other on a scatter plot. |
| I4 | Outlier. Particular instance with distinct variables values to the distribution, such as an instance relative apart from other instances in a scatter plot. |
| I5 | Compound fact. Meaningful composition of two or more insights types. |
3.1 Jumping Emerging Patterns Matrix

Before explaining the construction of the JEPs matrix, we first introduce and formalize the concept of Emerging Patterns (EPs) [18], [19], [45] and Jumping Emerging Patterns (JEPs) [15], [19] and how they are extracted from datasets.

3.1.1 Definitions

An EP is a descriptive rule combining dataset variables using relational operators to express emerging relationships (i.e., trends and differentiating behavior) in data [19], [46]. In formal terms, given a dataset $X = \{x_1, \ldots, x_N\}$ with each instance represented by $V = \{v_1, \ldots, v_M\}$ variables. A pattern $p$ is a conjunction of selectors (logical complex), each one defining a relational statement in the form of $v_i \otimes S_i$, where $S_i$ consists of one or more possible values (or a range) for the variable $v_i \in V$ and the relational operator $\otimes$ is $=, \neq, <, >, \leq, \geq$, or $\in, \notin$, $\leq$ [19], [59], that is

$$p = [v_1 \otimes S_1] \land [v_2 \otimes S_2] \land \ldots \land [v_M \otimes S_M] \quad (1)$$

An instance $x_i \in X$ (a.k.a. transaction, sample, example, or item) is said to be supported by a pattern $p$ if it satisfies all selectors of $p$ [19], [45], [59]. Moreover, given $X_a \subset X$ and $X_b \subset X$, with $X_a \cap X_b = \emptyset$, two disjoint partitions (i.e., sub-datasets) of $X$, a pattern $p$ is considered an EP in $X_b$ given $X_a$ if the percentage of instances supported by $p$ in $X_b$ divided by the percentage of instances supported in $X_a$ is larger than a threshold ($\geq 1$), that is, if the pattern $p$ is more “valid” for the instances in $X_b$ than in $X_a$. This ratio is called Growth Rate [19], [45], [59], and it is defined by

$$GR(X_a, X_b, p) = \begin{cases} 0 & \text{if } Supp(X_a, p) = 0 \land Supp(X_b, p) = 0 \\ \infty & \text{if } Supp(X_a, p) = 0 \land Supp(X_b, p) \neq 0 \\ \frac{Supp(X_b, p)}{Supp(X_a, p)} & \text{Otherwise} \end{cases} \quad (2)$$

where $Supp(X_i, p)$ is the support of $p$ in the dataset $X_i \subset X$, given by

$$Supp(X_i, p) = \frac{count(X_i, p)}{|X_i|} \quad (3)$$

with $count(X_i, p)$ the number of instances from $X_i$ supported by $p$, and $|X_i|$ the cardinality of $X_i$ [19], [45], [46]. The EP core idea is to discover patterns whose support increases (Growth Rate) from a partition $X_a$ to a partition $X_b$ [19], [45], [46]. For class-labeled datasets, where each instance $x_i \in X$ is associated with a class $y_i \in C = \{c_1, \ldots, c_J \geq 2\}$, the partitions can be composed by instances associated with the same label [19]. In this way, for binary-class datasets ($J = 2$), $X_a$ will contain instances of one class, while $X_b$ will contain the other class instances. For multi-class datasets ($J > 2$), an One-versus-All strategy can be used [19], and $X_b$ will contain the instances of a particular class, and $X_a$ will contain the instances of all remaining classes. In the remainder of this paper, the terms partition or class have the same meaning and will be used interchangeably.

For illustration, we use the synthetic dataset $X_S$ presented in Fig. 2, covering different scenarios. $X_S$ is composed of 500 instances, described by 2 real variables $(var_1, var_2)$, and equally distributed among 5 classes ($J = 5$), A, B, C, D, and E (100 instances per class). Three clusters can be spotted in $X_S$, where circles denote instances and color reflects their

![Fig. 2. The synthetic dataset $X_S$ used to illustrate our approach. It covers different scenarios where EPs can be used to describe a dataset, including instances belonging to strongly mixed (A and B), adjacent (C and D), and well-separated classes (E).](image-url)
class. Instances of class A (blue) and B (orange) strongly overlap, instances of class C (green) and D (brown) are adjacent, and instances of class E (yellow) are completely isolated.

Considering $X_A \subset X_S$ the partition containing all instances of class A (the same applying for the other classes B, C, D, and E), the pattern $p_{ex} = \{var_1 \in [96, 120]\} \land \{var_2 \in [78, 95]\}$ is an EP for class E instances (partition $X_E$) given the other classes. The pattern $p_{ex}$ Growth Rate equals to $GR((X_A \cup X_B \cup X_C \cup X_D) \setminus E, p_{ex}) = \infty$, since $Supp((X_E, p_{ex}) = 1.0$ and $Supp((X_A \cup X_B \cup X_C \cup X_D), p_{ex}) = 0$. The pattern $p_{ex}$ supports all class E instances $(count((X_E, p_{ex}) = |X_E| = 100)$, but none instances from classes A, B, C, or D (partitions $X_A, X_B, X_C,$ or $X_D$) as $count((X_A \cup X_B \cup X_C \cup X_D), p_{ex}) = 0$ and $|\{X_A \cup X_B \cup X_C \cup X_D\}| = 400$.

There are different types of EPs based on the relationships between variables [18], [19]. In this paper, we are interested in the so-called Jumping Emerging Patterns (JEPs) given their discriminative power among classes [15], [19]. A JEP is an EP observed in only one class given all other classes, defined by

**Definition 1 (Jumping Emerging Pattern – JEP).** Let $X_A \subset X$ and $X_B \subset X$ be two disjoint partitions of $X$, that is, $X_A \cap X_B = \emptyset$, and $p \in P$ be an Emerging Pattern (EP) of $X_B$ given $X_A$, a Jumping Emerging Pattern (JEP) is an EP with Growth Rate equals to $\infty$, that is, $\{p \in P|GR(X_A, X_B, p) = \infty\}$.

It is worth mentioning that JEPs also present a maximum value (1.0) of confidence (pattern precision) [19], defined by

$$Conf(X_A, X_B, p) = \frac{count(X_A, p)}{count(X_A, p) + count(X_B, p)}$$

Pattern $p_{ex} = \{var_1 \in [96, 120]\} \land \{var_2 \in [78, 95]\}$ is a JEP, since $GR((X_A \cup X_B \cup X_C \cup X_D) \setminus X_E, p_{ex}) = \infty$. Moreover, it presents confidence equals to $Conf((X_A \cup X_B \cup X_C \cup X_D) \setminus X_E, p_{ex}) = 1.0$, since $count((X_E, p_{ex}) = 100$ and $count((X_A \cup X_B \cup X_C \cup X_D), p_{ex}) = 0$. From this point forward, we use the synthetic dataset $X_E$ to help explain our approach, including JEPs extraction, selection, aggregation, and visualization.

