Visualization and Visual Analytic Techniques for Patterns

Wolfgang Jentner and Daniel A. Keim (authors' preprint)

Abstract This chapter surveys visualization techniques for frequent itemsets, association rules, and sequential patterns. The human is crucial in the process of identifying interesting patterns and thus, mining such patterns and visualizing them is important for the decision making. The complementary feedback loop that a user may use to refine parameters through inspecting the current mining results is broadly described as visual analytics. This survey identifies visual designs for patterns of each category and analyzes and compares their strengths and weaknesses systematically. The comparison and overview help decision-makers selecting the appropriate technique for their tasks and systems while knowing about their limitations.

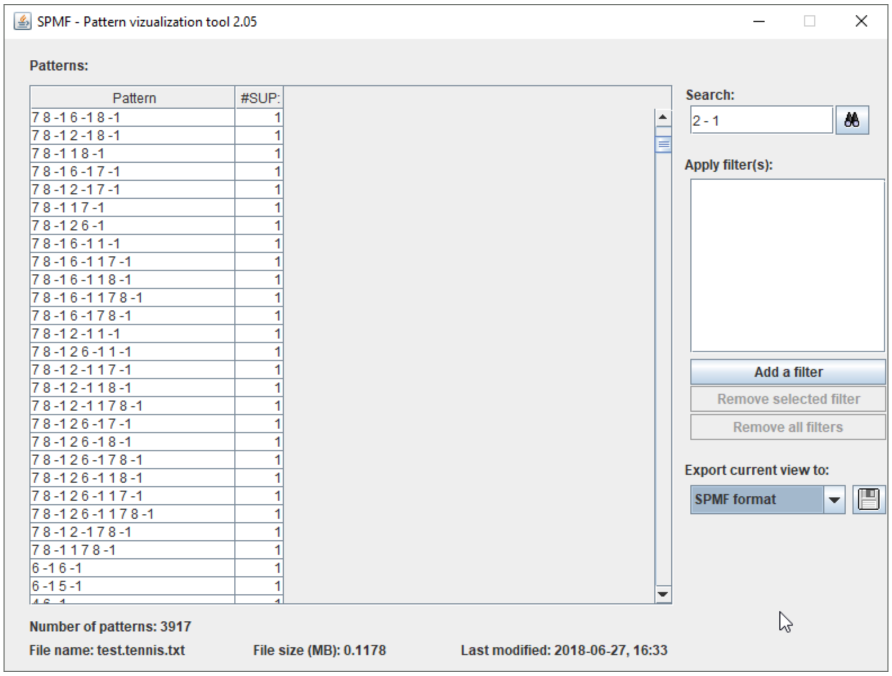
1 Introduction

"Exploratory data analysis isolates patterns and features of the data and reveals these forcefully to the analyst." [46] Frequent pattern mining is an important concept in data mining and exploratory data analysis [1]. Early on it became clear that visualization *is* required in the KDD process as only the human, as the ultimate decision maker, can identify the interesting information. The terms visual data exploration and visual data mining emerged describing efforts to integrate the user into the data mining process. In 2004, the term visual analytics gained recognition for a broader context of a multidisciplinary research field [24]. It is of special interest how users can integrate their knowledge into the data analysis process and eventually generate more knowledge [81].

As frequent pattern mining tends to produce many patterns, a lot of research has been devoted in finding interestingness measures filtering and ranking useful and interesting patterns for the user [31, 1]. Multiple surveys cover the algorithmic side of the mining process for itemsets [33, 16, 28], association rules [56, 45], and sequential patterns [102, 69, 29]. Implementations of these algorithms can be found in libraries such as WEKA [92] or SPMF [27] or for example in the FIMI repository [34, 52]. The output of the implemented algorithms is typically presented to the user in a textual form. This imposes, however, many limitations. In general, the cognitive load in identifying patterns, understanding their relations, and comparing

Wolfgang Jentner
Universität Konstanz; Universitätsstr. 10, 78467 Konstanz; e-mail: `wolfgang.jentner@uni.kn`

Daniel A. Keim
Universität Konstanz; Universitätsstr. 10, 78467 Konstanz; e-mail: `keim@uni.kn`



Patterns:

Pattern	#SUP:
7 8 -1 6 -1 8 -1	1
7 8 -1 2 -1 8 -1	1
7 8 -1 1 8 -1	1
7 8 -1 6 -1 7 -1	1
7 8 -1 2 -1 7 -1	1
7 8 -1 1 7 -1	1
7 8 -1 2 6 -1	1
7 8 -1 6 -1 1 -1	1
7 8 -1 6 -1 1 7 -1	1
7 8 -1 6 -1 1 8 -1	1
7 8 -1 6 -1 1 7 8 -1	1
7 8 -1 6 -1 7 8 -1	1
7 8 -1 2 -1 1 -1	1
7 8 -1 2 6 -1 1 -1	1
7 8 -1 2 -1 1 7 -1	1
7 8 -1 2 -1 1 8 -1	1
7 8 -1 2 -1 1 7 8 -1	1
7 8 -1 2 6 -1 7 -1	1
7 8 -1 2 6 -1 8 -1	1
7 8 -1 2 6 -1 7 8 -1	1
7 8 -1 2 6 -1 1 8 -1	1
7 8 -1 2 6 -1 1 7 -1	1
7 8 -1 2 6 -1 1 7 8 -1	1
7 8 -1 2 -1 7 8 -1	1
7 8 -1 1 7 8 -1	1
6 -1 6 -1	1
6 -1 5 -1	1
4 6 -1	1

Search: 2 - 1

Apply filter(s):

Export current view to: SPMF format

Number of patterns: 3917

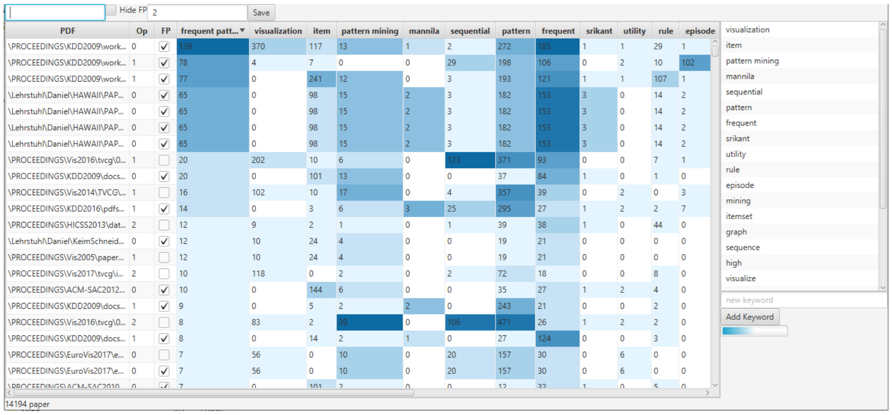
File name: test.tennis.txt File size (MB): 0.1178 Last modified: 2018-06-27, 16:33

Fig. 1: The *Pattern Viewer* of the Sequential Pattern Mining Framework (SPMF).

their interestingness measures (e.g., support) is high. Furthermore, these represented patterns are not visually aggregated which supports the user to browse and explore the generated pattern space.

The SPMF library features a *Pattern Viewer* which outputs patterns in a textual form inside a table with the interesting measures in the respective columns (Figure 1). The viewer supports the interactive exploration of patterns through filtering the patterns by a given string (positive and negative templating), by their interestingness measures (e.g., greater than a given value), and by ordering the rows. While such interaction capabilities are important to explore a pattern space, the textual representation does not allow a user to easily identify and relate patterns as well as viewing them in a more compact representation.

To the best of our knowledge, there exists no survey dedicated specifically to the visualization techniques of such patterns. There are, however, surveys in related fields such as set visualizations [8] or works on time-oriented data visualization [7, 84, 6] which is related to visualizing event sequences and sequential patterns. Other papers feature an extensive related work section, e.g., [48]. Hahsler and Chelluboina provide a survey of visualization for association rules in their paper for the R-package arulesViz [39]. In this chapter, we contribute a survey that thoroughly surveys visualization techniques for each of the specializations and systematically analyzes their strengths and weaknesses.



☐ Hide FP 2 Save

PDF	Op	FP	frequent patt...▾	visualization	item	pattern mining	mannila	sequential	pattern	frequent	srikant	utility	rule	episode
\PROCEEDINGS\KDD2009\work...	0	☑	139	370	117	13	1	2	272	185	1	1	29	1
\PROCEEDINGS\KDD2009\work...	1	☑	78	4	7	0	0	29	198	106	0	2	10	102
\PROCEEDINGS\KDD2009\work...	1	☑	77	0	241	12	0	3	193	121	1	1	107	1
\Lehrstuhl\Daniel\HAWAII\PAP...	0	☑	65	0	98	15	2	3	182	153	3	0	14	2
\Lehrstuhl\Daniel\HAWAII\PAP...	0	☑	65	0	98	15	2	3	182	153	3	0	14	2
\Lehrstuhl\Daniel\HAWAII\PAP...	0	☑	65	0	98	15	2	3	182	153	3	0	14	2
\Lehrstuhl\Daniel\HAWAII\PAP...	0	☑	65	0	98	15	2	3	182	153	3	0	14	2
\PROCEEDINGS\Vis2016\tvcg\0...	1	☐	20	202	10	6	0	123	371	93	0	0	7	1
\PROCEEDINGS\KDD2009\docs...	0	☑	20	0	101	13	0	0	37	84	1	0	1	0
\PROCEEDINGS\Vis2014\TVCG\...	1	☐	16	102	10	17	0	4	357	39	0	2	0	3
\PROCEEDINGS\KDD2016\pdfs...	1	☑	14	0	3	6	3	25	295	27	1	2	2	7
\PROCEEDINGS\HICSS2013\dat...	2	☐	12	9	2	1	0	1	39	38	1	0	44	0
\Lehrstuhl\Daniel\KeimSchneid...	0	☑	12	10	24	4	0	0	19	21	0	0	0	0
\PROCEEDINGS\Vis2005\paper...	1	☐	12	10	24	4	0	0	19	21	0	0	0	0
\PROCEEDINGS\Vis2017\tvcg\i...	2	☐	10	118	0	2	0	2	72	18	0	0	8	0
\PROCEEDINGS\ACM-SAC2012...	0	☑	10	0	144	6	0	0	35	27	1	2	4	0
\PROCEEDINGS\KDD2009\docs...	1	☑	9	0	5	2	2	0	243	21	0	0	2	0
\PROCEEDINGS\Vis2016\tvcg\0...	2	☐	8	83	2	39	0	106	471	26	1	2	2	0
\PROCEEDINGS\KDD2009\docs...	1	☑	8	0	14	2	1	0	27	124	0	0	3	0
\PROCEEDINGS\EuroVis2017\e...	0	☐	7	56	0	10	0	20	157	30	0	6	0	0
\PROCEEDINGS\EuroVis2017\e...	0	☑	7	56	0	10	0	20	157	30	0	6	0	0
\PROCEEDINGS\ACM-SAC2010...	0	☑	7	0	101	2	0	0	12	32	1	0	5	0

14194 paper

visualization
item
pattern mining
mannila
sequential
pattern
frequent
srikant
utility
rule
episode
mining
itemset
graph
sequence
high
visualize

Fig. 2: Our prototype to scan a large set of academic work.

The next Section explains our used methodology. Section 3 describes the visualization techniques for itemsets, association rules, and sequential patterns. Afterward, we compare and analyze the visualization techniques discovering their advantages and drawbacks. We conclude our chapter in Section 6 where we bridge the gap to utility mining and identify further research challenges.

2 Methodology

We use our internal database which includes conference and workshop proceedings (e.g., Vis[1], EuroVis[2], KDD[3]), journals (e.g., TVCG[4], Information Visualization[5]), and a variety of books. Using a set of keywords that are typical for the topic of pattern mining (e.g., sequence, pattern, mining, frequent, itemset) we perform a full-text search in our database and extract the full-text. With this method, we can find around 14000 papers. To efficiently scan this large amount of academic work we developed a prototype (Figure 2). The prototype indexes the full-text and extracts user-defined keywords. Each file is listed as a row. Keywords are represented as columns. Each cell is color-coded based on the frequency of the keyword. The so generated table can be filtered by any full-text query or keyword occurrences. Additionally, the table can be sorted by the frequency of keywords. The user can add new keywords interactively. Irrelevant papers can be marked not to be inspected twice. Relevant work is thoroughly scanned for further relevant related work. Additionally,

[1] http://ieeevis.org

[2] https://www.eurovis2018.org/

[3] http://www.kdd.org/

[4] https://www.computer.org/web/tvcg

[5] http://journals.sagepub.com/home/ivia

the ACM digital library, IEEE Xplore digital library, EG digital library, and DBLP computer science bibliography was used for keyword search.

