Visual Pattern Analytics for Event Sequences

Doctoral thesis for obtaining the academic degree Doctor of Natural Sciences (Dr. rer. nat.)

submitted by
Wolfgang Jentner

at the

Universität
Konstanz

Faculty of Sciences
Computer and Information Science

Konstanz, 2023

Day of the oral examination: April 27th 2023

Supervisors:

1. Prof. Dr. Daniel A. Keim, Universität Konstanz
2. Prof. Dr. Kwan-Liu Ma, University of California, Davis

Science may set limits to knowledge,
but should not set limits to imagination.

BERTRAND RUSSEL

Abstract

Pattern mining plays an essential role in unsupervised machine learning as it allows the clustering of structured data without requiring distance measures and purely relying on the definition of containment. Because it is unsupervised, it is predestined for exploratory analysis, and visual analytics offers a holistic perspective thoroughly involving the data, task, and especially the user in the decision-making process of designing tools for exploratory analysis. Pattern mining can easily generate millions of patterns since the search spaces are exponential. Additionally, the structures are often large and complex, which thwarts sense-making efforts by the user.

This dissertation explains how visual analytics can be leveraged to allow the effective exploration of sequentially structured data using pattern mining algorithms. The first focus is on interesting measures, a concept known from data mining that should quantify interestingness. Because interestingness is subjective and heavily depends on the task and the user, this work argues for understanding interestingness measures as features that quantify different properties of the patterns and the clusters they represent. It further presents an alternative taxonomy of available features that can be used in pattern mining and discusses their importance and limitations.

Secondly, this work surveys visualization techniques for structured data patterns, including their features, and highlights the differences between structured data as the input for the mining and the patterns themselves. Furthermore, it discusses the limitations of the visualization techniques, especially concerning scalability and the number of features.

Finally, well-known visual analytics concepts such as interactive visualizations, progressive visual analytics, or concepts from visual text analytics are being transferred for pattern mining and the exploration of patterns. It is explained and discussed how these concepts can be exploited and implemented to mitigate the effects of the exponential search spaces and the complexity of the patterns to ease the user's burden during the exploration process.

Even though this work focuses on event sequences and sequential patterns, all aspects can be transferred onto different data structures and pattern mining algorithms. Therefore, this dissertation provides a foundation for the exploratory analysis of structured data using pattern mining with countless possible extensions to inspire future research.

Abstract in simple language

In search of the most beautiful rainbow: If you let your imagination run wild, what would be the most beautiful rainbow you could imagine? If I give you seven colors to choose from, could you create the most beautiful rainbow from them? With up to seven colors there are already 127 different rainbows, with eight colors there are already 255, and with ten colors we can create more than 1000 different rainbows. Maybe you know the game Taboo, where you describe a term without being allowed to name this term or very similar terms directly. Your teammates then have to guess the term. We can set up similar game rules to describe our most beautiful rainbow. What if we had to describe the perfect rainbow without naming the colors? Could you create such a description? I would describe that my rainbow must consist of at least six colors and the colors should be arranged from light to dark. Now, as you can probably guess, this description applies to more than one rainbow. But this already helps, because instead of 1000 rainbows I only have to choose from the remaining 10. This choice is much easier. Of course, this means that the description has to be correct at first.

But why do we even bother and make the game so complicated? What increases the fun of playing Taboo does not necessarily make sense in real life? The more generally valid the description, the more powerful it is - as long as it is accurate. If we assume that my description always gives us the most beautiful rainbows, then I never have to adjust it at all, no matter what colors are given. But what happens if I give you only shades of gray as colors? You are probably disappointed by my lack of imagination or even think I am manic-depressive. But would your or my description still filter out the most beautiful rainbows? And what if we now no longer want to find the most beautiful rainbow, but instead the most delicious smoothie? You probably quickly realize that you won't get very far here with colors or descriptions of rainbows. Everything has its limits.

In my work, I delve into how such descriptions of a pattern (e.g., rainbows) are best created and also best understood. In the example, you probably already noticed that such descriptions often consist of several parts. But of course, we do not want to write a novel for our description of a beautiful rainbow. The shorter the better and the more general the more powerful. You probably also realized that your description and my description of the most beautiful rainbow are not necessarily the same. This makes things more complicated.

Another part of my work is how we can best represent patterns. For rainbows, this may be very simple, since we all have an idea of what a rainbow looks like. But if I presented you with 1000 rainbows on the screen, would you quickly find your most beautiful rainbow? It will probably take you quite a while and you will also quickly lose interest. But there are ways I can help you. For example, I can arrange the rainbows so that similar rainbows are

close to each other on the screen. I could also display different descriptions of a group of rainbows first and you pick the best group from that. The goal here is to be able to display as many patterns or descriptions of patterns as necessary without overwhelming you.

The last part of my work is about how you can interact with a computer program. For rainbows, for example, I might parts of a rainbow with fewer colors, from which you choose one that should definitely not be missing from your rainbow. After that, you get to choose some rainbows with more colors that you can choose from. From your answers, I can then create a selection that is much smaller than the original 1000 rainbows. And hopefully, you will quickly find the most beautiful one.

At the end of my work, unfortunately, no pot of gold awaits you. But at least a collection of different techniques and strategies, of how you can find it: the most beautiful of all rainbows.

Zusammenfassung

Das Pattern Mining spielt eine wesentliche Rolle beim unüberwachten maschinellen Lernen, da es das Clustern strukturierter Daten ermöglicht, ohne Abstandsmaße zu benötigen und sich lediglich auf die Definition der Eingrenzung zu verlassen. Da es unbeaufsichtigt ist, ist es prädestiniert für die explorative Analyse, und die visuelle Analyse bietet eine ganzheitliche Perspektive, die die Daten, die Aufgabe und vor allem den Benutzer in den Entscheidungsprozess bei der Entwicklung von Tools für die explorative Analyse einbezieht. Pattern Mining kann leicht Millionen von Mustern erzeugen, da die Suchräume exponentiell sind. Darüber hinaus sind die Strukturen oft groß und komplex, was die Bemühungen des Benutzers, einen Sinn zu finden, vereitelt.

In dieser Dissertation wird erläutert, wie die visuelle Analyse genutzt werden kann, um eine effektive Erkundung von sequentiell strukturierten Daten mit Hilfe von Pattern-Mining-Algorithmen zu ermöglichen. Der erste Schwerpunkt liegt auf interessanten Maßen, einem aus dem Data Mining bekannten Konzept, das die Interessantheit quantifizieren soll. Da Interessantheit subjektiv ist und stark von der Aufgabe und dem Benutzer abhängt, plädiert diese Arbeit dafür, Interessantheitsmaße als Merkmale zu verstehen, die verschiedene Eigenschaften der Muster und des Clusters, das sie repräsentieren, quantifizieren. Darüber hinaus wird eine alternative Taxonomie verfügbarer Merkmale vorgestellt, die beim Pattern Mining verwendet werden können, und ihre Bedeutung und Grenzen werden diskutiert.

Zweitens gibt diese Arbeit einen Überblick über Visualisierungstechniken für strukturierte Datenmuster, einschließlich ihrer Merkmale, und hebt die Unterschiede zwischen strukturierten Daten als Input für das Mining und den Mustern selbst hervor. Darüber hinaus werden die Grenzen der Visualisierungstechniken erörtert, insbesondere hinsichtlich der Skalierbarkeit und der Anzahl der Merkmale.

Schließlich werden bekannte Visual-Analytics-Konzepte wie interaktive Visualisierungen, progressive Visual Analytics oder Konzepte aus der visuellen Textanalytik für das Pattern Mining und die Exploration von Mustern übertragen. Es wird erläutert und diskutiert, wie diese Konzepte genutzt und implementiert werden können, um die Auswirkungen der exponentiellen Suchräume und die Komplexität der Muster abzumildern und den Benutzer während des Explorationsprozesses zu entlasten.

Obwohl sich diese Arbeit auf Ereignisabläufe und sequentielle Muster konzentriert, können alle Aspekte auf andere Datenstrukturen und Pattern Mining Algorithmen übertragen werden. Daher bietet diese Dissertation eine Grundlage für die explorative Analyse strukturierter Daten mit Hilfe von Pattern Mining mit unzähligen möglichen Erweiterungen, um zukünftige Forschung zu inspirieren.

