Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data

WOLFGANG JENTNER, University of Konstanz, Germany
GIULIANA LINDHOLZ, 4Soft GmbH, Germany
HANNA HAUPTMANN, Utrecht University, The Netherlands
MENNATALLAH EL-ASSADY, ETH AI Center, Switzerland
KWAN-LIU MA, University of California-Davis, United States of America
DANIEL KEIM, University of Konstanz, Germany

We present an approach that shows all relevant subspaces of categorical data condensed in a single picture. We model the categorical values of the attributes as co-occurrences with data partitions generated from structured data using pattern mining. We show that these co-occurrences are *a-priori*, allowing us to greatly reduce the search space, effectively generating the condensed picture where conventional approaches filter out several subspaces as these are deemed insignificant. The task of identifying interesting subspaces is common but difficult due to exponential search spaces and the curse of dimensionality. One application of such a task might be identifying a cohort of patients defined by attributes such as gender, age, and diabetes type that share a common patient history, which is modeled as event sequences. Filtering the data by these attributes is common but cumbersome and often does not allow a comparison of subspaces. We contribute a powerful **multi-dimensional pattern exploration approach (MDPE-approach)** agnostic to the structured data type that models multiple attributes and their characteristics as co-occurrences, allowing the user to identify and compare thousands of subspaces of interest in a single picture. In our MDPE-approach, we introduce two methods to dramatically reduce the search space, outputting only the boundaries of the search space in the form of two tables. We implement the MDPE-approach in an interactive visual interface (MDPE-vis) that provides a scalable, pixel-based visualization design allowing the identification, comparison, and sense-making of subspaces in structured data. Our case studies using a gold-standard dataset and external domain experts confirm our approach's and implementation's applicability. A third use case sheds light on the scalability of our approach and a user study with 15 participants underlines its usefulness and power.

The reviewing of this article was managed by associate editor Liang Gou.
This project has received funding from the Shift2Rail Joint Undertaking (JU) under grant agreement No 101008913. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Shift2Rail JU members other than the Union.
This publication reflects only the author's view and that the JU is not responsible for any use that may be made of the information it contains.
This project has received further funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826494.

Authors' addresses: W. Jentner and D. Keim, University of Konstanz, Universitätsstr. 10, 78457, Konstanz, Baden-Württemberg, Germany; emails: {wolfgang.jentner, keim}@uni.kn; G. Lindholz, 4Soft GmbH, Mittererstraße 3, 80336, München, Bayern, Germany; email: lindholz@4soft.de; H. Hauptmann, Utrecht University, Princetonplein 5, Utrecht, 3584, The Netherlands; email: h.j.hauptmann@uu.nl; M. El-Assady, ETH AI Center, Binzmühlestrasse 11/13, Zurich, 8092, Switzerland; email: melassady@ethz.ch; K.-L. Ma, University of California-Davis, 2063 Kemper Hall, One Shields Avenue, 95616-8562, Davis California, United States of America; email: ma@cs.ucdavis.edu.

2160-6455/2023/06-ART10 $15.00
https://doi.org/10.1145/3579031

Corrected Version of Record. V.1.1. Published September 13, 2023.

CCS Concepts: • **Human-centered computing** → **Visualization**; **Visual analytics;**

Additional Key Words and Phrases: Structured data mining, pattern mining, subspace search

ACM Reference format:
Wolfgang Jentner, Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim. 2023. Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data. *ACM Trans. Interact. Intell. Syst.* 13, 2, Article 10 (June 2023), 49 pages.
https://doi.org/10.1145/3579031

1 INTRODUCTION

A medical researcher is interested in analyzing patients and their medical histories, where she wants to find commonalities (patterns). The patients have additional attributes that describe the persons, such as gender, age, and diabetes type. The researcher is interested in finding patterns that are significant for all the patients and within specific groups of patients (called cohorts), for example, only female patients older than 80 years. One approach is to filter the patients by their attributes and re-run the same analysis for all the patients. While this might be feasible for a few defined cohorts, this method quickly becomes cumbersome for many cohorts or when there are many attributes involved, since the possible number of filter settings is exponential. Furthermore, the comparison of the cohorts and their medical history patterns is not trivial because *pattern mining*, a clustering approach for structured data, also faces an exponential search space and, thus, an exponential result set. Similar use cases are when a marketing expert is analyzing customers and their market baskets in combination with attributes describing the customers, or a pharmaceutical researcher analyzing the molecular structure of multiple drugs and their effects and side effects modeled as the attributes.

While the overall task stays the same, the structure of the data varies. In the first case, the patients' medical history can be modeled as event sequences, the market baskets are modeled as itemsets, and the molecular structure of the drugs is modeled as graphs. We further name these various types of structures *structured entities* as a generic term. Pattern mining, especially the well-studied frequent pattern mining [1], is a clustering approach for structured data finding commonalities in the form of sub-entities or rules of the structured entities in a database. For example, from itemsets, frequent sub-itemsets [22] can be mined as well as association rules [76]. For sequences, sub-sequences can be mined, better known as sequential patterns, but it is further possible to mine for episodes or sequential rules [23]. While many pattern mining algorithms for various types of structured data are available, a standard pattern mining algorithm is not capable of identifying significant patterns in subspaces [57].

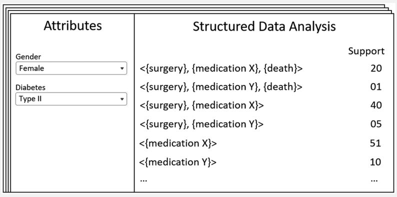
The use case of the medical researcher shows that this task is not trivial because of the two exponential search spaces: (i) the search space of the **structured data,** and (ii) the search space of the **subspaces**. A subspace is a subset of data attributes [41]. In this case, the attributes are assumed to be discrete, allowing a boolean function to evaluate whether a data record with its associated, discrete attributes meets the condition or not. This is also known as Iceberg Cubes [17]. The exponentiality of the search spaces has a significant impact on the runtime and the number of results, since either is an instance of the original search space. Therefore, a scalable approach is sought that also allows the comparison of the various subspaces.

Subspace clustering algorithms focus mainly on numerical data where distances are calculable between the data points. There are some algorithms that also focus on categorical data, however, they suffer from scalability issues [25]. In fact, also itemset mining can be considered a form of subspace clustering where the itemsets form the attributes characteristics of the subspaces. The support, the size of the cluster, is similar to the density measured in subspace clustering algorithms. Here, the major issue is the parameter estimation of the minimum support threshold and other

AT A GLANCE

We compare the common approach and our proposed MDPE-approach with the use case of diabetes patients where a medical doctor wants to discover subspaces (i.e., cohorts) of patients with similar medical histories.

The Common Approach



Attributes

Gender: Female

Diabetes: Type II

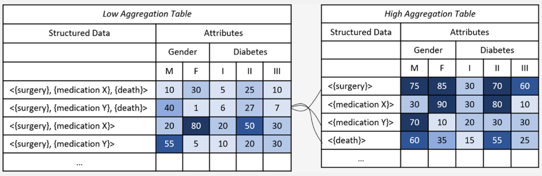
Structured Data Analysis

	Support
<{surgery}, {medication X}, {death}>	20
<{surgery}, {medication Y}, {death}>	01
<{surgery}, {medication X}>	40
<{surgery}, {medication Y}>	05
<{medication X}>	51
<{medication Y}>	10
...	...

Fig. 1. A dashboard design where the user must set filters (dropdowns) to filter the structured data. The result is then processed and visualized. The structured data analysis view shows sequential patterns and their corresponding support. Here, the sequential patterns, representing commonalities in patients' histories, are listed correspondingly with their support (i.e., frequency). The exponential amount of differently filtered structured data cannot be easily compared.

Common approaches (Figure 1) typically follow a dashboard design where their various views are linked through interactions by the user. For example, there are 40 *female* patients with *diabetes type II* that had a surgery followed by the administration of medication X. These approaches have two limitations: (i) a potentially high number of interactions with these filters is required by the user to explore all subspaces since there are an exponential number of possible filter settings; (ii) the subspaces cannot be directly compared as the structured data analysis view is updated each time the filters are being modified.

Our Approach: Multi-Dimensional Pattern Exploration



Low Aggregation Table

Structured Data	Attributes				
	Gender		Diabetes		
	M	F	I	II	III
<{surgery}, {medication X}, {death}>	10	30	5	25	10
<{surgery}, {medication Y}, {death}>	40	1	6	27	7
<{surgery}, {medication X}>	20	80	20	50	30
<{surgery}, {medication Y}>	55	5	10	20	30
...					

High Aggregation Table

Structured Data	Attributes				
	Gender		Diabetes		
	M	F	I	II	III
<{surgery}>	75	85	30	70	60
<{medication X}>	30	90	30	80	10
<{medication Y}>	70	10	20	30	30
<{death}>	60	35	15	55	25
...					

Fig. 2. Two tables represent the boundaries of the structured data mining search-space and juxtapose structured entities (rows) with attribute characteristics (columns) as co-occurrences (cells).

Our MDPE-approach (Figure 2) consists of two tables where structured entities and structured sub-entities (rows) are represented in combination with co-occurrence values (cells) of attribute characteristics (columns). In the previous example of patients with a surgery followed by medication X, it can be seen in the left table that the data consists of 80 *female* patients containing such a pattern, as well as 50 patients with *diabetes type II*. The combination of both attribute characteristics has a co-occurrence of 40 (see Figure 1). This is not represented in our approach because we will show that the co-occurrences are equivalent to the support using the *UniStruct-approach* (see Section 3) and are, thus, *a-priori*. Their removal is one of our measures to reduce the overall search space (see Section 4). Specifically, if the co-occurrence values for both attributes should be high, the individual co-occurrence values of both attribute characteristics must equal or higher. Medication X and Y seem to have high co-occurrences with the attribute gender (see right table) which can correlate with this drug in combination with a surgery (see left table). Because of the **significant search space reduction**, the MDPE-approach allows the visualization and **comparison of thousands of subspaces and multiple attributes simultaneously** while being **agnostic to the type of structured data**.

interestingness measures if they are supported by the algorithm. Such a parameter estimation is difficult, if not impossible, in an exploratory data analysis scenario.

We contribute a **general multi-dimensional pattern exploration approach** (MDPE-approach) that is **agnostic to structured data** and allows the analysis of **multiple discrete attributes** while **significantly reducing the two exponential search spaces**. While our approach generally provides for a minimum support threshold, the default is set to *two* which allows the user to see all relevant available subspaces in a single picture. Thanks to the implemented search space reduction, the result set stays as condensed as possible. We then **implement this approach** as an interactive visual interface, MDPE-vis, which uses a **scalable pixel-based representation** of the co-occurrence values and **includes multiple interestingness measures**, such as *support* and *length*. Our approach and visualization allow the **visual discovery and comparison** of **thousands of subspaces simultaneously**. Within the MDPE-vis, the user can further **interactively drill-down into the search space** while exploration and sensemaking is supported through detail-on-demand views, filters, and sorting options.

We showcase the power and scalability of our approach and interactive visualization using three use cases and a user study of 15 participants that explored a fourth dataset. The first use case, conducted by the authors of the paper, uses the *VAST Challenge 2017, Mini Challenge 1* dataset [69, 70] as a benchmark dataset as various patterns in this data are known and can be seen in our interactive visualization while the number of necessary interactions is dramatically reduced. A second use case consists of data that represents the eating behavior of 35 participants who logged their meals plus additional information such as their stress-level [65]. The attributes encode information such as the BMI, age, and whether the participants are vegetarian or vegan. We describe how the psychologists (domain experts) who conducted this study have used the implemented approach to explore their data and gain multiple insights. The third use case, conducted by the authors, uses a large dataset consisting of 2.5 million datapoints. The participants in the user study analyzed the “What We Eat In America” dataset and were able to find many insights and provided a lot of valuable feedback regarding the approach and tool. We derive a recommended workflow based on the feedback of our domain experts and the feedback of the user study.

2 BACKGROUND AND PROBLEM CHARACTERIZATION

Multiple domains and research fields are integrated in this work. We define our specific terms (Table 1) to avoid confusion in the remainder of the paper. Subsequently, we derive requirements for each aspect, which we strive to fulfill in our solution.

2.1 Requirement Analysis

Our MDPE-approach strives to accomplish exploratory data analysis as stated by Hoaglin et al.: “Exploratory data analysis isolates patterns and features of the data and reveals these forcefully to the analyst.“ [33] We summarize our goals with the following requirements:

R1: Agnostic to structured data. The examples show that the task of finding meaningful patterns in subspaces of data persists, while the data is often modeled in a variety of discrete structures. Thus, an approach is desirable that is agnostic to the type of structured data.

R2: Agnostic to interestingness measures. Depending on the data and task, different **interestingness measures (IMs)** are suitable and required. In the previous example, the only IM was the support, however, many other IMs are available which must be compatible with our approach. Therefore, the sought technique should be also agnostic to the interestingness measures.

R3: Agnostic and scalable to the attributes. Attributes represent additional properties for each structured entity. Their information varies greatly and thus, the sought approach shall be

Table 1. The Definitions and Terminology used in this Paper

Term	Definition
Structured Data	Any form of discrete structures. May be itemsets, event sequences, trees [39], or graphs [16].
Structured Entity	One entity of the structured data, namely one itemset, one event sequence, one tree, or one graph. We denote itemsets as $\{item1, item2\}$ and event sequences, which are sequences of itemsets, as $< \{item1, item2\}, \{item1\} >$.
Structured Sub-Entity	A structured entity that is contained in another structured entity. The definition of containment varies by the type of structured data. These are also often referred to as *patterns*. Examples are itemsets, event sequences, etc., but also more advanced sub-entities such as association rules or sequential rules.
Containment	This is an important property of structured data as structured entities can be contained in another [21]. The definition varies by the type of structured data. We use $e_a \sqsubseteq e_b$ when a structured sub-entity e_a is *contained* in a structured entity e_b. If e_a and e_b are itemsets then containment is equivalent to $e_a \subseteq e_b$.
Attribute	A categorical dimension or feature describing the structured entity. An example might be the *gender* of a patient. For the remainder of the paper, it is also important to understand that attributes can be represented as itemsets.
Attribute Characteristic	One categorical value of an attribute. Continuing the previous example this might be the value *male* of the attribute *gender*. We denote this as *Gender:M*.
Co-Occurrence	A number that shows how often a structured (sub-) entity is co-occurring with an attribute characteristic. An example is visible in Table 3 where the structured entity {bread, juice, milk, vegetables} co-occurs with attribute characteristic *Gender:W*. This is denoted with a 1 in the respective cell. Our approach also allows for real numbers to represent probabilities instead of counts and multiple co-occurrences per attribute, which denotes that multiple attribute characteristics of one attribute co-occur with one structured entity. An example of this might be the attribute *hobbies*, where more than one attribute characteristic is imaginable.
Pattern Mining Algorithm	Comprises all pattern mining algorithms for structured data such as frequent itemset mining [16], sequential pattern mining [23, 26, 49], episode mining [23], association rule mining [32], high utility mining [24], and compression methods [12], etc. Our system MDPE-vis implements the sequential pattern mining algorithm CM-SPAM [21] but the approach (Section 4) functions with any pattern mining algorithm and interestingness measure.
Interestingness Measure	A numerical feature of a structured entity. Well-known examples are *support* and *cardinality*, which we use in this paper. Other popular measures are *confidence*, which is used in association rule mining. We denote an interestingness measure as $f : E \rightarrow \mathbb{R}$, where the interestingness measure is a function that maps a structured entity E into a real value. We focus on objective interestingness measures [47].
Input Data	The input data for our approach contains structured entities and one or multiple attributes. A toy example can be seen in Table 2.

agnostic to the type of attribute data as well as show acceptable scalability to explore multiple data attributes and their characteristics simultaneously. We limit ourselves to discrete attribute data and consider continuous attributes in future work. Note that discretization can always be reached through binning or other measurements.

R4: Overview and comparison. Our approach shall provide an overview to the user and allowing to compare various subspaces simultaneously. Thus, an interactive visualization with good scalability is required to show the large amounts of subspaces.

R5: Minimize parameter estimation. Because additional IMs typically add new parameters, we want to limit ourselves to reduce the number of parameters a user must choose. In the optimal case, the user has to select no parameters before she can explore the data. This ensures that no data will be removed or filtered and thus cannot be explored in the first place.

3 RELATED WORK

We separate the related work into three sections. First, we cover algorithmic approaches for pattern mining which are also capable of handling attributes. The second part discusses visual analytics approaches for handling structured data and attributes, as well as interactive mining approaches. In the third part, we discuss subspace analysis in general which includes also numerical dimensions instead of only categorical ones.

3.1 Algorithmic Mining Approaches

The MDPE-approach is inspired and based on the work of Pinto et al., who describe the problem (i.e., task) of finding patterns in attribute-defined subspaces as multidimensional sequential pattern mining [57]. The work of Grahne et al. assesses a similar problem definition with association rules, using the term *circumstances* to describe the discrete associated attribute information [28]. Pinto et al. [57] detail three variants together with algorithms on how the two search spaces can be searched. We generalize these variants to any structured data and introduce two measures that effectively reduce the sizes of the two search spaces to fractions of the original search space. Pinto et al. focus on the algorithmic implementation of these variants and do not assess the exploratory data analysis and visual analytics aspect of their work. We briefly introduce the three variants, as well as our corresponding generalization. Note that Pinto et al. use the term dimension, whereas we use the term attribute (i.e., gender) and attribute characteristic (i.e., male, female).

UNISEQ Approach. UNISEQ is not an acronym but merely a name that stands for the embedding of the multidimensional discrete attributes in the event sequences. This can be achieved when the event sequences are prefixed or suffixed with an itemset that includes the discrete attribute characteristics. Using PrefixSpan [55] as the mining algorithm, this approach shows good scalability when the number of attribute characteristics is low. This approach can be generalized to any structure, as the attribute characteristics are represented as an itemset and can always be encoded in the original structured data. Although we do not use the UNISEQ approach directly, we use it to show several properties of the co-occurrence values, allowing us to eventually reduce the search spaces.

Dim-Seq Approach. This approach combines two mining algorithms for dimensions (i.e., attributes) and sequences (i.e., structured data). Mining the multidimensional attributes (i.e., attribute characteristics) is possible through iceberg cubing [17] and a sequential pattern mining algorithm. For this approach, the attribute characteristics are mined with the **Bottom-up cube (BUC)** algorithm [10] and the matching event sequences are then mined with an SPM algorithm. This approach shows poor scalability when performed automatically and can be compared to what we referred to as the "common approach" in Section 1 (see Figure 1). However, the automated BUC algorithm is exchanged with the manual labor defining the various filter settings for the attributes.

Seq-Dim Approach. The Seq-Dim approach switches the order in which the algorithms are applied compared to the Dim-Seq approach. For each mined subsequence that exceeds the threshold, a projected attribute characteristics database is built, and the various combinations of the attribute characteristics are mined using the BUC algorithm. This is the most efficient mining method when the event sequences and the number of attribute characteristics are dense. Our MDPE-approach is similar to this approach. However, the attribute characteristics are not automatically mined by a BUC algorithm. Instead, they are being encoded as co-occurrence values presented to the user in the form of two tables, allowing for exploratory data analysis. Our search-space reduction measures further increase the effectiveness of the MDPE-approach.

Another drawback of a fully automated process with no search-space reduction is the algorithm's runtime and, more importantly, the result set presented to the user. Iceberg cubing algorithms follow the idea of support as the primary interestingness measure. While a high threshold (i.e., minimum support) prunes the search space well due to the *a-priori*-property, the resulting combinations of attributes are of varying interestingness to the user. In a real use case, the user cannot easily define a threshold for the size of a subspace, which, for example, if the patient's use case would translate to the size of the cohort. Some rare diseases may only affect one or two patients contained in the data, which may not be a statistically significant amount. However, this may not necessarily defy the concept of interestingness to the user. Songram et al. use closed sequential pattern mining and closed frequent itemset mining algorithms in a Dim-Seq and Seq-Dim approach [58]. Closed pattern mining removes redundant patterns that hold the same information [1]. We do not apply this concept, as we exploit this redundancy to highlight subspaces.

3.2 Visual Analytics Approaches

A lot of research has been devoted to the analysis of structured data using interactive visual interfaces. We discuss related work in this section that allows the analysis of the attribute characteristics combined with the structured data.

Datajewel places the frequency distribution of single events into a pixel-based calendar view [3]. This effectively shows the distribution of events for multiple days, but is limited to a single attribute. Tools, such as *Lifelines2* [68], *Outflow* [73], *EventFlow* [50], *DecisionFlow* [27], and *EventPad* [14] use a design where distributions of the attributes are displayed in a separate panel. The panel is connected through linking and brushing capabilities. In the separate panel, the users can filter the data which updates the sequence data in the main panel. By this, the user can test hypotheses by defining appropriate filter combinations and inspecting the respective subspaces. Gotz and Stavropoulos use a third panel to provide a correlation statistic to a selected outcome measure [27]. The mentioned publications follow the "common approach" as sketched in Figure 1. UpSet by Lex et al. [45], is suitable for categorical and numerical dimensions, however, according to the authors, their visualization scales to only 40 set interactions which are equal to 40 patterns in our case. This is, of course, not suitable as we deal with exponential search spaces. According to the authors [44], higher scalability can be achieved with matrix-like visualizations which is exactly the approach we are following. The matrix representation alone does not provide enough scalability in our case. Additional measures are necessary.