In total, we could identify nine papers with visualization techniques for frequent itemsets, 11 works for association rules, and 20 relevant publications for sequential patterns and the visually closely related episodes.

3 Visualizations and Visual Analytics Techniques

Mining for itemsets, association rules, and sequential patterns has commonalities. This fact also holds true for visualizing such patterns. This section will explain visualization techniques and further elaborate about how visual analytics is applied with these visualizations. We distinguish between visualizing the resulting patterns of a mining algorithm, the input data, and whether intermediate patterns are visualized during the mining process.

3.1 Itemsets

Frequent itemset mining, originally developed for market basket analysis, is one of the most popular research areas and serves as a basis for association rule mining. The task is to find common sets of items (itemsets) that occur together in records, also called transactions. The size of an itemset refers to the number of items in the itemset and is also called k-itemset. Borgelt provides a good introduction and overview of algorithms, data structures, and extensions [16].

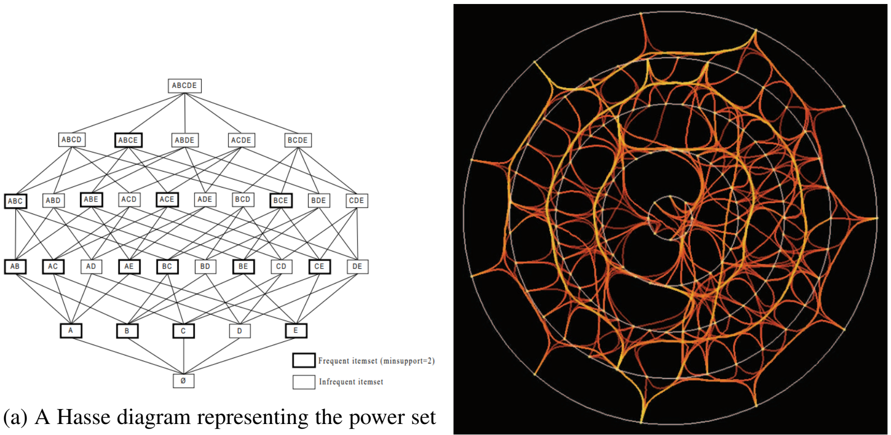
Visualizing sets is a heavily researched topic. For a comprehensive state-of-the-art survey we refer to Alsallahk et al. [8] and their companion website SetViz[6]. In the following, we focus on work that is specifically developed for frequent itemsets.

3.1.1 Lattice Representations

A representation of a lattice, or also concept hierarchy [40], of frequent itemsets, is often used to explain the concept of frequent itemset mining and the Apriori property. A typical representation is the Hasse diagram which shows the power set of an alphabet of items (Figure 3a). Note that in the Hasse diagram all possible itemsets are displayed. Frequent itemsets are highlighted. Additionally, itemsets can be annotated for example if they are closed [100] or maximal [19].

Klemettinen et al. already discuss the problem of clutter in directed graphs (see Section 3.2.2) due to too many edges [55]. The authors propose the Spiders technique [23] where multiple instances of one node are allowed. Bothorel et al. follow this idea and they display, similar to the *PowerSetViewer*, every frequent itemset

[6] `http://setviz.net`, accessed Feb. 2018



(a) A Hasse diagram representing the power set of A,B,C,D,E. The frequent itemsets are highlighted [76].

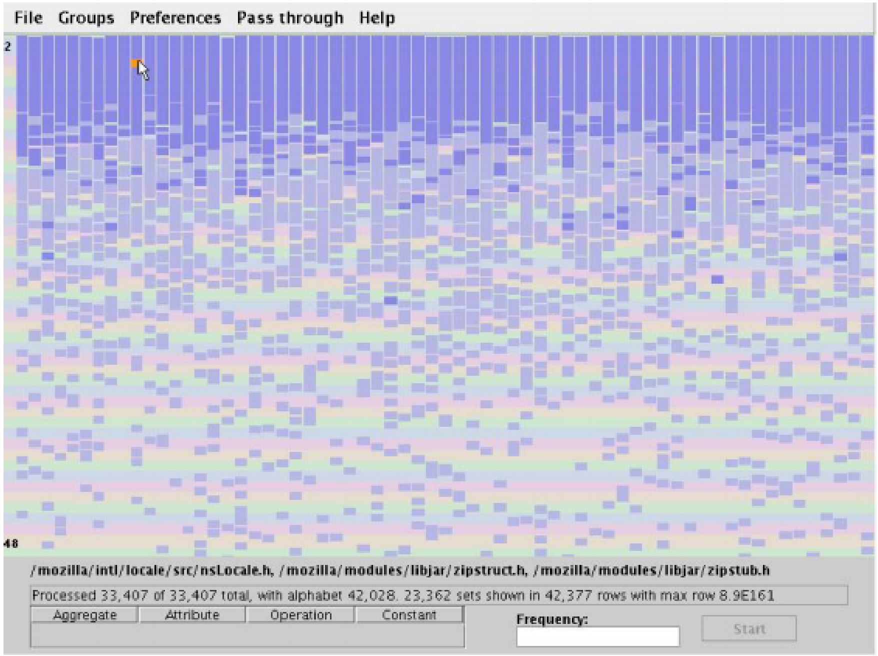
(b) A radial graph layout by Bothorel et al. [17]. Each frequent itemset is represented as a separate node. Infrequent itemsets are not shown in the graph.

Fig. 3: Two lattice representations showing frequent itemsets and their relations.

distinctly [17] (Figure 3b). The graph does not show the power set and rather only the frequent itemsets. They use a circular layout where itemsets of the same cardinality are placed on a concentric circle. The cardinality increases from the outside to the inside of the graph. A heuristic optimization strategy is used to place the nodes by reducing the length of the segment to reduce clutter. An additional measure is taken by an edge bundling strategy. The authors use three enhancements for their visualization. First, is a mapping of the support onto the alpha value for the colors (transparency) on the edges. Itemsets with high support are more opaque. The second measure is the accumulation of colors which is directly related to the edge bundling. The more edges are bundled, the brighter the color is assigned to the particular part of the edge. These two measures result in white, opaque edges for itemsets that have many supersets and high support. The last enhancement is a selection interaction. Multiple itemsets can be selected, and only their supersets and subsets are shown. Edges to 1-itemsets of supersets of the selected itemsets are displayed in a different color to reason about the origins of these supersets.

3.1.2 Pixel Based Visualizations

Munzner et al. use the accordion drawing technique that features guaranteed visibility and a rubber sheet navigation [75] (Figure 4). The itemsets are sorted vertically according to their size (number of items). Within one row they are lexicographically sorted. If there is not enough space, multiple itemsets are aggregated within one cell which is visualized as a darker, more saturated cell. The system provides a cell for



/mozilla/intl/locale/src/nsLocale.h, /mozilla/modules/libjar/zipstruct.h, /mozilla/modules/libjar/zipstub.h

Processed 33,407 of 33,407 total, with alphabet 42,028. 23,362 sets shown in 42,377 rows with max row 8.9E161

Fig. 4: PowerSetViewer (PSV) by Munzner et al. [75]. Each cell represents one or more itemsets (indicated by their saturation). The itemsets are ordered vertically by their size and lexicographically in horizontal direction.

every possible itemset that is generatable from the alphabet and, thus, supports the analysis for two different datasets that contain the same alphabet or analyzing the same dataset with different constraints. The support or frequency of an itemset is not mapped to a visual variable. However, the user can filter the result by this and other constraints. The visualization supports displaying up to 7 million itemsets and alphabet sizes of 40,000 [75].

3.1.3 Tree Visualizations

Most of the mined itemsets are redundant except for maximal frequent itemsets. This redundancy can be represented in a tree where each tree hierarchy represents a one or more k-itemsets. A total order of the items can be applied without the loss of generality and. Thus, a prefix tree can be generated which is also known as FP-tree [43].

The FP-Viz tool by Keim et al. [54] uses a sunburst [85] and interring [97] visualization technique to display the FP-tree [43]. Each circle segment represents a node of the FP-tree. The segments are ordered according to the tree hierarchy from the

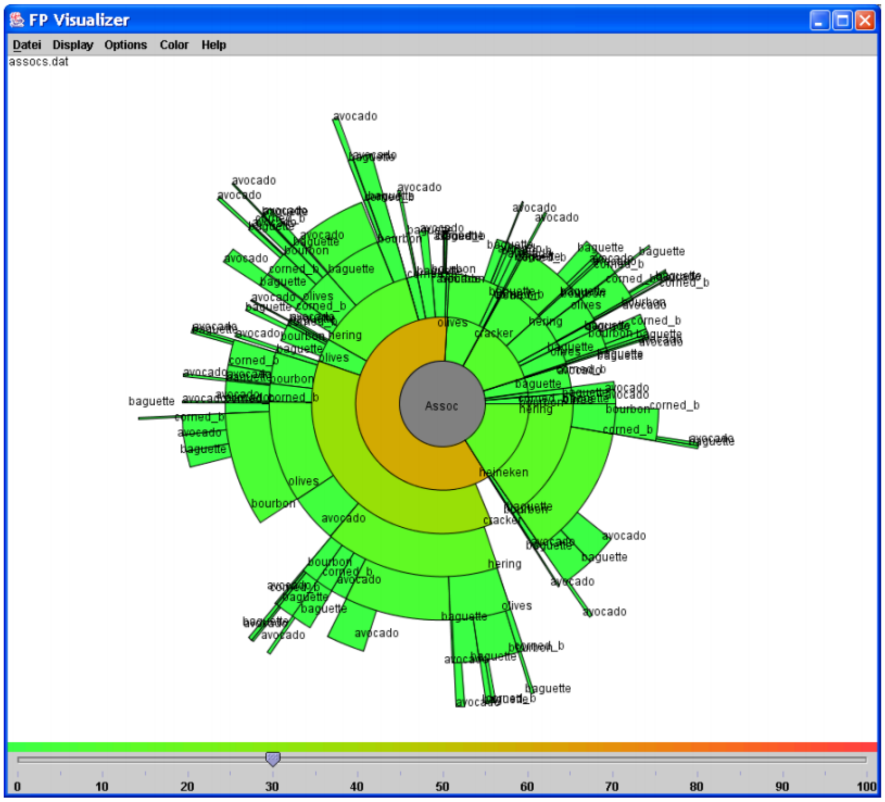


Fig. 5: FP-Viz [54] by Keim et al. showing the FP-tree in a hierarchical visualization known as sunburst. Each ring represents a tree hierarchy and each segment displays one item. Itemsets can be read by neighboring segments from in- to outside.

in- to the outside. Therefore, each pattern can be derived from the connected segments on each level. The support is mapped to color. The user can click any segment (item) to filter the tree such that only itemsets, where the clicked item is contained, are shown. The selected item is therefore represented as the root in the center.

Leung et al. provide a similar visualization [65]. In contrast to FP-Viz, the color is mapped onto the cardinality.

Leung et al. use FPMapViz [63] to visualize the hierarchy in a tree-map [82]. Itemsets of the same cardinality (size) are represented by rectangles of the same size. Therefore, itemsets can be read by going downwards the hierarchy as sub-rectangles. The support is displayed using colors.

In PyramidViz [64] (Figure 6) the prefix information is encoded from bottom to top in form of a pyramid. Each item that is part of a frequent itemset is represented

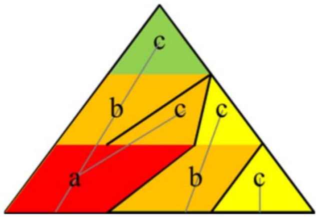


Fig. 6: The PyramidViz [64] by Leung et al. shows items as blocks of a pyramid. Connecting the blocks as shown by the thin gray lines forms the itemsets. The color references the support.

either as a trapezoidal block or triangle block depending on whether this item can be extended by another item or not. This impression is enforced by more vertical black lines to separate the blocks. Additionally, the items are connected by grey lines. The color hue of the blocks is used to display the frequency information. The visualization technique allows insights into the decreasing support of supersets of itemsets which is known as the Apriori property [4].