Zusammenfassung in einfacher Sprache

Auf der Suche nach dem schönsten Regenbogen: Wenn sie ihrer Fantasie freien Lauf lassen, was wäre der schönste Regenbogen, den sie sich vorstellen können? Wenn ich ihnen sieben Farben vorgebe, aus denen sie auswählen dürfen, könnten sie dann den schönsten Regenbogen daraus erzeugen? Mit bis zu sieben Farben gibt es bereits 127 verschiedene Regenbögen, bei acht Farben sind es schon 255, und bei zehn Farben können wir mehr als 1000 verschiedene Regenbögen erzeugen. Vielleicht kennen sie das Spiel Tabu, bei dem sie einen Begriff beschreiben, ohne diesen oder sehr ähnliche Begriffe direkt nennen zu dürfen. Ihre Mitspieler müssen dann den Begriff erraten. Wir können ähnliche Spielregeln aufstellen, um unseren schönsten Regenbogen zu beschreiben. Wie wäre es, wenn wir den perfekten Regenbogen beschreiben müssen, ohne die Farben zu benennen? Könnten sie eine solche Beschreibung erstellen? Meine Beschreibung wäre, dass mein Regenbogen aus mindestens sechs Farben bestehen muss und die Farben von hell nach dunkel angeordnet sein sollen. Wie sie sich nun vermutlich denken können, trifft diese Beschreibung auf mehr als nur einen Regenbogen zu. Aber auch das hilft, denn, wenn ich statt 1000 Regenbögen nur aus den verbleibenden 10 auswählen muss, so fällt mir meine Entscheidung deutlich leichter. Dies bedeutet aber natürlich, dass meine Beschreibung erst einmal stimmen muss.

Aber warum machen wir uns überhaupt die Mühe und das Spiel so kompliziert? Was bei Tabu den Spielspaß erhöht, muss im echten Leben ja nicht immer sinnvoll sein? Je allgemeingültiger die Beschreibung ist, desto mächtiger ist sie - solange sie zutrifft. Wenn wir annehmen, dass wir mit meiner Beschreibung immer die schönsten Regenbögen erhalten, dann muss ich sie gar nie anpassen, egal welche Farben vorgegeben sind. Aber was passiert, wenn ich ihnen z.B. nur Grautöne als Farben vorgebe? Sie sind wohl erst einmal von meiner Fantasielosigkeit maßlos enttäuscht oder halten mich sogar für manisch-depressiv. Aber würde meine oder ihre eigene Beschreibung immer noch die schönsten Regenbögen herausfiltern? Und wenn wir nun nicht mehr den schönsten Regenbogen finden möchten, sondern stattdessen den leckersten Smoothie? Sie merken schnell, dass sie hier mit Farben nicht mehr sehr weit kommen. Alles hat eben seine Grenzen.

In meiner Arbeit gehe ich darauf ein, wie man solche Beschreibungen eines Musters (z.B. Regenbögen) am besten erstellt und auch am besten versteht. In dem Beispiel haben sie vermutlich schon bemerkt, dass solche Beschreibungen öfters aus mehreren Teilen bestehen. Aber wir möchten natürlich auch nicht erst einen Roman schreiben für unsere Beschreibung eines schönsten Regenbogens. Je kürzer, desto besser und je allgemeiner, desto mächtiger. Sie haben sicherlich auch schon bemerkt, dass ihre und meine Beschreibung nicht unbedingt gleich sein müssen. Das macht die Sache natürlich noch komplizierter.

Ein weiterer Teil meiner Arbeit ist, wie wir am besten Muster darstellen können. Für Regenbögen mag das sehr einfach sein, da wir alle eine Vorstellung davon haben, wie ein Regenbogen aussieht. Aber wenn ich ihnen 1000 Regenbögen auf dem Bildschirm präsentiere, würden sie ihren schönsten Regenbogen schnell finden? Vermutlich brauchen sie eine ganze Weile und verlieren auch schnell die Lust. Aber es gibt Möglichkeiten, mit denen ich ihnen helfen kann. Zum Beispiel kann ich die Regenbögen so anordnen, dass ähnliche Regenbögen nahe bei einander sind. Ich könnte auch erst einmal verschiedene Beschreibungen einer Gruppe von Regenbögen darstellen und sie suchen sich daraus die Beste heraus. Das Ziel ist hier so viele Muster oder Beschreibungen der Muster wie notwendig darstellen zu können, ohne sie dabei zu überfordern.

Der letzte Teil meiner Arbeit beschäftigt sich damit, wie sie mit einem Computer-Programm interagieren können. Für Regenbögen könnte ich ihnen z.B. erst einmal vier Farben anzeigen, aus denen sie eine auswählen, die auf keinen Fall in ihrem Regenbogen fehlen darf. Danach dürfen sie noch zwei weitere Farben auswählen und dann frage ich sie noch, wie viele weitere Farben mindestens in ihrem Regenbogen sein müssen. Aus ihren Antworten kann ich dann eine Auswahl erzeugen, die sehr viel kleiner ist als die ursprünglichen 1000 Regenbögen. Und hoffentlich finden sie dann schnell den schönsten.

Am Ende meiner Arbeit erwartet sie nun leider kein Topf voll Gold. Aber immerhin eine Sammlung aus verschiedenen Techniken und Strategien, wie sie ihn denn nun finden können: den schönsten aller Regenbogen.

Acknowledgement

The Data Analysis and Visualization Group has been a part of my life for over a decade. When I started as a research assistant, I was amazed at how kind, creative, and inspiring the people in this group are. Paired with ever-exciting projects and opportunities, this eased my decision to pursue a Ph.D. in the group.

I am forever grateful for the guidance, mentoring, and always honest feedback of my supervisor, Daniel Keim. His kindness, humility and strive to set up people for success have inspired me ever since. Daniel's early trust in me to handle many of the group's grant projects has set me up to challenge myself. I have been rewarded with many experiences, working with amazing colleagues, and so many memories that will always put a smile on my face. I would also like to thank Kwan-Liu Ma, whose deep-provoking thoughts and conversations have helped me shape my dissertation significantly.

Fortunately, I was welcomed and integrated into a group that taught me everything necessary to be a successful researcher. I want to especially thank Florian Stoffel, Juri Buchmüller, Dominik Jäckle, and Dominik Sacha for their guidance, feedback, and friendship.

I was able to work with amazing colleagues in the group, which sparked many interesting thoughts and discussions. I would personally like to thank Matthias Kraus, Fabian Sperrle, Rita Sevastjanova, Udo Schlegel, Matthias Miller, Thilo Spinner, Frederik Dennig, Eren Cakmak, Maximilian Fischer, Daniel Seebacher, Dirk Streeb, Menna El-Assady, Geoffrey Ellis, Johannes Fuchs, Matthew Sharinghousen, and Benedikt Bäumle, for their input, great collaborations, papers, and projects. I want to thank Sabine Kuhr. I appreciated her support and experience with many administrative questions about research projects and academia.

I would also like to thank my external project collaborators and visiting researchers for their support in my endeavors. Their feedback and deep collaboration have shaped my academic path and experiences significantly.

I want to thank my parents, Regina and Bernhard, and my sister, Margrit, for their continuous support and love. Finally, I want to thank my beloved wife, Angie, for her love and support throughout these years. Even though it meant being apart for a long time, you have always encouraged, supported, and pushed me to continue my passion.

Declaration of the usage of AI

I hereby declare that I have composed this work independently and without using any generative artificial intelligence. This is also true for my publications used in this thesis. No figures in this thesis have been generated or modified by artificial intelligence. Artificial intelligence tools have only been used for spellchecking and to check for grammatical errors (i.e., Grammarly, LanguageTool, and Microsoft Word).

Contents

List of Figures

List of Tables

Introduction 1

We are surrounded by patterns and our mind constantly tries to recognize patterns from the input of our senses. A pattern is typically referred to as a regularity, thus helping us to make predictions and formulate theories and ideas. Our strive to observe and find patterns, trends, and correlations in nature was quickly picked up in computer science forming the fields of data mining, knowledge discovery in databases, and machine learning which are also summarized as the term artificial intelligence [1].

A vast amount of data is semi-structured and can be modeled as structured data. Data mining for structured data is known as structure mining or structured data mining [2, 3]. As with all data, we strive to identify patterns in such datasets based on their structural information. The task is named *pattern mining* and is classified as unsupervised machine learning and, more accurately, clustering. Standard clustering techniques cannot be easily adapted for structured data as these techniques require a distance measure which is not readily available for structured data. Instead, pattern mining focuses on structural *containment*. Therefore, a pattern in structure mining is nothing else than a cluster representative which is contained at least once in the structured entities it represents. Hence, a pattern is an abstraction or simplification of the data it represents.

Because of its peculiarities, specifically the unsupervised machine learning aspect, this field has a high correlation with exploratory data analysis (EDA). A term which has been heavily promoted by John Tukey [4]. Hoaglin et al. define it as:

> "Exploratory data analysis isolates patterns and features of the data and reveals these forcefully to the analyst." [4]

The research field of pattern mining has invented many measures that can serve as constraints for the algorithm to find patterns of relevance or interest. EDA also sparks some criticism because *exploration* or *exploratory* is vague and does not describe a specific task a user has to solve with a system [5].