Pure visual representations for structured entities are bound to smaller datasets, but it is challenging to identify subsequences even then. Pure algorithmic approaches are more efficient with larger datasets. Still, the parameter estimation for threshold and constraints is difficult, and interestingness measures to quantify the user's preferences are difficult to formalize or may not even be encoded in the structured data. As many of the pattern mining algorithms are bottom-up approaches, it makes sense to let the user assess the intermediate results and prioritize the mining based on the user's preferences or even discontinue the mining in certain areas of the search space.

Vrotsou et al. contribute an interactive query interface where a subsequence is represented as a graph and all available suffix- and prefix-events are visualized [64]. The user can manually expand the subsequence to build more complex sequences. A linked view displays the attribute dimensions of the matched event sequences. *Frequence* by Perer and Wang uses an interactive constraint-based mining approach, whereas a threshold based on a Pearson Correlation can be applied to one selected outcome measure which is identical to one attribute dimension in our case [56]. Stolper et al. extend this approach in their *Progressive Insights* system by allowing the user to interactively prioritize and prune the search space of the underlying mining algorithm [59]. The authors further establish design guidelines for progressive visual analytics. A similar drill-down approach into the search space of subsequences is described by Jentner et al. where the subsequences are projected to reveal their similarity and cluster information is utilized as subspace information [37]. Di Bartolomeo et al. recently published an intuitive and elegant design allowing the combination and exploration of event sequences with *one* attribute [5]. However, the design lacks scalability with larger alphabets as it is using the shortest common subsequence and cannot be easily applied to other types of structured data.

We strive to provide such information for multiple, automatically computed subsets of the structured entities and for various attribute dimensions simultaneously without the need to apply user-defined (cross-)filters.

3.3 Subspace Analysis

Subspace analysis describes a broad and highly-relevant area of research with a multitude of approaches. The goal is to identify relevant subspaces and to interpret and compare them. Fully automated approaches of subspace clustering [54] are capable of providing relevant subspaces while removing redundancy but do not consider the domain knowledge of the user. The dissertation of Stephan Günnemann [29] covers subspace clustering of complex data which includes mining vector data (numerical dimensions), incomplete data, and heterogenous data which combines numerical and categorical dimensions. Correlations are also covered but only with numerical dimensions. The heterogeneous data chapter deals with graph and network data and respective attributes per node. Such subspace clustering algorithms are tailored for numerical data as they search for dense regions using static and dynamic grids to evaluate the density of the data in various dimensions. The SURFING algorithm [6] uses a k-nearest-neighbor approach which implies an existing distance measure as well. For categorical data, density is equivalent to the support measure used in frequent itemset mining, however, due to the curse of dimensionality, data should be considered sparse as the number of dimensions increase. Because of the curse of dimensionality and too tight constraints in the mining algorithms, interesting subspaces may not be discovered. Visual approaches such as Parallel Coordinate Plots [34] typically do not scale well with an increasing number of dimensions, as the search space is exponential. Jäckle et al. contribute *Pattern Trails*, a 3D-based, visual analysis method for multivariate data revealing pattern transitions [35]. Moreover, the authors provide an excellent overview of available subspace analysis approaches in their related work. Both *PatternTrails* and the approach of Tatu et al. [61] rely on the SURFING algorithm which implies the need for numerical dimensions or an available distance measure for categorical dimensions. *PatternTrails* mentions sequences of dimensions which must not be confused with sequences in our approach as *PatternTrails* refers to the order of the dimensions to receive various visual patterns which are similar to ordering the dimensions of a parallel coordinate plot. In our approach, there is no automated ordering of attributes (i.e., dimensions) and the ordering can be defined by the user. The approach of Lehmann et al. [43] and EvoSets [60] are tools to track subspaces and their effect on dimensionality reductions which also requires numerical data or available distance matrices to compute the projections.

We visualize the subspaces as a pixel-based representation in a tabular layout [11] but in our approach every attribute characteristic is displayed separately instead of visualizing the mean or other statistics of one numerical dimension. Therefore, the discrete distributions are immediately visible.

Many approaches, applications, and commercial tools exist that allow semi-automatic filtering and aggregation of data [7]. Almost every available dashboard has cross-filter capabilities to allow the user to apply filters on various dimensions to update the data of the dashboard. Such filter options can be overwhelming to the user. Moritz Stefaner coined the terms "filter-" and "dropdown orgies"[1] to describe such an abundance of filter options in dashboards. Such cross-filter approaches follow the "common approach" (see Section 1), which implies an exponential number of filter settings to reveal underlying commonalities in the data. It is likely that a user misses an interesting subspace as the time using such an interactive dashboard for exploration is typically limited.

Our approach is tailored to categorical data and discrete structures, respectively. Furthermore, our approach does not make any assumptions about the interestingness of subspaces as this should be determined by the user. Interesting subspaces may be where the co-occurrence of one attribute characteristic is exceptionally high but may also be a uniform distribution of co-occurrences in one attribute. Similarly, a deviation from the distribution of all data may be interesting as well as similar or equal co-occurrence distributions. Our MDPE-approach empowers the user to see all relevant subspaces in a single, condensed pixel-visualization where a limited number of perspectives and some additional metrics allow the user to explore all subspaces to be found in the structured data.

4 MULTI-DIMENSIONAL PATTERN EXPLORATION APPROACH

In this section, we describe what type of data we expect as an input and how it is being transformed throughout the approach. Our approach uses an explicit encoding of the data which is similar to a one-hot encoding and then aggregates the data with two methods: (i) aggregating rows that contain the exact same structural information leaving only distinct structured entities, and (ii) using a modified constraint pattern mining algorithm to calculate broader patterns. We demonstrate this using a toy example of customer market basket analysis. We choose this example due to its simplicity and because itemsets have the smallest search space compared to other types of structured data. All of the following can be generalized to any type of structured data. The section starts with the input data, then continues with the explicit encoding transformation, further processing by two independent methods, and the output of our approach which consists of two tabular representations. We then prove the *a-priori* property of co-occurrences which is the reason that our approach is working. The section closes by describing the achieved search-space reduction.

4.1 Input

The input is shown in Table 2. In this example, the structured data is encoded as itemsets, where each item represents a product. Items in a set have no inherent order; however, any total order can be assumed without the loss of generality. The attributes encode information about the customers such as gender, age group, and country. Each row refers to a *transaction* of a customer referred to by the *ID*.

In general, the input must consist of an identifier, a known representation of structured data, and attributes. The attributes are assumed to be discrete. We refer to a value of an attribute as *characteristic*. Without the loss of generality, we can assume that an attribute may hold multiple

[1] https://medium.com/visualizing-the-field/there-be-dragons-dataviz-in-the-industry-652e712394a0.

Table 2. The Input Comprised of Structured Data and Discrete Attributes

Structured Data		*Attributes*		
ID	Transactions	*Gender*	*AgeGroup*	*Country*
1	{bread, juice, milk, vegetables}	*W*	> 18	DE
2	{bread, candy, soda}	*M*	≤ 18	*FR*
3	{bread, juice, vegetables}	*M*	> 18	*FR*
4	{bread, candy, soda}	*W*	≤ 18	*DE*

The data represents customers and their transactions modeled as itemsets. The attributes provide additional information for each customer.

Table 3. The Input Data Transformed Into the Explicit Encoding

Structured Data		*Attribute Characteristics*					
ID	Transactions	*Gender*		*AgeGroup*		*Country*	
		M	*W*	≤ 18	> 18	*DE*	*FR*
1	{bread, juice, milk, vegetables}	0	1	0	1	1	0
2	{bread, candy, soda}	1	0	1	0	0	1
3	{bread, juice, vegetables}	1	0	0	1	0	1
4	{bread, candy, soda}	0	1	1	0	1	0

Each characteristic is represented in a distinct column, whereas the values represent whether this characteristic is set or not. The cells containing a 1 are highlighted to improve the readability.

discrete values containing a set of characteristics. Let A be the set of attributes and $a \in A$ be an attribute. Then $|a|$ refers to the number of its distinct characteristics. For example, the attribute *Gender* contains two distinct characteristics ($|Gender| = 2$). Because the data is finite, the characteristics of each attribute are finite. This is also true for the *structure-entities* of the structured data, which are in this case the itemsets consisting of items. We refer to the set of all items as *alphabet* (Σ). As with the attribute characteristics, $|\Sigma|$ denotes the size of the alphabet. In the example, the size of the alphabet is 6 ($|\Sigma| = |\{bread, candy, juice, milk, soda, vegetables\}| = 6$).

4.2 Explicit Encoding

The input data, precisely the attributes, are transformed such that each row contains a binary vector for each characteristic having 0 and 1 denoting whether the characteristic is set or not. We name this explicit encoding. Table 3 displays the transformed data. The transformation affects only the attribute information. The structured data remains untouched. Each attribute characteristic is represented in a separate column, whereas the values determine whether this characteristic is set or not. Our approach does not assume or check for mutual exclusivity of the attribute characteristics. It assumes that attribute characteristics are items of a set and that one attribute can hold more than one attribute characteristic at once. It is possible to encode binary attributes (i.e., AgeGroup) with only one bit to denote whether their value is ≤ 18 or > 18. This assumes, however, that the values are dependent and cannot be independently true at the same time. We consider this a special case as the more general case, such as the person's hobbies, will likely have more than one value. Another reason to not encode binary attributes with only one bit is the problem of missing values. With only one bit, a missing value cannot be distinguished from a value that would result in the bit being 0. The resulting binary vectors can also be interpreted as *co-occurrences*. For example, the transaction with *ID* 4 has a co-occurrence with the characteristic *W* of attribute *Gender* of 1.

The transformation also nicely depicts the search space of the subspaces. Let m be the sum of all characteristics ($m = \Sigma_{a \in A}|a|$). In the example, m equals 6, which is also visible by the number

Table 4. The Data is Aggregated by Rows where the Structured Data is Equally Yielding a Table of Distinct Structured Data Entities

Structured Data		*Attribute Characteristics*					
IDs	Distinct Transactions	*Gender*		*AgeGroup*		*Country*	
		M	*W*	≤ 18	> 18	*DE*	*FR*
1	{bread, juice, milk, vegetables}	0	1	0	1	1	0
2 + 4	{bread, candy, soda}	1	1	2	0	1	1
3	{bread, juice, vegetables}	1	0	0	1	0	1

The co-occurrences are added. The IDs are propagated allowing a back-reference to the original data table.

of columns for the characteristics. Because these are binary vectors, the number of all possible combinations is 2^m. This also denotes the size of the search space of attributes. Note that in an *Attribute* → *Struct* approach, as it is commonly used, this is equal to the number of all possible filter combinations. However, many of these filter combinations would yield an empty result set because the combination of attributes does not occur in the data.

A valuable property of this transformation is the possibility to explicitly encode null values such as missing values occurring in the attributes. A missing value can be added as its own attribute characteristic, and co-occurrences to this missing value can be traced throughout the MDPE-approach and eventually back to the original transaction. Moreover, this can be extended to multiple attributes of the same attribute. For example, if the missing value occurring is known, and there exist several reasons.

It is also possible to encode any arbitrary number instead of 0 and 1, such as probability values for attribute characteristics. This is useful, for example, if a measurement is known to have an uncertainty (e.g., error range). Then, such uncertainties could be modeled as probabilities across multiple attribute characteristics.

4.3 Process

The previous transformation and treatment of the values as co-occurrences is the foundation of the MDPE-approach allowing row- and column-wise aggregations of the data. We opt for row-based aggregations only as detailed later in Section 4.6. To this extent, the structured data has not been considered, but the data is an essential aspect in solving the task of finding meaningful patterns in subspaces of the data.

The most straightforward form of aggregating structured data is by combining equivalent structures as a whole. A uniform representation of structured data simplifies that process. In the case of itemsets any total order can be defined over the items to achieve this. Other types of structured data are typically an extension and consist of multiple itemsets. For example, sequences are itemsets that occur in a sequence. Table 4 shows the resulting aggregated information. Only the itemset {bread, candy, soda} occurs twice in the input data (see Table 2) in rows 2 and 4. The respective co-occurrences are added. The table, for example, now clearly shows that all customers buying {bread, candy, soda} exclusively (without any other item) are in the *AgeGroup* of "≤ 18". This form of aggregation yields a table that enlists all distinct structured data entities that occur in the input data. It is possible that all structured data entities of the input are equal, which would result in a table with only one row. However, in real-world applications, it is more likely that only a little or even none of the structured data entities are equal, leading to only a little or no reduction of rows. We discuss this further in Section 4.6.

More aggregation is desirable, and pattern mining algorithms offer a well-studied possibility to do so - imposing additional challenges. Pattern mining algorithms cluster the data in the sense of

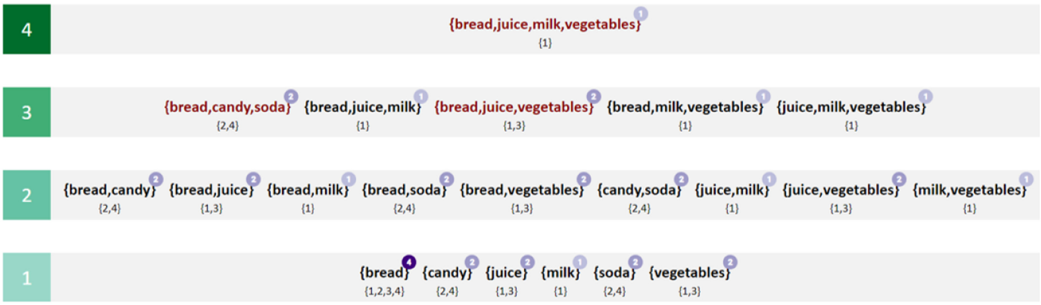


Fig. 3. The search space of an itemset mining algorithm visualized as a Hasse diagram [72]. The length (i.e., cardinality) is encoded in the vertical position. The support is denoted in purple. The red itemsets occur in the input data.

finding common sub-entities in the structured data. This is typically done in a depth-first-search, bottom-up approach where sub-entities containing only one item are combined until the combination can no longer be found in the data or any other termination criteria are met. Figure 3 sheds light on the significant challenge of pattern mining in structured data: the exponential search space. Note, that the figures in this section are not part of the visual interface but were added to support the reader in better understanding how we slice and reduce the search spaces. To create this search space, an itemset mining algorithm which does not use candidate generation (e.g., FPGrowth [30], ECIaT [75]) is being employed without any additional termination criteria. The resulting figure is comparable to a Hasse diagram [72]. The itemsets are sorted by their cardinality, which is also depicted by the green boxes on the left. The cardinality of an itemset is an IM and is often referred to as length or generation. An itemset I supports a transaction T if $I \sqsubseteq T$. The ids of the supported transactions are enlisted below each itemset (compare with Table 2). The number in the purple circles depicts the IM support which denotes the number of transactions the itemset supports. The itemsets with the red font occur in the transaction database (compare to Table 4). Using Figure 3, several observations can be made:

Observation 1: Diamond Shape. The search space has a diamond-like shape where only a few itemsets exist at the highest and lowest generation (top – bottom). The highest number of itemsets are in between (i.e., generation 2 & 3). Note that this observation cannot be made when the input data (Table 2) contains a transaction for every possible combination of items. This scenario is, however, unlikely in real-world applications. We discuss this edge case further in Section 4.6.

Observation 2: Redundancy. While every itemset only occurs once in the search space, multiple itemsets describe the same transactions as they support the same transactions. For example the itemsets {candy}, {soda}, {bread, candy}, {bread, soda}, {candy, soda}, and {bread, candy, soda} all describe the transactions with IDs 2 and 4. This is also depicted by their IM support, which is *two* for all of these itemsets.

Observation 3: Partial Order & *A priori*. The itemsets of the search space are partially ordered. An itemset I is contained in an itemset J, if $I \sqsubset J$. Thus, J is a superset of I. An essential property in the field of pattern mining is the *a priori* property of the IMs. Let I and J be itemsets of the transaction data T (Table 2) and $sup_T(I)$ be the function for the support. The *a priori* property states that:

$$\forall I : \forall J \sqsupseteq I : sup_T(J) \leq sup_T(I) \tag{1}$$

meaning that the support of all supersets of itemset I must be equal or lower than the support of itemset I. The same holds true for the IM length or generation, where:

$$\forall I : \forall J \sqsupset I : |J| > |I| \tag{2}$$

meaning that the cardinality of each superset must be larger than the cardinality of the itemset I.

Observation 4: Low Aggregation Table. The first table that has been produced in the MDPE-approach is the low aggregation table (see Table 4). The rows, specifically the transactions, of this table match the itemsets depicted in red in Figure 3. It is expected that these itemsets can be found at the top of the search space, defining the upper bounds.

These observations can be translated into three actions tackling the exponential search space problem in pattern mining of structured data.

Action 1: Search Space Reduction by support. This action is a result of observations 2, 3, and 4. Pruning the search space using thresholds applied to IMs is the core approach in pattern mining. If the IM is *a-priori*, the pruning can be implemented efficiently [2]. Observation 4 states that the itemsets occurring of the transaction data (depicted in red in Figure 3) occur mostly at the top of this search space and are already covered by the low aggregation table (Table 4). In conclusion, with observation 3, this means that these itemsets typically have a lower support with a minimum of 1. Thus, it is safe to apply a threshold in the form of a minimum support of *two*, which means that all itemsets with a support of *one* are removed. This will only remove duplicates as stated in observation 2 because if an itemset is equal to a transaction that has a support of *one* (e.g., {bread, juice, milk, vegetables}), then all subsets of this itemset that have a support of *one* will describe the same transaction and thus be redundant (e.g., {bread, juice, milk, }, {bread, milk, vegetables}, {juice, milk, vegetables}, {bread, milk}, {juice, milk}, {milk, vegetables}, {milk}). It is possible to increase this number to prune the search space even more, but this stands in contradiction to our requirement **R5** because it is possible that valuable information is being removed. The minimum support can be implemented as a parameter. However, we strongly suggest that a user only changes this parameter from its default value of *two* if the implications are crystal-clear.

Action 2: Search Space Reduction by length. This action follows the first action and is a result of observations 1, 2, and 4. While action 1 already removed some redundant information, observation 2 states that redundancy is more common in the search space and also occurs for higher supports. Observation 1 concludes that the highest number of itemsets are typically to be found at medium generations resulting in a typical diamond shape whereas the top part of this diamond is already covered by the low aggregation table (observation 4). Therefore, we employ a second parameter called *Initial Mining Depth* which terminates the pattern mining algorithm at a given generation. Similarly to the first parameter, a default value of *one* or *two* generations should be set. There are only edge cases where a deviation of this default value is necessary.

Action 3: Interactive Mining. Figure 4 depicts the search space after the first two actions. The itemsets that are crossed out are removed due to not meeting the minimum support of *two* (Action 1). The search space is further pruned by Action 2 with a parameter setting of the *Initial Mining Depth* of *one*, leaving only the itemsets of the first generation (highlighted in red). The remaining patterns cover the bottom of the search space and provide a lower bound. It cannot be assumed that the redundancy (Observation 2) holds for each of the patterns. Thus, the second action may have removed a non-redundant pattern. This is typically not a major problem because the co-occurrences are, in fact, *a priori*, which will be detailed in Section 4.5. To further mitigate this, an interactive mining technique can be used by using a user-defined selection of the already

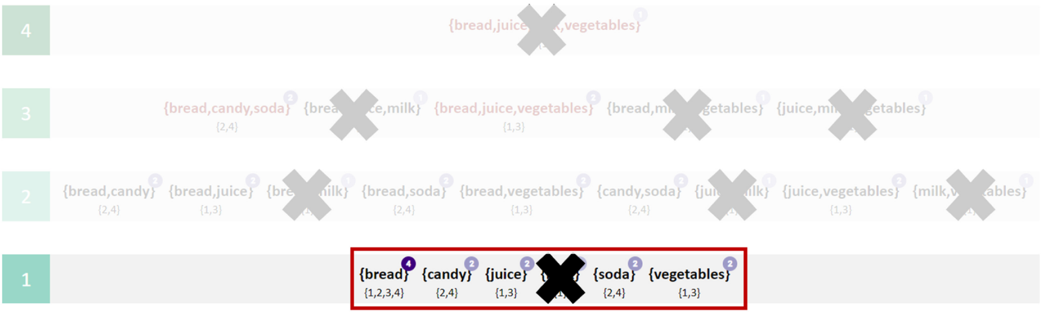


Fig. 4. The search space is pruned with a minimum support of *two* (Action 1) depicted by the crossed out itemsets. A second pruning step by length (cardinality) with a parameter setting of the *Initial Mining Depth* of *one* drastically reduces the size of the search space (Action 2). The remaining sub-itemsets are highlighted in the red box.