### 3.1.2 Extraction

Mining EPs is an NP-Hard problem resulting from the exponential number of candidate patterns that are generated as the number of variables grows [18], [19], [60], [61]. In this context, EPs can be mined using different heuristics [18], [19], [46]. One popular approach is to use Decision Tree (DT) models [6], [62]. The idea is to build varied DTs using diverse (random) factors and then extract a pattern from each decision path (the path from the root to a leaf node) [18], [19], [45], [46]. One of the most powerful techniques to mine EPs, and the method we use in this paper, is the Random Forest miner (RFm) [18], [48], [49]. RFm was developed based on the concept of Random Forests (RFs) [16], [17], and it has been proved to be a reasonable strategy for obtaining diversified patterns [48].

Shortly, let $V$ be the variables of a class-labeled dataset $X$. The RFm method builds $k$ unpruned DTs from $X$ selecting a random subset of variables at each internal node creation, where the variables subset size is $\log_2 |V|$. Unlike the RF method proposed in [16], the bagging process (random selection of training instances) [63] is not used in RFm, thus avoiding hidden dependencies among patterns and data instances [18], [48].

Algorithm 1 presents the extraction process, where a class-labeled dataset $X$ and the number of trees $k$ are input parameters. Function DecisionTree() creates a DT model [6], [16], [62], [63] while ExtractPatterns() extracts a pattern $p$ for each decision path (root to leaf node) of a given DT [18], [19], [46]. Since our approach focuses on descriptive analysis, we set $k$ as the minimum number of trees containing enough patterns to represent all data instances (this is later discussed in Section 4.1). Also, the entire dataset $X$ is employed to build the random DT models. Unlike predictive analysis, which focuses on creating generic models and usually splits a dataset into training and testing subsets [63], descriptive tasks are specific to a given dataset. They do not intend to be generic outside the scope of the data under analysis or rather offer explanations of the phenomena observed in a single dataset [4], [9].

**Algorithm 1. EPs Extraction Using Random DTs**

**Input:** $X$ - Class-labeled Dataset

$k$ - Number of Trees

**Output:** $P$ - Emerging Patterns

$P \leftarrow \emptyset$

for $1$ to $k$

$DT \leftarrow \text{DecisionTree}(Dataset = X, Subset Size = \log_2 k)$;

$P \leftarrow P \cup \text{ExtractPatterns}(DT)$;

end

One relevant factor is that the Fisher Exact Test (FET) can be applied to compute statistical significance per pattern [46], [64], [65]. Values above the significance p-value (usually 0.05) imply the null-hypothesis acceptance that there is no association between the pattern and a class [65]. Despite the ability to provide diversified patterns [18], [48], representing assorted class and variable-range combinations, random DT models usually extract redundant patterns [48], [49]. To address this problem, we present a novel pattern aggregation strategy to avoid overlapping users with repetitive information. It is next described.

### 3.1.3 Selection and Aggregation

Once all patterns are extracted from the $k$ random DT models (Algorithm 1), originating a set of patterns $P$, these must be selected and aggregated. Inspired by the selection method presented in [49], we analyze patterns in $P$ from the highest to the lowest support, choosing a subset of patterns with maximum confidence that does not support data instances supported by other patterns. This way, each instance is represented by only one JEP ($GR = \infty$). The main difference between our strategy and the one presented in [49] is that we do not discard supplementary patterns (different patterns that support the same instances). The supplementary patterns are aggregated, not losing their information.

The selection and aggregation process is summarized in Algorithm 2. Using an iterative greedy process applied to the set of patterns $P$ (resulted from Algorithm 1) ordered by
deciding support, we select as a candidate the first pattern (highest support) \( p_{\text{candidate}} \). If the confidence of \( p_{\text{candidate}} \) equals 1.0, it is selected as \( p_{\text{pivot}} \) and aggregated with the subset of patterns \( P' \subset P \) that support the same instances.

**Algorithm 2. JEPs Selection and Aggregation**

**Input:** \( P \) - Emerging Patterns  
**Output:** \( P \) - Jumping Emerging Patterns

\[
\text{SI} \leftarrow \emptyset; \\
P \leftarrow \text{OrderByDecreasingSupport}(\text{P}); \\
\text{while } p_{\text{candidate}} \leftarrow \text{Next}(\text{P}) \text{ do} \\
\quad \text{if Confidence}(p_{\text{candidate}}) = 1.0 \text{ and} \\
\quad \quad \text{SupportedInstances}(p_{\text{candidate}}) \notin \text{SI} \text{ then} \\
\quad \quad \quad \text{pivot} \leftarrow p_{\text{candidate}}; \\
\quad \quad \quad \text{SI} \leftarrow \text{SI} \cup \text{SupportedInstances}(\text{pivot}); \\
\quad \quad \quad \text{foreach } p_a \in \text{P} \text{ do} \\
\quad \quad \quad \quad \text{if SupportedInstances}(p_{\text{pivot}}) = \\
\quad \quad \quad \quad \quad \text{SupportedInstances}(p_a) \text{ then} \\
\quad \quad \quad \quad \quad p_{\text{pivot}} \leftarrow \text{Aggregate}(p_{\text{pivot}}, p_a); \\
\quad \quad \quad \quad \text{Remove}(p_a, P); \\
\quad \quad \quad \text{end} \\
\quad \quad \quad \text{else} \\
\quad \quad \quad \quad \text{Remove}(p_{\text{candidate}}, P); \\
\quad \quad \text{end} \\
\text{end} \\
\text{end}
\]

For the aggregation procedure, without loss of generality, we assume that the patterns in \( P \) have selectors in the form \( v \in S \), with \( S = [a, b] \) and \( a \) and \( b \in \mathbb{R} \). In this procedure, given a pattern \( p_a \in P \) supporting the same instances of \( p_{\text{pivot}} \), all patterns selectors for the same variable are aggregated as the intersection of the two defined real sets (\( S^{p_{\text{pivot}}} \) and \( S^{p_a} \)). So, supposing \( p_{\text{pivot}} \) and \( p_a \) having selectors for a variable \( v_i \in V \), the aggregation is \( S^{p_{\text{pivot}}} \cap S^{p_a} \). For the case where the selector for a variable \( v_i \in V \) is found in only one pattern \( p_{\text{pivot}} \) or \( p_a \), the missing real set is \( \min(X^{v_i}), \min(X^{v_i}) \), being \( \min(X^{v_i}) \) and \( \max(X^{v_i}) \) the minimum and maximum values for the variable \( v_i \) of \( X \). After the aggregation of the patterns in \( P' \) with \( p_{\text{pivot}} \), the patterns in \( P' \) are removed from \( P \). In this way, complementary patterns turn into a single pattern.