3.1.4 Linear Visualizations

Yang visualizes frequent itemsets as parallel coordinates [98] (Figure 7). Here, all items are placed on one axis (vertical coordinate). Their position is determined by their group (if a taxonomy is given) and further by the frequency of the item in descending order which is equal to the support of the 1-itemset which contains the respective item. There are as many axes as the longest frequent itemset that could be mined which is in the extreme case the size of the alphabet. All axes contain the same vertical order of the items. An itemset is visualized as a polyline. The strategy for drawing the polyline is similar to the vertical ordering of the items. Groups are arranged together, items within a group are sorted according to their frequency in descending order. This ensures positive slopes and reduces the clutter. The support can be either mapped to color or width of the line. The visualization effectively visualizes maximal frequent itemsets as all subsets are implicitly drawn as sub-segments of one polyline. This, however, means that the support of the subsets, which can differ greatly, is hidden from the user. For itemsets that share a common part, there is a chance that the polyline would be overplotted. Yang proposes the use of Bezier curves to solve this problem [99].

Leung et al. use a different mapping of the axis in their tool FIsViz [61] (Figure 8). Here, the support is shown on the y-axis and each item from the alphabet is mapped as a discrete dimension onto the x-axis. The items are placed in descending order in respect of their support. The items are connected with polylines indicat-

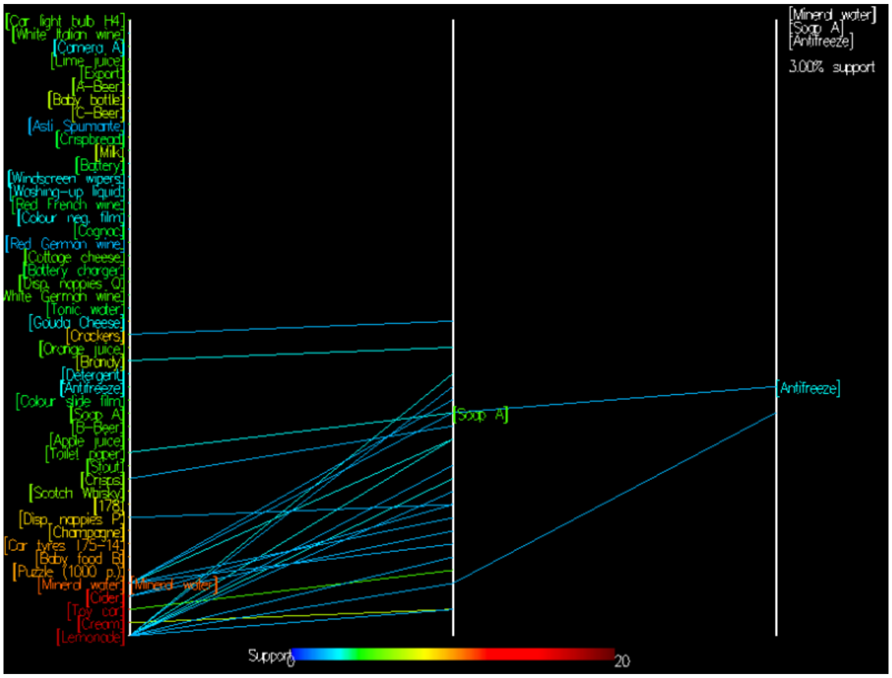
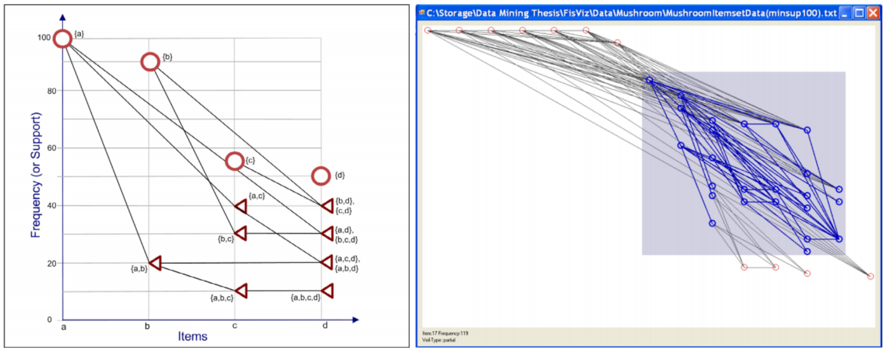


Fig. 7: Frequent itemsets displayed as parallel coordinates by Yang [98]. The support is mapped to color.

ing the itemsets and the according subsets or supersets respectively. The mapping onto the axes allows querying by support (vertically) and by cardinality in the horizontal direction. 1-itemsets are represented as a circle whereas all itemsets with a higher cardinality have a triangle icon. As in the parallel coordinate plots of Yang the polylines clutter as the size of the alphabet and the number of itemsets increases.

As a countermeasure Leung et al. propose WiFIsViz [62] where multiple itemsets are merged into horizontal lines called wiring-type diagrams. Itemsets are merged when they contain the same prefix, based on any total order of the items, and the same support. As shown in Figure 9, the tool consists of two views: an overview visualization on the left and a detail view on the right. The overview shows the merged patterns. As in FIsViz, the y-axis is used to display the support. The detail view uses a modified hierarchical view to display the itemsets where the vertical axis is used to span the tree and does not reflect the support. Both views are linked as shown in Figure 9. The wiring-type technique is also used in FpVAT [60] where it is combined with a raw data visualization module.



(a) Basic representation of FIsViz **(b)** Visualization of itemsets from mushroom data

Fig. 8: FIsViz [61] by Leung et al. represents itemsets in a scatterplot-like visualization where the y-axis conveys the support and the x-axis the cardinality of the itemsets. Polylines connect the items and, thus, build the itemsets.

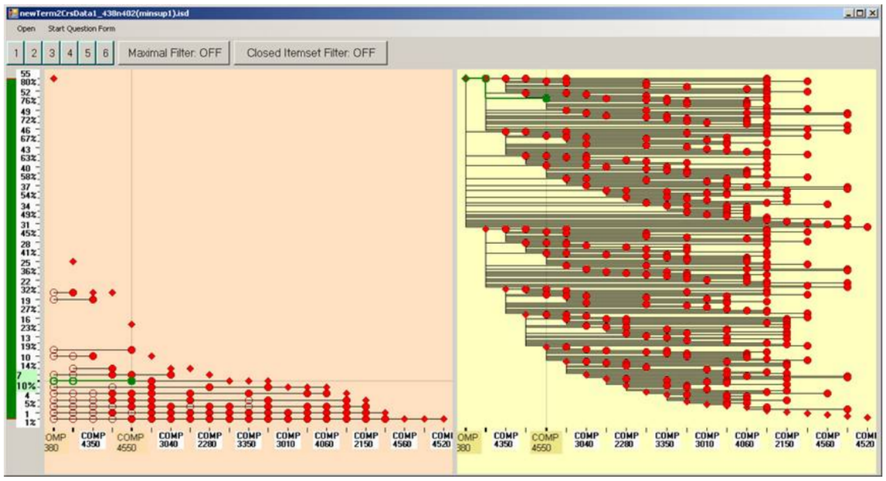
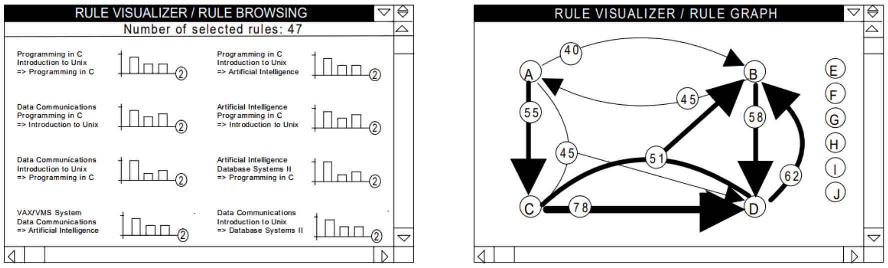


Fig. 9: WiFIsViz collapses multiple itemsets into horizontal lines to reduce clutter [62].

3.2 Association Rules

Association rules [2] are an extension of frequent itemset mining and an important concept in KDD. An association rule $X \rightarrow Y$, where X and Y are disjoint itemsets, indicates that items X (also called left hand side, body, or antecedent itemset) occurring in several records of a transaction database verify Y (also called right hand side, head, or consequent item). In general, two measures are applied for an association rule: (i) the support verifies that a rule does occur in at least x records and (ii)



RULE VISUALIZER / RULE BROWSING

Number of selected rules: 47

Programming in C
Introduction to Unix
=> Programming in C

Programming in C
Introduction to Unix
=> Artificial Intelligence

Data Communications
Programming in C
=> Introduction to Unix

Artificial Intelligence
Programming in C
=> Introduction to Unix

Data Communications
Introduction to Unix
=> Programming in C

Artificial Intelligence
Database Systems II
=> Programming in C

VAX/VMS System
Data Communications
=> Artificial Intelligence

Data Communications
Introduction to Unix
=> Database Systems II

RULE VISUALIZER / RULE GRAPH

Fig. 10: Association Rule Visualizer by Klemettinen et al. [55]. The left shows a browsing view where every barchart represents one rule. The right shows a directed graph where multiple association rules are displayed.

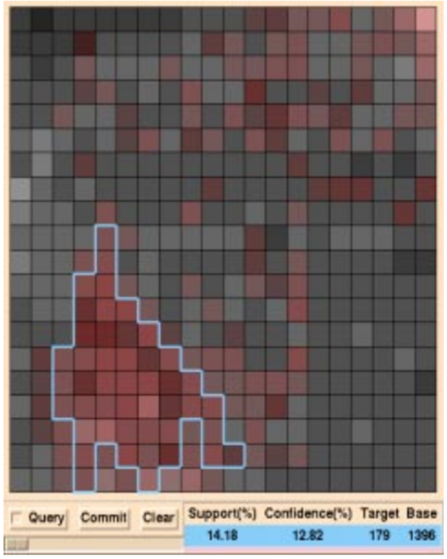
confidence measures the reliability of a rule (i.e., the probability that when X also occurs Y occurs). Association rule mining typically does not consider the order of items within a record or across records.

3.2.1 Individual Representations

Klemettinen et al. developed *Rule Visualizer* a tool to visualize and explore association rules [55]. Figure 10 shows two views of the tool: a browsing view on the left and a rule graph on the right (see Section 3.2.2). In the browsing view, every association rule is represented as one bar chart. The left bar represents the confidence, the right bar the support. The central bar displays the commonness. The number of attributes of the left-hand side of the rule is shown in the circled number. A textual representation of the association rule is shown left of each bar chart. The items are in separate lines. The left-hand side and right-hand side are distinguished by an arrow in front of the line.

3.2.2 Directed Graph

The rule graph (Figure 10, right) is a *directed graph* and visualizes several association rules simultaneously [55]. Here, each node represents an item. The arrows display the rule association where the thickness of the edge can be mapped to either the support or the confidence. Multiple items on the left-hand side are connected via an arc. For example, the rule $CD \rightarrow B$ can be found in the graph. The authors discuss in the paper that both properties of an association rule could be mapped to an edge by using color. Also using opacity is possible. It is clear that such a graph is difficult to draw and does not scale well to many association rules. Therefore, the authors offer four interaction possibilities: (i) an exclusion of association rules by removing items (as shown by the nodes E - J in Figure 10); (ii) an inclusion of items using templates; (iii) by letting the user set a maximal rule size for the itemset of



Support(%)	Confidence(%)	Target	Base
14.18	12.82	179	1396

Fig. 11: A matrix-based visualization for two-dimensional numerical association rules by Fukuda et al. [30].

the left-hand side; and finally (iv) letting the user join nodes (i.e., items) together. In general, even though such a graph may aggregate several association rules nicely, it is difficult to compare the properties (support or confidence) of different association rules. Therefore, the edges are labeled to compare the exact numbers.

Han et al. [42] (DBMiner) and Rainsford et al. [80] also use a directed graph to visualize association rules. Both use a radial layout for the items. Edges are thus connected within the so created circle. Rainsford et al. use gradient lines (yellow to blue or vice versa) to indicate the antecedent and consequent items. Bidirectional items are shown with a green line to easily spot them in the graph.

3.2.3 Matrix Views

Fukuda et al. are visualizing two-dimensional numerical association rules (e.g., $XY \to Z$) using a matrix-based visualization [30]. In this case, the left-hand side of the association rule is not based on binary features but numerical attributes (e.g., age). With an equal sized binning, the numerical features are discretized and can be mapped on a 2d grid (see Figure 11). The authors map the brightness onto the support of a mined rule and the confidence onto the color. Bright and red pixels stand for a rule with high support and confidence. A careful bin selection and an overall good correlation across the bins with the consequent item possibly results in a non-scattered view. The method scales well to small bin sizes or large ranges of the dimensions respectively.

AViz by Han et al. [41] extends the two-dimensional model to a three-dimensional space.