[1]: Zhou (2021), Machine Learning

[2]: Westphal et al. (1998), Data mining solutions: methods and tools for solving real-world problems

[3]: Liu (2011), Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Second Edition

[4]: Hoaglin et al. (1983), Understanding robust and exploratory data analysis

[5]: Adar (2017), Banning exploration in my infovis class

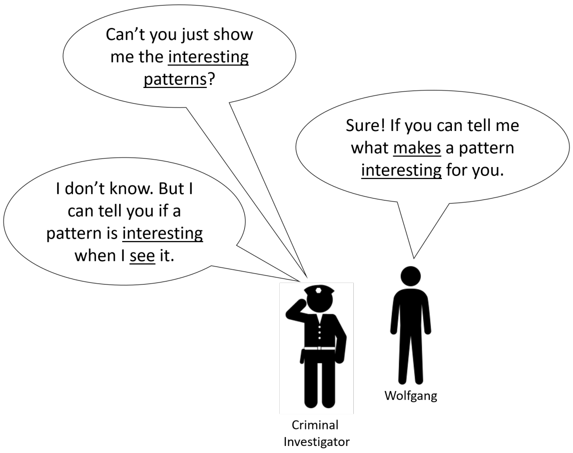

Can't you just show me the interesting patterns?
Sure! If you can tell me what makes a pattern interesting for you.
I don't know. But I can tell you if a pattern is interesting when I see it.
Criminal Investigator
Wolfgang

Figure 1.1: A conversation between a criminal investigator and me during the VALCRI project highlighting the motivation of my work.

The same article describes that an EDA system consists of two main parts: perceptual classification (i.e., pattern-finding) and perceptual clustering (i.e., pattern-making). The former is dedicated to identifying novel patterns in an unknown dataset whereas the latter refers to identifying whether a previously found pattern is repeating. Moreover, Eytan Adar states:

> "[...] a successful exploratory tool is one that lets the analyst find the patterns they are looking for in the data (quickly, accurately, reliably, scalably)." [5]

I strongly argue that any EDA system must additionally convey the semantics of the data such that a user can make an informed decision about whether a specific pattern is relevant.

In EDA, a user has typically little knowledge of the exact threshold and specific constraints which persuades the user to relax the constraints quickly causing a so-called *pattern explosion*. Because pattern mining relies on containment, search spaces are exponential, which in an unconstrained algorithm, can cause the mining of billions and trillions of patterns - even for small datasets. The number of possible pattern combinations can easily exceed the number of atoms in our observable universe which is currently estimated to be around 10^{82} [1]. Such a large amount of information can neither be displayed on a single screen nor cognitively processed by the user.

Figure 1.1 depicts a conversation I have had with a criminal investigator during an evaluation session in the VALCRI

[1] livescience.com/how-many-atoms-in-universe.html, accessed March 25, 2022

project[2]. The conversation nicely pinpoints the problem of pattern mining in EDA where a user is not able to tightly constrain the pattern mining algorithm but on the other hand, is overwhelmed by the sheer number of resulting patterns. However, the user is confident to identify an interesting pattern by seeing it which implies that it needs to be found first.

This challenging problem can be conquered using interaction. Jark van Wijk states that interaction is deemed necessary to explore data that does not fit onto a single screen [6]. The field of visual analytics expands on this process to fully integrate the user into the analysis process. Therefore, it can be stated that visual analytics combines human domain knowledge with artificial intelligence and machine learning methods through interactive visual interfaces.

[6]: Wijk (2006), Views on Visualization

This formulates my overall research question stated as:

> How to use visual analytics to foster exploration
> and sense-making of sequential patterns?

I have selected event sequences as a primary data structure throughout my research as they offer a variety of application areas and yet have extremely large search spaces. However, in much of my research, I have looked beyond and most of the conclusions can be transferred to any type of structured data.

1.1 Contributions and Outline

In order to answer the stated research question it needs to be broken down into more specific aspects which each form a chapter in this dissertation. The conversation with the criminal investigator sheds light on the fact that a large part of EDA in structure mining is to find and assess interesting patterns in the data. The data mining field endeavors to measure such interestingness of patterns to support the user in ranking or filtering for specific patterns. A plethora of measures have been proposed, often in conjunction with dedicated algorithms to solve a specific task. Chapter 3 critically assesses interestingness measures from a visual analytics perspective assuming that any system and its success depends on the data, task, and user. It provides an alternative interpretation and complementary dimension to the existing taxonomies of

[2] `valcri.org`, accessed March 25, 2022

interestingness measures. Furthermore, it describes several use cases based on my publications [7–10] to show how interestingness measures can be applied to certain domains and tasks.

Chapter 4 surveys visualization and visual analytics techniques for patterns. To the best of my knowledge, it is the first survey of its kind that focuses on patterns only while providing an overview of different types of structured data such as itemsets, association rules, sequential patterns, and episodes. Furthermore, this chapter contains a comparison of each technique concerning the scalability of certain aspects such as the size of the alphabet (number of items), as well as, the scalability of transactions. It underlines that if the structure is visualized in detail such that the individual items remain visible the overall scalability suffers whereas visualizations focusing on interestingness measures tend to scale better but may lack (critical) information. The chapter further contributes to the part of the sense-making of my research question in that it shows how structure can be effectively visualized to be intuitive for a user.

Chapter 5, called visual pattern analytics, then takes several techniques known from interactive visualization and visual analytics and transfers them for the exploratory analysis of patterns and pattern mining, effectively showing that the individual shortcomings of interestingness measures, structure, and their visual tradeoffs can be mitigated and their strengths boosted. It further discusses how progressive visual analytics can be leveraged to overcome the problem of the exponential search spaces and discusses the role of explainable artificial intelligence in the area of exploration of patterns. It further discusses how such approaches and applications can be evaluated and the limitations of user evaluations concerning pattern exploration.

This thesis provides the foundation for exploratory data analysis of sequential patterns. With the user's primary intent to find interesting patterns, the thesis provides insights into how interestingness measures and algorithms can be leveraged to mine for patterns of potential interest and how results can be visualized effectively and communicated to the user. Furthermore, it provides various approaches to how the user and the mining process can be more tightly coupled to improve the exploration and search while showing tradeoffs and limits of different aspects. Therefore, this thesis contributes a framework of techniques, designs, and approaches and shows how they can be combined to foster the exploration and sense-making of sequential patterns.

1.2 Publications

My research in visual pattern analytics allowed me to publish relevant results and techniques in competitive journals and conferences. This thesis is the summary of several of these publications and puts them into context. Since research is always a joint effort, in the following, I want to clarify my contributions to these publications. Throughout my dissertation, I will use the terms "taken from" and "based on" following the terminology of Daniel Seebacher [11]. Throughout my dissertation, these terms will label chapters and sections to clarify which of my publications have been used in the relevant parts. The labels will state which publication and section therein have been used and which co-authors have contributed.

Taken from Sections that are *taken from* one of my publications contain only minor edits in comparison to the publication. They may have been shortened, and edits are only made where deemed necessary to point to the correct references. For these parts, I have been the main contributor to the paper and have written these parts by myself. My co-authors and other reviewers have provided feedback on these sections, which I have implemented whenever possible.

Based on Sections that are *based on* my publications contain the essential content of the relevant part but not the written passage. These are mainly passages where I cannot confidently certify that I am the main author of these parts. I typically use the same references but typically add additional details in my thesis as some details have been stripped of the published paper.

If a section or chapter has not been labeled explicitly, it can be assumed that these parts have not been previously published in the same form. These are typically the parts that provide additional background and context to my publications.