The next candidate pattern \( p_{\text{candidate}} \in P \) is then analyzed. If \( p_{\text{candidate}} \) confidence value differs from 1.0, \( p_{\text{candidate}} \) is discarded, since it is not a JEP. However, if the confidence equals 1.0, the \( p_{\text{candidate}} \) supported instances are investigated. If \( p_{\text{candidate}} \) supports at least one instance already supported by the previously selected and aggregated patterns, \( p_{\text{candidate}} \) is discarded, removing a redundant pattern. If \( p_{\text{candidate}} \) supports only instances not supported by the previously selected and aggregated patterns, it is selected as \( p_{\text{pivot}} \) and aggregated with the set of patterns \( P' \) that support the same instances, removing from \( P \) this set of patterns after aggregation. This process is repeated until the end of \( P \) is reached, selecting and aggregating complementary patterns and discarding the redundant ones.

To illustrate the selection and aggregation process, consider the execution of Algorithm 1 to create \( P \) taking the synthetic dataset \( X_S \) and the number of trees \( k = 128 \) as inputs, resulting in 13,975 patterns. After Algorithm 2 is applied for selection and aggregation, 67 aggregated JEPs are produced, supporting all instances from the synthetic dataset \( X_S \). Fig. 3 presents how these 67 patterns segment the bi-dimensional space where \( X_S \) is contained. For the instances of class \( E \), the isolated groups of points, only one JEP is used to represent the entire class. For the adjacent classes \( C \) and \( D \), more JEPs are necessary since the overlapped border needs more patterns to describe it. For the two completed overlapped classes \( A \) and \( B \), several JEPs are used given the complexity to explain the differences between these classes.

In terms of computational complexity, the entire process of creating the aggregated JEPs can be split into two major components, extraction (Algorithm 1) and selection and aggregation (Algorithm 2). Considering the simple algorithm for DT induction, which orders the dataset variables (dimensions) before splitting the tree nodes, the complexity of inducting one tree is \( O(M \times \log N) \), where \( M \) is the number of variables and \( N \) instances. If \( k \) trees are created, the complexity of extraction is \( O(k \times M \times \log N) \). Given the employed greedy approach, the selection and aggregation complexity is \( O(|P|^2) \), where \( |P| \) is the number of extracted patterns. Since, in the worst-case scenario, the upper limit is to have \( N \) leaves in a DT, the maximum number of extracted patterns is \( k \times N \), and the complexity of the selection and aggregation process is \( O(k^2 \times N^2) \).

This selection and aggregation process results in meaningful aggregated JEPs supporting all instances of a dataset, explaining it through high support patterns, discarding the redundant ones, and using low support patterns to explain outliers.

### 3.1.4 JEPs Matrix Visualization

Once the JEPs are selected and aggregated, we use a matrix visual metaphor for presentation (insight 11). Instead of a matrix, we have also considered a graph metaphor, as used to visualize association rules [66], [67]. However, graphs do not scale well and cannot support comparisons of patterns considering the variables. Moreover, user tests have shown that rules organized into tables are easier to understand if compared to node-link arrangements [68]. Fig. 4 presents an example of our matrix representation. It places patterns as rows (1) and variables as columns (2). Classes are mapped to categorical colors. Matrix cells (3) present (local) variables’ histograms [69] (insight 12) showing the variables’
distributions of the instances supported by a particular pattern. Similarly, global histograms (insight I2) for each class are laid out on the top of the matrix (4). The number of bins is determined based on the Freedman-Diaconis rule [70], [71], although it can also be freely defined. We have also considered using violin plots instead of histograms, but they occupy more visual space, decreasing visual scalability. If a pattern (row) does not have a selector for a variable (column), the respective cell is left empty (no histogram) by default. A local histogram can also be displayed if the user wants. The patterns’ supports is mapped to a column on the matrix left side (5). The cumulative dataset coverage is mapped to a column on the matrix left side (6), representing the cumulative percentage of instances in a dataset captured by the patterns considering the matrix order (top to bottom). The variable importance is outlined above the global histograms, with the variables’ names at the bottom (7). Pattern support, cumulative coverage, and variable importance are mapped to brightness (linear grayscale) and size (rectangle width). The FET p-value for each pattern is displayed in a column to the matrix right side (8) using a binary color scheme, green for values below 0.05 (statistically significant) and purple otherwise.

We explore two aspects to help create more meaningful matrix-based visual representations for JEPs: filtering and ordering. Filtering is a common strategy when several patterns are available [18], [19]. JEPs can be filtered by support, data coverage, class, and supported instance(s) in our approach. Rows’ and columns’ order also plays an essential role in matrix visualization [57], [58]. Our implementation supports different ordering schemes to enhance users’ analytical capabilities. JEPs (rows) can be ordered by support, class, and class & support. Furthermore, variables (columns) can be ordered by importance, which is calculated based on [72] as follows

$$\text{Imp}(v_i, P) = \sum_{j=1}^{p_i} \left\{ \begin{array}{ll} \text{Supp}(X_{v_i}p_j) & \text{if } S_i \in p_j \\ 0 & \text{otherwise} \end{array} \right. $$

where the importance of a variable $v_i$ given a set of JEPs $P$ is the summation of the support of each pattern $p \in P$ having a selector for $v_i$. After calculating the importance for all $v \in V$, they are normalized between 0 and 1.

Twelve JEPs are displayed in Fig. 5 for the synthetic dataset $X_S$ out of the 67 previously discussed. The 67 patterns were filtered by data coverage ($p_{24}$ to $p_{57}$) and supported instances ($p_5$). The 12 resulting patterns are ordered by support. The first row contains the pattern $p_{24}$, where support is maximum, indicated by the filled rectangle on the support column. The cumulative coverage reflects the value of 0.20 (20%), as pattern $p_{24}$ supports all 100 class $E$ instances out of 500 dataset instances. Once pattern $p_{24}$ has selectors on variables $var_1$ and $var_2$, its cells show histograms for these variables considering only the instances (local) supported by $p_{24}$. For this case, global and local histograms are equal since $p_{24}$ supports all class $E$ instances. The second row represents the pattern $p_{34}$, with support equal to 0.87, meaning that this pattern supports 87% of class $C$ instances (87 instances out 100). The cumulative coverage indicates the value of 0.37, implying that patterns $p_{34}$ and $p_{31}$ together cover 37% of the instances in $X_S$. The last pattern $p_5$ has significantly lower support (0.01), not being statistically significant (purple for FET p-value). All the remaining 11 patterns are otherwise significant. From the first pattern $p_{24}$ to the last $p_5$, about 69% of the dataset $X_S$ is represented (indicated by the cumulative coverage of 0.69).

It is possible to observe in the JEPs of Fig. 5 variables values combinations associated with classes (insights I1 and I2). Class $E$ is represented by the strong pattern $p_{34}$, unique and supporting 100% of its instances (support value of 1.0). Instances from this class have median values for the variable $var_1$ and low for the variable $var_2$, as represented by the local histograms. Patterns $p_{34}$ and $p_{31}$ complete the list of

Fig. 4. The matrix-like visual metaphor. 1: JEPs are displayed as rows. 2: Variables are arranged as columns. 3: Cells show local normalized histograms. 4: Global histograms (one row per class) are placed on the top, also normalized. 5: Pattern support. 6: Cumulative coverage assuming the matrix order (top to bottom). 7: Variable importance. Both pattern support, cumulative coverage, and variable importance are mapped to size and brightness (grayscale). 8: FET (Fisher Exact Test) significance value colored as green (statistically significant) or purple (not significant).