Commercial tools like IBM QUEST [3][7] also provide matrices for association rules. In QUEST, item-to-item relationships (i.e., $X \rightarrow Y$) can be inspected in 2D views where either the support or the confidence is mapped to color. In a 3D view, the support is mapped to color, and the confidence is mapped to the height of a bar in the respective cell or vice versa. For 3-item association rules of form $XY \rightarrow Z$ QUEST offers a different 3D view where the plane on the bottom is similar to the 2D item-to-item visualization. The z-axis provides access to the second item of the antecedent itemset. The resulting cubes in the 3D space indicate the mined rules. The colors of the cubes can be mapped to the support or confidence. The 3D view may show clusters. The user can click on a box to see all other rules containing one of the items of the current rule. Additionally, spatial navigation is supported.

MineSet[8] [18] uses a similar 3D view for item-to-item association rules. Here, each cell contains a bar colored in a continuous color range. The height of the bar represents the confidence. The color shows the support. Additionally, each bar is sliced by one disk at different heights of the bar indicating the probability of the right-hand side of the rule.

3.2.4 Table Views

While the matrix views can be extended to show multiple item-to-item relationships by adding additional rows and columns for each itemset, Wong et al. point out that extending the matrix view in this way does not scale well with a large alphabet of items and that if one row contains many items it is difficult to compare it to another row with many items [93]. Another general problem of 3D views is the occlusion that may occur. The user must also adjust the perspective to compare, for example, the height of the bars – if this is not impossible due to occlusion. Wong et al. propose a rule-to-item 3D table-based view where each rule is displayed as a separate column of the matrix whereas the rows represent one single item (Figure 12). In the back two bar charts provide the support and confidence properties for each rule. Figure 12 shows, for example, the rule *james* & *michigan* → *nichols* on the rightmost column (next to the labels of the items) with a confidence of 100% and a support of 9%. It is clear that this visualization is capable of supporting many-to-many item association rules and scales well to larger alphabets as well as many rules. Navigating such large spaces might become difficult but can be supported through interaction by

[7] `http://www.almaden.ibm.com/cs/quest/demo/assoc/general.html`, accessed Feb. 2018

[8] `ftp://ftp.sgi.com/sgi/mineset/overview/mineset_overview.htm`, accessed Feb. 2018

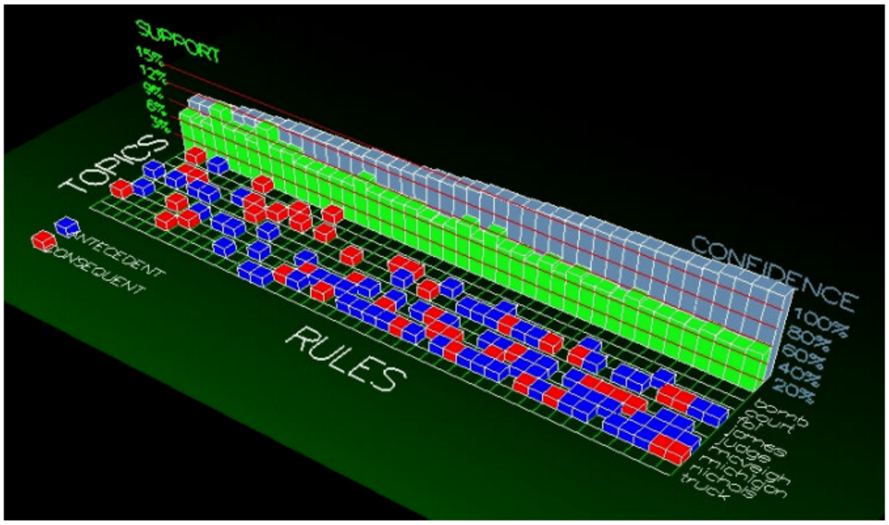


Fig. 12: A 3D table visualization by Wong et al. [93] representing each association rule as a separate column. The alphabet of items is shown as rows. Additionally, the confidence and the support for each association rule are provided.

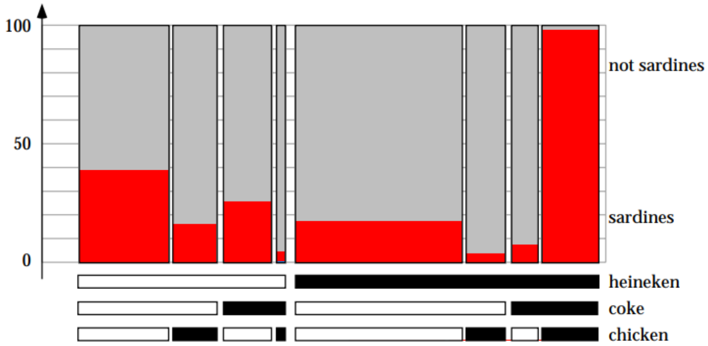


Fig. 13: Mosaic Plot of multiple association rules by Hofmann et al. [48].

highlighting the respective columns and rows. Lee et al. use the same technique to visualize association rules based on geo-tagged photos [59].

3.2.5 Mosaic Plots

Hofmann et al. build interactive Mosaic plots [44, 47], called Double Decker plots, based on contingency tables created from association rules enabling the user understanding the underlying structure [48]. Figure 13 shows such a plot for the rule

$$R1: \ heineken \ \& \ coke \ \& \ chicken \rightarrow sardines$$

as well as all subsets of this association rule, for example:

$$R2: \ heineken \ \& \ coke \ \& \ not \ chicken \rightarrow sardines$$

The two-colored horizontal bars on the bottom indicate whether the items are part of the rule (black) or not (white). Both measures for an association rule, support, and confidence, are shown above. The support is mapped to the width of one bar (grey and red part combined). The red highlighting in each bar shows the confidence. The confidence measure can be directly read from the visualization (e.g., R1: $\sim$ 98%, R2: $\sim$ 8%). The support is more difficult as there is no axis. However, it is possible to compare the support such as R1 has higher support than R2. This clever way of visualizing both measures simultaneously allows the user to compare multiple association rules, especially subsets, where the consequent part consists of one item or does not change. Comparing association rules with only a small intersecting set of the antecedent side is possible, however, increases the complexity of the visualization. In general, the visualization is not meant to aggregate a large set of rules nor scale well to large alphabets of items.

3.2.6 Linear Visualizations

Yangs idea to visualize itemsets in a parallel coordinate plot works as well for association rules [98]. To distinguish the antecedent and consequent side from items belonging to either side, Yang plots an arrow on the line where the sides are connected. For example, with a rule, $AB \rightarrow CD$ the arrow will be on the line between the second and third axis. The property of positive slopes on either side is still held true (see Section 3.1.4). A positive slope for the line connecting both sides cannot be guaranteed. Yang mentions that it is, however, more likely that this slope is negative. Besides the arrow, this feature also introduces a visible distinguishment. Two further visual variables are available: line width and color. For association rules both, confidence and support can be mapped at the same time. Note, that the line-width might increase clutter.

3.2.7 Mining with Subjective Interesting Measures

The previous visualization techniques focus on visualizing association rules including the visualization of the objective measure's support and confidence. Liu et al. explicitly focus on subjective measures and here, especially on the unexpectedness of a rule [66]. These are rules that are a contradiction to the user's knowledge or completely unknown. The user can, therefore, specify her knowledge and insert this into the *interestingness analysis system*. The user interface (Figure 14) separates four different types of identified rules: conforming rules, unexpected condition rules, un-

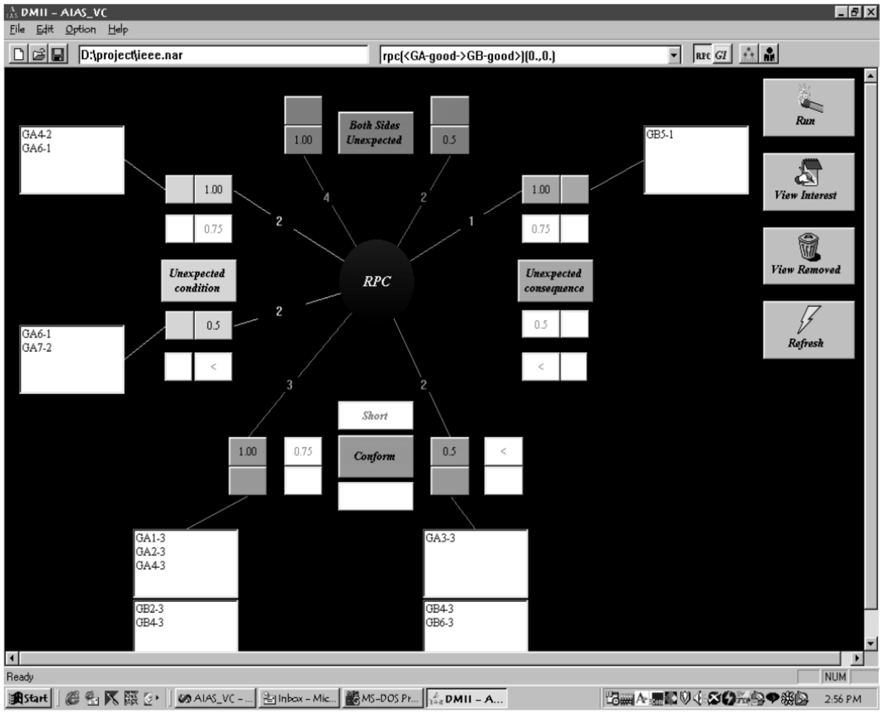
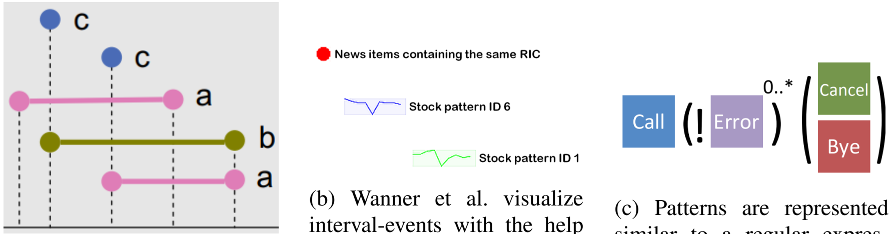


Fig. 14: The interestingness analysis system separates four different types of association rules based on their potential interestingness [66].

expected consequent rules, and both-side unexpected rules. The authors state that it is more important to only show the interesting part of the rule instead of the complete rule.

3.3 Sequential Patterns

Mining sequential patterns [5] describes an extension to frequent itemset mining where subsequences of a given sequence database are discovered [1]. In contrast to frequent itemsets and association rules, the order is defined in the data which puts limitations onto the visualizations since they should represent this order of the events in a pattern. An event is equal to a set of items. These itemsets are extended to hold an additional property which defines their order and occurrence in time. In application areas such as DNA sequences, web logs, and click streams [69] there are most of the times only 1-itemsets involved which simplifies visualizations. Sequential patterns are also useful for mining trajectories [32] and find common patterns in movement [9]. In general, a pattern can be viewed as a prototype of multiple event



(a) A representation for interval and point event patterns by Shin et al. [96].

(b) Wanner et al. visualize interval-events with the help of sparklines generated by a SOM [89].

(c) Patterns are represented similar to a regular expression [20].

Fig. 15: Point and interval patterns can be represented as vertically separated point and line constructs (a). More information can be encoded using glyphs (b). Annotating events with constructs known from regular expressions enables to create queries and rules (c).

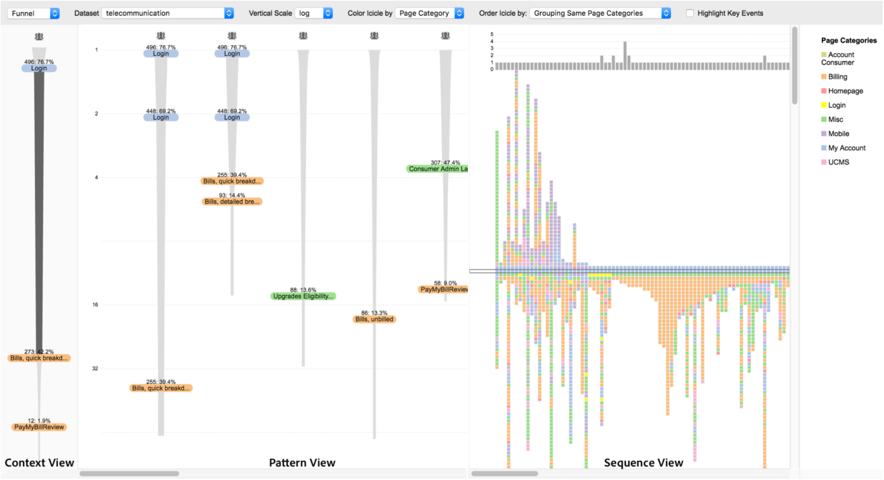
sequences and, therefore, be visualized in such a manner. In a geo-spatial context, this allows, for example, to identify common travel routes in traffic.