Table 5. The Data is Aggregated by Rows where a Pattern is Contained in the Original Transaction

Structured Data		*Attribute Characteristics*					
IDs	Pattern	*Gender*		*AgeGroup*		*Country*	
		M	*W*	≤ 18	> 18	*DE*	*FR*
1 + 2 + 3 + 4	{bread}	2	2	2	2	2	2
2 + 4	{candy}	1	1	2	0	1	1
1 + 3	{juice}	1	1	0	2	1	1
2 + 4	{soda}	1	1	2	0	1	1
1 + 3	{vegetables}	1	1	0	2	1	1

The co-occurrences are added. The IDs are propagated, allowing a back-reference to the original data table (Table 2).

mined patterns and mining for the patterns of the next, higher generation obeying the partial order. For example, the user might select the patterns {candy} and {soda} of generation one and interactively mines for all patterns of the second generation that contains *either* of the selected patterns. This would result in the patterns {bread, candy}, {bread, soda}, and {candy, soda} (see Figure 3 or Figure 4, respectively). It is important to mention that the minimum support for the interactive mining is further set to the initial parameter, which is, by default, *two*. This means that patterns that are already removed due to the minimum support constraint will not be part of the interactive mining result. Algorithmic details will be explained in Section 5.5.

The remaining patterns are used to generate a table analog to the low aggregation table (Table 4). We call this table high aggregation table (see Table 5). The major difference is that the second column does not contain transactions anymore, but patterns which are sub-itemsets of the original transactions. The mining algorithms do return not only the patterns themselves but also the transactions they support (that they are contained in). This is depicted in the ID column, where all transaction IDs are enlisted that support this pattern. The co-occurrences are added using the explicit encoding of the data (see Table 3).

4.4 Output

Our MDPE-approach's output is two tables that are equal in their structure and contain co-occurrences of discrete attribute characteristics to either distinct transactions or patterns. Distinct transactions and patterns can be simply regarded as aggregations of the original

Table 6. This Output is Optional and Represents a Vector of Co-occurrences which are Aggregated using all of the Data

Structured Data		*Attribute Characteristics*					
IDs	Pattern	*Gender*		*AgeGroup*		*Country*	
		M	*W*	≤ 18	> 18	*DE*	*FR*
1 + 2 + 3 + 4	–	2	2	2	2	2	2

This vector can then be subtracted from the other vectors which are used by some normalizations (see Section 5.2).

transactions (aggregations of rows). The tables are derived from the initial transformation of the data into an explicit encoding (see Table 3 and Section 4.2, respectively). We name the first generated table *low aggregation table* (Table 4) which displays the distinct transactions and aggregates their co-occurrences. This table typically covers the top part of the search space (see red patterns in Figure 3).

The second table is called *high aggregation table* (Table 5) as typically many transactions of the input data are aggregated within one row. This table is generated using a modified pattern mining algorithm in conjunction with pruning strategies (see Action 1 & 2 in Section 4.3). The high aggregation table covers the bottom part of the search space (see Figure 4).

A third optional output is generated, which aggregates the co-occurrences of all data into a single vector. This is represented in Table 6. This vector can then be subtracted from other vectors of the output to calculate a divergence because, unlike in this example, it cannot be assumed that co-occurrences are uniformly distributed. We use this method in some of our normalizations as described in Section 5.2.

4.5 Proof: Co-Occurrences are *A-Priori*

We have previously described that the IMs support and *length* are *a-priori*. The same is also true for the co-occurrences and an integral part of why the MDPE-approach of generating two tables covering the boundaries of the structured data search space is working. In this section, we prove that the co-occurrences are *a-priori* by showing that the co-occurrences are equivalent to support measures using the *UniStruct*-approach. Let A be one attribute and c be a characteristic of attribute A ($c \in A$). For example, c = AgeGroup:≤18 or c = Gender:W. The *UniStruct* approach shows that it is possible to model this problem by treating attribute characteristics as items and adding them to the transaction data (see Table 7). This can be done for all of the transactions, and a standard pattern mining algorithm can be applied. The problem here is that the algorithm now has two combined, exponential search spaces, which are typically tackled by tightening the constraints, e.g., by increasing the minimum support. This is, however, in contradiction to our requirement R5. Let I and J be itemsets of the search space S (see Figure 3) and c be a defined attribute characteristic. Let further $cooc_S(I, c)$ be a function returning the co-occurrence value, e.g., $cooc_S(\{\text{candy}\}, \text{AgeGroup:}{\leq}18) = 2$ (compare to the cell of row {candy} and column AgeGroup:≤18 in Table 5). Let $sup_S(I)$ be the function evaluating the support of a pattern. Using the *UniStruct* approach, we can show that:

$$sup_S(I \cup c) = cooc_S(I, c) \tag{3}$$

This means that the support of a pattern combined with the attribute characteristic is equal to the co-occurrence of the pattern and the attribute characteristic. The above example can be used to generate this pattern of $I \cup c$: {candy, AgeGroup:≤18}. This itemset is a subset of the transactions 2 and 4, thus, $sup_S(\{\text{candy}, \text{AgeGroup:}{\leq}18\}) = 2$. Because of Equations (1) and (3) we can conclude that:

$$\forall I : \forall J \sqsupseteq I : cooc_S(J, c) \leq cooc_S(I, c) \tag{4}$$

Table 7. The Input Data of Table 2 Modeled by the *UniStruct* Approach

UniStruct: Structured Data combined with the Attribute Characteristics	
IDs	Transactions
1	{bread, juice, milk, vegetables, Gender:W, AgeGroup:>18, Country:DE}
2	{bread, candy, soda, Gender:M, AgeGroup:≤18, Country:FR}
3	{bread, juice, vegetables, Gender:M, AgeGroup:>18, Country:FR}
4	{bread, candy, soda, Gender:W, AgeGroup:≤18, Country:DE}

Every attribute characteristics is treated as an item and merged with the itemset of the structured data.

This means that a co-occurrence value of a pattern in the high aggregation table (Table 5) can only be higher or equal to the co-occurrence of any transaction that is a superset of the pattern in the low aggregation table (Table 4). Because these two tables cover the boundaries of the search space (see Figures 3 & 4) it can further be concluded that the high aggregation table will always hold the maximum of the co-occurrence numbers whereas the low aggregation table will always hold the minimum of the co-occurrence numbers.

4.6 Search-Space Reduction

We show the search-space reduction, providing the combined search space's upper bounds in contrast to our MDPE-approach's upper bounds. Let n be the number of transactions as provided by the input (Table 2) and n' be the number of distinct transactions (Table 4). Let further be Σ the alphabet of items of the structured data and m be the number of all attribute characteristics of all attributes A: $m = \sum_{a \in A} |a|$. We also define two functions $g(x)$ and $g_k(x)$ where $g(x)$ calculates the size of a pattern mining search space of structured data and $g_k(x)$ the maximal possible number of patterns for one generation k. We showcase this using the search space of itemset mining where $g(|\Sigma|) = 2^{|\Sigma|}$ and $g_k(|\Sigma|) = \binom{|\Sigma|}{k}$. This is also known as the power set of Σ. Note that the search space of itemset mining is the smallest, as other types of structured data such as sequences and graphs allow more combinations of structured entities due to their structure. As discussed in Section 4.2, the number of all possible combinations of attributes is 2^m which defines the search space of the possible subspaces. Therefore, the combined search space is:

$$g(|\Sigma|) * 2^m = 2^{|\Sigma|+m} \tag{5}$$

which can be also trivially shown using the *UniStruct* approach.

Two measures reduce the size of the search space in the MDPE-approach. Firstly, we do not calculate any co-occurrences for combinations of multiple attribute characteristics but only for combinations of patterns in the structured data and one attribute characteristic. The main reason for this is that the co-occurrence of any combination of attribute characteristics can only be equal or lower than the co-occurrences of each of the combined attribute characteristics (see Equation (4)). The second reason is the partial order that occurs when various attribute characteristics are combined. The partial order is visible in Figure 3 which is equivalent to the search space of attribute characteristics when the *UniStruct* approach is being used. Because we want to keep the tabular layout as detailed in the next section, an intuitive linearization of this partial order is not trivial. This measure reduces the search space to:

$$g(|\Sigma|) * m = 2^{|\Sigma|} * m \tag{6}$$

So far, the search space has been reduced by limiting the number of combinations of attribute characteristics. This is equivalent to the number of columns in the respective tables (Table 4 & 5). As described in Section 4.3, actions 1 and 2 reduce the search space of the structured mining, which

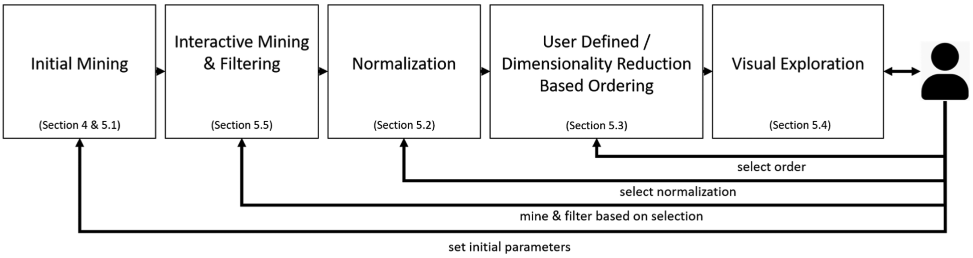


Fig. 5. The processing pipeline of our implemented approach. The user can parameterize and influence every step. The results are propagated to the MDPE-vis.

is equivalent to the rows of the tables. Therefore, it is now required to determine the maximum number of rows possible for each of the tables. The maximum number of rows for the low aggregation table is in fact defined by the size of the alphabet of the structured data but more importantly also limited by the number of rows of the input (see Table 2):

$$\mathcal{O}(n‘) = min(n, g(|\Sigma|)) \tag{7}$$

It is important to understand that the case where $n > g(|\Sigma|)$ does not necessarily mean that $n‘ = g(|\Sigma|)$. A trivial edge case underlines this where the input table consists of an unlimited number of rows, but each row contains the same structured data entity. Because the low aggregation table only holds the distinct structured entities, this means that $n‘ = 1$. Furthermore, it is not possible that $n‘ > g(|\Sigma|)$ because the function evaluates the number of all possible combinations of structured entities. Thus, the worst case in terms of search space is when $n \geq g(|\Sigma|)$ and $n‘ = g(|\Sigma|)$. Finally, it can be concluded that the case of $n > g(|\Sigma|)$ is not typical and does not occur in many real-world datasets because the structured data search space is, in fact, exponential. In contrast, the number of data rows increases linearly.

The maximum possible number of rows ($\mathcal{O}(z)$) for the high aggregation table only depends on the search space of structured data and specifically, the parameter *Initial Mining Depth* defined as d:

$$\mathcal{O}(z) = \mathcal{O}\left(\sum_{x=1}^{d} g_x(|\Sigma|)\right) = \mathcal{O}\left(\sum_{x=1}^{d} \binom{|\Sigma|}{x}\right) \tag{8}$$

Assuming that the second parameter, the *Initial Minimum Support* is *two*, we can construct another edge case to show that the number of data rows n is independent of $\mathcal{O}(z)$: let there be 2 input data rows ($n = 2$) and both rows contain itemsets that complete all possible items of the alphabet (Σ), then all possible combinations of subsets can be constructed where all of these subsets satisfy the minimum support of 2 and all subsets of cardinality lower or equal to the *Initial Mining Depth* (d) will be contained in the high aggregation table.

5 MULTI-DIMENSIONAL PATTERN EXPLORATION VISUALIZATION

Our interactive visual interface (MDPE-vis) implements the MDPE-approach described in the previous section. Figure 5 shows how the output from the *initial (pattern) mining*, in the form of two tables, is further processed before being visualized to the user. The *interactive mining & filtering* step is optional and will be described at the end of this section to improve the readability and clarity of the process.

5.1 Initial Mining

The initial mining step has been covered in the previous section (Section 4). Two parameters are both defaulted to **two** which the authors strongly recommend keeping. Increasing the *initial minimum support parameter* higher than two is possible and will speed up the mining process significantly but may prune subspaces that may have been interesting to the user. The effect of this parameter is the same as interactively increasing the filter option of the support (Figure 7 D_2) which leads to fewer rows in the right table. Once the application is started, the user cannot set this filter lower than the value of the initial minimum support. We, therefore, recommend leaving the default setting and use the interactive filter setting with the drawback that the initial mining process may take longer.

The opposite is true for the *initial mining depth parameter*. The higher this number, the more rows will be added to the right table - exponentially. This is more severe for datasets with a large alphabet (Σ) in the structured data part. However, with these datasets, the curse of dimensionality predicts that the data is more sparse which means that shorter structured sub-entities are more descriptive. It is always possible for the user to increase the generation of the sub-entities through the interactive mining capabilities (Section 5.5) and because of the *a-priori* property of the co-occurrences (Section 4.5). For small alphabets < 10, increasing this parameter to *three* might be useful whereas for large alphabets > 50 reducing this parameter to *one* may be sufficient. It is difficult to estimate this parameter as, in the end, the best value depends on how the structured data is distributed according to their sub-entities.

5.2 Normalization

The co-occurrence values can be normalized in various ways to highlight different aspects of the data. Hence, from a user's point of view, these normalizations offer various perspectives on the data. As shown in Figure 6, our system supports six different perspectives. The figure shows the same part of the data for each perspective. The perspectives are distinguished between *absolute* and *relative/deviation* whereas the absolute perspectives represent the frequency information of the attribute characteristics with varying normalizations. The values are mapped onto a linear binned color map. The *relative* perspectives show the difference of the populations' distributions compared to the subsets (rows) which is done by subtracting the vectors of the output tables (Tables 4 & 5) by the global vector (Table 6). Therefore, positive and negative deviations are possible which are mapped onto a diverging color map. As with the *absolute* perspectives, the normalizations vary.

All color maps are taken from the *ColorBrewer 2.0* online tool [31]. The three different normalizations are labeled by their visual effect on various aspects of the data. The *subspace* perspective normalizes the data per attribute such that the share of a characteristic is reflected. This perspective is invariant to the overall support of the respective row and thus highlights the characteristics (columns). Attributes with fewer characteristics are likely to be more visible by this. The *subset* perspective linearly normalizes all values within each attribute. This specifically highlights attributes that only have a small variance in their co-occurrence distribution. Furthermore, on a more overview level, this perspective allows the identification of equal rows forming the so-called visual blocks. Globally normalized values comprise the *support* perspective, which is correlated to the support. This is visible as the support is also mapped onto the bar chart to the right of each row, respectively. This perspective also supports the identification of visible blocks. A seventh perspective visualizes the **normalized pointwise mutual information (nPMI)**. The scale ranges from -1 (dark red) over 0 (grey, light colors) to 1 (dark blue). If the nPMI is -1, it means there are no co-occurrences (co-occurrence = 0). If the value is around 0, the values are independent. And for 1, they are correlated.

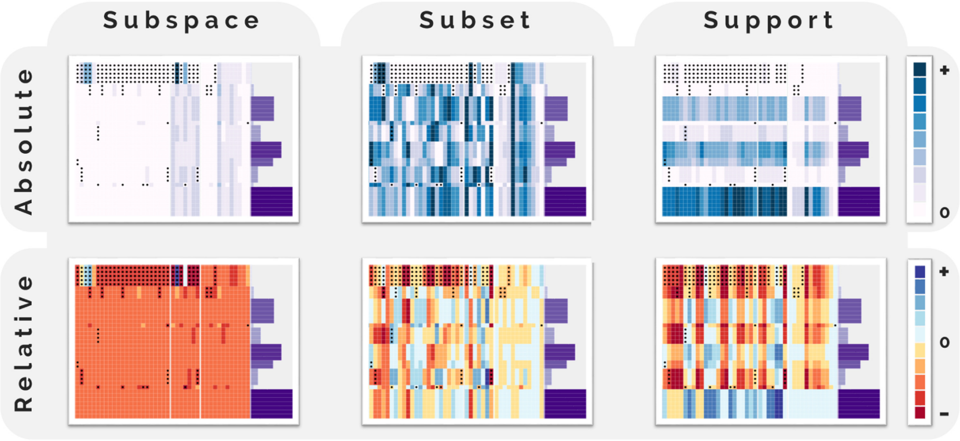


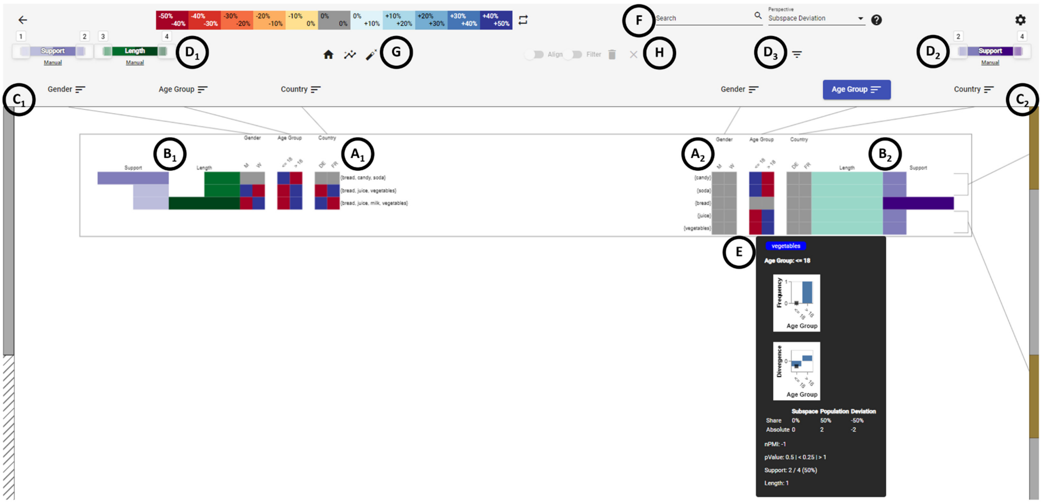
Fig. 6. Six distinct perspectives on the same data. The subspace view highlights attribute characteristics (columns). The subset view highlights the visible blocks. The support view correlates with the support of the aggregation, which is noticeable by the corresponding purple bar chart. The relative views show the difference in comparison to the population.

5.3 User Defined and Dimensionality Reduction based Ordering

The columns of the tables represent attribute characteristics and can be ordered arbitrarily. Some attribute characteristics follow a natural total order, such as age groups. Each row represents a structured entity where no *total* order is given, as structured entities are only partially ordered as discussed in observation 3 in the previous section. Therefore, the user can define the order for one or more columns (i.e., attribute characteristics) which is useful for some tasks as later detailed in the use case section (Section 6). The MDPE-approach also always employs a default order. We experimented with four algorithms: **Principal Component Analysis (PCA)**, **Multi-Dimensional Scaling (MDS)**, **t-Distributed Stochastic Neighbor Embedding (t-SNE)**, and **Locally Linear Embedding (LLE)** [62]. For each technique, we set the number of output dimensions to one. We discard t-SNE and LLE due to their varying necessary parameter estimations and runtime. We use the MDS with a Euclidean distance measure based on the co-occurrences. The PCA provides a less visually coherent result and is therefore discarded. As our observation 2 (redundancy) states, multiple structured sub-entities can describe the same structured entities (e.g., transactions). This also leads to the effect that the co-occurrence vectors (i.e., rows of the tables) contain the exact same values and are, thus, placed together by applying the MDS. We call these rows with equal co-occurrence vectors *visible blocks* (see, for example, in Figure 6).

5.4 Visual Exploration

The visual exploration is enabled by MDPE-vis (Figure 7). The figure shows the example data from Section 4, more specifically Table 4 (A_1) and Table 5 (A_2). Note, that the order of the rows is different because of the applied dimensionality reduction as discussed in Section 5.3. The pixel-based tables (A) and the bar charts representing interestingness measures (B) are drawn onto a zoomable and pannable canvas, which is framed on each side by an overview pane displaying rows of equal (normalized) co-occurrences (C). On top, a static header provides sorting and filtering options (D) as well as a legend, a search field to filter for specific structured entities, as well as an option to change the normalization (F). Note, that the information about the structured entity itself is only visible as a label for each row and in the tooltip (E). In the following, we will detail all components of our interactive visualization. They are prefixed by a letter referencing the labels in Figure 7.



	Subspace	Population	Deviation
Share	0%	50%	-50%
Absolute	0	2	-2

nPMI: -1

pValue: 0.5 | < 0.25 | > 1

Support: 2 / 4 (50%)

Length: 1

Fig. 7. The overview of MDPE-vis displaying the output Table 4 (A_1) and Table 5 (B_1) of our approach (see Section 4.4). The co-occurrence tables of the MDPE-approach are visualized as pixel-based tables (A) on a pannable and zoomable canvas, where each row is accompanied by two interestingness measures displayed as a bar chart (B). The canvas is framed on each side by an overview pane that highlights rows of equal co-occurrences (C). The static header features filter and sorting options, as well as pixel filters (D). The structured entities are only visible as a label next to each row and as a detail-on-demand view in the form of a tooltip (E). The user can search for specific structured entities and change the perspective on the co-occurrence data by changing the normalization (F). A statistical overlay and guidance can be accessed (G). Options when selecting rows become available in the top center (H). This example can be accessed here: https://mdspe.dbvis.de/app/paper-example (username: dbvis, password: $mdspe$).