1.2.1 Used publications and contributions

Journal Articles

- **W. Jentner**, D. Sacha, F. Stoffel, G. Ellis, L. Zhang, 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 [8]

  This paper reports on our work in the VALCRI project [12]. It is building upon several publications that report earlier and intermediate progress [7, 13, 14]. Florian Stoffel contributed to the concept extraction and underlying NLP pipelines. Dominik Sacha contributed to early prototypes of the similarity space selector (S^3) and the crime cluster table (CCT). I integrated updated versions of the S^3, CCT components into the VALCRI prototype and added the sequence modeling. I further added the sequence similarity space selector (S^4), as well as, the weight observer component (WOC). The ontology vis component was contributed by Yevgen Kuzmenko who was supervised by Dominik Sacha and me. A co-occurrence matrix for the features, an early prototype of the S^4, was contributed by Raphael Buchmüller who was supervised by Dominik Sacha and me. The paper reports on the design process of each component and the integrated components in the VALCRI prototype. I took primary responsibility for this publication. The contributions of the publication emerged through many group discussions and feedback including Leishi Zhang and Geoffrey Ellis. The paper was written in a collaborative manner where each person focused on their contributions (i.e., components description, their design history, and related work). Daniel Keim commented on the paper draft multiple times. I revised the entire paper multiple times and therefore use it in my dissertation in:

  - Section 1, Introduction → Section 3.4.1.1, Task Description (based on)

  - Section 2, Related Work → Section 3.4.1.2, Limitations of existing work (taken from)
  - Section 3.1, Feature Generation → Section 3.4.1.3, From Modus Operandi to Sequential Patterns (based on)
  - Sections 3 & 4, Design Study Methodology & The Concept Explorer → Section 3.4.1.4, Description of Interestingness Measures (based on)
  - Section 3 & 4, Design Study Methodology & The Concept Explorer → Section 5.2.1, VALCRI Concept Explorer (taken from)

- R. Sevastjanova, **W. Jentner**, F. Sperrle, R. Kehlbeck, J. Bernard, M. El-Assady; *QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling*; Association for Computing Machinery, Transactions on Interactive Intelligent Systems (ACM TIIS); 2021 [9]

  I have specifically contributed to the sequence modeling, sequential pattern and sequential rule mining components. I provided a pattern mining code library, that I have written, to Rita Sevastjanova which she used in the QuestionComb-tool. I suggested the design of the rules, specifically the distance preservation using the triangles which are based on the design of Chen et al. [15]. Furthermore, I suggested exploiting closed patterns for mining and maximal patterns for the detail-on-demand functionality. The application and visual system have been implemented mainly by Rita Sevastjanova with the support of Fabian Sperrle, Rebecca Kehlbeck, and me. I have authored Sections 4.1 and 4.2 in the publication and contributed to Section 5.4. I have revised the entire paper upon request and provided comments throughout the entire creation phase. Jürgen Bernard contributed to the Visual Interactive Labeling, and Mennatallah El-Assady contributed to the XAI aspects. Daniel Keim provided comments on the paper draft several times. I am using parts of this publication in my dissertation:

  - Section 1, Introduction → Section 3.4.2.1, Task Description (based on)
  - Section 4.1, Data as Sequences of Words → Section 3.4.2.2, Data Modeling (based on)
  - Section 4.2, Sequential Pattern Mining for an Ex-

plainable Classifier → Section 3.4.2.3, Description of Interestingness Measures (based on)
- Section 5, QuestionComb: The Interface → Section 5.4.3 (based on)

► **W. Jentner**, G. Lindholz, H. Hauptmann, M. El-Assady, K.L. Ma, D. A. Keim; *Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data*; Association for Computing Machinery, Transactions on Interactive Intelligent Systems (ACM TIIS); accepted 2022 [10]

I first presented the idea at my proposal talk in 2018. Giuliana Lindholz (Dehn) helped me to create the first prototype under my supervision. A second, web-based prototype was created by myself. I have received comments and feedback on the idea from my colleagues at the DBVIS group as well as visiting researchers such as George Grinstein, Jürgen Bernard, Remco Chang, and Tobias Schreck. I received further helpful feedback from various anonymous reviewers and Liang Gou (Assistant to the Editor-in-Chief at ACM TIIS). I have authored the entire publication myself and received comments from Hanna Hauptmann (Schäfer) and Mennatallah El-Assady. Kwan-Liu Ma and Daniel Keim commented on the paper draft as well. I am using several parts of this publication in my dissertation, specifically:

- Section 1, Introduction → Section 3.4.3.1, Task Description (taken from)
- Section 3, Related Work → Section 3.4.3.2, Limitations of existing work (taken from)
- Section 4, Multi-dimensional Pattern Exploration Approach → Section 3.4.3.3, Description of Interestingness Measures (taken from)
- Section 5.2, Normalization → Section 3.4.3.4, Additional variations based on the Interestingness Measures (taken from)
- Section 5.3, Interactive Mining & Filtering → Section 5.5.3, Multi-selection-based Mining (taken from)

Book Chapters

► **W. Jentner**, D. A. Keim; *Visualization and Visual Analytic Techniques for Patterns*; Book chapter High-Utility

Pattern Mining: Theory, Algorithms, and Applications; 2019 [16]

I have gathered the relevant information and written the book chapter by myself. Florian Stoffel and Mennatallah El-Assady provided feedback on an early draft. Daniel Keim provided continuous feedback on the paper draft. I received further comments from Philippe Fournier-Viger, editor of the book "High-utility pattern mining" [17] where this chapter appeared in. I am using the entire book chapter in Chapter 4: Visualization Techniques for Structured Data Patterns.

Workshop Articles

- **W. Jentner**, G. Ellis, F. Stoffel, D. Sacha, D. A. Keim; *A Visual Analytics Approach for Crime Signature Generation and Exploration*; The Event Event: Temporal & Sequential Event Analysis, IEEE VIS 2016 Workshop; 2016 [18]

  This paper reports on an early prototype developed for the VALCRI project [12]. I have developed the prototype myself and received feedback from my co-authors. The paper has been authored by me whereas Geoffrey Ellis, Florian Stoffe, and Dominik Sacha edited and provided feedback. Daniel Keim provided comments on the paper draft. I have revised the entire publication several times. I am using content from this publication in my dissertation:

  - Sections 4, Visual Analytics Approach → Section 3.4.1.4, Description of Interestingness Measures (based on)

- J. Buchmüller, **W. Jentner**, D. Streeb, D. A. Keim; *ODIX: A Rapid Hypotheses Testing System for Origin-Destination Data*; IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2017 MC1); 2017 [19]

  This publication was a group effort of Juri Buchmüller, Dirk Streeb and myself in participation at the VAST Challenge 2017, Mini Challenge 1 [20]. I implemented the code of the prototype. Juri Buchmüller contributed to the Neo4J implementation that allowed us to model the raw data as sequences. Dirk Streeb contributed to a second prototype. I specifically contributed Section 3 to the paper that describes the application I developed. I use parts of this publication in my dissertation:

- Section 3, Application → Section 5.1.1, Selection of Transactions (taken from)

- **W. Jentner**, R. Sevastjanova, F. Stoffel, D. A. Keim, J. Bernard, M. El-Assady; *Minions, Sheep, and Fruits: Metaphorical Narratives to Explain Artificial Intelligence and Build Trust*; Workshop on Visualization for AI Explainability, IEEE VIS 2018 Workshop; 2018 [21]

  This paper has been a group effort of several discussions in our small group (Rita Sevastjanova, Florian Stoffel, Mennatallah El-Assady, and Jürgen Bernard as a visiting researcher). Florian Stoffel initiated the discussion with the question "What do you understand as trust?". I took over the responsibility of the primary author and coordinated all writing. I specifically contributed to the didactic reduction parts, as well as, the minion metaphorical narrative, and the trust-building model. Most of the figures were drawn and edited by Rita Sevastjanova. All authors participated in the writing. Daniel Keim provided comments on the draft. I revised the entire paper several times. I am using parts of this publication in my dissertation:

  - Section 3.2 & 3.3, Trust-Building Model & Exemplary Metaphorical Narratives → Section 5.6.4, Understanding Concepts & Metaphorical Narratives (taken from)

1.2.2 Other peer-reviewed publications

Several publications that I co-authored are not included in this dissertation. The following entails an extensive list of these publications sorted by year.

1. M. El-Assady, D. Hafner, M. Hund, A. Jäger, **W. Jentner**, C. Rohrdantz, F. Fischer, S. Simon, T. Schreck, D. A. Keim; *Visual Analytics for the Prediction of Movie Rating and Box Office Performance*; VAST Challenge 2013 - Award for Effective Analytics, 2013 [22]
2. F. Wanner, T. Schreck, **W. Jentner**, L. Sharalieva, D. A. Keim *Relating interesting quantitative time series patterns*

with text events and text features; (Best Paper Award); IS&T/SPIE Electronic Imaging; 2014 [23]