This section focuses on visualization techniques specifically for sequential patterns. For a broader view on time-oriented data we refer to the book of Aigner et al. [7] and other surveys [84, 6].

3.3.1 Individual Representations

Patterns can be represented in form of text, for example $\langle\{a\},\{b,c\}\rangle$ [26] where an item a occurs before items b and c which occur at the same time. In case of interval events that do not only occur at a specific point in time but rather over a given time span with a defined start and ending point Shin et al. [96] use a representation where the start of an event is marked with a $+$ and the end with a $-$. The following number is used to identify the starts and ends if the same event occurs multiple times. For example, $a^{+1} < b^{+1} < b^{-1} < c < a^{-1} = d$ where an interval event a starts before another interval event b, b ends before a point event c which is followed by the end of a. The point-event d occurs at the same time as a ends. Textual representations are less fitting for interpreting and understanding a pattern can be complex which is especially true for patterns with many items and when items also occur at the same time.

Shin et al. also present another representation which is more intuitively comprehensible (Figure 15a). The order is maintained from left to right whereas events are separated vertically. Colors plus the additional labels allow the identification of the events. This representation can be modified to use different symbols for the events or even glyphs which allows visualizing additional information for an event [89] (Figure 15b). In case of a larger alphabet, the visual representation does not scale well regarding understandability [50]. Furthermore, as a single pattern representa-



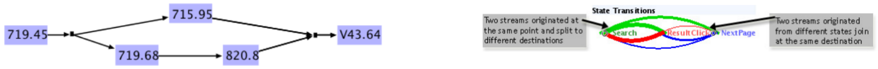
Funnel; Dataset: telecommunication; Vertical Scale: log; Color Icicle by: Page Category; Order Icicle by: Grouping Same Page Categories; □ Highlight Key Events

Fig. 16: Maximal sequential patterns from clickstream data are visualized on the left. The right side shows an aligned sequence view based on a selected event [68].

tion, it is not desirable to visualize many patterns at the same time. In none of these representations, additional properties such as the support are mapped. This can be added for example in a separate representation such as a bar, the size of the pattern can be modified, or the background color (hue, saturation). The latter two have to be applied carefully as it might distort colors and hinder the user to compare different patterns.

Cappers et al. use a representation which is very similar to regular expressions [20] (Figure 15c). This is especially powerful to create rules and queries to simplify and filter a large set of rules. It also allows to simplify the patterns themselves by reducing the number of events that are displayed. However, a user must know and understand the concept of regular expressions. It is, however, difficult to automatically generate such a representation from a given pattern.

Liu et al. mine for maximal sequential patterns in clickstream data [68]. As shown in Figure 16, patterns are represented as linear, vertical constructs of different lengths where the temporal flow is indicated from top to bottom. The length encodes the average sequence length. The vertical position of the event reflects the average number of clicks needed to reach the specific event. The grey line connecting the events resembles a funnel visualization and indicates the decreasing support or in other words, the percentage of people reaching the event through the click sequence. The exact number is also shown above of each event. The view on the right visualizes each sequence individually. Clicking on an event of a pattern will align all sequences by that event. Metrics of a sequence are represented by the bar chart on top. Without any alignment and through sorting the events by category an icicle plot [57] resembles. It is noteworthy to mention that icicle plots are quite popular to


719.45
715.95
719.68
820.8
V43.64
State Transitions
Two streams originated at the same point and split to different destinations
Search
ResultClick
NextPage
Two streams originated from different states join at the same destination

(a) The pattern diagram by Patnaik et al. [77] shows patterns from electronic medical records.

(b) The *state transition panel* in the tool *Session Viewer* places all events in one line but allows splits in the streams [58].

Fig. 17: Two node-link visualizations that use a state transition analogy to display and aggregate multiple sequential patterns.

visualize event sequences [58, 94, 95, 11] and that aligning sequences leads to an better overview for comparison [73, 20, 58, 68]

3.3.2 Flow Diagram Visualizations

Flow diagrams, also known as flowcharts, are frequently used to display complex systems. They are especially useful to represent different states or components of a system. Transition or connections in between the states are connected with lines or arrows. Sequential pattern mining works on discrete event data. Therefore, the analogy to represent events as states connecting successive events to indicate the transition is given.

Mannila et al. extract global partial orders from event sequence data [71]. While this is not the same as sequential patterns, there are many similarities. Visualizing and aggregating event sequences in such a way provides an intuitive representation that moderately scales to a growing number of events. No frequency information (support) is shown in this visualization.

Similar to the tree visualization of Mannila et al., Patnaik et al. provide a *pattern display* in their tool which is designed for electronic medical records [77]. The pattern display is laid out horizontally and can split up into different events which might converge later on (Figure 17a).

Lam et al. also use the idea of a state transition representation [58] in their tool *Session Viewer* (Figure 17b). For logs, such state transition representations are well known especially when Markov models are being used [38]. The width of the arcs represents the frequency of the respective transition.

Hu et al. mine sequential patterns based on words from tweets and visualize the words in a node-link diagram [49]. In their tool *SentenTree*, the nodes (words) have different sizes and colors representing their frequencies (double-encoded).

As for individual patterns, nodes of a flow diagram can also integrate glyphs which enables the user to intuitively understand complex features and their temporal order [12].

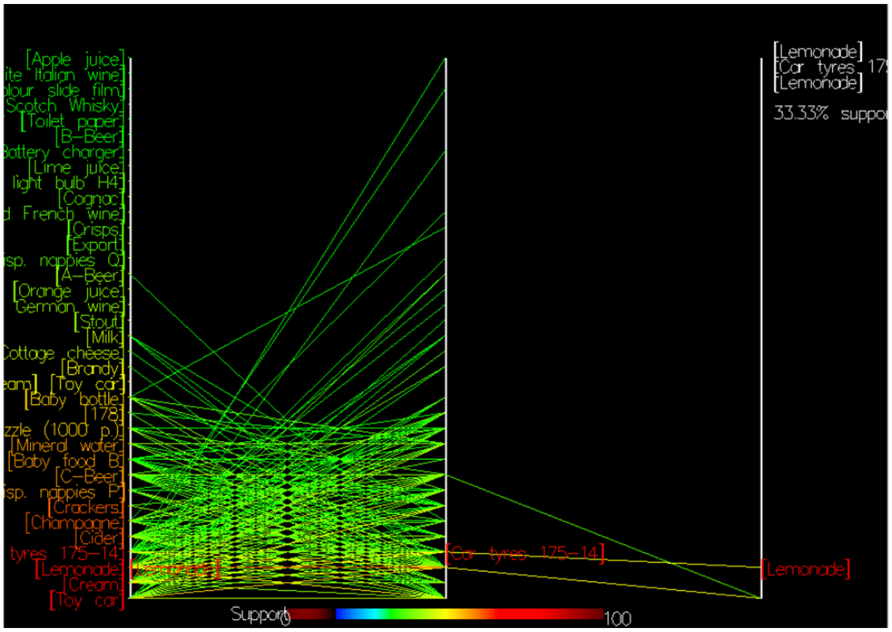


Fig. 18: A parallel coordinates plot to visualize sequential patterns [98].

Parallel coordinates are also suitable for sequential patterns as their axes promote a natural ordering. In contrast to visualizing itemsets (Section 3.1.4) and association rules (Section 3.2.6) each element on the axis represents an itemset instead of an item. This is necessary because in general, sequential patterns may have items occurring at the same time. As the order is given, no assumption can be made that the polylines have positive slopes. The property that subsequences are absorbed remains true. The support of the pattern is not mapped onto the complete polyline. Instead, each 2-itemset segment is colored according to its respective support resulting in a polyline with multiple colors. Yang claims that this approach provides more information [98]. The fact that itemsets are mapped to distinct elements on the axes as well as the nonuniform slopes increases the clutter.

Sankey diagrams extend flow diagrams by using a visual mapping on the width of the line that shows the flow. Typically the quantity of the flow is being mapped. In contrast to state transition representations, sankey diagrams use larger bars (rectangles) as a vertex. Perer and Wang use sankey diagrams to represent sequential patterns [78] (Figure 19). This type of visualization can aggregate multiple patterns and implicitly showing subpatterns. The support of each pattern is mapped onto the width of the line. In case of convergence, the slightly transparent line overlaps and allows the distinction of the originating path. An additional attribute can be mapped onto the color which further helps to distinguish the different flows. Chou et al. use a variant of a sankey diagram to visualize privacy preserving event sequence

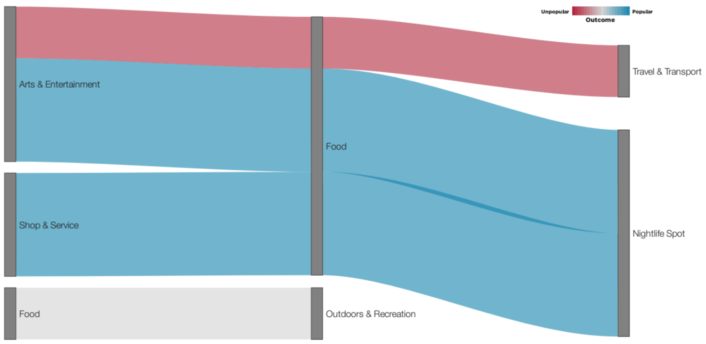


Fig. 19: Sequential patterns visualized using a sankey diagram [78].

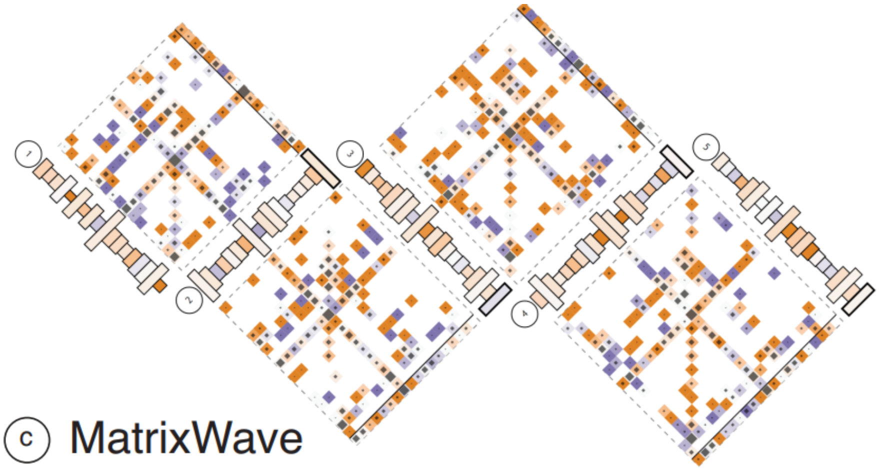


Fig. 20: *MatrixWave* [101], a zig-zag layout of multiple matrices which can be derived from the sankey-diagram visualization.

data [22]. Perer et al. combine their sankey diagram representation with a bubble chart which shows the most frequent 1-event patterns [79].

Zhao et al. point out that sankey diagrams, as other node-link diagrams, tend to produce clutter due to many overlapping edges [101]. They propose to use multiple transition matrices in a zig-zag layout to represent event sequence data. Each transition matrix replaces two vertical stages (set of nodes) and their links in between them. The height of the node-bars from the sankey diagram are replaced by bars on the sides of the matrix. The links connecting the nodes are visualized in the matrix where the metric that is mapped to the line width is now mapped to the size of a square in the grid of the matrix. Additional design elements such as the color can be

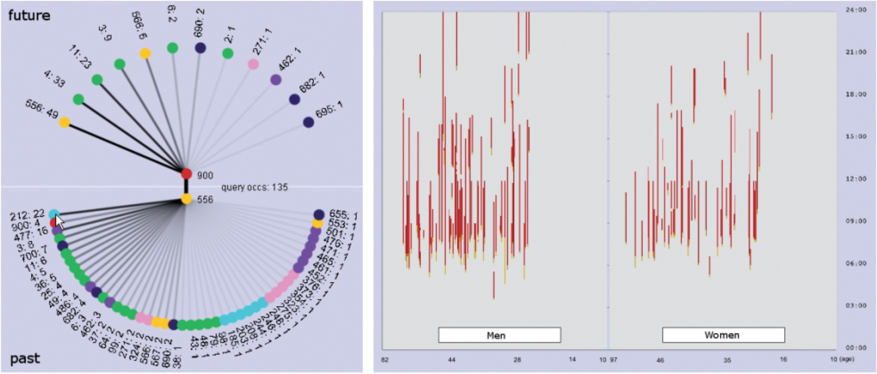


Fig. 21: The *ActiviTree* interface [88]. The left shows a graph where the central part is the current pattern in focus. Preceding and succeeding events are placed below and above the central part whereas the support is mapped onto the opacity of the lines. The right part visualizes the sequences where the yellow parts represent the selected pattern.

used to map additional attributes, for example, to compare two datasets.