Fig. 5. Twelve aggregated JEPs filtered out of the 67 obtained for the synthetic dataset $X_S$. Variables values combinations and classes associations are presented, such as the strong pattern $p_{34}$, which supports all instances of class $E$ and indicates that these instances have median values for variable $var_1$ and low values for variable $var_2$. These 12 JEPs represent about 69% of the dataset $X_S$ (cumulative support), where only the low support pattern $p_5$ is not statistically significant (purple for FET p-value).
high support patterns. The $p_{34}$ supports 87% of class $C$ instances, having low values for variable $v_{1}$ and high for variable $v_{2}$. Pattern $p_{12}$ supports 80% of class $D$ instances, with low values for variable $v_{3}$ and median for variable $v_{2}$. These three patterns ($p_{34}$, $p_{31}$, and $p_{12}$) cover more than half (53%, cumulative coverage of 0.53) of the dataset. Pattern $p_{1}$ refers to a small subset of class $D$ instances (14%), differing from pattern $p_{32}$ on variable $v_{2}$ (higher values). The patterns providing the highest support for classes $A$ and $B$ ($p_{60}$ and $p_{10}$) represent only 11% of each class. Classes $A$ and $B$ strongly overlap (Fig. 2), so high support patterns for these classes are not attainable. Pattern $p_{1}$ represents a small subset of class $C$ instances (8%), differing from pattern $p_{34}$ on variable $v_{2}$ (lower values). For class $D$, the pattern $p_{5}$ supports only one instance, representing an outlier compared to $p_{32}$ and $p_{1}$ on variable $v_{2}$ (higher value).

3.2 Similarity Map

To support clusters and outliers analyses (insights $\text{I3}$ and $\text{I4}$) from a class-labeled dataset $X$, we use DR layouts leveraging the space extension approach proposed in [73]. Our primary goal in using such an extension is to provide a straightforward way to navigate between patterns and data instances as “viewed” or represented by the patterns (insight $\text{I5}$), incorporating the JEPs’ perspectives into the obtained layout. If only the original data is used, the pattern information is ignored. Using such an extension, it is easier to see the groups of instances that are well described by a pattern and the instances that are not. In general lines, a new variable is created for each data variable where the value of a given instance is calculated considering the pattern supporting it. Hence, instances supported by the same pattern receive the same value derived from the original variable. Formally, let $X \in \mathbb{R}^{N \times M}$ be the dataset, where $N = |X|$ is the number of instances in $X$ and $M = |V|$ the number of variables, the key idea is to create an extended dataset $X' \in \mathbb{R}^{N \times 2M}$ as

$$X' = [X, \bar{X}]$$

with $\bar{X} \in \mathbb{R}^{N \times M}$ composed by the centroids (mean values) of disjoint groups of instances. For the instances subset $X_h \subset X$ belonging to a group $h$, the extension equals to

$$\bar{x} = \frac{1}{|X_h|} \sum_{x \in X_h} x$$

Since each instance $x_i \in X$ is supported by only one aggregated JEP, and we want to represent the relationships between instances from the patterns’ perspective (insight $\text{I3}$), we use JEPs to define the groups of instances to extend $X$. Hence, $X'$ incorporates a new set of variables $\bar{V}$, representing the mean of the instances subset supported by each pattern.

To help users control the influence of the extension into the similarity relationships, a real parameter $\lambda \in [0,1]$ is used to control the gradual transition [73] between the dataset $X$ and the extended part $\bar{X}$ with $X_{\text{weight}} = X'W_\lambda$, where matrix $W_\lambda \in \mathbb{R}^{2M \times 2M}$ is defined by

$$W_\lambda = \begin{pmatrix} (1-\lambda)I & 0 \\ 0 & \lambda I \end{pmatrix}$$

After the normalization of $X_{\text{weight}}$ (z-score), a DR technique is used to create the similarity map. By varying the parameter $\lambda$, it is possible to create maps representing the distance relationships considering only the original dataset $X$ ($\lambda = 0$), only the extended part $\bar{X}$ ($\lambda = 1$), or in between ($0 < \lambda < 1$), defining the similarity among instances using a combination of data and patterns information (in Section 4.1 we present a heuristic to determine $\lambda$). In this paper, we use the Multiplicative Scaled (MDS) technique [74] to project $X_{\text{weight}}$ since it faithfully preserves the global relationships between groups of instances. Other faster global techniques could also be used, for instance, the Landmarks MDS [75] to speed up this process, especially for larger datasets.

For creating the extended version of the synthetic dataset $X_S$, the 67 JEPs previously discussed are used; that is, 67 groups are considered in the extension process [73]. The result synthetic dataset extension $X_S$ has then four variables $V' = \{v_{1}, v_{2}, v_{3}, v_{4}\}$, where $v_{1}$ and $v_{2}$ are the mean values of $v_{1}$ and $v_{2}$ from each instances subset supported by the 67 patterns. Fig. 6 presents instances maps for $X_{\text{weight}} = X_SW_\lambda$ with $\lambda = 0.70$. Two versions are present, one with colors given by the original dataset classes and another colored according to some JEPs (with gray indicating instances not supported by any pattern of interest). Three highly dense clusters can be identified (insight $\text{I3}$), one for each high supported pattern ($p_{24}$, $p_{31}$, and $p_{32}$). By inspecting the JEPs matrix (Fig. 5), it is possible to observe (insight $\text{I5}$) that the one formed by pattern $p_{24}$ (pink) for class $E$ (pink) contains instances with high values for
variable \( var_1 \) and low for variable \( var_2 \) (insight 12) compared to those established by patterns \( p_{31} \) (emerald) and \( p_{32} \) (red). The difference between these two latter clusters is found in variable \( var_2 \) (insight 12), where class \( C \) instances \( (p_{31}) \) have higher values than class \( D \) instances \( (p_{32}) \). Furthermore, an outlier for class \( D \) (peach) can also be noticed (insight 14). It resulted from the low support pattern \( p_5 \) (insight 11), with a low value for variable \( var_1 \) and a high for variable \( var_2 \) (insight 12) in contrast (exception) to the class \( D \) cluster (red) settled by pattern \( p_{32} \) (insight 15).

Similarity maps can be used to filter the JEPs matrix visualization, allowing users to select instances of interest, and obtaining the patterns supporting them. Similarity maps can play an essential role when dealing with a high number of patterns.

4 Use Cases

This section presents two use cases showing how to use VAX for multivariate exploratory analysis. The two use cases involve datasets with and without ground truth labels (classes). VAX is implemented using Python and is also available as a code package. The source code for the two use cases is accessible as Python notebook pages.