(A) Pixel-based Tables. The pixel-based tables represent the co-occurrence values of the tables based on our MDPE-approach in Section 4. A pixel-based representation allows the highest density to represent information. In combination with the zoomable and pannable canvas, this design provides the highest degree of scalability while the search space reduction of the MDPE-approach reduces the necessity of drawing too many rows and columns (i.e., pixels) in general. A recent study by Yang et al. also found that panning and zooming are the fastest interactions and provide as good of a context as overview+detail designs for matrix visualizations [74]. This design is also in line with our requirements **R3 - agnostic and scalable to the attributes** and **R4 - overview and comparison**. The left table (A_1, Figure 7) refers to the output Table 4 where the distinct structured entities and their co-occurrences are listed. The right table (A_2) represents the data of Table 5. The order of the columns is identical in both tables; however, the order of the rows is different due to the dimensionality reduction step (see Section 5.3). The pixels are colored using a diverging colormap because of the relative subspace perspective. In the example case (Table 2), the co-occurrence vector for the entire population would contain 2 everywhere (Table 6). Note, that this is a random case of the example that the co-occurrences of the population are uniformly distributed. As each attribute contains two attribute characteristics, the normalized value for each cell of the population vector is calculated by $\frac{2}{2+2} = 50\%$, where the numerator is the co-occurrence value of the cell and the denominator is the sum of all co-occurrence values for this attribute. The middle row of the right table (A_2) represents the co-occurrences of the itemset $\{bread\}$ and all values have a 0% deviation because the item $\{bread\}$ occurs in all the transactions and thus, its co-occurrence vector is identical to the co-occurrence vector of the population. A deviation is, for example, visible for $\{candy\}$ which is the bottom row of the right table (A_2). As visible in Table 5 in

the attribute *Age Group*, the co-occurrence value for the attribute characteristic *Age Group*:$\leq$ 18 is 2, which is 100% when it is normalized. Thus, the deviation to the population where the normalized value is 50% is +50% which is represented as a dark-blue pixel. Because the attribute contains only two characteristics, the other cell representing the attribute characteristic *Age Group*:$>$ 18 to the right shows a deviation of −50%, which is colored as dark-red.

Space is reserved between the two tables. The area is used to draw Bézier curves to link both tables visually. This is shown in Figure 10 where the two tables are additionally vertically aligned by the selection. To avoid clutter, the connecting curves are only shown when the user selects one or multiple rows in either table. The color of the lines refers to one selection. The colors are uniformly selected from the HSV color space. As the right table holds the sub-entities of the structured data, a selection there show all entities where the currently selected sub-entity is contained (see Section 2, Structured Data). Similarly, a selection in the left table reveals all contained sub-entities. The current selection and linked entities are additionally highlighted. This reduces the opacity of all non-selected or non-linked rows. The user can additionally align the tables vertically based on the selection.

(B) Interestingness Measures Bar Charts. Interestingness measures (see Section 2) are represented as bar charts. This design is agnostic to any specific interestingness measure (**R2**). In the example, as shown in Figure 7, there are two **interestingness measures (IMs)** represented: length (green) and support (purple). Because IMs can be, and in this case *are*, *a-priori*, we use a double encoding. The maximum for the IM length in the high aggregation table (B_2) is typically very low and much smaller than the maximum length of the low aggregation table (B_1). The opposite is true for the support where high values are typically in the high aggregation table (B_2) and low values occur in the low aggregation table (B_1). Another impression of these phenomena is shown in Figures 3 and 4 where we use the same colors to depict the IMs. Note, that with larger search spaces of the structured data, the differences in the value ranges are typically much larger. The width of the bar charts is normalized to the minimum and maximum of the respective table. If the width was normalized globally (across both tables), the values for one IM would be so small in one of the tables that they would be almost invisible to the user. The brightness of a color encodes the value of the interestingness measure and is normalized globally across both tables. For example, in Figure 7 the bar displaying support of the middle row of the right table (B_2) representing *{bread}* is outstanding and in a dark purple color because the support of 4 is the highest value not only for this table (A_1) and in general (see Figure 3). On the left side (B_1) the middle row also has an outstanding bar for the support. However, in this case, the underlying value is 2 which is the highest support of this table but not the maximum globally, therefore, represented in a lighter purple color. Note, that the initial mining depth parameter was set to *one* resulting in sub-entities only of length 1 (see Figure 4) and, thus, forming a uniform distribution for the green bar charts (B_2). Gray transparent overlays over the bar charts indicate when a minimum and maximum threshold is set. This is visible in Figure 10 where the maximum threshold for the IM length has been lowered to 26 for the left table and the support has been lowered to 6407 for the right table.

(C) Overview Panes. Two overview panes are placed on both sides of the canvas. Larger datasets cannot be viewed entirely on the canvas, and the overview panes support the user in navigating the pannable and zoomable canvas. In Figure 10, the functionality is more intuitively visible because of the larger dataset. The opaque white background displays the overall length (i.e., the number of rows) for each table. Because in this dataset the left table contains less than half of the rows than the right table, the white background only covers approximately 43% of the height. The remaining space below is displayed with a striped pattern indicating empty space. The gray overlays on both sides indicate the vertical position of the canvas for each table. The more the user zooms out, the

larger the gray overlays grow within a vertical direction. In Figure 7, the gray overlays cover the whole space, as both tables are entirely visible.

The second functionality of the overview panes is the display of *visible blocks* which occur because the co-occurrence-vectors of these rows are equal (see Observation 2 in Section 4.3) and placed together because of the dimensionality reduction step (see Section 5.3). A *visible block* is formed if two or more rows have an equal co-occurrence vector. These blocks are not always standing out in the pixel-based representation due to visual noise. Therefore, *visual blocks* are indicated by the ocher-colored blocks in the overview panes. The larger the ocher-colored block in the overview pane, the more rows belong to the block. When hovering the blocks, the respective rows in the table are brushed and highlighted. A tooltip further indicates the number of rows with equal co-occurrence vectors. This is, for example, visible in Figure 7 as the first two rows of the right table (A_2) form a *visible block* which is indicated also in the overview pane (C_2). A gray line connects the rows with the indicator in the overview pane. The second *visible block* is not shown in this figure as it is cropped, however, the gray connection line is still visible. To avoid clutter, the user can parameterize and filter the indicators by a minimum threshold, which hides all indicators of *visible blocks* that have fewer rows than the set threshold. The default value for this parameter is set to the minimum allowed value of *two*.

(D) Filtering and Sorting. The filter and sorting options are located in the static header, which does not zoom and pan with the canvas (Figure 7, *D*). The user can also filter for specific values (D_3), or visually spoken, filter the tables based on more blue or red pixels. The user can do this for a specific attribute characteristic or any attribute characteristic of a specific attribute. The attribute labels are located at the bottom of the static header. Grey lines connect the static labels with the dynamic canvas blocks. This supports the user in navigating the canvas and preserving the overview in the horizontal direction. In the canvas, we adapt the gap size between the attribute column blocks to be increased with smaller zoom levels to make the distinction between attributes even clearer. In the header, the user can click on an attribute that shows all the underlying attribute characteristics, which are the columns of the pixel-based table in the canvas. For each of the attribute characteristics, the user can sort the respective table by this column. The user can determine a sorting order, which is reflected by small numbers in the header. This allows sorting the table specifically by multiple columns. The sorting options for both tables are independent, allowing a higher degree of flexibility for the analysis. The default order with the lowest priority is always determined by the dimensionality reduction step.

With the same interactions, the user can also filter the table for values of specific attribute characteristics. All employed filtering options result in a reduction of rows only. This implies that no aggregations are changing, and therefore already visible colors do not change. Such behavior is desirable for the consistency of dynamic visualizations [59]. An attribute characteristic filter can set a minimum and maximum threshold. Only rows that have all cells within the defined range are retained. The user can further filter the rows by IMs. Figure 7 at locations D_1 and D_2 shows range-sliders where the user can define the minimum and maximum threshold for the IMs length (green) and support (purple). Note, that the sliders for the left table (D_1) are inverted to align with the bar charts of the canvas, as their baseline is on the right (towards the center of the canvas).

Lastly, the user may select specific rows of interest and may remove these rows explicitly or keep only the selection and remove all other rows. In combination with the iterative mining, this is similar to a templating approach with positive (i.e., interesting) and negative (i.e., uninteresting) templates [40, 46]. This is also described as a design goal by Stolper et al. [59].

When the user opens the interactive visualization, no filters and no sorting options are set initially allowing the user to get a first overview impression of the data, or more specifically, of the co-occurrences and interestingness measures that follows our requirements **R4** and **R5**.

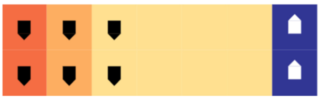
(E) Detail-on-Demand View. As previously mentioned, the structured data itself is only visible in the form of labels aligned with each row and additionally in the detail-on-demand view in the form of a tooltip. This design decision follows our requirement **R1** and also supports the requirements **R3** and **R4** as visualizing structured data in a scalable manner is difficult if not impossible due to the various constraints inferred by the structure itself. The simple structured data representation in the tooltip shows items of an itemset vertically stacked, whereas the itemset themselves is aligned horizontally. For the VAST Challenge 2017 use case (see Section 6), we added a map representing the structured data where the blue lines represent all routes and a red line shows the specifically structured sub-entities (see Figure 10). The tooltip also contains two bar charts that depend on the row and the hovered attribute (column group). The upper bar chart shows the histogram of the attribute, whereas the lower bar chart shows the deviations of the histogram compared to the global histogram of all data.

Besides the structured data, the detail-on-demand view also displays the statistics of the co-occurrences as well as the other IMs. A table represents the differently normalized co-occurrence values (see Section 5.2) for the respective subspace (i.e., row), the co-occurrence value of the overall population, and the deviation of the subspace to the population. Below the table in the detail-on-demand view, the remaining IMs are represented which are in this implementation support and length. As shown in the example of Figure 7, the support for the sub-entity *{candy}* is 2, which is 50% relative support because the input data consists of four rows. The length is 1 because this itemset contains only one item (i.e., the cardinality of the itemset).

(F) Querying, Perspectives, and Miscellaneous Options. The top part of the static header provides various features for the user's workflow. In the center (*F*) is a search query field, allowing the user to filter the rows of both tables by their structured data. For example, the query *candy* will only leave rows where the structured entities, i.e., itemsets contain the item *candy*. Also, more complex, regular-expression-like, queries are possible. This feature is useful for users to verify preexisting knowledge about the data and better learn and understand how the application works and how visual patterns can be interpreted (see 6.5).

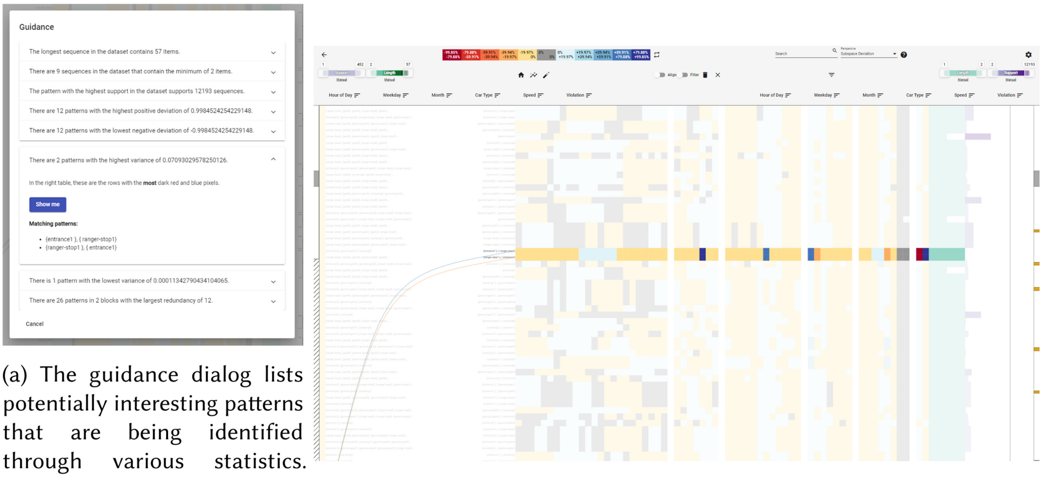
Next to the search field, a drop-down menu provides all available and implemented perspectives on the data. A change here will only adapt the colors of the pixels, while the current viewport of the canvas is retained. The legend on the left is updated accordingly and shows the minimum and maximum value for each colored bin.

On the right side are two buttons located that hold all current sorting and filtering settings. The sorting settings can be reordered by dragging and dropping, and both settings for each filter or sorting option can be deleted individually.



(G) Statistical Overlay and Guidance. The left most button (house symbol) resets the canvas back to a default position. The button for the statistical overview to the right opens a dialog where the user can specify the thresholds for three variants of the binomial test (two-sided, left-sided, right-sided). The glyph is displayed on a pixel when the p-value of the statistical test is below the threshold. The glyph is composed of three parts, which refer to the three different types of tests. The rectangle is displayed if the two-sided test is significant. The triangle on top is displayed if the right-sided test is significant if the value is significantly greater than the average. The triangle on the bottom is displayed for the left-sided test. The color of the glyph is chosen based on the background color of its pixel. The glyph is filled with white color if it's relatively dark; otherwise black.

We implemented a preliminary guidance system. The rightmost button (Figure 7 (G)), opens a dialog window (Figure 8(a)) which entails a list of possibly interesting patterns. Their interestingness



Guidance

The longest sequence in the dataset contains 57 items.

There are 9 sequences in the dataset that contain the minimum of 2 items.

The pattern with the highest support in the dataset supports 12193 sequences.

There are 12 patterns with the highest positive deviation of 0.9984524254229148.

There are 12 patterns with the lowest negative deviation of -0.9984524254229148.

There are 2 patterns with the highest variance of 0.07093029578250126.

In the right table, these are the rows with the **most** dark red and blue pixels.

Show me

Matching patterns:

- {entrance1 }, { ranger-stop1}
- {ranger-stop1 }, { entrance1}

There is 1 pattern with the lowest variance of 0.00011342790434104065.

There are 26 patterns in 2 blocks with the largest redundancy of 12.

Cancel

(a) The guidance dialog lists potentially interesting patterns that are being identified through various statistics. The summary describes the statistical property. The details reveal a visual description and a list of matched patterns.

(b) A click on the "show me" button closes the dialog, zooms, and pans the canvas automatically to the respective rows. Additionally, the rows are highlighted. The highlighted rows refer to the finding "Cars Violating the Park Rules" (Section 6.1 / Figure 11).

Fig. 8. A guidance system is available as a dialog. It contains multiple elements, whereas each guidance element consists of a statistical description, a description of the corresponding visual effects, and a list of matching patterns. Therefore, the guidance feature can be easily extended with more elements using various statistics and machine-learning strategies.

is determined using basic statistics such as the pattern with the highest support and the longest pattern (most items). Other statistics are based on the co-occurrences to find the pattern with the highest positive and negative deviation, which is visible by the darkest blue or red pixels in a row. Also, the variance across all co-occurrences for a pattern can be determined, which is equal to the row that contains the bluest and reddest pixels. The guidance system is implemented as an extendible interface that allows for any statistics and machine learning method to be added. Each calculation based on the whole dataset must result in one element that contains a statistical description of relevance, a summary that describes the visual effects in the interface, and a list of matched patterns. Each element contains a "show me" button that closes the dialog, zooms, and pans the canvas to the matched patterns. The patterns are automatically selected such that they are highlighted in the overview. In Section 7, we will discuss further possibilities for extensions and limitations.

(H) Options for Selection. When the user selects one or multiple rows, these options become active. The user can align the tables vertically such that the selection and connected rows on the other table align vertically. Additionally, the filter can be enabled, which removes all non-selected rows from the canvas. A remove button deletes the selected rows from the canvas, which can be useful if the sub-sequences and their correlations are deemed uninteresting or previously explored. The right-most button clears the selection.

5.5 Interactive Mining & Filtering

The user can drill down into the search space using interactive mining, which is the implementation of the described Action 3 in Section 4.3. This is done by selecting rows and choosing, via a context menu, the option drill down. In the upcoming dialog, the user may choose how many generations (see Section 2) should be mined. The default and recommended value is to mine only the next higher generation of the selection.

Our algorithmic basis is the CM-SPAM algorithm by Fournier et al. as it shows good performance with dense datasets [21]. The CMAP approach of Fournier et al. is an extension to the SPAM algorithm [4] and functions as a bloom filter reducing the amount of the more computationally expensive candidate generation routines of the SPAM algorithm. SPAM uses an efficient vertical database layout, representing the occurrence of an item across all event sequences in a bitmap. The algorithm mines the search space by extending prefixes with an *Itemset-Extension* and *Sequence Extension*. We introduce several modifications to make the CM-SPAM algorithm interactive. There is a difference in the mining technique for each table as the left table (A_1) contains the original, distinct structured entities and the right table (A_2) contains already mined sub-entities. While our MDPE-approach is generalized for any type of structured data, our implementation currently only works with event sequence data and itemsets as the former data type is an extension to itemsets. For any other type of structured data or mining type such as rules, a different type of algorithm has to be selected and implemented. Our various modifications are however similar for any type of pattern mining algorithm.

To not run into a so-called pattern explosion caused by the exponential search space, we constrain the mining algorithm to a maximum length l_{MAX} (i.e., generation or cardinality). We modify the mining from a depth-first-search to a breadth-first-search similar to Perer et al. [56] which allows us to effectively mine structured sub-entities by generation (see Figure 3). Let S be the structured entities contained in the input data D and P be the set of structured sub-entities of D that satisfies the minimum support (s_{MIN}) and does not exceed a maximum length l_{MAX}. Let further P_k be the set of structured sub-entities of the current highest generation: $p \in P_k \subset P | length(p) = k = l_{MAX}$. The user can drill down into the search space using two interactions, which effectively adds more structured sub-entities to P. P can be considered as the rows of the right table. A drill-down operation will add more rows to this table. Each drill down mines for the next generation(s) of sub-entities P_{k+x}. The user can define this by increasing the maximum length l_{MAX}, which must be at least $k + 1$.

The first interaction is based on the selection of structured entities $S_{SEL} \subseteq S$. This is a selection of rows S_{SEL} in the left table which holds the distinct transactions S. This is trivial to accomplish with the algorithm. The user selects rows of the low aggregation table (Table 4, Figure 7 A_1). This table only holds distinct structured entities, therefore the following algorithmic procedure is required: (1a:) All already mined sub-entities P of the selected entities (i.e., rows) S_{SEL} are considered ($P_{SEL} \sqsubseteq S_{SEL}$). (1b:) As the SPAM algorithm is prefix-based, only the highest generation of the considered sub-entities is used (P_k). (1c:) The minimum support threshold is determined by the minimal support of P_k: $minSup = \min_{p \in P_k} support(p)$. The smaller the number of selected structured entities (i.e., rows) by the user, the faster the algorithm can mine additional sub-entities, as the selection simply serves as a projection of the original data.

The second drill-down interaction is based on a selection of structured sub-entities P_{SEL} which are stored in the high aggregation table (Table 5, Figure 7 A_2). The initial assumption to use this selection and filter for the highest generation to mine for longer sub-entities is, however, not correct as the SPAM approach is prefix-based. Therefore, a prefix $< \{a\}, \{b\} >$ (a occurs before b) can only be extended with c yielding $< \{a\}, \{b\}, \{c\} >$. It is, however, not possible to receive $< \{a\}, \{c\}, \{b\} >$ or $< \{c\}, \{a\}, \{b\} >$ even though $< \{a\}, \{b\} >$ is contained in both of these sub-entities (i.e., subsequences). To mine all desired sub-entities of the higher generation, two additional steps have to be included: (2a:) All structured entities have to be considered where P_{SEL} is contained: $S_{SEL} \subseteq \{s \in S, p \in P_{SEL} \mid p \sqsubseteq s\}$. (2b:) Afterward, steps 1a - 1c can be executed based on S_{SEL}. The result of the $k + 1$ generation may include sub-entities where some sub-entities of the user's selection P_{SEL} are not contained in the mined sub-entities. This requires another pruning step. (2c:) Let the mined subsequences with a threshold length l be $\{p_l \in P_l \mid support(p) \geq s_{MIN} \land length(p) = l\}$. All desired sub-entities must be contained in P_{SEL}: $\{p_l \in P_l, p \in P_{SEL} \mid p \sqsubseteq p_l\}$. The sub-entities where this condition does not hold are still kept in the result set as these sub-entities are used when further drilling down into the search space. They are, however, hidden from the user and only become visible if another drill-down interaction verifies their condition. The check for containment of $s_a \sqsubseteq s_b$ runs in $O(m)$ whereas $m = length(s_b)$. We employ additional heuristics acting as a bloom filter to speed up this process.