Vrotsou et al. use a graph layout to let a user interactively mine for patterns also allowing the generation of infrequent patterns [88] (Figure 21, left). The currently focused part is shown in the center of the graph with events preceding the pattern below and succeeding events above. The events are ordered according to their significance in descending order from left to right. The frequency information (support) is mapped onto the opacity of the edge. The events (nodes) are colored according to a classification of the underlying data. The user can interactively extend the pattern by clicking on preceding and succeeding events. Similarly, nodes can be removed from the current pattern to enable the user to explore a different pattern. The right view (Figure 21) displays the sequences whereas the highlighted part (yellow) refers to the selected pattern from the left side. The time is also represented on the y-axis from bottom to top. The x-axis is separated by sex, and the sequences are ordered by age within their group.

The earlier mentioned icicle plot visualization could also be used for *branching patterns* [67]. Liu et al. can alternatively display a node-link diagram or a combination of both as shown in Figure 22. The width of the rectangles and the links reflect the number of sequences. The node-level is double-mapped onto the vertical position as well as the color. Only links leading to exit nodes are colored in a specific grey color to make them distinctive.

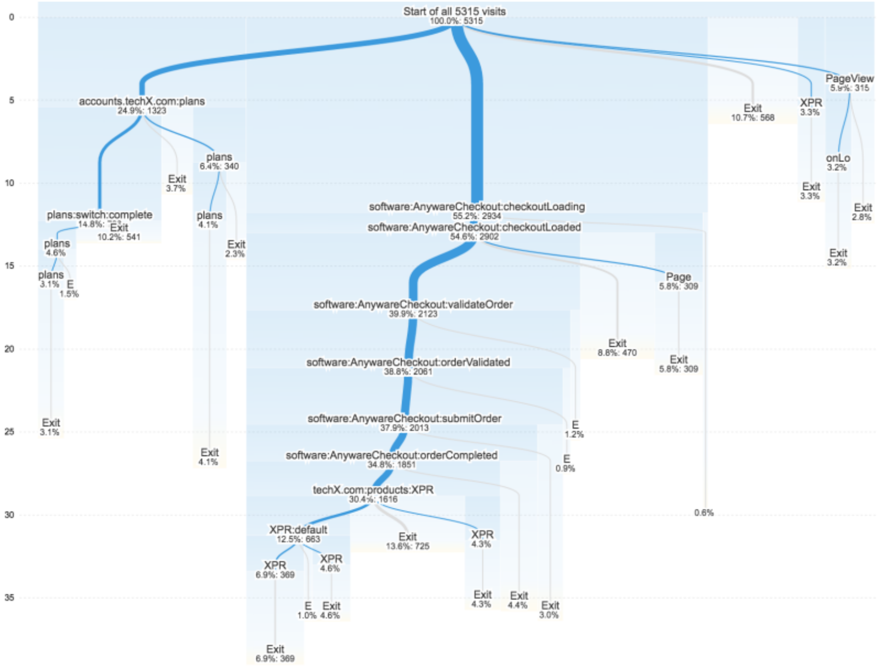


Fig. 22: A combination of an icicle plot visualization and node-link diagram for branching patterns [67].

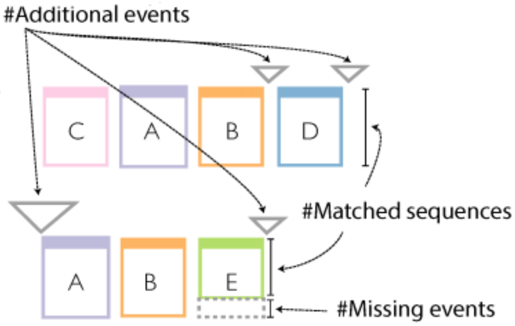
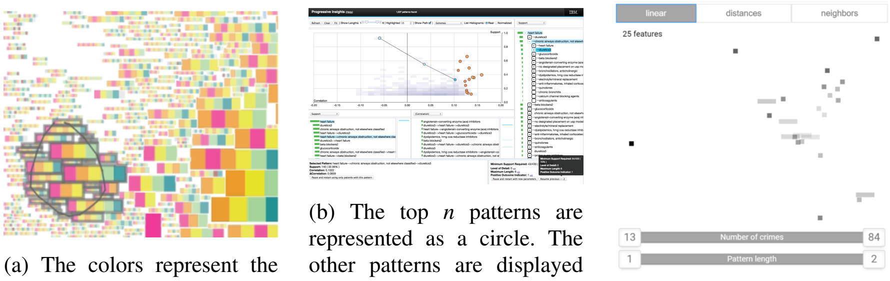


Fig. 23: Two aggregated patterns represent two disjoint sets of similar patterns [21].

3.3.3 Aggregated Patterns

The previous section introduced various pattern visualizations suitable to represent several patterns synonymously. Another method is to aggregate similar patterns and visualize only the prototype.



(a) The colors represent the events of a pattern. The patterns are clustered using a SOM and an additional placement strategy is applied to guarantee an overlap-free layout [91].

(b) The top *n* patterns are represented as a circle. The other patterns are displayed as a heatmap in the scatterplot. Different metrics can be mapped to the axes. The line connecting the patterns indicates sub- and superpatterns [86].

(c) Patterns are represented through rectangles which are placed on a 2D plane using projection methods. Similar patterns are closer together [51].

Fig. 24: Patterns can be clustered and placed on a 2D plane reflecting their similarity or different metrics can be used to layout patterns.

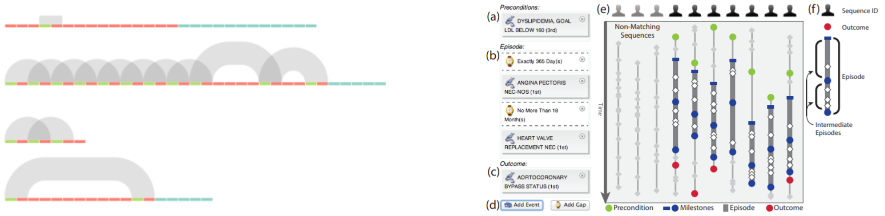
Chen et al. summarize multiple patterns according to the minimum description length [37] and visualize the so gained prototype including the lost information [21]. This is achieved using corrections that insert or delete events. The visualization, as shown in Figure 23, encodes the number of matched events which is mapped to the height of the rectangle and the number of additional events that can occur before, after, or in-between the pattern which is displayed by a triangle glyph of varying size.

3.3.4 Pattern Placement Strategies

This section focuses on the placement of multiple sequential patterns in a 2D layout. This enables the users to quickly identify similar patterns or finding interesting patterns by placing them according to metrics.

Wei et al. use differently colored, horizontally placed rectangles to represent web click stream patterns [91]. They use a SOM with Markov chains to define the 2D positions for each pattern such that clusters become visible. As this does not guarantee an overlap-free layout, Wei et al. use an additional placement strategy which is also used to create word clouds [87]. The significance of a pattern is mapped onto the size.

Stolper et al. use a scatterplot to plot patterns [86]. The user can select three metrics. The first defines the size of the circles whereas the top *n* patterns are drawn. The rest is aggregated in a heatmap. The other two metrics are mapped to the axes of the scatterplot. In the use case described in the paper, the support is mapped to the y-axis, and a correlation measure is used to align the patterns horizontally. Through a line connector, sub- and super patterns of an inspected pattern can be found.



(a) The EventExplorer interface of Bodesinsky et al. uses a technique similar to arc diagrams to visualize serial episodes in event sequences and thus highlighting recurring patterns and their distributions [15].

(b) *DecisionFlow* shows episodes using glyphs and highlights the respective space in between the beginning and ending of the episode [35].

Fig. 25: Two techniques to visualize serial episodes in event sequences.

Gotz et al. use a scatterplot where the support is mapped to the x- and the y-axis [36] of different patients' outcome. The size of the circles (patterns) is mapped to the correlation to the outcome.

In previous work we use a scatterplot to depict sequential patterns on a 2D space [51]. In contrast to Stolper et al. the axes were not defined by some metric, but we use various projections to reduce the high-dimensional space and map it on a 2D plane. The view is, therefore, showing similarities of patterns. Each pattern is represented by a rectangle whereas the width of the rectangle defines the number of items that are contained in the sequential pattern.

3.3.5 Episode Visualization

In episode mining [72], patterns are mined in a single sequence. A pattern, therefore, is a partial order of events that can be found multiple times in a given sequence of events. Typical application examples are logs of alarms, user interactions, and medical events of a patient. Episodes can, therefore, show the semantically meaningful connection of events as well as predict the behavior of a sequence in future. Although mined differently, episodes are visually closely related to sequential patterns. This section will report on visualization techniques that display the position of serial and parallel episodes.

Arc diagrams [90] are a popular method for visualizing recurring episodes in a sequence. They are especially powerful to highlight regular recurring patterns which can be useful in various domains such as text or music. Bodesinsky et al. make use of a variation of arc diagrams to visualize recurrent patterns of serial episodes in event sequences [15] (Figure 25a).

In *DecisionFlow*, Gotz et al. use a glyph representation to visualize episodes in event sequences [35] (Figure 25b). The user defines a query of a precondition (green), milestones (blue) that resemble the episode, and an outcome (red). The glyph for the first milestone of the episode uses a rectangular shape which differs from circle shapes of the following milestones. The serial episode is additionally highlighted.

In a previous work [50], we use horizontal lines above or below of an event sequence to indicate the occurrences of serial and parallel episodes. The horizontal lines reduce clutter as there is no overplotting as in the arc diagrams but the length of a serial episode covers is not directly visible. Multiple occurrences of the same episode are indicated by small vertical shifts of one pixel within the horizontal lines.

4 Comparison

This section compares the different visualization techniques within their specific domain. We hereby use visual variables that Bertin identified [14] and their ranking [70, 74]. We use this ranking to depict for example the task of comparing the support for different patterns. Additionally, we provide an analysis how scalable the techniques are regarding alphabet size ($\Sigma-$Scalability) and the number of patterns (Pattern-Scalability). Furthermore, we compare the task of identifying a pattern. This means that a user is capable to completely identify all items of a pattern in their respective context without using any additional interaction technique such as tooltips. Lastly, we compare whether the hierarchy of the patterns can be examined. We use a scale from $--$, $-$, $+$, $++$ and $/$ if this is not supported. Note that these ratings were obtained by surveying visualization and visual analytics experts. [9]

4.1 Frequent Itemsets

Table 1 shows that the pixel-based visualization features the best scalability. However, it does not provide insight into the support of an itemset nor can the itemsets be directly interpreted from the visualization. Lattice visualizations are well suited to represent the relationships of itemsets, i.e., pattern containment. It is possible to map the support to another visual variable, but the graph structure is prone to be cluttered due to many crossing lines. The tree visualizations provide a better comparison for support by using color or hue [54, 63, 17, 64] as the visual variable. Linear visualizations provide the best support comparison as they map this feature to the position.

		Year	Σ-Scalability	Itemset-Scalability	Support Comparison	Identification
lattice	[76]	1999	-	-	/	+
	[17]	2013	+	+	-	-
pix	[75]	2005	++	++	/	/
tree	[54]	2005	- -	+	+	-
	[63]	2011	-	+	-	-
	[65]	2012	- -	+	+	-
	[64]	2016	+	-	+	++
linear	[98]	2003	+	+	++	-
	[61]	2008	+	-	++	+
	[62]	2008	+	++	+	+

Table 1: A comparison of visualization techniques for frequent itemsets categorized as pixel-based visualizations, tree visualizations, and linear visualizations.

					Comparison			
		Year	Σ-Scalability	AR-Scalability	Support	Confidence	Both	Identification
ind	[55]	1994	- -	- -	++	++	++	++
graph	[55]	1994	- -	-	+	+	-	-
	[42]	1997	- -	-	+	+	/	+
	[80]	2000	-	+	/	/	/	+
matrix	[30]	1996	+	+	+	+	+	-
	[3]	1996	-	+	+	-	-	-
	[18]	1997	-	+	-	+	-	-
	[41]	2000	+	+	+	+	-	- -
tab	[93]	1999	+	++	+	+	+	+
mos	[48]	2000	-	- -	+	++	+	+
lin	[98]	2003	+	+	-	-	/	+

Table 2: Visualization techniques for association rules that are categorized as individual representations, graph visualizations, matrix- and table-based visualizations, mosaic plots, and linear visualizations.