4.1 Use Case I—US Presidential Election

The first use case involves the analysis of the 2016 US presidential election using the dataset from a survey conducted in 2018 by the independent social research organization NORC of the University of Chicago \([4], [76]\). It contains the participants’ answers about different political and societal aspects of the United States and what candidate they supported for the election (Donald Trump or Hillary Clinton). Following the steps described in \([4]\), the nationally representative subset is used, resulting in 4,913 data instances (registered voters) and 67 variables. After removing missing values and keeping only data instances with a revealed vote (Donald Trump or Hillary Clinton), the number of variables is reduced to 60 and instances to 3,754, 43.3% pro-Donald Trump (1,625), and 56.7% pro-Hillary Clinton (2,129).

The first step in the VAX pipeline is to induct the random DTs to extract the patterns (Algorithm 1). In this process, the only parameter that needs to be set is the number of trees. To reduce pattern duplication, we use the minimum number of trees that result in patterns representing all data instances (after Algorithm 2), starting with a small number of DTs and successively inducting more trees until the data coverage is 100%. Fig. 7a presents the obtained data coverage for different numbers of trees, from \( 2^1 = 2 \) to \( 2^{14} = 16,384 \). In this experiment, 2,048 is the minimum number of DTs capable of providing enough diversified patterns resulting in 100% of coverage after selection and aggregation. After extraction, from the 856,900 discovered patterns, 255 are selected, 3,042 aggregated, and 853,603 discarded (Algorithm 2).

For the similarity map creation, \( \lambda \) must be defined for the dataset extension \([73]\). In this process, we propose to select \( \lambda \) by analyzing the Kruskal Stress \([74]\) and Silhouette Coefficient \([77]\). The idea is to find the value that maximizes distance preservation (stress) and group cohesion/separation (silhouette) so that groups defined by patterns can be spotted while minimizing the distortion of distances conveyed by the original data. To make our analysis more straightforward, we normalize and invert the original Silhouette Coefficient, using \( sc' = 1 - \frac{sc}{2} \), where \( sc \) is the original silhouette, resulting in values in \( sc' \in [0,1] \) (the lower, the better). Fig. 7b shows the Kruskal Stress \( st \in [0,1] \) (the lower, the better) and the inverted Silhouette Coefficient \( sc' \) of different extended datasets (using the 255 selected and aggregated patterns) varying \( \lambda \). The best tradeoff between distance preservation and group cohesion/separation is found between 0.60 and 0.70. Hence, we empirically set \( \lambda = 0.65 \). Notice that \( \lambda \) is calculated after the logic rules inference, selection, and aggregation, so this process involves the projection of the extended datasets and the stress and silhouette computation only.

Fig. 8a presents 14 JEPs from the resulting 255, representing 70% of the dataset. The patterns are ordered by support and variables by importance. Despite the number of variables and complexity of the dataset, about half of its instances (48%) are described by only two patterns, \( p_{181} \) and \( p_{152} \) (first two rows). These are the highest support patterns for Hillary and Trump voters, respectively. Interestingly, it is possible to notice, given the differences in the local histograms of both patterns, that half of the Hillary voters (50% from pattern \( p_{181} \) and about half of Trump voters (45% from pattern

1. https://pypi.org/project/vaxm/
2. https://popolinneto.gitlab.io/vaxm/papers/2022/tvcg/methodology/
3. https://popolinneto.gitlab.io/vaxm/papers/2022/tvcg/usecasei/
4. https://popolinneto.gitlab.io/vaxm/papers/2022/tvcg/usecaseii/
(a) Fourteen JEPs filtered out the 255 selected and aggregated. Rows (JEPs) are ordered by support and columns (variables) by importance.

\( p_{152} \) differ fundamentally in three aspects (first three columns). While Hillary voters are in favor of expanding the Affordable Care Act – Obamacare (HEALTHLAW variable with majority answers “Expand the law”), against the wall with Mexico (IMMWALL variable with majority answers “Strongly oppose”), and believe that the Trump election campaign was coordinated with Russia (RUSSIA variable with majority answers “Yes”), Trump voters have opposite opinions (HEALTHLAW variable with majority answers “Repeal the law entirely”, IMMwall variable with majority answers “Strongly favor”, and “RUSSIA” with majority answers “No”). Notice that, although these three points compose the more generic differences between almost half of the voters, the differences are not evident for the other half, which can be observed on the global histograms for these three variables.

The support drops considerably for the subsequent patterns. However, revealing a more heterogeneous scenario for the other half of the voters. For instance, pattern \( p_{220} \) (third row) diverges from the strong pattern \( p_{152} \) (second row), both representing Trump voters, on political ideology (variable IDEO fourth column) with voters described by \( p_{220} \) considering themselves as moderate instead of conservative. Pattern \( p_{116} \) (fourth row) differs from the strong pattern \( p_{184} \) (first row), both describing Hillary voters, regarding the Affordable Care Act (variable HEALTHLAW first column), advocating to repeal the law at least in parts. The last pattern \( p_{150} \) supports only 2% of instances from its associated class. All the remaining 241 patterns have equal or less than 2% of support.

In addition to the JEPs matrix, two similarity maps are created to help the analysis, one in Fig. 8b considering the
original dataset ($\lambda = 0$) and another in Fig. 8c using an extended version ($\lambda = 0.65$). In both figures, the maps at the top are colored using voting information (Donald Trump or Hillary Clinton), and the maps at the bottom using pattern information, considering the 4 patterns with the highest support ($p_{1184}$, $p_{1522}$, $p_{2120}$, and $p_{1169}$). Instances colored in gray represent voters not supported by these 4 patterns. These high support patterns compose clusters in the similarity maps of Fig. 8c but not of Fig. 8b, indicating that adding JEPs information can reveal clusters in DR layouts that are otherwise not found using only the original dataset, which helps users navigate between patterns and data.

Finally, while the majority of the US electorate can be split into two major groups, one supporting Hillary and another Trump, a fascinating picture emerge. The usual simplistic division between two extremes is indeed much more complex in real life, with several subgroups diverging from the most general “common sense” within a group. This is clear in the similarity map where the major groups are spotted, but different patterns conveyed by the JEPs also define mixed subgroups.

4.2 Use Case II – World Happiness

The second use case involves the dataset of the 2019 World Happiness Report made available by the Sustainable Development Solutions Network (SDSN) [78]. This dataset presents the “Happiness Score” of 156 countries, indicating how happy their citizens perceive themselves, and six other variables capturing different perspectives. These variables are “GDP per capita”, “Social support”, “Healthy life expectancy”, “Freedom to make life choices”, “Generosity”, and “Perceptions of corruption”.

Since the world happiness dataset does not present class labels, we discretize the happiness score into three groups, indicating low, medium, and high happy countries. The idea is to verify the differences between countries based on perceived happiness levels regarding the other six variables. The happiness score is discretized in three equally sized bins. In other words, we are transforming a regression problem into a classification [79]. We have chosen discretization, but approaches such as clustering [63] and instances subset selection [80] are also suitable.