In either of the two cases, additional sub-entities are being mined which will add the resulting sub-entities to the high aggregation table (Figure 7 A_2). If the user selected all rows and would mine to the highest possible generation, the size of the original search space would still not be reached as in our MDPE-approach no combinations of attribute characteristics are being mined and displayed. In other words, only the number of rows can be increased but not the number of columns. However, the scenario that a user would drill down in the entirety of the search space is highly unlikely because in many applications visual patterns can be early determined because of the *a-priori* property of the co-occurrences and thus, mining for additional sub-entities that would only reveal the exact same co-occurrence distributions is not useful.

6 USE CASES

We present two use cases using different datasets to show the applicability of our approach. The first use case uses the VAST Challenge 2017 Mini Challenge 1 dataset as it provides ground truth. The second dataset originates from a study that psychologists conducted to investigate the eating behavior of persons and correlate it to various user groups.

6.1 VAST Challenge 2017 Mini Challenge 1

The VAST Challenge 2017 Mini Challenge 1 [69, 70] data consists of traffic in the "Boonsong Lekagul Nature Preserve." The traffic is captured at certain locations in the park, monitoring a specific ID for each vehicle, as well as its type and time. This information can be modeled as event sequences where each trip in the park is a sequence, and the locations in combination with the time form the events.

Table 8 shows the data containing three sample entries. In order to draw the maps and calculate the speeds, we further have the coordinates for each point (e.g., gate6) of a fictional coordinate system available, as the average speed of a car is continuous and is, therefore, binned. The challenge describes that no traffic besides rangers are allowed to visit ranger stops. If this is violated, we mark the attribute *violation* with *yes*.

Pattern mining is a clustering task for structured data. Mapping the vehicles' routes as event sequences allows us to aggregate the data by the routes as well as partial routes. Note that the sequential pattern mining algorithm is not constrained and allows gaps in the routes. Such a constraint would be possible and would reduce the number of route segments significantly; however, we did not want to adjust our methods too much to the use case. The co-occurrences of the attribute characteristics, therefore, represent distributions per route (i.e., distinct event sequences, left table) and route segment (i.e., sub-sequences, right table). The visualization shows the user whether routes and route segments correlate with behaviors, such as that a route is only driven at certain times, by certain vehicles, and at certain speeds. The data contains 18,739 event sequences with 577 distinct event sequences. The number of distinct events (locations) is 40 which is the size of the alphabet of symbols ($|\Sigma|$). Most importantly, the VAST Challenge data contains defined and intentionally created hidden patterns which are now available since they were discovered by the participants of the challenge and are available in the published solution [71]. The data, therefore,

Table 8. The Input Data for the VAST Challenge Dataset with Three Example Entries

Structured Data		*Attributes*					
SID	Event Sequence	Hour	Weekday	Month	Car Type	Avg. Speed	Violation
1	< {entrance4}, {gate1}, . . . >	4am	Monday	March	Truck	20–25	Yes
2	< {ranger-base}, {gate5}, . . . >	2pm	Tuesday	October	Ranger	10–15	No
3	< {entrance2}, {general-gate2}, . . . >	3pm	Friday	May	Bus	25–30	No

The full dataset contains 18,739 event sequences. Each sequence consists of multiple itemsets with exactly one item referring to a specific location in the preserve. The average speed attribute is binned. The sequences reflect the positions of the car logged at a specific time and location within the preserve. This table is structurally equal to Table 2.

contains ground truth. We show how it is possible to detect these hidden patterns with our provided interface. We especially focus on the *unusual* patterns, as these occur only in small subsets of the data and are thus difficult to find automatically. It is noteworthy to mention that, in order to find the patterns that we present in the following, mostly no interactions besides zooming and panning are necessary. The whole dataset can be loaded at once. With a minimum support $s_{MIN} = 2$ and maximum length $L_{MAX} = 2$, 1332 sub-sequences can be mined. This effectively means that 577 rows are being drawn in the low aggregation table (Table 4) and $1,332$ rows are visualized on the high aggregation table (Table 5). This use case is conducted by the authors of this paper. Figure 9 shows that three out of four findings are immediately visible when the data is being loaded. The user may use the pan and zoom capabilities to find the respective visual patterns, however, no other interactions are necessary. Only for the fourth finding which we have named "The Race", one interaction is necessary which requires sorting the left table by the highest bin of the attribute "Speed". This is because this visual pattern is only visible by a single pixel and thus, does not show a strong visual stimulus.

Suspicious Truck. The suspicious truck is visible in Figure 10. The finding is largely visible in the data as multiple rows contain dark blue and red pixels whereas the surrounding rows only contain light pixels (see Figure 9, bottom right). The specific sub-sequences in the right table have been selected which causes the brushing of one single line in the left table. This shows that, despite the redundant sub-sequences, there is only a single, distinct event sequence representing the truck's route. In other words, the truck always took the same route. Blue pixels represent a positive deviation from the average of all data. The most prominent pixels are the *violation:yes* (1) and the *car type:truck* (2). Furthermore, it can be seen that the truck only drives on Tuesdays and Thursdays (3) and only between 2 am to 5 am (4). The route is also displayed in the tooltip on the map as a blue line (5). The co-occurrence distributions for all attributes are equal in both of the tables. Because of the redundant sub-sequences and the dimensionality reduction-based ordering, these distributions are better visible in the right table. The tooltip reveals the support of 23 for this sub-sequence. This means that the truck drove 23 times using this route in the preserve. The relative support of $23/183739 = 0.123\%$ underlines that this pattern is not common in the data. Yet, it is prominently visible in our interactive visual interface.

Cars Violating the Park Rules. This use case is similar to the suspicious truck, but even more extreme in the sense of the frequency and shorter trip length. As with the suspicious truck, two rows contain dark-colored pixels, which are in strong contrast to the surrounding areas (see Figure 9, top right). Figure 11 shows a cropped screenshot of the interface after the initialization. As in the previous use case, the anomaly is immediately visible in the *violation* attribute (right-most column group). The column that shows cases, where cars have violated the rules, shows a positive deviation (blue pixel). Furthermore, the attributes *weekday* and *hour of day* reveal that the cars visited the preserve on a Friday between 10 am and 3 pm. The absolute and relative support of

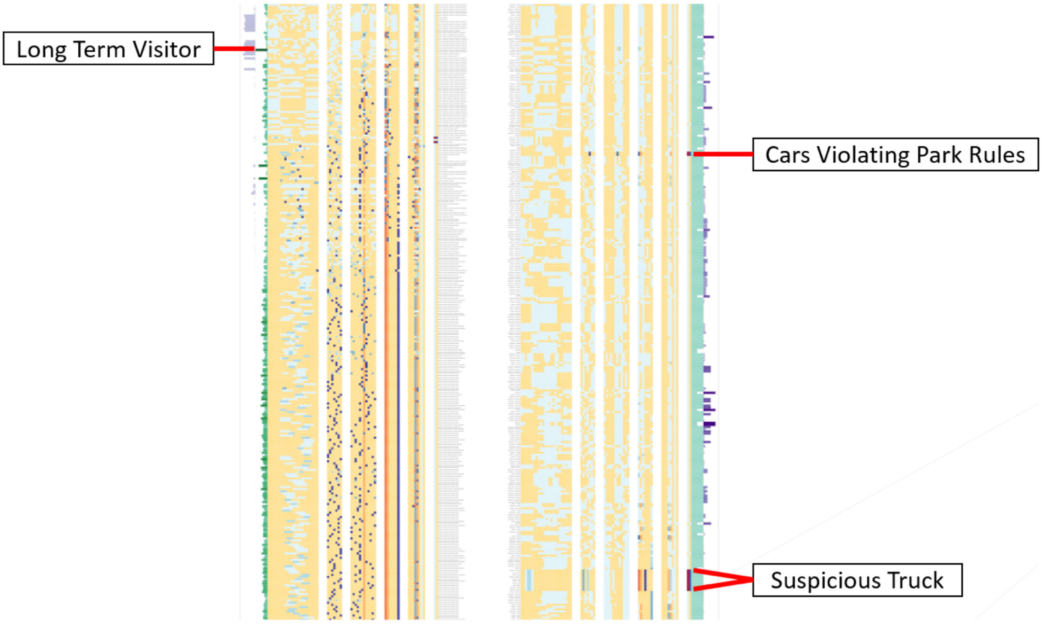
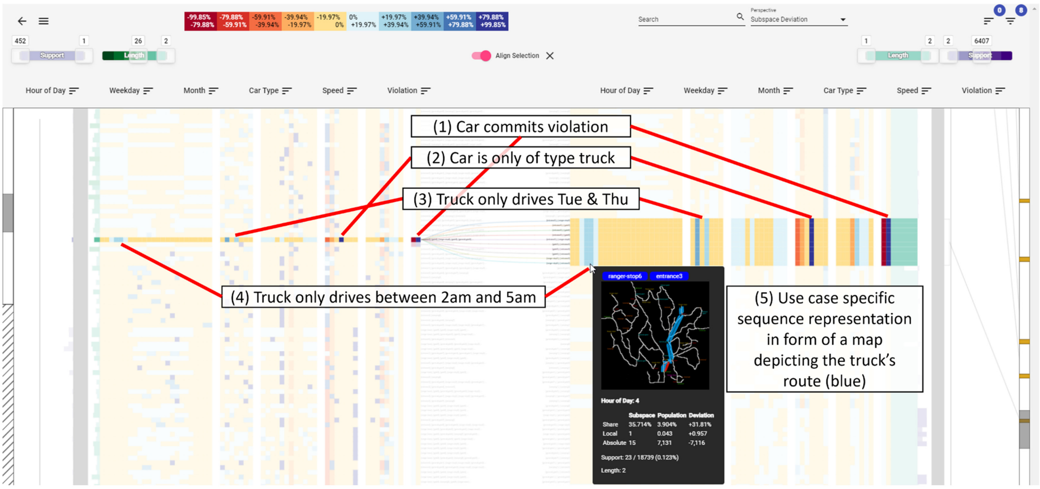


Fig. 9. Without any user interactions, the overview immediately reveals three out of four findings that can be discovered in the VAST Challenge 2017 data set. The UI has been cropped. The patterns in the right table are visible because there are multiple rows that contain several darker pixels which stand largely in contrast to the rows above and below. The finding in the left table can be identified because it is the longest sequence in the dataset which is indicated by the green bar chart to the left of the table. For all of these findings, filters can be applied to reduce the tables to these specific rows.

this pattern is even lower than in the previous use case, as only 6 car trips out of 18, 739 show this behavior. The fact that the stated event sequence is only of length three reduces the possible number of redundant sub-sequences. As visible in the Figure 11, there are only two redundant sub-sequences.

Long Term Visitor. The third use case shows an anomaly in the interestingness measures. It can be found by inspecting the left table using the green length bar chart (see Figure 9, left side). The length of the sequence is much longer than any other sequence in the dataset, thus it is visible. Additionally, the length slider on the left can be increased to filter for this row. An atypical long event sequence can be found as shown in Figure 12. With the help of the (green) bar chart, this anomaly can also be simply identified. The perspective mode is switched to the subset view revealing that the visitor is staying the whole time, that the dataset covers, in the preserve.

The Race. This finding is not easily visible in the dataset but can be found using a hypothesis-driven approach (Figure 13). The tables can be sorted by the highest speed bin (40–45) which then reveals the respective rows. Therefore, the user clicks on the attribute *speed* and then sorts the tables according to the columns with the highest speed bins in descending order. Alternatively, the co-occurrence filter can be used to filter for the rows. The highlighted area shows one row with two bins of an increased speed (blue pixels). Based on the descending ordering, it can further be verified that no other car trips have such a high average speed. The tooltip reveals (not shown in the figure) that the support of this distinct event sequence is two, whereas the other attributes show that both routes were driven at the same time by two different cars. Hence, this pattern is



(1) Car commits violation

(2) Car is only of type truck

(3) Truck only drives Tue & Thu

(4) Truck only drives between 2am and 5am

(5) Use case specific sequence representation in form of a map depicting the truck's route (blue)

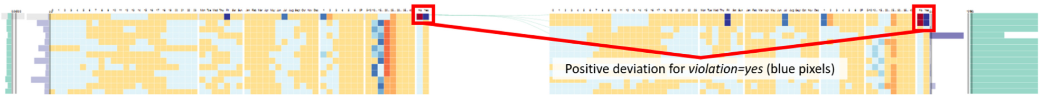
Hour of Day: 4

	Subspace	Population	Deviation
Share	35.714%	3.904%	+31.81%
Local	1	0.043	+0.957
Absolute	15	7,131	-7,116

Support: 23 / 18739 (0.123%)

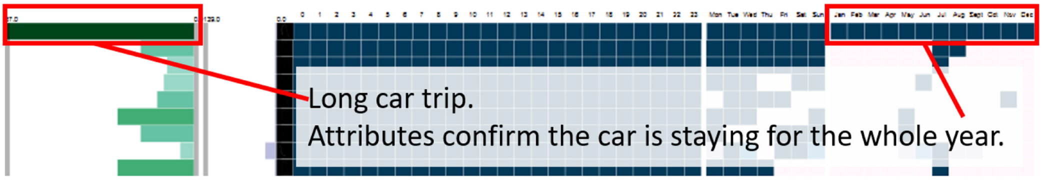
Length: 2

Fig. 10. In the VAST Challenge 2017 narrative, the suspicious truck illegally dumps waste in a northern lake of the preserve (blue route in the map of the tooltip). The respective rows are selected and, thus, highlighted. The truck makes a total of 23 trips and only drives between 2 am and 5 am and only on Tuesdays and Thursdays. The left table only contains one distinct row representing the truck, as it always takes the same route and, therefore, the event sequences are identical. The right table contains multiple identical rows with sub-entities (i.e., sequential patterns) that are redundant, generating a largely visible block. This example can be accessed here: https://mdspe.dbvis.de/app/vc (username: dbvis, password: $mdspe$).



Positive deviation for *violation=yes* (blue pixels)

Fig. 11. Six cars driving illegally to a ranger station at one point in time. Note that the order of the attributes is equal to Table 8. A clear positive deviation is visible in the last pixel of the upper rows (*violation* = *yes*) which has a dark blue color. The characteristic *violation* = *no* therefore has a negative deviation (dark red).


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Mon Tue Wed Thu Fri Sat Sun
Jan Feb Mar Apr May Jun Jul Aug Sept Oct Nov Dec

Long car trip.
Attributes confirm the car is staying for the whole year.

Fig. 12. The top line shows an event sequence of unusual length. The attributes in the subset perspective reveal that this specific car is staying the whole year in the preserve.

called the race. Note that the speed calculation is based on a fictive coordinate system and therefore no units are available.

6.2 Expert Study: Eating Behavior

This use case utilizes data conducted in a study by psychologists where 35 participants logged their meals over the period of one week by using the smartphone application SMARTFOOD [13, 63, 65, 66]. Additionally, the participants reported their subjective stress levels right before and after their

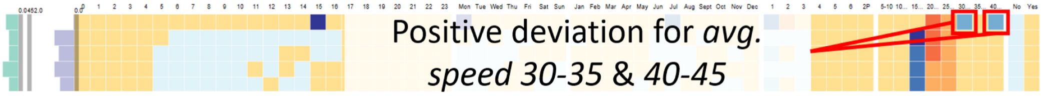


Fig. 13. A race of two cars through the preserve, showing higher speed than usual. Based on the descending order of rows based on the columns representing the highest speed bins, it can also be confirmed that no other traffic is driving as fast.

meals using a 7-point Likert scale ranging from 0 (high stress) to 6 (low stress). The data shows three fictional sample participants that are displayed in Table 9. The reported stress levels before and after the meal and the type of the meal are combined in one itemset. Even though the stress levels take place sequentially in relation to the meals (before and after), the symbols are combined in one itemset. Their labels *str-pre* and *str-post* indicate their order within the unordered itemset. The reason for this is that a sub-sequence < {str-pre:1}, {meal:lunch} > where a pre-stress level of 1 is followed by lunch does not necessarily mean that it is directly followed, as numerous gaps are possible. Instead, with this encoding, the sub-sequence would be < {str-pre:1, meal:lunch} > where the pre-stress level and the meal are items in one itemset. In the latter case, the stress level certainly belongs to the meal within the same itemset. Again, the first encoding could be also combined with an additional window constraint for the pattern mining algorithm, but such constraints prune the search space tremendously and the consequence might be that important information is lost such as whether stress levels in the morning influence the stress levels at dinner time. There are two possibilities for how the participants' meals can be represented. Either each of the 35 participants' meals is represented as one event sequence (i.e., a row of the table) where all meals of the week occur sequentially, or the meals are split by day and participant, which would result in $7 * 35$ rows in the table. The latter encoding does not allow any insight in whether stress levels or meals influence over multiple days but, for example, if a participant has a snack before breakfast it can be certain that the snack was actually taken before breakfast and is not an artifact that the snack was consumed the day before.

The attributes consist of the *gender*, *age*, *vegetarian*, *vegan*, and ***Body-Mass-Index (BMI)***. The attribute *age* is not binned, and each value is displayed as a separate characteristic. The *BMI* is binned according to the scale of the World Health Organization [52].

The size of the alphabet for this dataset is 20 which is half the size of the previous use case. None of the participants have a completely overlapping history of meals which is why the left table contains only 35 rows. Because the structured data contains sequences, there are many combinations possible which results in over 200 rows in the right table with an initial mining depth of *two*. Note, that because of data privacy reasons we cannot publish an overview screenshot of the whole dataset. Instead, we select three out of nine findings the psychologists reported and display and describe the visual patterns using cropped images.

The implemented tool for this approach was tested in three phases by the psychologists that conducted the study:

(1) In this phase, the tool was demonstrated to the domain experts using the VAST challenge dataset as their own data was still being gathered. Several interactions of the tool were demonstrated and the visual patterns were explained to the domain experts. The meeting lasted for about one hour.
(2) This phase consisted of a short paired analytics session where the data of the psychologists was loaded into the tool. The domain experts identified some of the already known correlations that are contained in their data. This meeting was only 30 minutes.
(3) The psychologists used the tool on their own without any of the authors being present. The domain experts prepared a document consisting of screenshots and descriptions with

Table 9. The Data of the Eating Behavior Study Shows Three Artificial Participants

Structured Data		*Attributes*				
SID	Event Sequences	Gender	Age	Vegetarian	Vegan	BMI
1	< {str-pre:4, breakfast, str-post:5}, {str-pre:3, lunch, . . . >	m	20	yes	no	normal
2	< {str-pre:1, lunch, str-post:3}, {str-pre:1, snack, . . . >	m	30	no	no	obesity III
3	< {str-pre:5, lunch, str-post:6}, {str-pre:4, cake, . . . >	f	21	yes	yes	obesity I

The participants log their stress levels (*str-pre & str-post*) before and after every meal (0 = highest, 6 = lowest). Every meal, including the pre- and post-stress level, is encoded as an itemset. The itemsets of meals over the days form the event sequence.

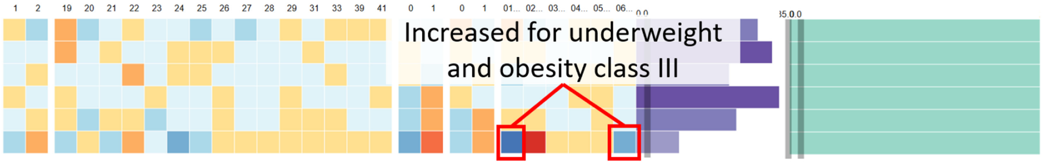


Fig. 14. The highlighted line shows the behavior of six participants where a *high stress level* was reported followed by a *snack*. The relatively higher prevalence of this behavior (i.e., sub-sequence) can be visually compared across *BMI* categories.

findings of potential interest. In total, they reported nine findings. Note that these patterns are labeled as patterns of potential interest, as they have not yet been verified as statistically significant by the domain experts. It is unknown to the authors how much time the psychologists spent exactly, but in the post-interview session, the psychologists mentioned they spent more than one hour with the tool.

(4) A 30 minute interview was conducted with the domain experts where they provided feedback about the tool.

The psychologists were associated with the authors from previous projects and during an informal meeting, the idea was brought up that their own dataset would be suitable for analysis with our approach. The psychologists were not compensated but asked to continue using the tool for their own datasets. In the following, we present three out of the nine patterns of potential interest identified by the domain experts. All of these nine patterns of potential interest were novel to the psychologists and were not discovered in their own analysis, which was mainly done using R Stats.

Stress and Snacks. This pattern of interest, as depicted in Figure 14 shows a behavior (i.e., subsequence) where a meal with a high pre-stress level (*str-pre:1*) is followed by a *snack*. Out of the 35 participants, 6 participants show this behavior. As Figure 14 shows, this behavior seems to be over-represented in participants with a relatively low or high BMI. For instance, only 2 of the 35 participants (5%) are classified as slightly underweight, but both participants show this specific behavior. A similar visual pattern can be seen for participants with a *BMI* of *obesity class III*.