3. M. El-Assady, **W. Jentner**, M. Stein, F. Fischer, T. Schreck, D. A. Keim; *Predictive Visual Analytics - Approaches for Movie Ratings and Discussion of Open Research Challenges*; Proceedings of the IEEE VIS 2014 Workshop Visualization for Predictive Analytics; 2014 [24]
4. F. Wanner, **W. Jentner**, T. Schreck, A. Stoffel, L. Sharalieva, D. A. Keim; *Integrated visual analysis of patterns in time series and text data - Workflow and application to financial data analysis*; Information Visualization; 2015 [25]
5. **W. Jentner**, M. El-Assady, D. Sacha, D. Jäckle, F. Stoffel; *Dynamite: Dynamic Monitoring Interface for Task Ensembles*; IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2016 MC1) (Award - Notable Support for Streaming Analysis); 2016 [26]
6. M. El-Assady, V. Gold, **W. Jentner**, M. Butt, K. Holzinger, D. A. Keim; *VisArgue - A Visual Text Analytics Framework for the Study of Deliberative Communication*; Proceedings of The International Conference on the Advances in Computational Analysis of Political Text (PolText2016); 2016 [27]
7. F. Stoffel, **W. Jentner**, M. Behrisch, J. Fuchs, D. A. Keim; *Interactive Ambiguity Resolution of Named Entities in Fictional Literature*; Eurographics Conference on Visualization (EuroVis 2017); 2017 [28]
8. D. Sacha, **W. Jentner**, L. Zhang, F. Stoffel, G. Ellis; *Visual Comparative Case Analytics* EuroVis Workshop on Visual Analytics (EuroVA); 2017 [13]
9. **W. Jentner**, M. El-Assady, B. Gipp, D. A. Keim; *Feature Alignment for the Analysis of Verbatim Text Transcripts*; EuroVis Workshop on Visual Analytics (EuroVA); 2017 [29]
10. D. Streeb, J. Buchmüller, U. Schlegel, **W. Jentner**, M. Behrisch, B. Schneider, D. Seebacher; *Uncovering the Mistford Toxic Conspiracy*; Conference on Visual Analytics Science and Technology, VAST; 2017 [30]
11. **W. Jentner**, D. Jäckle, U. Engelke, D. A. Keim, T. Schreck; *A Concept for Consensus-based Ordering of Views*; EuroVis Workshop on Visual Analytics (EuroVA); 2018 [31]
12. **W. Jentner**, F. Stoffel, D. Jäckle, A. Gärtner, D. A. Keim; *DeepClouds: Stereoscopic 3D Wordle based on Conical Spirals*; Workshop on Visualization as Added Value in the Development, Use, and Evaluation of Language

Resources (VisLR III) @LREC; 2018 [32]

13. E. Cakmak, G. Castiglia, **W. Jentner**, J. Buchmüller, D. A. Keim; *Visualization For Train Management: Improving Overviews in Safety-critical Control Room Environments*; 4th International Symposium on Big Data Visual and Immersive Analytics; 2018 [33]
14. N. Weiler, M. Kraus, T. Kilian, **W. Jentner**, D. A. Keim; *Visual Analytics for Semi-Automatic 4D Crime Scene Reconstruction*; 4th International Symposium on Big Data Visual and Immersive Analytics; 2018 [34]
15. I. Piljek, G. Dehn, J. Frauendorf, Z. Salem, Y. Niyazbayev, J. Buchmüller, E. Cakmak, **W. Jentner**, F. Stoffel, D. A. Keim; *Identifying Patterns and Anomalies within Spatiotemporal Water Sampling Data*; IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC2 // Award for Elegant Design of an Interactive Display); 2018 [35]
16. B. Bäumle, I. Boesecke, R. Buchmüller, Y. Metz, J. Buchmüller, E. Cakmak, **W. Jentner**, D. A. Keim; *Interactive Webtool for Tempospatial Data and Visual Audio Analysis*; IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC2 // Honorable Mention for Interactive Analytic Tool MC1); 2018 [36]
17. E. Cakmak, U. Schlegel, M. Miller, J. Buchmüller, **W. Jentner**, D. A. Keim; *Interactive Classification Using Spectrograms and Audio Glyphs*; IEEE Conference on Visual Analytics Science and Technology (VAST Challenge 2018 MC1); 2018 [37]
18. U. Schlegel, **W. Jentner**, J. Buchmüller, E. Cakmak, G. Castiglia, R. Canepa, S. Petralli, L. Oneto, D. A. Keim, D. Anguita; *Visual Analytics for Supporting Conflict Resolution in Large Railway Networks*; 2019 INNS Big Data and Deep Learning (INNSBDDL 2019); 2019 [38]
19. M. El-Assady, **W. Jentner**, F. Sperrle, R. Sevastjanova, A. Hautli-Janisz, M. Butt, D. A. Keim *lingvis.io - A Linguistic Visual Analytics Framework*; ACL (3); 2019 [39]
20. M. Dose, N. Wendt, M. Mühling, T. Pollok, **W. Jentner**, S. Schindler, A. L. Tilling, R. King, F. Fest, T. Philipp, M. Kastelitz; *FLORIDA: Analyse von Videomassendaten im Kontext terroristischer Anschläge*; Crisis Prevention; 2019 [40]
21. **W. Jentner**, J. Buchmüller, F. Sperrle, R. Sevastjanova, T. Spinner, U. Schlegel, D. Streeb, H. Schäfer; *N.E.A.T. - Novel Emergency Analysis Tool*; IEEE Conference on

Visual Analytics Science and Technology (VAST Challenge 2019 Grand Challenge); 2019 [41]

22. M. El-Assady, **W. Jentner**, R. Kehlbeck, U. Schlegel, R. Sevastjanova, F. Sperrle, T. Spinner, D. A. Keim; *Towards XAI: structuring the processes of explanations*; ACM Workshop on Human-Centered Machine Learning; 2019 [42]
23. T. Pollok, M. Kraus, C. Qu, M. Miller, T. Moritz, T. Kilian, D. A. Keim, **W. Jentner**; *Computer Vision Meets Visual Analytics: Enabling 4D Crime Scene Investigation from Image and Video Data* ICDP 2019; London; 2019 [43]
24. M. Kraus, T. Pollok, M. Miller, T. Kilian, T. Moritz, D. Schweitzer, J. Beyerer, D. A. Keim, C. Qu, **W. Jentner**; *Toward Mass Video Data Analysis: Interactive and Immersive 4D Scene Reconstruction*; Sensors; Special Issue Selected Papers from the 9th International Conference on Imaging for Crime Detection and Prevention (ICDP-19); 2020 [44]
25. L. Martí-Bonmatí, Á. Alberich-Bayarri, R. Ladenstein, I. Blanquer, J. D. Segrelles, L. Cerdá-Alberich, P. Gkontra, B. Hero, JM García-Aznar, D. Keim, **W. Jentner**, K. Seymour, A. Jiménez-Pastor, I. González-Valverde, B. Martínez de las Heras, S. Essiaf, D. Walker, M. Rochette, M. Bubak, J. Mestres, M. Viceconti, G. Martí-Besa, A. Cañete, P. Richmond, K. Y Wertheim, T. Gubala, M. Kasztelnik, J. Meizner, P. Nowakowski, S. Gilpérez, A. Suárez, M. Aznar, G. Restante, E. Neri; *PRIMAGE project: predictive in silico multiscale analytics to support childhood cancer personalized evaluation empowered by imaging biomarkers*; European radiology experimental 4-1 1-11; 2020 [45]
26. M. T. Fischer, S. D. Hirsbrunner, **W. Jentner**, M. Miller, D. A. Keim, P. Helm; *Promoting Ethical Awareness in Communication Analysis: Investigating Potentials and Limits of Visual Analytics for Intelligence Applications*; Proceedings of FAcct '22 : 2022 ACM Conference on Fairness, Accountability, and Transparency; 2022 [46]
27. **W. Jentner**, F. Sperrle, D. Seebacher, M. Kraus, R. Sevastjanova, M. T. Fischer, U. Schlegel, D. Streeb, M. Miller, T. Spinner, E. Cakmak, M. Sharinghousen, P. Meschenmoser, J. Görtler, O. Deussen, F. Stoffel, H.-J. Kabitz, D. A. Keim, M. El-Assady, J. F. Buchmüller; *Visualisierung der COVID-19-Inzidenzen und Behandlungskapazitäten mit CoronaVis*; Resilienz und Pandemie: Handlungsempfehlungen anhand erster Erfahrungen mit

Covid-19; 2022 [47]

1.2.3 Technical Reports

The main effort during my Ph.D. was to participate in and manage projects. One of the outcomes of these projects is technical reports, also called deliverables, that describe intermediate and final project results. Their main purpose is to ensure the transparency and quality of the work performed in the project. While these are not considered to be peer-reviewed, there are actually several checks performed. Multiple partners write the reports, then typically checked by two partners within the project that have not co-authored. The steering committee of the project performs the next check. Finally, the stakeholder (funding agency) acquires anonymous reviewers for a project that also provides a profound review of the deliverable. I want to include these reports as they reflect large parts of my work and research during my time as a Ph.D. student. A short list of all the projects that I was part of follows:

1. VisArgue - Analysis, and Visualization of Political Communication; BMBF; 2014 - 2016
2. VALCRI - Visual Analytics for Sense-Making in Criminal Intelligence Analysis; EU-FP7; 2016 - 2018
3. FLORIDA - Flexible, semi-automatic analysis system for the evaluation of mass video data; BMBF; 2016 - 2019
4. IN2DREAMS - INtelligent solutions 2ward the Development of Railway Energy and Asset Management Systems in Europe; EU-Horizon2020; 2017-2020
5. ASGARD - Analysis System for Gathered Raw Data; EU-Horizon2020; 2016 - 2020
6. VICTORIA - Video Analysis for Investigation of Criminal and Terrorist Activities; EU-Horizon2020; 2017 - 2021
7. PRIMAGE - PRedictive In-silico Multiscale Analytics to support cancer personalized diaGnosis and prognosis, Empowered by imaging biomarkers; EU-Horizon2020; 2018 - present
8. PEGASUS - Police extraction and analysis of heterogeneous mass data for combating organized crime structures; BMBF; 2020 - present

9. DAYDREAMS - Development of prescriptive AnalYtics baseD on aRtificial intElligence for iAMS; EU-Horizon2020; 2020 - present
10. VIKING - Trusted Artificial Intelligence for Police Applications; BMBF; 2022 - present
11. KTBW-RPM - Remote Patient Monitoring; Baden- Württemberg - State Funding; 2022 - present

Not all technical reports have the authors explicitly stated. Some of them only mention the partners' institutions, others only the first author. The deliverables where I am the main author (i.e., first author) have their titles underlined. This list does not contain deliverables that I have reviewed in the internal quality assurance process.