The “high happy” class encloses 42 countries with happiness scores in $[6.13, 7.76]$, the “medium happy” contains 79 countries with scores in $[4.49, 6.13]$, and the “low happy” class groups 35 countries with scores in $[2.85, 4.49]$. Since the variable “Happiness Score” is employed to define the classes (discretization), it is not used to create the patterns. The number of trees $k$ and $\lambda$ are set following the procedures specified in Section 4.1. From 2,893 extracted patterns ($k = 64$), 29 were selected, 252 aggregated, and 2,612 discarded. For the similarity map creation, we use $\lambda = 0.65$.

Out of the 29 JEPs, 6 were selected to create the JEPs matrix of Fig. 9a, one per class with the highest supports per class and 3 others representing the highlighted instances in Fig. 9b. These 6 JEPs represent 47% of the dataset. The patterns are ordered by support and variables by importance in the resulting matrix. Pattern $p_5$ represents 64% of the “high happy” countries, indicating that those countries have high values in “GDP per capita”, “Healthy life expectancy”, and “Social support”. Moreover, $p_5$ supports countries with varied values for “Freedom to make life choices”. Pattern $p_{11}$ describes “low happy” countries, accounting for 43% of that class. These countries have low values for “GDP per capita” and “Perceptions of corruption” and median values for “Health life expectancy”, “Social support”, and “Generosity”. Compared to “high happy” countries (pattern $p_5$), it is clear that those “low happy” countries present lower values for “GDP per capita”, “Health life expectancy”, and “Social support”. Pattern $p_{27}$ supports 32% of “medium happy” countries, having median values for “GDP per capita”, low values for “Perceptions of corruption”, and flat distributed values for “Freedom to make life choices”. These “medium happy”
countries are between “high happy” and “low happy” in terms of “GDP per capita”, but the other two employed variables, “Freedom to make life choices” and “Perceptions of corruption”, assume values not much different from the other two classes.

The first three patterns $p_5$, $p_{11}$, and $p_{27}$ of Fig. 9a encode countries’ general behavior (43% of the dataset), following the tendencies of the global histograms. The patterns $p_5$ and $p_{11}$ result in two clusters in the similarity map presented in Fig. 9b. The map on the top presents instances colored by class (“high”, “medium”, and “low”), and the bottom by supported pattern ($p_5$, $p_{11}$, $p_{27}$, $p_{27}$, $p_{11}$, and $p_{19}$). The map allows users to reason about clusters and outliers by selecting instances of interest and visualizing their respective patterns. From Fig. 9b, the 6 countries placed at the center of the map are selected, presenting their patterns ($p_{12}$, $p_{21}$, and $p_{19}$) in Fig. 9a together with the other 3 high supported patterns ($p_5$, $p_{11}$, and $p_{27}$). The 3 “high happy” countries supported by pattern $p_{12}$ (emerald) differ from the general group of high happy countries ($p_5$) by having median values for “GDP per capita”. The single “low happy” country supported by pattern $p_{21}$ (yellow) diverges from the general group of low happy countries ($p_{11}$) by holding a higher value of “GDP per capita”, “Healthy life expectancy”, and “Social support”. The 2 “medium happy” countries supported by pattern $p_{19}$ (gold) are different from the general group of medium happy countries ($p_{27}$) by yielding high values for “Perceptions of corruption”. Furthermore, these 6 countries with different happiness levels are similar in “Healthy life expectancy”. In summary, browsing the instances similarity map and visualizing patterns from selected instances makes it possible to reason about countries’ differences and similarities under the optics of happiness levels, allowing the analysis of the most general behavior per class and the particularities of sub-groups or outliers that deviate from the norm.

This explicit discovery and representation of patterns enables VAX to help users uncover intricate relationships that are hard to capture by employing usual multidimensional visualization techniques. As an example, a Parallel Coordinates Plot (PCP) for the 2019 World Happiness Report dataset is presented in Fig. 10. In this visualization, it is possible to see general trends/patterns, conveying similar information as the patterns $p_5$, $p_{11}$, and $p_{27}$ (Fig. 9a). However, quantifying the number of countries for which this information is true, including the ranges of the variables where this holds, is challenging. The analysis becomes even more troublesome when representative secondary patterns exist but deviate from the general norm, e.g., the pattern $p_{12}$ (Fig. 9a). This pattern is valid for 7% of highly happy countries and involves several different variable ranges. This intricate information, explicitly represented by the VAX pattern, is very hard, if possible, to be discovered using a PCP. Selection (interaction) may help to mitigate the problem but finding such patterns may be complex and time-consuming in practice.

Similar observations can be made about Scatterplot Matrices (SPLOMs). An intuition about general patterns can be acquired, especially about the individual variables’ distributions. However, it is difficult to quantify the number of countries for which this information is valid and the range of variables and distribution where this holds – in the accompanying supplemental material, we present such SPLOM in addition to the PCP and SPLOM visualizations of the US Election dataset to emphasize the advantages of explicitly capturing and representing data patterns using VAX opposed to letting users find them using typical multidimensional visual representations.

5 DISCUSSION AND LIMITATIONS

Prediction Versus description. The distinction between model interpretability (focus on prediction) and data explanation (focus on description) is blurred. At first glance, any combination of a prediction model and interpretability strategy could also be used for data description. While it is true that some (global) interpretability techniques can be adapted, a crucial difference must be observed for the general case. While prediction focuses on finding the best way to separate classes with generalization in mind so unknown instances can be properly classified, description (through JEPs) focuses on specialization since the goal is to understand all data of a given dataset, with overfitting an acceptable and positive property [4], [9], [49]. The result is that the inter-class separation is usually well represented by interpretability strategies. Still, intra-class information is typically lost, such as the existence of subgroups of instances within a class.

Existing Solutions. EPs visualization has great potential [18], [46], and VAX provides a powerful method to visualize JEPs focusing on multivariate data analysis via automated insights [56]. VAX stands out from the existing data explanation approaches, such as Explainers [9] and Visual Neural Decomposition (VND) [4], by allowing patterns with more than three variables, non-linear relations, and class “pure” patterns. Nevertheless, such patterns may have lower support when compared with patterns supporting instances from different classes (as found in VND [4]). If desirable, VAX is readily adaptable to display different types of EPs (not only on JEPs), mining patterns where instances belong to multiple classes as well. VAX differs from model explanations solutions, such as ExMatrix [11] and RuleMatrix [10] by conveying visual representations that use local histograms (instead of variables ranges) and DR layouts that represent patterns perspectives. Furthermore, by using a matrix-like visual metaphor, VAX allows for producing meaningful visual representations by filtering.
and ordering patterns (rows) and variables (columns) beyond what was explored in this paper.

Pattern Extraction, Selection, and Aggregation. VAX involves extracting, selecting, aggregating, and visualizing JEPs. The RFm method [18], [48] is used for building random DT models (Algorithm 1), and it has been proved to be a valuable approach to mine diversified patterns [48]. The JEPs selection and aggregation process (Algorithm 2) is inspired by an existing approach [48], reducing the initial set of extracted patterns. These are selected by their confidence and support and aggregated with other meaningful patterns. Those not fulfilling the specifications of being representative (high support) or an exception (for particular instances) are discarded. This way, complementary patterns are aggregated, not losing their information, whereas redundant patterns are ignored.