Low Stress-Levels for Vegetarians. When sorting the table in descending correlation for vegetarians, multiple behaviors (rows) can be visually detected, as shown in Figure 15. Row 1, 2, and 4 show the attribute characteristics of behaviors that have varying combinations of stress levels 5 and 6 (low). In all of these rows, the attribute characteristic (column) *vegetarian=yes* shows a positive deviation (blue pixels).

Snack and Breakfast. The bottom row in Figure 15 represents the behavior <{snack}, {breakfast}> (*snack* followed by *breakfast*) which seems to be higher correlated with participants that are classified as *pre-obese* and *obese class II*. Note in this case the data was encoded by participants and day, therefore, it can be concluded that the participants had a *snack* right before *breakfast*.

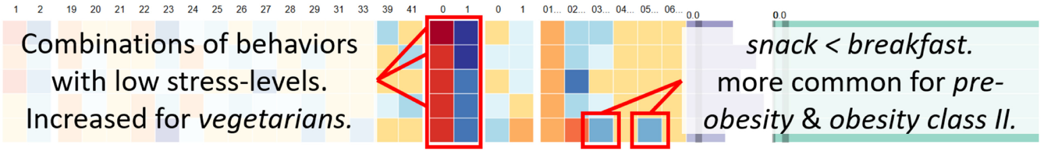


Fig. 15. Various behaviors (rows) that seem to have a high correlation with participants that are *vegetarian*. Row 1, 2, and 4 represent behaviors that have various combinations of *low stress levels*. The bottom row shows the attributes of behavior *snack* followed by *breakfast* which shows a relatively higher prevalence with people classified as *pre-obese* and *obesity class II*.

Table 10. The Transformed and Filtered Dataset of the US Weather Events Contains Data for 2019 and 2020 with Around 2.5 Million Records

Structured Data		*Attributes*				
SID	Event	Year	Month	Day of Month	Hour of Day	US-State
1	$< \{\text{Rain}, \text{Heavy}\} >$	2020	03	15	20	CA
2	$< \{\text{Snow}, \text{Moderate}\} >$	2021	01	31	23	MI
3	$< \{\text{Hail}, \text{Light}\} >$	2021	04	01	03	WI

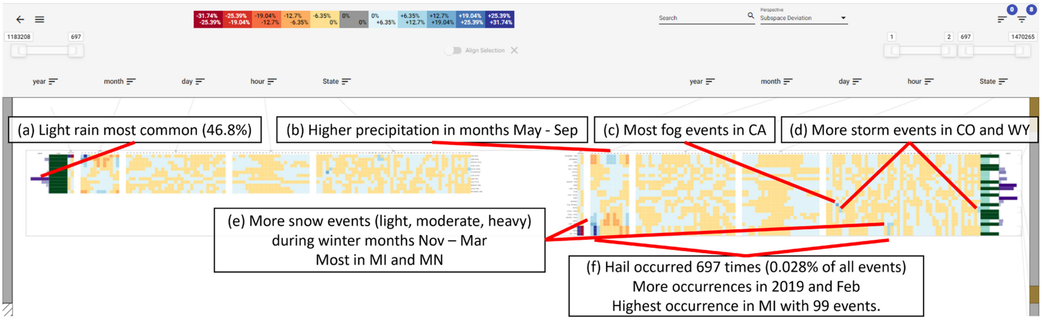
6.3 Kaggle Dataset: US Weather Events

This use case, conducted by the authors, demonstrates some scalability capabilities of our approach and highlights why the alphabet size has a higher impact on the number of rows in the tables than the number of rows in the input. The original dataset consists of 6.3 million rows containing weather events collected at airports in the United States[2] However, this is too large for our implementation as we use the SPAM algorithm which creates a bit vector for each item in the transactions and the length would cover all 6.3 million transactions. As we compute all structured sub-entities in memory, the memory consumption exceeds our hardware capabilities. With a more efficient implementation, it will be possible to load the entire dataset. This is only an engineering effort and does not change any core parts of our approach. Instead, we filter the weather events for the years 2019 and 2020 which still leaves about 2.5 million data rows remaining. The original data covers the years 2016–2020.

We transform the data as shown in Table 10 where one event consists of the type of weather that has been observed and its severity. The starting time of the event has been transformed into multiple discrete attributes: *year*, *month*, *day of month*, and *hour of day*. An additional attribute contains the *US-State* that represents where the event has been recorded. The execution of the pipeline takes about 8 minutes and consumes around 41*GB* of memory. An important point is that the size of the alphabet is relatively small with only 13 items which would be the same if all years would have been used. Furthermore, only 12 distinct events (rows) are contained in the data. Due to the small alphabet and a low number of distinct events, the visual output is rather small (Figure 16) and much smaller than the previous two use cases even though this use case has a much larger input.

As shown in Figure 16, multiple visible patterns can be found as soon as the data is loaded. Light rain is the most frequent in the dataset (a). More precipitation events than average were recorded in May - September (b). Most fog events occurred in the state of California (c) and more storm events than average were recorded in Colorado and Wyoming (d). Snow events occur more often during the winter months from November until March (top red line) which is not surprising as

[2] https://www.kaggle.com/sobhanmoosavi/us-weather-events.



(a) Light rain most common (46.8%)

(b) Higher precipitation in months May - Sep

(c) Most fog events in CA

(d) More storm events in CO and WY

(e) More snow events (light, moderate, heavy)
during winter months Nov – Mar
Most in MI and MN

(f) Hail occurred 697 times (0.028% of all events)
More occurrences in 2019 and Feb
Highest occurrence in MI with 99 events.

Fig. 16. An annotated snapshot of the interface after the data is loaded. No further interactions have been performed. The visualization shows various correlations of weather events reported at US airports with the temporal domain as well as the location encoded by the US-State. The order of the attributes (blocks of columns) is equal to the attributes shown in Table 10.

the United States is located in the northern hemisphere (e). Most snow events were recorded in Michigan (MI) and Minnesota (MN, lower red line). Even though the event hail occurs infrequently with only 697 times (0.028%), it is prominently visible at the bottom of the right table (f). The blue pixels show a positive deviation where there have been more hail events recorded in 2019 and in the month of February (red line to the left). The slightly darker blue pixels in the most right column block of the right table represent a higher occurrence of the event in the state of Michigan (MI, red line to the right).

6.4 User Study: What We Eat In America Dataset

We have conducted a user study with 15 participants. The participants were recruited from the Computer and Information Science Department at the University of Konstanz, and every participant has experience with visualizations and visual analytics tools. Out of the 15 participants, two are on a Bachelor's level, two are on a Master's level, 10 are Ph.D. students, and one is a PostDoc. None of the participants was the author of the paper. All of the studies have been conducted by the first author of this paper. The participants were compensated with some chocolate bars but received no compensation otherwise. The study was conducted online via Zoom. The participants could choose whether they would like to enable the camera. Some of the participants agreed to record the call in video and audio. It was made clear that the recordings will not be published and deleted in an appropriate time frame. The interviewer's camera was turned off during the study. The participants provided their answers using an anonymous online form that was open and filled out during the study. They could decide freely whether they speak aloud about what answers they write down or tell the interviewer that they had finished answering the question. In most of the studies, the participants used both approaches depending on the questions. The answers were provided in bullet points in English and German. Each user study lasted about two hours and was divided into five parts. Note that the guidance feature was disabled during the study as it was not stable enough as a feature.

(1) Introduction and overall task. The interviewer explained the process of the study and then introduced the overall task of subspace search and correlation analysis in categorical data. At the end of this part, the participants were asked whether they understood the task and had any additional questions. Afterward, the participants were asked how they would tackle this task if presented with such a dataset. This part lasted for roughly 10 to 15 minutes, depending on the questions.

(2) Introduction of the tool. The participants were asked whether they knew the VAST Challenge 2017 MC1 dataset. Nine participants reported that they had heard of it, and six reported that they didn't know it. None of the participants had previously worked with this dataset. Then the concepts of the approach were introduced by the interviewer using PowerPoint slides, and afterward, the tool was introduced using screen sharing with the same dataset. The interviewer explained interaction possibilities, possible perspectives, the statistical overlay, and the filtering and sorting capabilities of the tool. The participants could ask questions on the spot and had been asked whether they had any additional questions regarding the tool. This part lasted around 25 to 30 minutes, depending on the questions.

(3) Estimation about the search spaces. This part consisted of estimation questions about the search spaces. The participants were not expected to know or derive the correct formulas but give a rough estimate of how large the respective search space would be. Therefore, they were asked to provide their answers as the power of a natural number (i.e., 10^X). The parameters of the VAST Challenge dataset have been used to estimate the various search spaces. The first question was to estimate the theoretically possible number of combinations using 60 attribute characteristics. The second question was to estimate the theoretically possible number of sequential patterns in the VAST challenge dataset ($\Sigma = 40$ and longest sequence = 57). It was underlined that no assumptions regarding the data should be made for this question. The third question was to guess the actual number of patterns that exist in the dataset, including assumptions such as "cars may not be at two locations at the same time" and "cars must follow the roads in the park (i.e., cannot jump)". It was also made clear that there exists no formula for such a case, and the answer could be only provided if the patterns were mined. The answers to each question were provided after the respective question. Most participants chose not to tell the interviewer their answers but provided their answers in anonymous form. Afterward, it was explained that the approach, as described in this paper, only uses patterns of generations one and two (length), which greatly reduces the number of patterns. The VAST Challenge dataset shows 114, 540 values (pixels) to the user using this approach. This part lasted about 10 to 15 minutes, depending on the questions.

(4) Paired analytics session. The mode was switched, and the participants were asked to share their screens and use the tool in a paired analytics manner and a think-aloud fashion. Before that, the interviewer introduced the dataset to the participant, which is called "What We Eat In America (WWEIA)" [51]. For the study, the dataset from 2017–2018 had been prepared and consisted of 7,640 participants logging their intake of one day [18]. More specifically, the sequences consist of the "Combination Food Type" (DR1CCMTX) and the "Name of eating occasion" (DR1_030Z). Note that the eating occasion is provided in English and Spanish. The dataset has been joined with the dataset "Body Measures (BMX_J)" [19] which contains the body mass index of the persons that participated in the study. Finally, the dataset was also joined with the dataset of "Demographic Variables, and Sample Weights (DEMO_J)" [20] which contains the gender and age information of the participants. Altogether, the prepared dataset for the study contained the following attributes for each participant:

- Gender (DEMO_J - RIAGENDR) with male and female as characteristics
- Age (DEMO_J - RIDAGEYR) in age groups of 5 years (0–5, 5–10, . . ., 80+)
- Body Mass Index (BMX_J - BMXBMI) in categories by the World Health Organization (underweight, normal, overweight, obese I–III, missing)
- Intake Day (DR1IFF_J - DR1DAY); Sunday - Saturday
- Intake Hour (DR1IFF_J - DR1_020); 0–23
- Breastfed infant (DR1IFF_J - DRABF); Yes, No

Table 11. The WWEIA Dataset Consists of 7,640 Sequences Containing what People Eat for what Occasion Associated with Attributes About the Person's Gender, Age, BMI, Intake Day, Intake Hour, and Whether they were Breastfed

Structured Data		*Attributes*					
SID	Event Sequences	Gender	Age	BMI	Day	Hour	Breastfed
1	< {o:Lunch, t:Cereal}, {o:Snack, t:Chips}, . . . >	m	20–25	normal	Mo	{8,10, . . .}	No
2	< {o:Cena, t:Meat}, {o:Botana, t:Ice cream}, . . . >	f	40–25	obese I	Fr	{18,22, . . .}	No
3	< {o:Breakfast, t:Cereal}, {o:Lunch, t:Salad}, . . . >	f	15–20	normal	We	{6,12, . . .}	No

Table 11 shows a summary of the dataset. None of the participants had previously worked with the dataset or a similar dataset. Before starting their analysis, they were asked to write down some expectations and hypotheses about the dataset. The participants then spent around 30 to 45 minutes using the tool. They could ask questions and discuss their findings with the interviewer.

(5) Post interview. First, the participants were asked to write down the insights they found in the WWEIA dataset using the tool and, more specifically, whether their initial hypotheses could be verified, falsified, or not answered. Furthermore, they were asked whether they had missed anything significantly in the dataset and, if yes, what they would have needed (e.g., specific features, more time, etc.). Finally, they were asked to revisit their answers to the first part of the study about what approaches, algorithms, and tools they would use now that the participants had more insights about such datasets and their search spaces. The next section in the post-interview dealt with specific UI features used. For each feature, the participants could indicate whether it was helpful for their analysis or unhelpful/distracting. For each of the questions, they could also write an optional answer specifying their problems. The UI features included canvas navigation, different perspectives, filtering features, sorting capabilities, and interactions based on the selection of rows. The last four questions were more general, where the participants were asked to write down what they found most difficult using the tool, what they most liked about it, whether they would use it again, and finally, what they would improve. The post-interview part took around 20 to 30 minutes, depending on the discussions.

Results. This section reports the most significant results of the study. Regarding their own approaches on how to tackle the dataset, none of the participants know of a readily available tool for such datasets and tasks. Five participants reported they would try out tools such as KNIME, Tableau, Charticulator, or spreadsheets to gain some insights into the data. Most of the participants would try to use Python and Jupyter notebooks to get insight into the data, such as manually filtering the data by attributes based on hypotheses and then running pattern-mining algorithms to gain insight. Four of the participants reported they would try to run pattern mining first and then use correlation measures with the attributes. Two participants state they would transfer the data into vector space to apply correlation measures and dimensionality reduction to find patterns and outliers. The second time this question was asked in the post-interview six participants answered they would not use their previously mentioned approaches at all. Three participants answered that they would still use Python/Jupyter lab approaches but only for hypothesis testing and not for exploration.

The questions about the search spaces revealed that the search space size for 60 attribute characteristics was much more intuitive for the participants to estimate, as seven out of 15 guessed correctly with the formula 2^{60} (Figure 17). However, the search space of sequential pattern mining was greatly underestimated. Only one person was able to derive the formulas and provide the correct answer. All other participants answered in ranges from 10^3 to 10^{200}, whereas the correct answer is 10^{686} (Figure 18). The actual number of subsequences in the VAST Challenge dataset

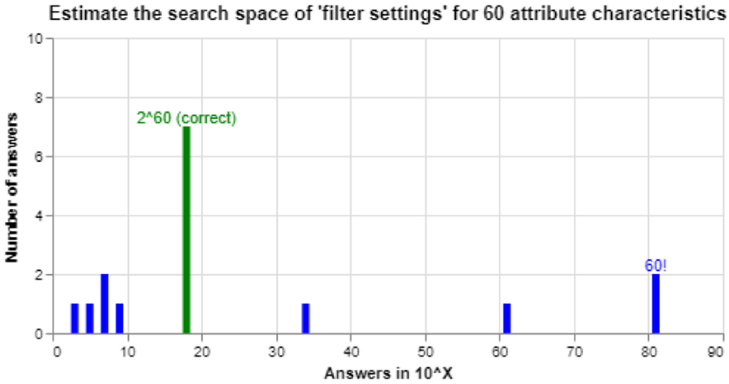


Fig. 17. Participants' estimates about the search space of attribute characteristics. The correct answer is marked in green. Note that the x-axis is logarithmic.

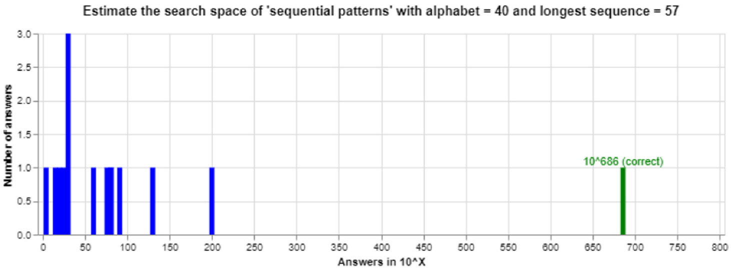


Fig. 18. Participants' estimates about the search space of sequential pattern mining. The correct answer is marked in green. Note that the x-axis is logarithmic. Only one participant estimated the correct amount by deriving the correct formulas. All other participants greatly underestimated the size of the search space.

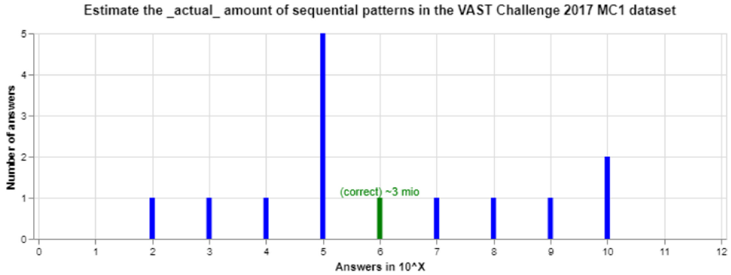
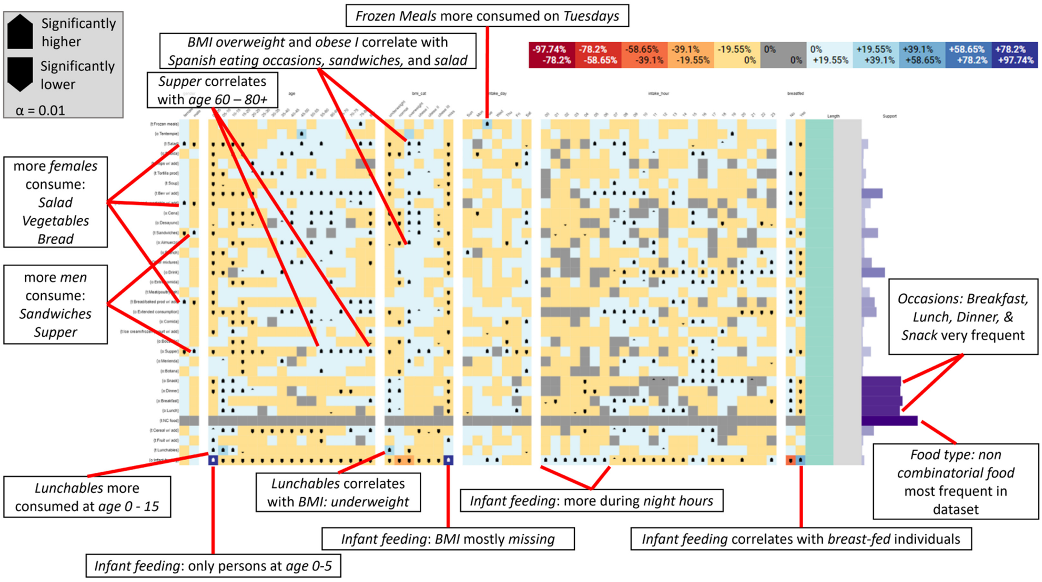


Fig. 19. Participants' estimates about the search space of sequential patterns that are actually in the data. The correct answer is marked in green. Note that the x-axis is logarithmic. Note that the participants may have been biased, as the correct answer to the previous question was given to them before answering it.

was guessed more correctly, however, eight of the participants responded with 10^2 to 10^5 and five participants with 10^7 to $10^1 0$ whereas only one participant estimated correctly with 10^6 (Figure 19).

During the paired analytics phase, the participants spent the most time with the view, as shown in Figure 20. As recommended by the interviewer, they reduced the length of the right table to 1, resulting in the table only displaying single occasions and food types. Furthermore, most participants activated the statistical overlay to understand better which correlations are significant with an alpha threshold of 5%. The annotations in the figure represent only some of the insights



Frozen Meals more consumed on *Tuesdays*

BMI overweight and *obese I* correlate with *Spanish eating occasions, sandwiches,* and *salad*

Supper correlates with *age 60 – 80+*

more *females* consume: *Salad* *Vegetables* *Bread*

more *men* consume: *Sandwiches* *Supper*

Occasions: Breakfast, Lunch, Dinner, & *Snack* very frequent

Food type: non combinatorial food most frequent in dataset

Lunchables more consumed at *age 0 - 15*

Lunchables correlates with *BMI: underweight*

Infant feeding: more during *night hours*

Infant feeding: *BMI mostly missing*

Infant feeding correlates with *breast-fed* individuals

Infant feeding: only persons at *age 0-5*

Fig. 20. The view the participants spent most of their time in. The length of the right table is reduced to 1 leaving only single occasions and food types per row (no combinations). The column groups represent the attributes of gender, age group, BMI category, intake day, intake hour, and breastfed. The statistical overlay is activated with all three possible tests with $\alpha = 0.01$. Several insights can be generated using this view. The notes only represent some of the insights of the participants. All of the annotated insights are significant by two of the statistical tests. As it is visible by the black glyphs, many more insights that can be derived only based on this view.

that the participants generated. Many more significant findings can be made as it is visible by the black glyphs on the pixels. However, even though they are significant, some findings were deemed noise or random occurrences, such as persons who consume frozen meals reported their meals on a Tuesday. Other findings, such as Lunchables are consumed more by children and teenagers, confirmed expectations. The BMI category underweight is overrepresented for the same group as the BMI has been designed for adults and tends to be too low for children. Later on, the participants increased the length filter again to two, which shows all data. Using browsing, filtering, and sorting, they analyzed the data in a hypothesis-driven manner.