1. D. Sacha, **W. Jentner**, L. Zhang, F. Stoffel, G. Ellis, D. A. Keim; *Applying Visual Interactive Dimensionality Reduction to Criminal Intelligence Analysis*; VALCRI; 2017 [14]
2. R. Canepa, S. Petralli, L. Oneto, **W. Jentner**, J. Buchmüller, C. Ducuing, I. Emanuilov, M. Swiatek, D. Anguita; *D5.1: Data Analytics Scenarios*; IN2DREAMS; 2018
3. **W. Jentner**, L. Oneto, R. Spigolon; *D5.3: Visual Analytics of Railway Data and Models*; IN2DREAMS; 2018
4. C. Ducuing, I. Emanuilov, M. C. Janssens, R. Spigolon, L. Oneto, R. Canepa, S. Petralli, **W. Jentner**; *D4.5: Legal analysis of the placing on a blockchain of a data marketplace in the railways*; IN2DREAMS; 2018
5. L. Oneto, M. Swiatek, C. Ducuing, I. Emanuilov, J. Buchmüller, **W. Jentner**, D. Anguita; *D5.2: Assessment metrics and rule-based data analytics tools Proof-of-Concept*; IN2DREAMS; 2019
6. **W. Jentner**, M. Kraus, N. Weiler; *D6.5: Visual Analytics system for semi-automatic 4D crime scene reconstruction*; VICTORIA; 2018
7. M. Dose, R. Zhou, T. Pollock, G. Roman Jimenez, **W. Jentner**, M. Kraus; *D7.2: Graphical User Interface*; VICTORIA; 2018
8. M. Dose, N. Wendt, R. Zhou, **W. Jentner**, M. Kraus; *D7.6: VICTORIA Video Analysis Platform*; VICTORIA; 2019
9. M. T. Fischer, D. Seebacher, M. Worring, D. Streeb, **W. Jentner**; *D7.5: Visual Analytics Framework and Techniques*; ASGARD; 2019
10. **W. Jentner**, E. Cakmak, J. Buchmüller, G. Castiglia, L. Oneto; *D5.4: Rule-based and Visual analytics knowledge*

extraction demonstrator; IN2DREAMS; 2020

11. **W. Jentner**, M. Kraus, N. Weiler, T. U. Kilian, M. Miller, F. Stoffel, D. A. Keim; *Teilvorhaben: Visual Analytics zur Semi-automatischen Tatortrekonstruktion (Schlussbericht)*; FLORIDA; 2020
12. M. Dose, N. Wendt, R. Zhou, **W. Jentner**, M. Kraus; *D7.6: VICTORIA Video Analysis Platform, prototype V2*; VICTORIA; 2021
13. L. Oneto et al.; *D5.1: IAMS Prototype Integration Guidelines, User Requirements and Scenarios*; DAYDREAMS; 2021
14. **W. Jentner**; *D4.1: Analysis of the context and state-of-the-art*; DAYDREAMS; 2022
15. L. Oneto et al.; *D2.1: Learning from Data and Human Behaviour*; DAYDREAMS; 2022
16. **W. Jentner**; *D4.2: Context-driven Dynamic HMI Design and Prototype*; DAYDREAMS; 2022
17. **W. Jentner**, R. Canepa; *D4.3: Context-driven Dynamic HMI Assessment*; DAYDREAMS; 2022
18. M. Anastasopoulos; **W. Jentner**; G. Chevaleyre *D3.2: Multi-Objective Decision Optimisation Tools Assessment*; DAYDREAMS; 2022

Pattern Mining 2

This chapter introduces definitions and lays the background for this thesis.

2.1 Definitions

Pattern mining is a special form of clustering for structured data that does not require any distance measure and instead relies on the definition of *containment*. Structured data, in its simplest form, is defined as a set of items or an *itemset*.

Definition 2.1.1 (Itemset)

$$I = \{i_1, i_2, ..., i_n\}$$

All the items, also called symbols, of a dataset, are called *alphabet*, denoted by the symbol Σ. Therefore, all itemsets I of a dataset must be a subset of the alphabet ($I \subseteq \Sigma$). Without loss of generality, a total order can be defined over the items in the set. Such a total order can be, for example, a lexicographic order. Throughout this thesis, a total order is assumed to improve readability.

For itemsets, containment is straightforward as it is equal to a subset. In this dissertation, we define containment as:

Definition 2.1.2 (Containment)

$$A \sqsubseteq B$$

whereas a pattern A is contained in another pattern or structured data B. If A and B are itemsets then the containment is equal to

Definition 2.1.3 (Containment for itemsets)

$$A \sqsubseteq B \equiv A \subseteq B$$

A sequence is a more complex data structure that is based on itemsets and is defined as an ordered list of itemsets denoted as

Definition 2.1.4 (Sequence)

$$s = \langle I_1, I_2, ..., I_n \rangle$$

Each of the itemsets of a sequence must be a subset of the alphabet

$$\forall k | 1 \leq k \leq n, I_k \subseteq \Sigma$$

In the context of event sequences, the items of the itemsets are also often referred to as *events*. Let there be an alphabet $\Sigma = \{a, b, c\}$ and a sequence $s = \langle \{a, b\}, \{c\} \rangle$, then the sequence s is interpreted as events a and b occur at the same time and event c occur afterward. Frequently, each event is annotated with a timestamp or a number to encode the distance between the events further.

Containment of sequence $s_a = \langle A_1, A_2, ..., A_n \rangle$ in another sequence $s_b = \langle B_1, B_2, ..., B_m \rangle$ is defined as:

Definition 2.1.5 (Containment of sequences)

$$s_a \sqsubseteq s_b$$

$$\iff$$

$$\exists i | 1 \leq i_1 < ... < i_n \leq m, A_1 \subseteq B_{i1}, A_2 \subseteq B_{i2}, ..., A_n \subseteq B_{in}$$

For example, a sequence $s_a = \langle \{a\}, \{b\} \rangle$ is contained in $s_b = \langle \{a, c\}, \{d\}, \{b, c\} \rangle$ but not in $s_c = \langle \{b, c\}, \{a, d\} \rangle$ because itemsets $\{a\}$ and $\{b\}$ do not occurr in the correct order.

The *length* of an itemset or sequence is referred to as the cardinality of the itemset or the summed cardinality of the itemsets in the sequence. In several works *length* is also defined as the number of itemsets of the sequence. Because of this ambiguity, this dissertation uses the term *generation* for the former definition of length.

Definition 2.1.6 (Generation (itemset))

$$g = |I|$$

For a sequence $s = \langle I_1, I_2, ..., I_n \rangle$, the generation is defined as:

Definition 2.1.7 (Generation (sequence))

$$g = \sum_{i=1}^{n} |I_i|$$

Therefore, a g-itemset is an itemset of generation g and a g-sequence a sequence of generation g.

The structured data to be mined is provided as a database or dataset D. Each row consists of an ID and the structured data in an encoded form. For itemsets, rows are also sometimes called transactions. However, this is more task-specific. $|D|$ defines the size of the database or the number of rows.

Association rule mining is an extension to itemset mining. It allows finding the correlation of itemsets within transactions. A rule consists of two itemsets A and B and is denoted as:

Definition 2.1.8 (Association Rule)

$$A \rightarrow B | A \cap B = \emptyset \wedge A \cup B \neq \emptyset$$

The left-hand side of the rule is called *antecedent* side, and the right-hand side of the rule is called the *consequent* side. Although this is called a rule, it should be understood as a correlation that can be observed with a certain probability in the data.

The generation of an association rule is defined as:

Definition 2.1.9 (Generation (association rule))

$$g = |A| \quad |B|$$

Like association rules and itemsets, sequential rules are an extension of sequential patterns.