In this paper, we suggested a simple approach to define $k$ for the extraction process to discover enough patterns covering all data instances. This is indeed an expensive approach, and usually setting $k$ to a large number (proportional to the dataset size), the usual literature’s approach [19], [48], [49], typically brings good results. Another possibility to speed up our process is setting a data coverage threshold (smaller than 100%). In this case, the downside is that some instances (probably outliers) may not be supported by patterns.

DR Layouts Extension. DR layouts are great for analyzing multidimensional data [3], [26], [73]. Still, the quality of a layout is directly related to the DR technique’s efficiency in revealing clusters and outliers presented in the original space. To help in this process, we extend the original space [73] creating new data variables that add pattern information so that the resulting layout also reflects the patterns. The DR layout can be browsed for clusters and outliers, presenting the patterns supporting instances of interest. With JEPs matrix visualization and the similarity maps, compound facts [56] can be generated, enabling linked data insights involving descriptive patterns, clusters, and outliers. Compound facts are highly desirable, providing more nuanced insights about multivariate data [56].

Although not a constrain, we used the statistical mean in this process of variable creation, as suggested in [73], and the classic MDS technique [74] for generating DR layouts. Other statistical measures or DR techniques could also be used. The study of how these choices impact the produced DR layouts is fascinating but out of our scope. We leave such investigation as future work.

Scalability. Since histograms require a certain vertical (height) display space, it is impossible to visualize at the JEPs matrix a significant number of patterns at once. Potential solutions are selecting groups of interest, like the patterns that combined attain minimum dataset support (e.g., 70%) or focusing on particular instances (e.g., outliers). The visual metaphor is also limited by the number of variables used by a single pattern regarding horizontal space for displaying variables. For this matter, filtering variables by importance may be an option.

JEP’s Quality and Ambiguous Data. The quality of the extracted JEPs is related to DTs’ ability to capture the differences between distinct classes. Although JEPs have proven to be capable of capturing diversified relations [48], non-linear relationships are represented through local linear orthogonal approximations, and such approximations may result in too many low-support JEPs. In the worst-case scenario, one JEP per instance if the separation between classes is non-existent, defeating the purpose of generating descriptions for general patterns in the data – for instance, Classes A (orange) and B (blue) in Fig 2. This issue derives from DTs’ limitation of learning generic (complex) patterns or their nonexistence in the dataset in question, which is desirable since it is better not to create artifacts.

JEPs’ quality is not related to the usual definition of models’ performance. Since “all” data is used for training, the testing accuracy is irrelevant. However, the training accuracy is crucial. The generated random DTs have to present 100% training accuracy; with less than that, misclassified instances will be wrongly explained and may impact the variables’ ranges on the extracted patterns, generating spurious information. One solution would be to ignore misclassified instances, but they may be essential for the task at hand. In our case, we guarantee 100% accuracy for every DT by removing ambiguous instances (instances with the same values but belonging to different classes) as part of a pre-processing stage and by overfitting the models using not pruned DTs – in our implementation, if the randomly selected variables to construct a tree do not result in a 100% accurate model, other variables are added until such accuracy is reached. The downside is the high number of resulting low support patterns and the lack of support for ambiguous instances. In any case, since JEPs are class-pure patterns, it is unwanted (or even impossible) to have instances of different classes supported by the same pattern. So, removing ambiguous instances avoids creating erroneous explanations.

6 Conclusion

This paper presents multivariate data explanation (VAX), a new visual analytics (VA) approach for analyzing multivariate datasets. VAX employs aggregated jumping emerging patterns (JEPs) to capture intricate patterns in class-labeled datasets. A matrix-like visual metaphor is used for JEPs visualization, where patterns are rows, variables are columns, and data distribution conveyed using histograms are matrix cells. Based on matrix visualization, meaningful visual representations can be reached by filtering and ordering patterns (rows) and variables (columns). Furthermore, similarity maps produced by DR techniques aim to convey a better overall image of a dataset (e.g., clusters and outliers) using the JEPs lens. VAX can be applied to different domains, which can address phenomenon comprehension through knowledge acquisition, being a valuable tool for extracting data insights. We plan as future work to develop new approaches for filtering and ordering JEPs and variables to enhance VAX visual representations. Moreover, we intend to pursue new methods for diversified pattern extraction and comparisons involving different DR techniques and statistical measures for similarity map creation.

Acknowledgments

The authors wish to thank the valuable comments and suggestions obtained from the reviewers, as well as the support received from the Qualification Program of the Federal Institute of São Paulo (IFSP).
REFERENCES

[1] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, Mastering the information age solving problems with visual analytics, 2010. [Online]. Available: http://diglib.igib.org/handle/10.2312/14803

[2] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim, “Knowledge representation and visualization,” IEEE Trans. Vis. Comput. Graph., vol. 20, no. 12, pp. 1604–1613, Dec. 2014.

[3] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci, “Visualizing high-dimensional data: Advances in the past decade,” IEEE Trans. Vis. Comput. Graph., vol. 23, no. 3, pp. 1249–1268, Mar. 2017.

[4] J. Knittel, A. Lalama, S. Koch, and T. Ertl, “Visual neural decomposition to explain multivariate data sets,” IEEE Trans. Vis. Comput. Graph., vol. 27, no. 2, pp. 1374–1384, Feb. 2021.

[5] T. N. Dang and L. Wilkinson, “Scagexplorer: Exploring scatter-plots by their scagnostics,” in Proc. IEEE Pacific Visualization Symp., 2014, pp. 73–80.

[6] P.-N. Tan, M. Steinbach, and V. Kumar, Introduction to Data Mining, 1st ed. London, U. K.: Pearson, 2005.

[7] D. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, “Why should i trust you?: Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 1135–1144.

[8] B. Kane, B. Cuissart, and B. Crémilleux, “Minimal jumping emerging patterns: Computation and practical assessment,” in Proc. Conf. Adv. Knowl. Discov. Data Mining, 2015, pp. 722–733.

[9] L. Breitman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[10] G. Biau and E. Scornet, “A random forest guided tour,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2014, pp. 291–300.

[11] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan, “EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2009, pp. 1283–1292.

[12] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model explanation for global and local interpretability,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2018, pp. 1905–1915.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should i trust you?: Explaining the predictions of any classifier,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 1135–1144.

[14] L. Breitman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[15] J. Talbot, B. Lee, A. Kapoor, and D. S. Tan, “EnsembleMatrix: Interactive visualization to support machine learning with multiple classifiers,” in Proc. SIGCHI Conf. Hum. Factors Comput. Syst., 2009, pp. 1283–1292.

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” 2018. [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16982/15850

[17] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 4768–4777.