4.2 Association Rules

Association rules typically contain two interestingness measures. Table 2 shows that visualizing both simultaneously is difficult while preserving their structure and be scalable at the same time. Graph-based systems can aggregate multiple patterns but quickly get cluttered [55]. A solution to this is to generate a node for each itemset instead for each item [42, 80]. Matrices feature pixel-like visualizations and thus, scale better. Implementing both interestingness measures is difficult here, as mapping them onto the opacity and the hue at the same time might infer perceptual

[9] A quantitative comparison would require the implementation of each technique, standardized datasets, as well as a methodology to measure the scalability, for example, by measuring the occlusion through pixel overplotting [25]. We consider this as future work.

		Year	Σ-Scalability	Pattern-Scalability	Support Comparison	Identification
ind	[96]	2009	- -	- -	/	++
	[89]	2016	- -	- -	/	++
	[20]	2018	+	+	/	+
flow	[71]	2000	-	-	/	+
	[98]	2003	-	+	+	-
	[58]	2007	-	-	-	-
	[88]	2009	+	-	+	-
	[77]	2011	-	-	/	-
	[78]	2014	+	-	+	+
	[101]	2015	+	+	+	-
	[67]	2017	-	-	+	+
	[49]	2017	-	-	+	+
agg	[21]	2018	+	++	+	+
place	[91]	2012	-	++	+	+
	[86]	2014	++	++	++	/
	[36]	2014	++	++	++	/
	[51]	2018	++	++	+	/
episode	[35]	2014	-	–	/	++
	[15]	2015	+	+	- -	+
	[50]	2017	+	++	/	/

Table 3: A comparison of visualization techniques for sequential patterns using individual representations, flow diagrams, aggregated pattern visualizations, placement strategies, and episode visualizations.

biases [30]. The table view maps association rules differently, and by using a 3D view, both, the support and the confidence can be visualized using a 3D bar chart (position) at the same time [93]. The double-decker plots [48] aggregate similar rules featuring a good comparison for both interestingness measures and enable the user to compare different rules. Hahsler et al. provide a similar comparison of visualization techniques for association rules [39].

4.3 Sequential Patterns

Compared to frequent itemsets and association rules we could identify the most visualization techniques for sequential patterns (Table 3). This may be because many real-world applications can be abstracted to event sequence data. Individual representations of sequential patterns do not scale well regarding alphabet size and the number of patterns which is expected. Flow diagrams provide slightly better scalability while most of them also include the interestingness measure *support* in the visual representations. The aggregated pattern visualization technique visualizes the number of missing events that are not covered by the pattern itself. The technique is interesting for maximal, closed, or generator patterns as it reveals how much information is lost by the compression. The visual representations used in the pattern placement strategies scale best overall which is due to the abstract visualizations of the pattern. Most of the techniques do not allow direct identification of the pattern but merely provide information about the support or the length of a pattern. The techniques used in the episode visualization category reveal the occurrences of patterns within an event sequence and also allow the identification of periodic occurrence-patterns.

5 Discussion and Opportunities for Research

The comparison reveals that there is a tendency of decreasing scalability visible when the identification of a pattern is required. This means that every instance (i.e., item, rule-item, event) must be visualized. Visualizations for frequent itemsets offer the greatest degree of freedom as the items within an itemset typically have no natural order and thus, can be placed in a manner to reduce clutter. For association rules, this degree of freedom is already limited to some extent, as a rule, especially the antecedent and consequent itemset must be distinguishable. The itemsets themselves can be again ordered freely. Sequential patterns are the most restrictive entities because the events (itemsets) have a predefined order which must be retained to not loose information. Note, that this observation is not true for abstract visual representations where the individual entities of a pattern are hidden.

Similarly, the additional display of an interestingness measure impacts the designs. There exists only a finite number of visual variables [13] on which information can be mapped to. Designs for frequent itemsets offer again the highest degree of freedom because it is possible to map the interestingness measure onto the powerful position variable [70, 74] while retaining the identifiability and limiting the clutter. Association rules that typically hold two interestingness measures (support and confidence) nicely show the challenges of representing both simultaneously without impacting the scalability or the identifiability.

Many visual designs are embedded into larger systems that provide the opportunity for the user to not only explore the mined patterns but also the input data (e.g., items, event sequences) as well as allowing the user to filter and query the input, apply transformations such as merging or substituting events up to creating rules to match patterns to newly simplified events of a higher semantic level. Simplifying and aggregating events does not only reduce the information that needs to be displayed, but also simplifies the identification and especially the re-identification of patterns by the user. Icons are, for example, a powerful tool to represent semantical patterns. Such operations are crucial to allow the user including her domain knowledge into the mining process. However, it does not guarantee that a pattern mining algorithm only reveals the interesting patterns which imposes the visual exploration of patterns.

To the best of our knowledge, there exists no dedicated visual design for high utility patterns. It is, however, clear that this additional information requires the use of another visual variable that is not already occupied and imposes the least perceptional bias and clutter. Additionally, the user should be able to gain insights into the lowest entities of the pattern (e.g., items, events) and their respective utilities to make informed decisions on the appropriate parameters and thresholds. As this will have a great impact on the scalability of the system, it will be useful to display the utility of a pattern first and only provide details on demand [83]. A visual analytics system must be able to give the user insight into the input data, i.e., the utilities of the entities. This also includes distributions as well as the aforementioned operations that are typically available in related systems. These possibilities and others such as templating will aid the user in setting the correct parameters and eventually empower the system to include the user's domain knowledge.

Besides high utility pattern mining, other extensions of the pattern mining methods covered in this survey exist. Examples are sequential rule mining, periodic pattern mining, and sequence prediction. As for high utility pattern mining, the commonality of these extensions is that the degree of freedom is further restricted and the complexity of the visualization must increase as more information and constraints are available. The comparison provided in this survey sheds light on the necessity to adjust the visualization technique towards the tasks and users. This aspect remains inevitable and probably becomes even more essential for these extensions.

To overcome the limitations of one visual design a system can include multiple different visual representations of the same data allowing to view the data from multiple perspectives. This has to be carefully designed, for example through linking and brushing [53] to provide a benefit to the user. Such systems often raise the complexity and steepen the learning curve.

Another interesting approach is the application of visual designs that offer a higher degree of freedom. For example, a preliminary analysis of sequential patterns can be gained by using techniques for visualizing itemsets and thus disregarding the order of the events at first. Then, the selection of the user could be displayed with visualization techniques for sequential patterns.

Interactions impose great benefits for the exploration and sense-making of patterns. Highlighting techniques can be useful to show similar or related patterns for example in a lattice based representation but also identifying the occurrence of items in other patterns. Filtering and search techniques support the user in verifying whether an expected pattern or similar patterns can be found in the result set.

Pattern mining is an essential part of the data mining field with many possible application domains. Visualization, on the other hand, is required for the exploration and sense-making of the mined patterns. As mentioned in Section 1, software libraries are available providing pattern mining algorithms. However, less support is available on the visualization side. Commercial visualization suites have no or only limited capabilities in visualizing this data type [10]. The arulesViz R-package [39] features different types of visualizations for association rules. This might be because no best-practice visualizations established that are superior to the other available techniques.

6 Conclusions

We provide a survey of visualization and visual analytic techniques for frequent itemsets, association rules, and sequential patterns which systematically compares the visual designs in each category to highlight their strengths and weaknesses.

This survey and the comparison of techniques reflects well that the perfect visual design does not exist and that compromises have to be made. Limitations can be mitigated using multiple different designs as well as interactive visualizations that feature filtering, sorting patterns according to various interestingness measures, templating, and provide details on demand.

References

1. C. C. Aggarwal and J. Han, editors. *Frequent Pattern Mining*. Springer, 2014.
2. R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.
3. R. Agrawal, M. Mehta, J. C. Shafer, R. Srikant, A. Arning, and T. Bollinger. The quest data mining system. In *KDD*, pages 244–249. AAAI Press, 1996.
4. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*, pages 487–499. Morgan Kaufmann, 1994.
5. R. Agrawal and R. Srikant. Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, pages 3–14. IEEE, 1995.
6. W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visualizing time-oriented data - A systematic view. *Computers & Graphics*, 31(3):401–409, 2007.
7. W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, 2011.
8. B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. J. Rodgers. The state-of-the-art of set visualization. *Comput. Graph. Forum*, 35(1):234–260, 2016.
9. G. L. Andrienko, N. V. Andrienko, P. Bak, D. A. Keim, and S. Wrobel. *Visual Analytics of Movement*. Springer, 2013.
10. M. Behrisch, D. Streeb, F. Stoffel, D. Seebacher, B. Matejek, S. H. Weber, S. Mittelstädt, H. Pfister, and D. Keim. Commercial visual analytics systems - advances in the big data analytics field. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2659–2668, to appear 2018.
11. J. Bernard, D. Sessler, T. May, T. Schlomm, D. Pehrke, and J. Kohlhammer. A visual-interactive system for prostate cancer cohort analysis. *IEEE Computer Graphics and Applications*, 35(3):44–55, 2015.
12. J. Bernard, N. Wilhelm, B. Krüger, T. May, T. Schreck, and J. Kohlhammer. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2257–2266, 2013.
13. J. Bertin. Sémiologie graphique: Les diagrammes-les réseaux-les cartes. 1973.
14. J. Bertin. Semiology of graphics: diagrams, networks, maps. 1983.
15. P. Bodesinsky, B. Alsallakh, T. Gschwandtner, and S. Miksch. Exploration and assessment of event data. In *Proceedings of EuroVis Workshop on Visual Analytics*, 2015.
16. C. Borgelt. Frequent item set mining. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 2(6):437–456, 2012.
17. G. Bothorel, M. Serrurier, and C. Hurter. Visualization of frequent itemsets with nested circular layout and bundling algorithm. In *ISVC (2)*, volume 8034 of *Lecture Notes in Computer Science*, pages 396–405. Springer, 2013.

18. C. Brunk, J. Kelly, and R. Kohavi. Mineset: An integrated system for data mining. In *KDD*, pages 135–138, 1997.
19. D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: A maximal frequent itemset algorithm for transactional databases. In *ICDE*, pages 443–452. IEEE Computer Society, 2001.
20. B. C. M. Cappers and J. J. van Wijk. Exploring multivariate event sequences using rules, aggregations, and selections. *IEEE Trans. Vis. Comput. Graph.*, 24(1):532–541, 2018.
21. Y. Chen, P. Xu, and L. Ren. Sequence synopsis: Optimize visual summary of temporal event data. *IEEE Trans. Vis. Comput. Graph.*, 24(1):45–55, 2018.
22. J. Chou, Y. Wang, and K. Ma. Privacy preserving event sequence data visualization using a sankey diagram-like representation. In *SIGGRAPH Asia Symposium on Visualization*, pages 1:1–1:8. ACM, 2016.
23. G. H. Collier. Thoth-ii: Hypertext with explicit semantics. In J. B. Smith and F. G. Halasz, editors, *Hypertext'87 Proceedings, November 13-15, 1987, Chapel Hill, North Carolina, USA*, pages 269–289. ACM, 1987.
24. G. Ellis and F. Mansmann. Mastering the information age solving problems with visual analytics. In *Eurographics*, volume 2, page 5, 2010.
25. G. P. Ellis and A. J. Dix. The plot, the clutter, the sampling and its lens: occlusion measures for automatic clutter reduction. In *AVI*, pages 266–269. ACM Press, 2006.
26. P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas. Fast vertical mining of sequential patterns using co-occurrence information. In *PAKDD (1)*, volume 8443 of *Lecture Notes in Computer Science*, pages 40–52. Springer, 2014.
27. P. Fournier-Viger, J. C. Lin, A. Gomariz, T. Gueniche, A. Soltani, Z. Deng, and H. T. Lam. The SPMF open-source data mining library version 2. In *ECML/PKDD (3)*, volume 9853 of *Lecture Notes in Computer Science*, pages 36–40. Springer, 2016.
28. P. Fournier-Viger, J. C. Lin, B. Vo, T. C. Truong, J. Zhang, and H. B. Le. A survey of itemset mining. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 7(4), 2017.
29. P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas. A survey of sequential pattern mining. *Data Science and Pattern Recognition*, 1(1):54–77, 2017.
30. T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. *ACM SIGMOD Record*, 25(2):13–23, 1996.
31. L. Geng and H. J. Hamilton. Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3):9, 2006.
32. F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory pattern mining. In *KDD*, pages 330–339. ACM, 2007.
33. B. Goethals. Survey on frequent pattern mining. *Univ. of Helsinki*, 19:840–852, 2003.
34. B. Goethals and M. J. Zaki, editors. *FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA*, volume 90 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2003.
35. D. Gotz and H. Stavropoulos. Decisionflow: Visual analytics for high-dimensional temporal event sequence data. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1783–1792, 2014.
36. D. Gotz, F. Wang, and A. Perer. A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data. *Journal of Biomedical Informatics*, 48:148–159, 2014.
37. P. Grünwald. A tutorial introduction to the minimum description length principle. *CoRR*, math.ST/0406077, 2004.
38. M. Guzdial, C. Walton, M. Konemann, and E. Soloway. Characterizing process change using log file data. Technical report, Georgia Institute of Technology, 1993.
39. M. Hahsler and S. Chelluboina. Visualizing association rules: Introduction to the r-extension package arulesviz. *R project module*, pages 223–238, 2011.
40. J. Han. Mining knowledge at multiple concept levels. In *CIKM*, pages 19–24. ACM, 1995.
41. J. Han and N. Cercone. Aviz: A visualization system for discovering numeric association rules. In *PAKDD*, volume 1805 of *Lecture Notes in Computer Science*, pages 269–280. Springer, 2000.