Ten of the participants reported that they gained 5–10 new insights, and four participants reported that they could gain more than ten new insights. Only one participant reported gaining 1–2 new insights into the WWEIA dataset. Twelve of the participants were sure or assumed that they had missed or overlooked something in the data. The participants overwhelmingly reported needing more time with the tool and the dataset. Another limiting factor was the steep learning curve, especially in the interpretation of the various perspectives. Our participants found it difficult to understand which perspective is most suitable for a certain task. Another difficulty was memorizing the attributes and attribute characteristics (column groups and columns). The participants mentioned that they felt more comfortable navigating the tool at the end as they memorized the order. In the beginning, they needed to use the header and tooltip for orientation which slowed them down. Only three participants wished for more filtering options, such as filtering based on p-values. Four stated they would like to use a guidance feature. Again, this was turned off during the study as it was not stable enough.

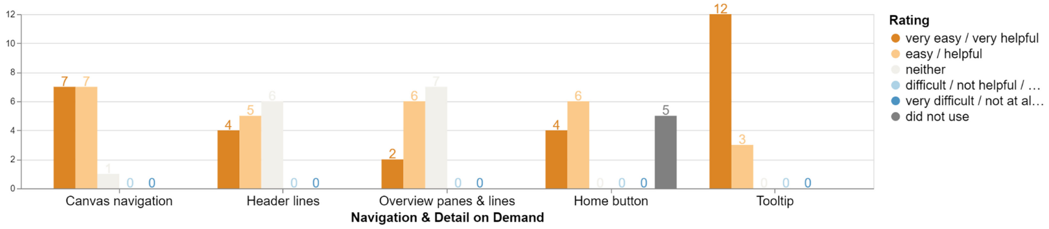


Fig. 21. The various ratings of the canvas navigation and features to support navigating the canvas. Note the slightly different answer-type for each question and note that the did not use option was only available for the home button.

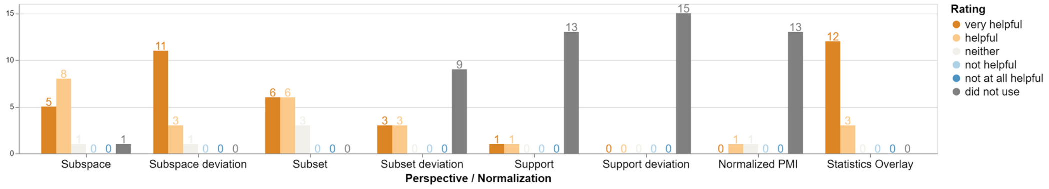
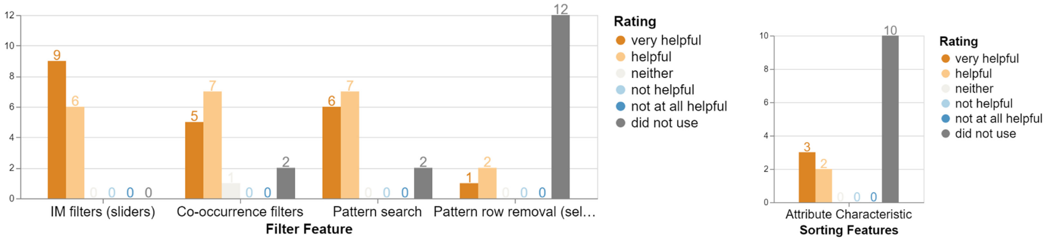


Fig. 22. The various ratings for the different perspectives/normalizations and the statistics overlay.

The canvas navigation was deemed as very easy or easy by 14 of the participants (see Figures 21 and 22). The study revealed that panning and zooming in non-chromium browsers are much laggier, as well as that the zoom levels jump quite a lot on MacOS. The feedback regarding the lines from the headers and the overview panes on either side was mixed, as six and seven participants reported they did not use them as an aid for navigation. Otherwise, these components were rated helpful or very helpful. Similarly, five of the participants have not used the "fly home" button. The feedback regarding the tooltip was overwhelmingly positive, with 12 participants stating "very helpful" and three participants rating it "helpful." It was, however, mentioned that the tooltip can be distracting during panning and zooming as it covers a large area of the space.

The feedback for the perspectives (normalizations, see Section 5.2) varied significantly (see Figure 22). The subspace, subspace deviation, and subset perspective were most used and rated helpful or very helpful by the majority. The subset deviation, support, support deviation, and normalized PMI perspective were mostly not used by the participants. All participants answered that the statistical overlay was very helpful or helpful for their analysis. Likewise, all participants reported that they did not miss any perspective or did not know whether any important perspective was missing. Again, many of the participants reported that they have difficulties interpreting the perspectives and choosing the appropriate perspective based on their task. They suggested more time/experience with the tool or more training as a countermeasure.

The interestingness measure filters (sliders) were rated helpful or very helpful by all participants (Figure 23(a)). The co-occurrence filters that allow filtering for specific pixel colors and the pattern search were less used but rated positively. The pattern row removal for a selection of rows was mostly not used by the participants. Only three participants used that feature and rated it very helpful or helpful. A similar reaction is visible for the attribute characteristic sorting as it was mostly not used, but the persons that used it rated it positively. Ten participants reported that they did not miss any additional filtering capabilities, three answered "I don't know," and two answered "Yes," where they specified that they wished for filtering based on p-values and that the filtering could be more specific such that AND and OR queries are possible. The filters are combined using



(a) Filter Features (b) Sorting Feature

Fig. 23. The ratings for filter and sorting features.

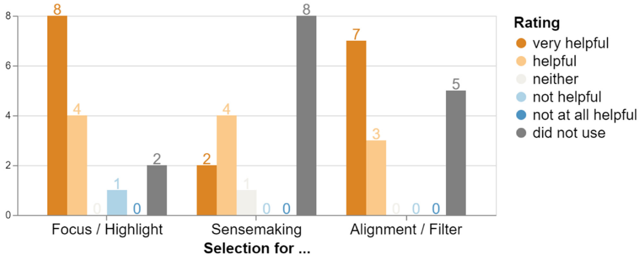


Fig. 24. The various selection features and use cases.

AND in the current state, whereas only the co-occurrence filters are ORed. Multiple participants expressed the wish to search for multiple filter queries in the pattern search. This would have been helpful for English and Spanish occasions or related food types. The ratings for missing sorting capabilities were similar, and the participants expressed the wish to sort patterns alphabetically and sort by multiple attribute characteristics of one attribute. Furthermore, it was remarked that the flipping of the tables (because of the MDS) could be confusing.

The selection features and use cases were also rated positively (Figure 24). The first question referred to a selection to highlight specific rows to follow the pixels along the horizontal axis better. Most participants rated this as helpful, only one as not helpful, and two did not use this feature. The participants stated that it was unintuitive that the selection only works on pixels and not on the row labels and that the tooltip can be distracting in this case. If too many rows are selected, the edge bundles connecting the two tables become overplotted, and in the worst case, the canvas navigation becomes laggy. Otherwise, they rated this feature as useful and necessary for exploration. The selection for sensemaking was mostly not used and referred to selecting one or multiple rows and then browsing the highlighted rows in the other table *without* using any additional filters. The participants reported that this is too tedious. However, combined with the alignment or filter (third question), this feature becomes incredibly valuable and is rated positively by most. The participants agreed that the selection and filtering should be two separate interactions as otherwise, it would be too confusing to follow the tool. No other feedback was provided regarding missing selection features.

The last part of the interview concerned more general feedback, where the first question was about what the participants found most difficult in using the tool/approach. The majority reported that the steep learning curve is an obstacle, especially with the many available perspectives and various options. They furthermore reported that a good entry point for the analysis would be helpful, such as an initial tour or guidance feature (which was disabled at the time). Otherwise, the long loading times of the tool were noted negatively.

Regarding the question of what the participants liked the most, the overwhelming answer was the good overview of the dataset combined with the many available correlations/visual patterns in a single image, allowing them to test the hypotheses extremely rapidly without much user interaction. Several users rated the tool as a truly exploratory analysis. Furthermore, it was appreciated that the filters and statistical overview helped quickly reduce the tables to find specific areas of interest and verify/falsify hypotheses. While the steep learning curve was remarked for the previous question, the participants acknowledged that once the tool and its capabilities are understood, the analysis becomes very easy and quick as it does not change between datasets even if the structured data is different. The option to flip the colormap was rated positively as for many participants, it was unintuitive that blue colors have a positive deviation as their mental state compared the colors to a temperature where a red color (warmer) equals higher values. Furthermore, it was appreciated that a CSV upload is provided, and 14 participants stated in a separate question that they would use the tool again for another analysis. One participant answered with "I don't know," as they were unsure whether they would have the right datasets in the future. The question on what they would improve was answered similarly to what they found most difficult. One participant mentioned that it would be good to hide specific attributes or attribute characteristics to better focus on the remaining ones. Four participants would like to see a recommender/guidance system to find a better entry point for the tool. The last question about miscellaneous feedback was answered by one participant with "A very powerful tool!"

In a post-study, we demonstrated the guidance system to four of the user study participants that specifically suggested such a feature for the tool. The participants agreed that the guidance system is useful as an onboarding process for the tool to showcase to the user what is possibly interesting and what statistical effects cause the visual effects in the interface. The four participants were satisfied with the current set of statistics and could not suggest any further statistics without making any specific assumptions about the dataset.

6.5 Lessons Learned and Recommended Workflow

In this section, we report about the lessons learned from our own usage with our implementation, as well as the paired analytics sessions with the domain experts and during the user study. Our approach is quite complex to understand at first as the paradigm on how to look at this type of data and task is very much different from what we entitled as the common approach using filters and analyzing the structured data separately. However, the participants report that the effort is rewarding as the tool is very powerful and can generate many insights within minutes. Furthermore, the workflow and concepts can easily be transferred to other datasets.

We derive a workflow for our approach based on our experiences plus the insights that we gained during paired analytics sessions with the domain experts and the user study. In general, we recommend removing the respective rows from the tables after a finding has been discovered to place the focus on the remaining data. We want to clarify that this derived workflow must not be strictly followed and should be more regarded as a recommendation. For exploratory analysis, many different workflows can lead to successful and interesting findings.

(1) Entry Point. In the beginning, the left table should be ignored. Additionally, it is helpful to reduce the length (green slider) of the right table to one. This will leave only single events per row which are easier to interpret (see, for example, Figure 20). The small table can then be explored and the visual representation can be better understood by the user. It also helps to turn on the statistical overlay.

(2) Verification and Rapid Hypothesis Testing. This phase is important to verify existing knowledge, learn to understand the visual properties and interaction possibilities of the tool, and build

trust [38, 42]. The search field is useful to filter for specific subsequences and sequences and then check the distributions of the attribute characteristics. A tooltip for each pixel reveals the absolute and relative frequencies, as well as the support and length of the respective (sub-)sequence.

(3) Extreme Frequencies & Diversions. Extreme frequencies and diversions are visualized with dark and saturated pixels. Even when zoomed out far, such pixels remain visible and can be easily identified. An example can be found in Figure 9. The user must be careful as extreme values may also occur for small subsets in the data. The support filter (purple bar charts) and significance visualization and filter can be exploited to remove irrelevant rows. Note that this phase also includes the search for extreme values of the interestingness measures (e.g., support and length) as shown in Figure 12.

(4) Visible Blocks. The dimensionality reduction forms visible blocks of equal attribute characteristics distributions (mostly in the subsequence table). Large blocks are typically due to long and specific subsequences that correlate with specific characteristics of attribute characteristics. Such *pixel-blocks* can be easily discovered with this approach (for example, in Figure 10). The "Overview Visualization" on each side supports the search for these blocks. The visible linking to the event sequence table and the detail-on-demand views help the user find the relevant sequences.

(5) Targeted Attribute Characteristics Exploration. In this phase, the user searches for deviations of specific attributes and attribute characteristics (i.e., column-blocks and single columns). This is supported by sorting the rows according to a characteristic or filtering by their frequency or significance. However, we recommend sorting because determining the appropriate thresholds for the filters can be quite difficult. This part of the workflow is used for example in Figure 13.

(6) Drill Down. When the attributes do not show any interesting distributions or diversions of distributions, this may be due to too general sub-entity. The user can now drill down into the search space by selecting specific rows in the low aggregation table and check whether the aggregations of the selected distinct structured entities show any interesting behavior. Another possibility is to select rows in the high aggregation table and drill down based on this selection. The selection can be made based on tendencies of diversions of the distributions, even though these diversions may not be significant yet. This effectively generates more specific sub-entities out of the selected structured (sub-) entities. At last, the user has also the possibility to mine for longer sub-entities within the whole dataset. This may lead, however, to a so-called pattern explosion where thousands of new rows may be added. The length filter (green horizontal bar chart) can be used to filter for a specific generation of structured (sub-) entities after the drill-down. Now, the user may start over with step one of the presented workflow.

7 DISCUSSION AND OUTLOOK

We report some properties of our approach and the implemented interactive visual interface and discuss limitations, scalability, and generalizability. Note that our implementation is just one possibility to further process and visualize our approach's output and that many other implementations or further processing steps such as subspace clustering, correlation measures, and dimensionality reduction techniques are imaginable.

7.1 Properties

The findings in the use cases nicely show that our approach is independent of the support of an structured entity or structured sub-entity, enabling a user to quickly find anomalies or unusual patterns in small subsets of the data. We argue that this is an essential property for many application areas, as the data is typically not uniformly distributed. Furthermore, the user often has great difficulties in defining such thresholds that essentially determine an interesting subspace's minimum size. This information is part of the exploration process itself. Our approach works with the lowest possible threshold of two for sub-entities and one for distinct structured entities avoiding this pitfall entirely. By increasing the thresholds, the approach works equally well for more frequent or otherwise significantly occurring patterns. A scenario with higher thresholds is equally possible with our approach and speeds up the pattern mining as more structured sub-entities are pruned, and it reduces the number of rows in the table, providing a better overview.

The pixel-based display enables the user to explore many distributions for multiple attributes simultaneously. The dimensionality reduction supports this analysis task. The available perspectives support the user in assessing the distributions of the attribute characteristics from different point-of-views. The statistics overview and guidance features support the user in finding meaningful insights in the dataset. In many cases, it is not necessary to mine all structured sub-entities as sub-entities of a lower generation are often specific enough to describe interesting subsets in the data, whereas the interestingness is defined by the distribution of the attribute characteristics and the objective IMs.

The structured sub-entities that have the same distributions in their attributes are grouped to form visual cues for the user. The larger such blocks are, the longer and more specific the respective sub-entity that can be mined. Even if deviations in the distributions are not significant, a tendency is often visible. In combination with the user's domain knowledge, drill-down operations can reveal the significant and interesting subspaces that correlate with the possibly even more interesting structured entity or sub-entity that are provided on demand.

7.2 Limitations

Our approach is based on the core assumption that specific aggregations of structured entities correlate with certain subspaces defined by structured sub-entities and thus show a deviation from the population of all loaded data or share a common interesting distribution in their attribute characteristics. This means that with our approach, outlier distributions that share the same structure as a larger cluster of structured entities can typically not be easily detected. One example might be that 100 patients share the exact same patient history (i.e., identical event sequence), but only one or two patients show a deviation in their attribute characteristics. This limitation can be mitigated with specific normalizations that highlight small co-occurrence values.

Another conceptual limitation that remains is over-aggregation. If a sub-entity has a large support (i.e., in the extreme case it is 100%), then there will be no deviation visible from the global population. Of course, this may hold some valuable information for the user, but this is a known pattern oftentimes. However, even with smaller supports, a sub-entity may aggregate too much data to show an interesting deviation from the global vector. The user always has the option to drill down into the search space. However, this requires that the user first identifies this sub-entity (i.e., row) as potentially interesting. Three arguments mitigate this limitation. First, real-world data is often sparse (curse of dimensionality), which causes redundancies in sub-entities and eventually forms what we call visible blocks in the tables. Second, because the co-occurrences are *a-priori*, attributes might not show significant deviations in the co-occurrence-distributions of the attribute characteristics, but tendencies are typically visible as the aggregation of multiple rows is simply a vector sum. Third, the low aggregation table that holds the distinct structured entities can be exploited to find interesting co-occurrence distributions, which the user can then aggregate by selecting the respective rows and drilling down into the search space. As a last resort, the user always can increase the *initial mining depth parameter* or select all rows and drill down into the search space, resulting in the same result.

Another limitation is the necessity for discrete attributes. This requires that continuous attributes (e.g., age) need to be binned by the user or by a (semi-) automatic approach. However, binning has a tremendous effect on the distribution and may also lead to over-aggregation. Therefore, the user must choose the binning appropriately and adjust it to the task and data at hand using her domain knowledge. Alternative representations as done in SMARTExplore [11] or UpSet plots [45] for numerical dimensions are possible. We consider this future work.

7.3 Scalability

We first have to mention that our proposed approach's scalability is more limited by our implementation than by the approach itself, as the interactive visual interface is quite sensitive to drawing hundreds of thousands of rows and columns per frame (i.e., triggered by an interaction). Also, our current implementation requires a lot of memory as the computation is performed in memory. The third use case demonstrates that the size of the alphabet has a much larger impact on the computation runtime than the number of input rows. We are, for example, currently not able to display the Kaggle dataset "Online Retail II UCI[3]" even though it only contains around 44.000 rows. However, the alphabet, determined by the product names, is greater than 3000, which produces around 60.000 structured sub-entities. This exceeds our available memory.

Because pixel-based representations allow the highest compression of information in a 2D space, this representation on a pannable and zoomable canvas scales in general well with the number of rows and columns and compares hundreds of distributions in different subsets of the data. The number of rows and columns is reduced by our approach, which would mean an exponential increment in both dimensions. The overview visualization helps the user in identifying visible blocks of identical or similar subspaces that indicate the existence of longer or many structured sub-entities that share a common subspace.

The number of attribute characteristics is proportional to the number of columns. Therefore, attributes containing many attribute characteristics for example the attribute *country of origin* may result in 195 attribute characteristics representing each country individually which would lead to 195 columns in both tables. One way to mitigate this is using more coarse hierarchies and allowing the user to interactively and individually unfold these hierarchies. In our example, this could mean that continents or other task-specific regions first aggregate the countries, and the user can extend each region to see the individual countries then. Because these columns are then aggregated, patterns may be missed. However, the aggregation only causes the co-occurrences to be summed such that general tendencies remain visible. Although this requires engineering efforts in the implementation, it is trivial to accomplish as the co-occurrences can be summed. In none of our experiments we saw the need to aggregate attribute characteristics as in today's screen sizes many columns can be displayed at once.

Although the search space reduction of our approach dramatically reduces the number of sub-entities as well as attribute characteristics, the resulting fraction of the general search space remains exponential. More specifically, the fraction of the structured data search space remains exponential, whereas the attribute search space fraction is linear. This means that adding an attribute characteristic will add another column to both tables whereas adding another attribute will add as many columns as attribute characteristics it contains. Therefore, there is no difference between adding one attribute with 100 attribute characteristics or adding two attributes with each 50 attribute characteristics. In conclusion, this means that with larger alphabets in the structured data, an exponentially large number of rows has to be calculated and visualized. However, this also heavily depends on the data's sparseness (i.e., is every possible combination represented in the data?).

[3] https://www.kaggle.com/mashlyn/online-retail-ii-uci/version/3.

Possible mitigations to reduce the number of rows are to increase the minimum thresholds for the respective IMs, introduce or tighten other (structural) constraints, and remove certain items from the alphabet or combine them into new items.

7.4 Generalizeability

While our proposed approach only requires structured data in combination with discrete attributes, any implementation must be tailored to the type of structured data and useful IMs for the task at hand. As we describe in Section 5.5, state-of-the-art algorithms must be slightly modified to support interactive mining. The required modifications should be similar in design and mostly independent of the selected algorithm. Because our interactive visualization does not display any structured data in the overview, it is quite agnostic to any data structure, as well as IM. Furthermore, other IMs that follow our broad definition (Table 1) can be simply added as additional bar charts. The highest degree of customization is implemented in the detail-on-demand views such as the tooltip where, in our implementation, event sequences and sub-sequences are displayed in a textual form as well as a map for the VAST Challenge 2017 Mini Challenge 1 use case. Such customizations to the data structure and the underlying semantics are useful to the user [36].

Regarding the data-users-tasks design-triangle of Miksch and Aigner [48], our approach and interactive visualization are most constrained to the task. As we have briefly addressed already, the task of finding interesting subspaces in structured data is most often combined with other tasks at hand. Therefore, we envision our approach and visualization as a pre-selection to find interesting subspaces, where the user can identify and select these subspaces to continue in another view or window to analyze these subspaces in further detail.

It is also possible to completely abandon the visualization and further process our approach's output using statistics, subspace clustering, and other machine-learning techniques to identify interesting subspaces. With this, our approach can function as a preprocessing step during the analysis.