[18] D. Mazumdar, M. P. Neto, and F. V. Paulovich, “Random forest similarity maps: A scalable visual representation for global and local interpretation,” Electronics, vol. 10, no. 22, 2021, Art. no. 2862. [Online]. Available: https://www.mdpi.com/2079–9292/10/22/2862

[19] J. Li and C.-Q. Zhou, “Incorporation of human knowledge into data embeddings to improve pattern significance and interpretability,” IEEE Trans. Vis. Comput. Graph., to be published, doi: 10.1109/TVCG.2022.3209382.

[20] Q. Liu, Y. Ren, X. Ma, and Q. Li, “RankAxis: Towards a systematic combination of projection and ranking in multi-attribute data exploration,” IEEE Trans. Vis. Comput. Graph., to be published, doi: 10.1109/TVCG.2022.3209463.

[21] G. Dong and J. Li, “Efficient mining of emerging patterns: Discovering trends and differences,” in Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 1999, pp. 43–52.
[64] A.-L. Boulesteix, G. Tutz, and K. Strimmer, “A CART-based approach to discover emerging patterns in microarray data,” in *Advances in Knowledge Discovery and Data Mining*, vol. 29, no. 1, pp. 68–77, Dec. 2019.

[65] P.-M. Law, A. Endrenyi, and J. Staszek, “Characterizing and visualizing complex data patterns,” in *Proc. IEEE Vis. Conf.*, 2020, pp. 171–175.

[66] C.-H. Chen et al., “Matrix visualization and information mining,” in *Proc. Comput. Statist.*, 2004, pp. 85–100.

[67] H.-M. Wu, S. Tzeng, and C.-H. Chen, *Matrix Visualization*. Berlin, Germany: Springer, 2008, pp. 681–708.

[68] R. Michelasi and R. Stepp, “Revealing conceptual structure in data by interactive itemset mining,” in *Machine Learning 10*, D. Michie, J. E. Hayes, and Y.-H. Pao, Eds., Hoboken, NJ, USA: Wiley, 1982.

[69] L. Wang, H. Zhao, G. Dong, and J. Li, “On the complexity of finding emerging patterns,” *Theor. Comput. Sci.*, vol. 335, no. 1, pp. 15–27, 2005.

[70] J. Li, T. Manoukian, G. Dong, and K. Ramamohanarao, “Incremental maintenance on the border of the space of emerging patterns,” *Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 116–144, Jul. 2004.

[71] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. London, U.K.: Chapman and Hall/CRC, 1984.

[72] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*. Berlin, Germany: Springer, 2013.

[73] A.-L. Boulesteix, G. Tutz, and K. Strimmer, “A CART-based approach to discover emerging patterns in microarray data,” *Bioinformatics*, vol. 19, no. 18, pp. 2465–2472, 2003.

[74] E. Loekito and J. Bailey, “Using highly expressive contrast patterns for classification - is it worthwhile?,” in *Advances in Knowledge Discovery and Data Mining*, T. Theeramunkong, B. Kijirikul, N. Cercone, and T.-B. Ho, Eds., Berlin, Germany: Springer, 2009, pp. 483–490.

[75] C. H. Yamamoto, M. C. F. Oliveira, and S. O. Rezende, “Including the user in the knowledge discovery loop: Interactive itemset-driven rule extraction,” in *Proc. ACM Symp. Appl. Comput.*, 2008, pp. 1212–1217.

[76] C. Yamamoto, M. de Oliveira, M. Fujimoto, and S. Rezende, “An itemset-driven cluster-oriented approach to extract compact and meaningful sets of association rules,” in *Proc. 6th Int. Conf. Mach. Learn.asy.*, 2007, pp. 57–92.

[77] J. Huysmans, K. Dejaegher, C. Mues, J. Vanthienen, and B. Baesens, “An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models,” *Decis. Support Syst.*, vol. 51, no. 1, pp. 141–154, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923610002368

[78] T. Munzner, *Visualization Analysis and Design*, Boca Raton, FL: CRC Press, 2014.

[79] D. Freedman and P. Diaconis, “On the histogram as a density estimator:2 Theory,” *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, vol. 57, no. 4, pp. 453–476, Dec. 1981.

[80] M. Correll, M. Li, G. Kindlmann, and C. Scheidegger, “Looks good to me: Visualizations as sanity checks,” *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 830–839, Jan. 2019.

[81] W. Jüpf, “A decision rule based approach to generational feature selection,” in *Advances in Data Mining. Applications and Theoretical Aspects*, P. Perner, Ed., Berlin, Germany: Springer, 2018, pp. 220–239.

[82] D. Pérez, L. Zhang, M. Schaefer, T. Schreck, D. Keim, and I. Díaz, “Interactive feature space extension for multidimensional data projection,” *Neurocomputing*, vol. 150, pp. 611–626, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092523121402879

[83] J. B. Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis,” *Psychometrika*, vol. 29, no. 1, pp. 1–27, Mar. 1964.

[84] V. Silva and J. Tenenbaum, “Sparse multidimensional scaling using landmark points,” Stanford Univ., Stanford, CA, USA, Tech. Rep., 2004. [Online]. Available: https://graphics.stanford.edu/courses/cs468-05-winter/Papers/Landmarks/Silva_landmarks5.pdf

[85] T. Tompson and J. Benz, “AP votecast 2018.” [Online]. Available: https://www.openicpsr.org/openicpsr/project/109687/version/V2/view

[86] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, 1987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0377042787901257

[87] J. Helliwell, R. Layard, and J. Sachs, *World Happiness Report 2019*. New York, NY, USA: Sustainable Development Solutions Network, 2019. [Online]. Available: https://worldhappiness.report/ed/2019/

[88] P. Salmin and Y. Kecman, “Regression as classification,” in *Proc. IEEE Conf. Southeastcon*, 2012, pp. 1–6.

[89] F. Cao and E. T. Brown, “DRIL: Descriptive rules by interactive learning,” in *Proc. IEEE Vis. Conf.*, 2020, pp. 256–260.

Mario Popolin Neto received the PhD degree in computer science from the University of São Paulo, Brazil, in 2021. Currently, he is a professor with the Federal Institute of São Paulo - IFSP, Araraquara - SP, Brazil. His research interests include information visualization and visual analytics. His work contributes towards visualization methods for classification models’ interpretability, promoting visual explanations of models and data.

Fernando V. Paulovich (Member, IEEE) is an associate professor in visual analytics for data science with the Department of Mathematics and Computer Science, Eindhoven University of Technology (TU/e), the Netherlands. Before moving to the Netherlands, he was a professor and Canada research chair with Dalhousie University, Canada (2017-2022), and an associate professor with the University of São Paulo, Brazil (2009-2017). He has been researching information visualization and visual analytics, focusing on integrating machine learning and visualization tools and techniques, taking advantage of the automation provided by machine learning approaches, and user knowledge through interactions with visual representations to help people understand and take advantage of complex and massive data collections. In the past years, his research aims at designing and developing visual analytics techniques for the general public to advance the concept of data democratization, promoting unconstrained access to data analysis and widening the analytic capability of lay users in transforming data into insights.