Definition 2.1.10 (Sequential Rule)

$$\langle A_{i_1}, A_{i_2}, ..., A_{i_n} \rangle \rightarrow \langle B_{j_1}, B_{j_2}, ..., B_{j_m} \rangle | i_n < j_1$$

Because sequential patterns maintain the order of events, a sequential rule is defined as that all events of the antecedent side must be before the events of the consequent side. Sequential rules represent temporal correlations that if one or more events happen, they are followed by certain events by a certain probability (called confidence).

Definition 2.1.11 (Generation (sequential rule))

$$g = \sum_{i=1}^{n} |A_i| \quad \sum_{j=1}^{m} |B_j|$$

2.2 Search Spaces

Understanding search spaces is fundamental to this work. The exponentiality of a search space can drive up the numbers so quickly that it becomes unimaginable to the human brain. The underlying reason for this is the containment definition of patterns as it allows for transitive and ubiquitous containment of patterns.

First, the search spaces do not depend on the amount of data (number of transactions) in the database. Their size is only determined by the alphabet, which is all items in the database. For sequences the maximal length of the longest sequence in the database is important.

For itemsets, calculating the search space is straightforward. The search space of an alphabet $\Sigma = \{a, b, c\}$ ($n = |\Sigma| = 3$) consists of the following patterns:

- $\emptyset$
- $\{a\}$
- $\{b\}$
- $\{c\}$
- $\{a, b\}$
- $\{a, c\}$
- $\{b, c\}$
- $\{a, b, c\}$

This is the set of all subsets of $\{a, b, c\}$ which is also known as the *power set*. Another representation of this power set is shown in Figure 2.1 in the form of a Hasse-diagram [48]. The itemsets are ordered in a partial order (i.e., no total order exists) due to the containment definition, therefore this search space is also known as a *lattice*. The itemsets are visually ordered vertically in layers according to their generation

[48]: authors (2021), Hasse diagram

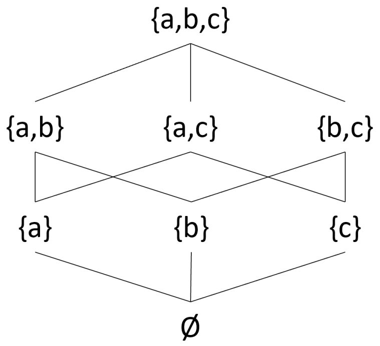


Figure 2.1: The search space (power set) of $\{a, b, c\}$ represented as a Hasse-diagram [48]. The lines indicate the containment.

whereas the lowest generation is on the bottom and the highest generation (3) is on top. Note that this diamond-like shape is characteristic of search spaces in pattern mining and that, specifically, the generations in the vertical center account for the highest number of patterns within the search space (i.e., highest entropy).

An itemset can be represented as a binary vector whereas, for example, the itemset $\{a, c\}$ can be represented as 101. Therefore, the size of the search space can be simply derived as 2^n. Typically, the empty set is considered to be trivial and uninteresting which is why it is often discarded reducing the search space to:

$$2^n - 1 \tag{2.1}$$

For sequences, another limit is necessary as the alphabet itself does not account for the maximal length of a sequence. The length of a sequence is the number of itemsets within that sequence. Let it be m. For an alphabet $\Sigma = \{a, b\}$ and a maximal length of $m = 2$, the following sub-sequences can be generated:

- $\langle\{a\}\rangle$
- $\langle\{b\}\rangle$
- $\langle\{a, b\}\rangle$
- $\langle\{a\}, \{a\}\rangle$
- $\langle\{a\}, \{b\}\rangle$
- $\langle\{b\}, \{a\}\rangle$
- $\langle\{b\}, \{b\}\rangle$
- $\langle\{a, b\}, \{a\}\rangle$
- $\langle\{a, b\}, \{b\}\rangle$
- $\langle\{a\}, \{a, b\}\rangle$
- $\langle\{b\}, \{a, b\}\rangle$
- $\langle\{a, b\}, \{a, b\}\rangle$

Note that the maximum generation is always $n * m$. The first three list items refer to sequences of length one since they only contain one itemset. The number of possible patterns is, therefore, identical to $2^n - 1$. For sequences of length two, this can be extended to $2^n - 1^2$. Therefore, the search space can be defined as:

$$\sum_{i=1}^{m} 2^n - 1^i = \frac{2^n - 12^n - 1^m - 1}{2^n - 2} \tag{2.2}$$

[1] While this formula may not be impressive, it shows how

[1] This part is based on my publication "Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data" (Section 6.4: User Study: What We Eat In America Dataset) [10]. I have been the main author of this publication and section and have written all the contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.

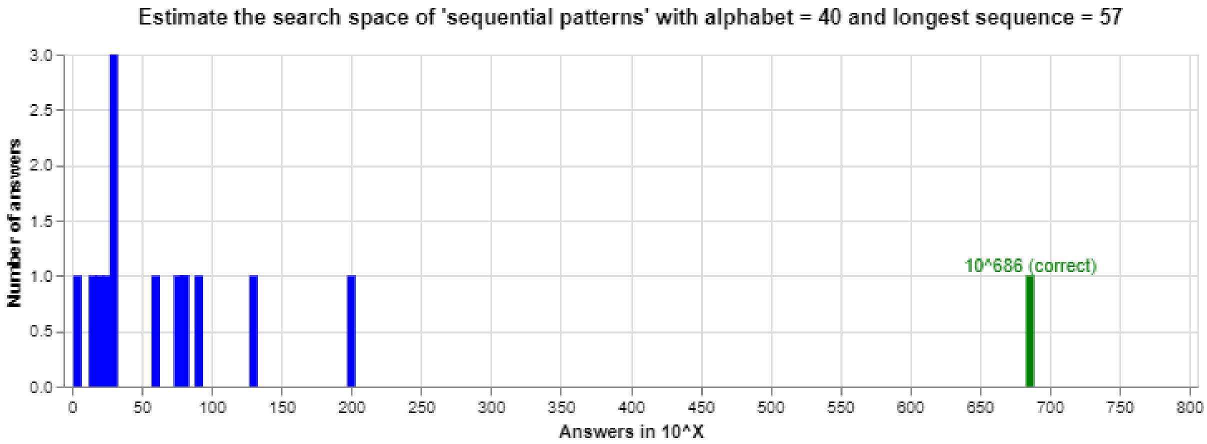


Figure 2.2: 14 persons were asked to estimate the search space of sequential pattern mining for an alphabet size of 40 and the longest sequence of 57. The estimates are magnitudes below the actual result. Note the logarithmic x-axis.

quickly search spaces in pattern mining can grow. Humans tend to underestimate exponentiality as they are not very intuitive. For an evaluation of one of my publications [10], I have conducted interviews with 15 master-, Ph.D.-students, and PostDocs (see subsection 5.7.1). They were asked to estimate the search space of sequential patterns for an alphabet of $n = 40$ and the longest sequence $m = 57$ without knowing the above formula. The size of the alphabet and the longest sequence are from a real-world dataset from the VAST Challenge 2017 Mini Challenge 1 [20]. Their answers are depicted in Figure 2.2 and show all but one participant greatly underestimates the size of the search spaces. Note the logarithmic x-axis in the plot. The one participant who estimated correctly derived a similar to the above formula to then estimate the result. Large numbers such as 10^686 cannot be intuitively understood anymore since any point of reference to our real world is missing. As mentioned earlier, the estimated amount of atoms in our visible universe is estimated to be around 10^{82} to 10^84. [2] The size of the search space is magnitudes greater than this number.

Such search spaces can neither be computed, stored nor ever fully analyzed or visualized. Thanks to the curse of dimensionality (see section 2.4) and approaches such as the apriori-algorithm [49] this is not necessary to generate all possible patterns.

[20]: Whiting et al. (2017), VAST Challenge 2017 Mini Challenge 1

[2] `livescience.com/how-many-atoms-in-universe.html`, accessed March 25, 2022

[49]: Agrawal et al. (1994), Fast Algorithms for Mining Association Rules in Large Databases

2.3 Equivalence Classes

Equivalence classes are sets of patterns that represent identical clusters. Table 2.1 shows a simple database from a market basket analysis. It displays four customers and their market baskets, i.e., which items they have bought at the grocery store. Pattern mining generates patterns that are contained in the transactions. Since the data is structured as itemsets the patterns themselves are itemsets or sub-itemsets. All possible patterns that can be generated from this dataset are shown in Figure 2.3. In total, 21 distinct patterns can be generated, however, the input data only consists of four transactions. The patterns are visualized in the form of a Hasse-diagram [48] where the patterns of the lowest generation (see Definition 2.1.7) are placed on the bottom and the highest-generation-patterns are placed on top. The number in the purple circles indicates the *support* of a pattern, an interestingness measure describing how many transactions the pattern is contained in. Below each pattern, the transaction IDs are displayed (see Table 2.1). The patterns are outlined in various colors (labeled A-D) that represent the four equivalence classes. All patterns within one equivalence class describe the same set of transactions in the input data. Even though none of these patterns are equal, they describe the same data which shows that they describe redundant information.