Our guidance system is implemented in an extendible fashion such that arbitrary statistics and machine learning methods may be used to identify possibly interesting findings in the data. It is, however, quite difficult to implement such statistics without making specific assumptions about the dataset. That is because the notion of interestingness is not only based on statistics but equally influenced by the semantics of the patterns in the structured data and the attribute characteristics that form the co-occurrences. A good example is shown in Figure 20, where the participants of our study identified a correlation of *frozen meals* are being consumed more on *Tuesdays*. Even though the corresponding pixel is statistically significant and has a relatively dark color compared to most other pixels, our study participants labeled this correlation as noise or a random occurrence. On the other hand, other correlations such as eating occasions written in Spanish correlating with higher BMIs, or *Lunchables* correlating with children and therefore with the BMI class *underweight* were deemed more interesting even though the visuals are less expressive (yet still visible) in the overview. By encoding such information in the data before the analysis, it is possible to identify potentially better findings concerning the users' interests. However, encoding such knowledge is tedious and likely more of a burden to the user than spending time in an overview visualization where all (required) correlations are visible at once and can be interactively filtered and sorted based on attribute characteristics and patterns.

7.5 Outlook

An obvious extension to this work is further use cases and applications using various types of structured data as well as IMs. Specifically, rule mining such as association rule mining or sequential rule mining could introduce new IMs that can be combined with our approach. This could also result in novel normalizations enabling different perspectives on the co-occurrences (Section 5.2). Another interesting extension would be high-utility pattern mining [24]. Furthermore, algorithms, such as Ditto [9], could be explored which in the optimal case would produce the optimal number of rows and would not require the user to drill down into the search space manually. On the other hand, we show that our best practice of leaving the initial mining depth parameter (length) at 2 works well for most real-world datasets. Only for datasets with a smaller alphabet, this parameter should be increased.

Because our approach only utilizes vector sums, other numbers such as uncertainty values could be reliably propagated throughout our proposed approach. This could be achieved either as values accompanying the co-occurrence values or as dedicated attributes that represent uncertainty. Using the pixel-based approach, multiple uncertainties could be modeled and compared across subspaces.

Representative learning [15, 67] and representation learning [8] may be a useful addition to providing a more concise overview than our approach is capable of. Learning respective kernels and other encodings for structured data is possible [53], yet, one strength of our approach is its applicability to any arbitrary data structure without relying on an ensemble of pre-trained models and the fact that our approach is deterministic. Similarly, representation learning may be used to find better and more concise representations for the attribute characteristics, eventually reducing the number of columns. In any case, we strongly argue that the semantics of the structured data and the semantics of the attribute characteristics should remain as our evaluation depicts how much these semantics influence the users in their determination of relevance and interestingness for a finding. Machine learning can be exploited to learn such patterns based on labeled data, but it is quite difficult to transfer these models onto new data sets and structures.

8 CONCLUSIONS

Our contribution is two-fold. We first propose a novel approach with a paradigm shift to process structured data in combination with discrete attributes to find interesting subspaces based on co-occurrence values and traditional interestingness measures using state-of-the-art, slightly modified, pattern mining algorithms. Subspace clustering algorithms typically require numerical dimensions and are sensitive to the number of dimensions. The latter is also true for dashboards that support the common "cross-filtering" which we describe as the common approach in the introduction. Our approach turns the problem around and, thus, scales linearly with additional attribute dimensions, while showing all relevant subspaces in a single, condensed picture that is to be explored by the user. Our approach is tailored only to the task of identifying subspaces but agnostic to any structured data type, interestingness measure, and attributes. Because this data and task require the analysis of two exponential search spaces, it is often difficult to process and visualize these amounts of data. Therefore, we introduce two measures to reduce the search spaces dramatically, outputting two tables that represent the boundaries of said search space. We prove this search space reduction's legitimacy by showing that the used co-occurrence values are *a-priori* and explain how this can be exploited with high dimensional search spaces (curse of dimensionality). We then implement our approach into an interactive visual interface supporting the user in the exploratory data analysis task of identifying interesting subspaces. Our design of a pixel-based representation on a zoomable and pannable canvas shows good scalability, further mitigating the exponential search spaces' effects. The four use cases using a gold-standard dataset, a small study with domain experts, a large dataset to showcase the scalability, and a user study with 15 participants underline our approach and visualization's applicability and show that all of our requirements are satisfied. Based on the domain experts and the participants of our user study, we derive a suitable workflow for our approach combined with the interactive visual interface.

REFERENCES

[1] Charu C. Aggarwal and Jiawei Han (Eds.). 2014. *Frequent Pattern Mining*. Springer. https://doi.org/10.1007/978-3-319-07821-2

[2] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. 1996. Fast discovery of association rules. In *Advances in Knowledge Discovery and Data Mining*, Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy (Eds.). AAAI/MIT Press, 307–328.

[3] Mihael Ankerst, Anne Kao, Rodney Tjoelker, and Changzhou Wang. 2008. DataJewel: Integrating visualization with temporal data mining. In *Visual Data Mining*. Lecture Notes in Computer Science, Vol. 4404. Springer, 312–330.

[4] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. 2002. Sequential pattern mining using a bitmap representation. In *KDD*. ACM, 429–435.

[5] Sara Di Bartolomeo, Yixuan Zhang, Fangfang Sheng, and Cody Dunne. 2021. Sequence braiding: Visual overviews of temporal event sequences and attributes. *IEEE Trans. Vis. Comput. Graph.* 27, 2 (2021), 1353–1363. https://doi.org/10.1109/TVCG.2020.3030442

[6] Christian Baumgartner, Claudia Plant, Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. 2004. Subspace selection for clustering high-dimensional data. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1–4 November 2004, Brighton, UK*. IEEE Computer Society, 11–18. https://doi.org/10.1109/ICDM.2004.10112

[7] M. Behrisch, D. Streeb, F. Stoffel, D. Seebacher, B. Matejek, S. H. Weber, S. Mittelstaedt, H. Pfister, and D. Keim. 2018. Commercial visual analytics systems-advances in the big data analytics field. *IEEE Transactions on Visualization and Computer Graphics* (2018), 1–1. https://doi.org/10.1109/TVCG.2018.2859973

[8] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 8 (2013), 1798–1828.

[9] Roel Bertens, Jilles Vreeken, and Arno Siebes. 2016. Keeping it short and simple: Summarising complex event sequences with multivariate patterns. In *KDD*. ACM, 735–744.

[10] Kevin S. Beyer and Raghu Ramakrishnan. 1999. Bottom-up computation of sparse and iceberg CUBEs. In *SIGMOD Conference*. ACM Press, 359–370.

[11] Michael Blumenschein, Michael Behrisch, Stefanie Schmid, Simon Butscher, Deborah R. Wahl, Karoline Villinger, Britta Renner, Harald Reiterer, and Daniel A. Keim. 2018. SMARTexplore: Simplifying high-dimensional data analysis through a table-based visual analytics approach. In *VAST*. IEEE, 36–47.

[12] Francesco Bonchi and Claudio Lucchese. 2004. On closed constrained frequent pattern mining. In *Proceedings of the 4th IEEE International Conference on Data Mining (ICDM 2004), 1–4 November 2004, Brighton, UK*. IEEE Computer Society, 35–42. https://doi.org/10.1109/ICDM.2004.10093

[13] Simon Butscher, Yunlong Wang, Katrin Ziesemer, Karoline Villinger, Deborah Wahl, Laura König, Gudrun Sproesser, Britta Renner, Harald T. Schupp, and Harald Reiterer. 2016. Lightweight visual data analysis on mobile devices: Providing self-monitoring feedback. In *VVH 2016-1st International Workshop on Valuable Visualization of Healthcare Information*. 28–34.

[14] Bram C. M. Cappers and Jarke J. van Wijk. 2018. Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE Trans. Vis. Comput. Graph.* 24, 1 (2018), 532–541.

[15] Mayanka Chandrashekar and Yugyung Lee. 2021. Class representative learning for zero-shot learning using purely visual data. *SN Comput. Sci.* 2, 4 (2021), 313.

[16] Hong Cheng, Xifeng Yan, and Jiawei Han. 2014. Mining graph patterns. In *Frequent Pattern Mining*, Charu C. Aggarwal and Jiawei Han (Eds.). Springer, 307–338. https://doi.org/10.1007/978-3-319-07821-2_13

[17] Leah Findlater and Howard J. Hamilton. 2003. Iceberg-cube algorithms: An empirical evaluation on synthetic and real data. *Intell. Data Anal.* 7, 2 (2003), 77–97.

[18] U.S. Centers for Disease Control and Prevention. 2022. National Health and Nutrition Examination Survey; 2017–2018 Data Documentation, Codebook, and Frequencies; Individual Foods, First Day (DR1IFF_J). https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DR1IFF_J.htm.

[19] U.S. Centers for Disease Control and Prevention. 2022. National Health and Nutrition Examination Survey; 2017–2018 Data Documentation, Codebook, and Frequencies; Body Measures (BMX_J). https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/BMX_J.htm.

[20] U.S. Centers for Disease Control and Prevention. 2022. National Health and Nutrition Examination Survey; 2017–2018 Data Documentation, Codebook, and Frequencies; Demographic Variables and Sample Weights (DEMO_J). https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO_J.htm.

[21] Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas. 2014. Fast vertical mining of sequential patterns using co-occurrence information. In *PAKDD (1) (Lecture Notes in Computer Science)*, Vol. 8443. Springer, 40–52.

[22] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Bay Vo, Tin Chi Truong, Ji Zhang, and Hoai Bac Le. 2017. A survey of itemset mining. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 7, 4 (2017). https://doi.org/10.1002/widm.1207

[23] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. 2017. A survey of sequential pattern mining. *Data Science and Pattern Recognition* 1, 1 (2017), 54–77.

[24] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Roger Nkambou, Bay Vo, and Vincent S. Tseng. 2019. *High-Utility Pattern Mining*. Springer.

[25] Guojun Gan and Jianhong Wu. 2004. Subspace clustering for high dimensional categorical data. *SIGKDD Explor.* 6, 2 (2004), 87–94. https://doi.org/10.1145/1046456.1046468

[26] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Philip S. Yu. 2019. A survey of parallel sequential pattern mining. *ACM Trans. Knowl. Discov. Data* 13, 3 (2019), 25:1–25:34. https://doi.org/10.1145/3314107

[27] David Gotz and Harry Stavropoulos. 2014. DecisionFlow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 1783–1792.

[28] Gösta Grahne, Laks V. S. Lakshmanan, Xiaohong Wang, and Ming Hao Xie. 2001. On dual mining: From patterns to circumstances, and back. In *ICDE*. IEEE Computer Society, 195–204.

[29] Stephan Günnemann. 2012. *Subspace Clustering for Complex Data*. Ph.D. Dissertation. RWTH Aachen University. http://darwin.bth.rwth-aachen.de/opus3/volltexte/2012/4103.

[30] Jiawei Han, Jian Pei, and Yiwen Yin. 2000. Mining frequent patterns without candidate generation. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16–18, 2000, Dallas, Texas, USA*, Weidong Chen, Jeffrey F. Naughton, and Philip A. Bernstein (Eds.). ACM, 1–12. https://doi.org/10.1145/342009.335372

[31] Mark Harrower and Cynthia A. Brewer. 2003. ColorBrewer. org: An online tool for selecting colour schemes for maps. *The Cartographic Journal* 40, 1 (2003), 27–37.

[32] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. 2000. Algorithms for association rule mining - A general survey and comparison. *SIGKDD Explorations* 2, 1 (2000), 58–64. https://doi.org/10.1145/360402.360421

[33] David Caster Hoaglin, Frederick Mosteller, and John Wilder Tukey. 1983. *Understanding Robust and Exploratory Data Analysis*. Vol. 3. Wiley New York.

[34] Alfred Inselberg and Bernard Dimsdale. 1990. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *IEEE Visualization*. IEEE Computer Society Press, 361–378.

[35] Dominik Jäckle, Michael Hund, Michael Behrisch, Daniel A. Keim, and Tobias Schreck. 2017. Pattern trails: Visual analysis of pattern transitions in subspaces. In *VAST*. IEEE Computer Society, 1–12.

[36] Wolfgang Jentner and Daniel A. Keim. 2019. *Visualization and Visual Analytic Techniques for Patterns*. Springer International Publishing, Chapter 12, 303–337. https://doi.org/10.1007/978-3-030-04921-8

[37] Wolfgang Jentner, Dominik Sacha, Florian Stoffel, Geoffrey P. Ellis, Leishi Zhang, and Daniel A. Keim. 2018. Making machine intelligence less scary for criminal analysts: Reflections on designing a visual comparative case analysis tool. *The Visual Computer* 34, 9 (2018), 1225–1241.

[38] Wolfgang Jentner, Rita Sevastjanova, Florian Stoffel, Daniel A. Keim, Jürgen Bernard, and Mennatallah El-Assady. 2018. Minions, sheep, and fruits: Metaphorical narratives to explain artificial intelligence and build trust. In *Workshop on Visualization for AI Explainability*.

[39] Aída Jiménez, Fernando Berzal, and Juan Carlos Cubero Talavera. 2010. Frequent tree pattern mining: A survey. *Intell. Data Anal.* 14, 6 (2010), 603–622. https://doi.org/10.3233/IDA-2010-0443

[40] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. 1994. Finding interesting rules from large sets of discovered association rules. In *CIKM*. ACM, 401–407.

[41] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. 2012. Subspace clustering. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2, 4 (2012), 351–364.

[42] John D. Lee and Katrina A. See. 2004. Trust in automation: Designing for appropriate reliance. *Human Factors* 46, 1 (2004), 50–80. https://doi.org/10.1518/hfes.46.1.50.30392

[43] Dirk J. Lehmann and Holger Theisel. 2016. Optimal sets of projections of high-dimensional data. *IEEE Trans. Vis. Comput. Graph.* 22, 1 (2016), 609–618. https://doi.org/10.1109/TVCG.2015.2467132

[44] Alexander Lex and Nils Gehlenborg. 2014. Points of view: Sets and intersections. *Nature Methods* 11, 8 (2014), 779.

[45] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. 2014. UpSet: Visualization of intersecting sets. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 1983–1992.

[46] Bing Liu, Wynne Hsu, Shu Chen, and Yiming Ma. 2000. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems* 15, 5 (2000), 47–55.

[47] Kenneth McGarry. 2005. A survey of interestingness measures for knowledge discovery. *Knowledge Eng. Review* 20, 1 (2005), 39–61. https://doi.org/10.1017/S0269888905000408

[48] Silvia Miksch and Wolfgang Aigner. 2014. A matter of time: Applying a data-users-tasks design triangle to visual analytics of time-oriented data. *Comput. Graph.* 38 (2014), 286–290. https://doi.org/10.1016/j.cag.2013.11.002

[49] Girivar Modi, Sanjay Bansal, and Mr. Anil Patidar. 2018. A survey on sequential rule mining techniques. *International Journal For Technological Research In Engineering* 6, 3 (2018).

[50] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. 2013. Temporal event sequence simplification. *IEEE Trans. Vis. Comput. Graph.* 19, 12 (2013), 2227–2236.

[51] U.S. Department of Agriculture. 2022. What We Eat In America (WWEIA) Database. https://data.nal.usda.gov/dataset/what-we-eat-america-wweia-database.

[52] World Health Organization. 2019. World Health Organization - Body mass index. http://www.euro.who.int/en/health-topics/disease-prevention/nutrition/a-healthy-lifestyle/body-mass-index-bmi. Accessed: 2019-03-27.

[53] Benjamin Paaßen, Claudio Gallicchio, Alessio Micheli, and Alessandro Sperduti. 2019. Embeddings and representation learning for structured data. In *ESANN*.

[54] Lance Parsons, Ehtesham Haque, and Huan Liu. 2004. Subspace clustering for high dimensional data: A review. *SIGKDD Explorations* 6, 1 (2004), 90–105.

[55] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. 2001. PrefixSpan: Mining sequential patterns by prefix-projected growth. In *ICDE*. IEEE Computer Society, 215–224.

[56] Adam Perer and Fei Wang. 2014. Frequence: Interactive mining and visualization of temporal frequent event sequences. In *IUI*. ACM, 153–162.

[57] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal. 2001. Multi-dimensional sequential pattern mining. In *CIKM*. ACM, 81–88.

[58] Panida Songram, Veera Boonjing, and Sarun Intakosum. 2006. Closed multidimensional sequential pattern mining. In *ITNG*. IEEE Computer Society, 512–517.

[59] Charles D. Stolper, Adam Perer, and David Gotz. 2014. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Trans. Vis. Comput. Graph.* 20, 12 (2014), 1653–1662.

[60] Guodao Sun, Sujia Zhu, Qi Jiang, Wang Xia, and Ronghua Liang. 2021. EvoSets: Tracking the sensitivity of dimensionality reduction results across subspaces. *IEEE Transactions on Big Data* (2021).

[61] Andrada Tatu, Fabian Maass, Ines Färber, Enrico Bertini, Tobias Schreck, Thomas Seidl, and Daniel A. Keim. 2012. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *7th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2012, Seattle, WA, USA, October 14–19, 2012*. IEEE Computer Society, 63–72. https://doi.org/10.1109/VAST.2012.6400488

[62] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. 2009. Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.* 10 (2009), 66–71.

[63] Karoline Villinger, Deborah R. Wahl, Gudrun Sproesser, Harald T. Schupp, and Britta Renner. 2017. A visual analysis of the behavioral signature of eating: The case of breakfast. *The European Health Psychologist* 19, Supp (2017), 689.

[64] Katerina Vrotsou, Jimmy Johansson, and Matthew D. Cooper. 2009. ActiviTree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Trans. Vis. Comput. Graph.* 15, 6 (2009), 945–952.

[65] D. R. Wahl, K. Villinger, M. Blumenschein, L. M. König, K. Ziesemer, G. Sproesser, H. T. Schupp, and B. Renner. 2018. Why we eat what we eat: Dispositional and in-the-moment eating motives. Manuscript submitted for publication.

[66] Deborah R. Wahl, Karoline Villinger, Gudrun Sproesser, Harald T. Schupp, and Britta Renner. 2017. The behavioral signature of snacking: A visual analysis. *The European Health Psychologist* 19, 5 (2017), 355–357.

[67] Jianlong Wang, Biao Hou, Licheng Jiao, and Shuang Wang. 2021. Representative learning via span-based mutual information for PolSAR Image classification. *Remote. Sens.* 13, 9 (2021), 1609.

[68] Taowei David Wang, Catherine Plaisant, Ben Shneiderman, Neil Spring, David Roseman, Greg Marchand, Vikramjit Mukherjee, and Mark S. Smith. 2009. Temporal summaries: Supporting temporal categorical searching, aggregation and comparison. *IEEE Trans. Vis. Comput. Graph.* 15, 6 (2009), 1049–1056.

[69] Mark A. Whiting, Kris Cook, R. Jordan Crouser, John Fallon, Georges Grinstein, Jereme Haack, Cindy Henderson, Kristen Ligget, Diane Staheli, Jana Strasburg, Jerry Tagestad, and Carrie Varley. 2017. VAST Challenge 2017: Mystery at the Wildlife Preserve. http://vacommunity.org/dl516. Accessed: 2018-12-12.

[70] Mark A. Whiting, Kris Cook, R. Jordan Crouser, John Fallon, Georges Grinstein, Jereme Haack, Cindy Henderson, Kristen Liggett, Diane Staheli, Jana Strasburg, Jerry Tagestad, and Carrie Varley. 2017. VAST Challenge 2017 Mini Challenge 1. http://www.vacommunity.org/VAST+Challenge+2017+MC1. Accessed: 2018-12-12.

[71] Mark A. Whiting, Kris Cook, R. Jordan Crouser, John Fallon, Georges Grinstein, Jereme Haack, Cindy Henderson, Kristen Liggett, Diane Staheli, Jana Strasburg, Jerry Tagestad, and Carrie Varley. 2017. VAST Challenge 2017 Reviewer Guide: Mini-Challenge 1. https://www.cs.umd.edu/hcil/varepository/VAST%20Challenge%202017/challenges/Mini-Challenge%201/solution/VAST%20Challenge%202017%20MC1%20Solution%20Final.pdf. Accessed: 2019-03-27.

[72] Wikipedia. 2021. Hasse diagram. https://en.wikipedia.org/wiki/Hasse_diagram.

[73] Krist Wongsuphasawat and David Gotz. 2012. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Trans. Vis. Comput. Graph.* 18, 12 (2012), 2659–2668.

[74] Yalong Yang, Wenyu Xia, Fritz Lekschas, Carolina Nobre, Robert Krüger, and Hanspeter Pfister. 2022. The pattern is in the details: An evaluation of interaction techniques for locating, searching, and contextualizing details in multivariate matrix visualizations. *CoRR* abs/2203.05109 (2022).

[75] Mohammed Javeed Zaki. 2000. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.* 12, 3 (2000), 372–390. https://doi.org/10.1109/69.846291

[76] Mingzhu Zhang and Changzheng He. 2010. Survey on association rules mining algorithms. In *Advancing Computing, Communication, Control and Management*. Springer, 111–118.

Received 1 March 2021; accepted 15 October 2022