42. J. Han, Y. Fu, W. Wang, J. Chiang, W. Gong, K. Koperski, D. Li, Y. Lu, A. Rajan, N. Stefanovic, B. Xia, and O. R. Zaïane. Dbminer: A system for mining knowledge in large relational databases. In *KDD*, pages 250–255. AAAI Press, 1996.
43. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein, editors, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.*, pages 1–12. ACM, 2000.
44. J. A. Hartigan and B. Kleiner. Mosaics for contingency tables. In *Computer science and statistics: Proceedings of the 13th symposium on the interface*, pages 268–273. Springer, 1981.
45. J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mininga general survey and comparison. *ACM sigkdd explorations newsletter*, 2(1):58–64, 2000.
46. D. Hoaglin, F. Mosteller, and J. Tukey. 1983, understanding robust and exploratory data analysis.
47. H. Hofmann. Exploring categorical data: interactive mosaic plots. *Metrika*, 51(1):11–26, 2000.
48. H. Hofmann, A. Siebes, and A. F. X. Wilhelm. Visualizing association rules with interactive mosaic plots. In R. Ramakrishnan, S. J. Stolfo, R. J. Bayardo, and I. Parsa, editors, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*, pages 227–235. ACM, 2000.
49. M. Hu, K. Wongsuphasawat, and J. T. Stasko. Visualizing social media content with sententree. *IEEE Trans. Vis. Comput. Graph.*, 23(1):621–630, 2017.
50. W. Jentner, M. El-Assady, B. Gipp, and D. A. Keim. Feature alignment for the analysis of verbatim text transcripts. In *EuroVA 2017: EuroVis Workshop on Visual Analytics*, pages 13–18, 2017.
51. W. Jentner, D. Sacha, F. Stoffel, G. Ellis, L. Zhang, and D. A. Keim. Making Machine Intelligence Less Scary for Criminal Analysts: Reflections on Designing a Visual Comparative Case Analysis Tool. *The Visual Computer Journal*, 2018.
52. R. J. B. Jr., B. Goethals, and M. J. Zaki, editors. *FIMI '04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004*, volume 126 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2005.
53. D. A. Keim. Information visualization and visual data mining. *IEEE Trans. Vis. Comput. Graph.*, 8(1):1–8, 2002.
54. D. A. Keim, J. Schneidewind, and M. Sips. Fp-viz: Visual frequent pattern mining. In *InfoVis*, 2005.
55. M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *CIKM*, pages 401–407. ACM, 1994.
56. S. Kotsiantis and D. Kanellopoulos. Association rules mining: A recent overview. *GESTS International Transactions on Computer Science and Engineering*, 32(1):71–82, 2006.
57. J. B. Kruskal and J. M. Landwehr. Icicle plots: Better displays for hierarchical clustering. *The American Statistician*, 37(2):162–168, 1983.
58. H. Lam, D. M. Russell, D. Tang, and T. Munzner. Session viewer: Visual exploratory analysis of web session logs. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, IEEE VAST 2007, Sacramento, California, USA, October 30-November 1, 2007*, pages 147–154. IEEE Computer Society, 2007.
59. I. Lee, G. Cai, and K. Lee. Mining points-of-interest association rules from geo-tagged photos. In *46th Hawaii International Conference on System Sciences, HICSS 2013, Wailea, HI, USA, January 7-10, 2013*, pages 1580–1588. IEEE Computer Society, 2013.
60. C. K. Leung and C. L. Carmichael. Fpvat: a visual analytic tool for supporting frequent pattern mining. *SIGKDD Explorations*, 11(2):39–48, 2009.
61. C. K. Leung, P. Irani, and C. L. Carmichael. Fisviz: A frequent itemset visualizer. In T. Washio, E. Suzuki, K. M. Ting, and A. Inokuchi, editors, *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008, Osaka, Japan, May*

20-23, 2008 Proceedings, volume 5012 of *Lecture Notes in Computer Science*, pages 644–652. Springer, 2008.

62. C. K. Leung, P. Irani, and C. L. Carmichael. Wifisviz: Effective visualization of frequent itemsets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*, pages 875–880. IEEE Computer Society, 2008.
63. C. K. Leung, F. Jiang, and P. P. Irani. Fpmapviz: A space-filling visualization for frequent patterns. In M. Spiliopoulou, H. Wang, D. J. Cook, J. Pei, W. Wang, O. R. Zaïane, and X. Wu, editors, *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*, pages 804–811. IEEE Computer Society, 2011.
64. C. K. Leung, V. V. Kononov, A. G. M. Pazdor, and F. Jiang. Pyramidviz: Visual analytics and big data visualization for frequent patterns. In *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress, DASC/PiCom/DataCom/CyberSciTech 2016, Auckland, New Zealand, August 8-12, 2016*, pages 913–916. IEEE Computer Society, 2016.
65. C. K.-S. Leung and F. Jiang. Radialviz: an orientation-free frequent pattern visualizer. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 322–334. Springer, 2012.
66. B. Liu, W. Hsu, S. Chen, and Y. Ma. Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems*, 15(5):47–55, 2000.
67. Z. Liu, B. Kerr, M. Dontcheva, J. Grover, M. Hoffman, and A. Wilson. Coreflow: Extracting and visualizing branching patterns from event sequences. *Comput. Graph. Forum*, 36(3):527–538, 2017.
68. Z. Liu, Y. Wang, M. Dontcheva, M. Hoffman, S. Walker, and A. Wilson. Patterns and sequences: Interactive exploration of clickstreams to understand common visitor paths. *IEEE Trans. Vis. Comput. Graph.*, 23(1):321–330, 2017.
69. N. R. Mabroukeh and C. I. Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.*, 43(1):3:1–3:41, 2010.
70. J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. Graph.*, 5(2):110–141, 1986.
71. H. Mannila and C. Meek. Global partial orders from sequential data. In R. Ramakrishnan, S. J. Stolfo, R. J. Bayardo, and I. Parsa, editors, *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*, pages 161–168. ACM, 2000.
72. H. Mannila, H. Toivonen, and A. I. Verkamo. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, 1(3):259–289, 1997.
73. M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *IEEE Trans. Vis. Comput. Graph.*, 19(12):2227–2236, 2013.
74. T. Munzner. *Visualization Analysis and Design*. A.K. Peters visualization series. A K Peters, 2014.
75. T. Munzner, Q. Kong, R. T. Ng, J. Lee, J. Klawe, D. Radulovic, and C. K. Leung. Visual mining of power sets with large alphabets. *Department of Computer Science, The University of British Columbia*, 2005.
76. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Efficient mining of association rules using closed itemset lattices. *Inf. Syst.*, 24(1):25–46, 1999.
77. D. Patnaik, P. Butler, N. Ramakrishnan, L. Parida, B. J. Keller, and D. A. Hanauer. Experiences with mining temporal event sequences from electronic medical records: initial successes and some challenges. In C. Apté, J. Ghosh, and P. Smyth, editors, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*, pages 360–368. ACM, 2011.
78. A. Perer and F. Wang. Frequence: interactive mining and visualization of temporal frequent event sequences. In T. Kuflik, O. Stock, J. Y. Chai, and A. Krüger, editors, *19th International Conference on Intelligent User Interfaces, IUI 2014, Haifa, Israel, February 24-27, 2014*, pages 153–162. ACM, 2014.

79. A. Perer, F. Wang, and J. Hu. Mining and exploring care pathways from electronic medical records with visual analytics. *Journal of Biomedical Informatics*, 56:369–378, 2015.
80. C. P. Rainsford and J. F. Roddick. Visualisation of temporal interval association rules. In K. Leung, L. Chan, and H. Meng, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2000, Data Mining, Financial Engineering, and Intelligent Agents, Second International Conference, Shatin, N.T. Hong Kong, China, December 13-15, 2000, Proceedings*, volume 1983 of *Lecture Notes in Computer Science*, pages 91–96. Springer, 2000.
81. D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. P. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1604–1613, 2014.
82. B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.
83. B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL*, pages 336–343. IEEE Computer Society, 1996.
84. S. F. Silva and T. Catarci. Visualization of linear time-oriented data: A survey. In *WISE*, pages 310–319. IEEE Computer Society, 2000.
85. J. T. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *INFOVIS*, pages 57–65. IEEE Computer Society, 2000.
86. C. D. Stolper, A. Perer, and D. Gotz. Progressive visual analytics: User-driven visual exploration of in-progress analytics. *IEEE Trans. Vis. Comput. Graph.*, 20(12):1653–1662, 2014.
87. F. B. Viégas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *IEEE Trans. Vis. Comput. Graph.*, 15(6):1137–1144, 2009.
88. K. Vrotsou, J. Johansson, and M. D. Cooper. Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. *IEEE Trans. Vis. Comput. Graph.*, 15(6):945–952, 2009.
89. F. Wanner, W. Jentner, T. Schreck, A. Stoffel, L. Sharalieva, and D. A. Keim. Integrated visual analysis of patterns in time series and text data - workflow and application to financial data analysis. *Information Visualization*, 15(1):75–90, 2016.
90. M. Wattenberg. Arc diagrams: Visualizing structure in strings. In *INFOVIS*, pages 110–116. IEEE Computer Society, 2002.
91. J. Wei, Z. Shen, N. Sundaresan, and K. Ma. Visual cluster exploration of web clickstream data. In *IEEE VAST*, pages 3–12. IEEE Computer Society, 2012.
92. I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.
93. P. C. Wong, P. Whitney, and J. Thomas. Visualizing association rules for text mining. In *IEEE Symposium on Information Visualization 1999 (INFOVIS'99), San Francisco, California, USA, October 24-29, 1999.*, pages 120–123. IEEE Computer Society, 1999.
94. K. Wongsuphasawat, J. A. G. Gómez, C. Plaisant, T. D. Wang, M. Taieb-Maimon, and B. Shneiderman. Lifeflow: visualizing an overview of event sequences. In *CHI*, pages 1747–1756. ACM, 2011.
95. K. Wongsuphasawat and J. J. Lin. Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at twitter. In *IEEE VAST*, pages 113–122. IEEE Computer Society, 2014.
96. S. Wu and Y. Chen. Discovering hybrid temporal patterns from sequences consisting of point- and interval-based events. *Data Knowl. Eng.*, 68(11):1309–1330, 2009.
97. J. Yang, M. O. Ward, E. A. Rundensteiner, and A. Patro. Interring: a visual interface for navigating and manipulating hierarchies. *Information Visualization*, 2(1):16–30, 2003.
98. L. Yang. Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates. In *ICCSA (1)*, volume 2667 of *Lecture Notes in Computer Science*, pages 21–30. Springer, 2003.
99. L. Yang. Pruning and visualizing generalized association rules in parallel coordinates. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):60–70, 2005.
100. M. J. Zaki and C. Hsiao. CHARM: an efficient algorithm for closed itemset mining. In *SDM*, pages 457–473. SIAM, 2002.

101. J. Zhao, Z. Liu, M. Dontcheva, A. Hertzmann, and A. Wilson. Matrixwave: Visual comparison of event sequence data. In *CHI*, pages 259–268. ACM, 2015.
102. Q. Zhao and S. S. Bhowmick. Sequential pattern mining: A survey. *ITechnical Report CAIS Nayang Technological University Singapore*, 1:26, 2003.