[48]: authors (2021), Hasse diagram

To eliminate these redundancies closed and maximal patterns have been introduced.

[50]: Pasquier et al. (1999), Discovering Frequent Closed Itemsets for Association Rules

Closed Patterns This was first defined by Pasquier et al. [50]. Let P be the set of patterns in the lattice. A closed pattern p_c is closed if and only if there exists no *super-patterns* that contain p_c that also describe the same data. Since this can be expressed with the interestingness measure *support* the formal definition is:

Table 2.1: A market basket database showing four customers and their market baskets with the items they have bought.

ID	Transactions
1	$\{\text{bread}, \text{juice}, \text{milk}, \text{vegetables}\}$
2	$\{\text{bread}, \text{candy}, \text{soda}\}$
3	$\{\text{bread}, \text{juice}, \text{vegetables}\}$
4	$\{\text{bread}, \text{candy}, \text{soda}\}$

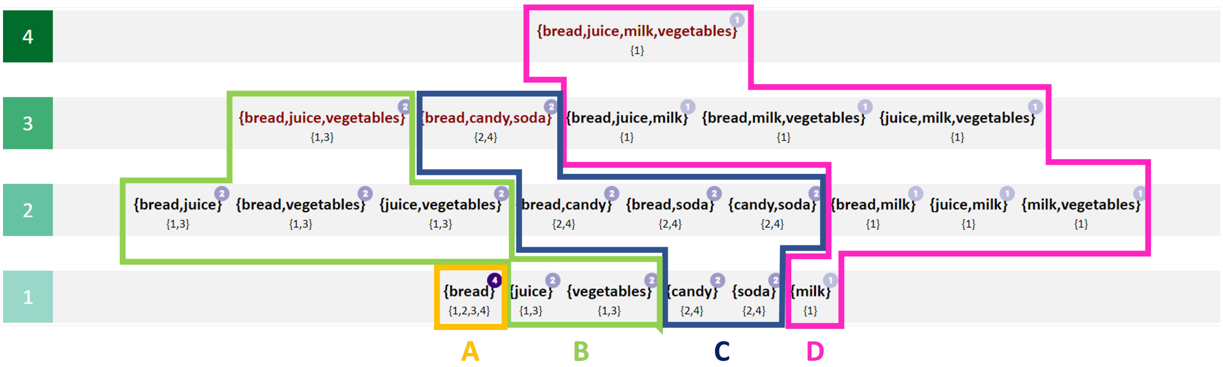


Figure 2.3: Four equivalence classes (A-D) visually represented in a Hasse-diagram.

Definition 2.3.1 (Closed Pattern)

$$p_c | p_c, p \in P \wedge p : p_c \sqsubseteq p \wedge support p_c = support p$$

In our example $\{bread\}$ would be a closed pattern since there are no super-patterns that contain $\{bread\}$ that also have a support of 4. The other closed patterns are:

- $\{bread, juice, vegetables\}$
- $\{bread, candy, soda\}$
- $\{bread, juice, milk, vegetables\}$

In other words, closed patterns are the patterns of the highest generation within one equivalence class. Unlike in this example, there can be multiple closed patterns within one equivalence class. A naive approach would first mine all patterns and then filter for the closed patterns. However, there exist several algorithms to mine for closed patterns efficiently [51]. Mining for closed itemsets is said to be lossless regarding the information since only redundant descriptions are being removed.

[51]: (2014), Frequent Pattern Mining

Generator Patterns Generator patterns, or also key patterns, are the opposite of closed patterns in the sense that here only the patterns of the lowest generation of one equivalence class are being retained. A generator pattern p_g is a generator if and only if there exists no *sub-patterns* that are being contained in p_g that describe the same data.

Definition 2.3.2 (Generator Pattern)

$$p_g | p_g, p \in P \wedge p : p \sqsubseteq p_g \wedge supportp_g = supportp$$

Note that in the previous definition, it was stated as $p_c \sqsubseteq p$ and here it is $p \subseteq p_g$. In the example, the following patterns would be generator patterns:

- $\{bread\}$
- $\{juice\}$
- $\{vegetables\}$
- $\{candy\}$
- $\{soda\}$
- $\{milk\}$

Looking at the lattice (Figure 2.3), these are the patterns on the bottom of each equivalence class (i.e., the patterns of the lowest generation). Note that in some literature the empty set is considered a generator pattern that describes all transactions in the database. In our example, this would replace $\{bread\}$ with the empty set since the empty set is contained in all patterns but its support is also 4. As with closed patterns, generator patterns are said to be lossless since only redundant information is being removed. Another interesting observation exists for rule mining: taking a generator pattern for the antecedent (left-hand side) and the closed pattern minus the generator pattern on the consequent side (right-hand side), the confidence will always be 100% since the rule exists only in the same equivalence class. Also, generator patterns can be mined efficiently [52].

[52]: Fournier-Viger et al. (2017), A survey of itemset mining

Maximal Patterns Maximal patterns are a simplification of closed patterns in the sense that the support constraint is removed. Therefore, a maximal pattern p_m is maximal if and only if there exists no super-patterns p that contain p_m.

Definition 2.3.3 (Maximal Pattern)

$$p_m | p_m, p \in P \wedge p : p_m \sqsubseteq p$$

In the example, the maximal patterns would be:

- $\{bread, candy, soda\}$
- $\{bread, juice, milk, vegetables\}$

Because of the definition, the set of maximal patterns P_M is always a subset of closed patterns P_C which are a subset of the lattice (all patterns) P: $P_M \subseteq P_C \subseteq P$. Concerning the amount of information, maximal patterns are considered lossy since not all information of the clusters (with the patterns as their representants) is retained.

While the definition of a minimal pattern is equivalent to generator patterns without the support constraint, they are not to be found in the literature as they are not considered useful. If the empty set is considered, the empty set itself is always minimal as it is contained in any pattern. If it is removed, then all first-generation patterns are minimal.

Note that the definitions of closed-, generator-, and maximal patterns are based on the lattice (i.e., the mined patterns) which is typically also constrained by other interestingness measures such as the *support* and *confidence* for rules. Any constraints on the lattice modify the lattice itself and thus influence the sets of closed-, generator-, and maximal patterns.

Equivalence classes and the definitions of closed-, generator-, and maximal patterns are considered useful to reduce the exponential amounts of patterns drastically.

2.4 Curse of Dimensionality

The curse of dimensionality is a term for a set of well-studied phenomena [53–56] in high dimensional data analysis impacting all areas of machine learning and artificial intelligence. The reason for this is that the volume grows exponentially about the number of dimensions. A general notion in machine learning is, therefore, that with an increasing number of dimensions (i.e., features) the amount of data needed to find a general approximation (i.e., model) grows exponentially.

[53]: Indyk et al. (1998), Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality
[54]: Köppen (2000), The curse of dimensionality
[55]: Kuo et al. (2005), Lifting the curse of dimensionality
[56]: Verleysen et al. (2005), The Curse of Dimensionality in Data Mining and Time Series Prediction

One of the major effects is that data points in high dimensional space become equidistant. However, Houle et al. show that ranked-based similarity measures are more robust than distance measures [57]. Another major impact is that almost all datasets in high-dimensional data can be considered sparse.

[57]: Houle et al. (2010), Can Shared-Neighbor Distances Defeat the Curse of Dimensionality?

Pattern mining is not only a clustering technique but moreover a *subspace clustering* technique. The idea behind subspace clustering is that only certain dimensions (instead of all dimensions) can be useful to form clusters. In fact, frequent pattern mining influenced the research field of subspace clustering which later has been generalized and grown independently. However, many of the core concepts of subspace clustering are born in the field of frequent pattern mining [58]. In frequent itemset mining, every item can be considered as a discrete dimension that every transaction either possesses or not. Testing out every possible combination of n dimensions would result in $2^n - 1$ tests (leaving out the empty set which would be equal to no dimension being considered). The a-priori algorithm states that all supersets of an infrequent itemset must also be infrequent and thus this combination of dimensions can be discarded early. Although subspace clustering focuses on numerical dimensions, early algorithms such as CLIQUE [59] and MAFIA [60] use grids to discretize the numerical data whereas the discrete cells (and the fact that a datapoint is within or not) then form the items and the apriori algorithm can be reused.

[58]: Zimek et al. (2014), Frequent Pattern Mining Algorithms for Data Clustering

[59]: Agrawal et al. (1998), Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications
[60]: Nagesh et al. (2001), Adaptive Grids for Clustering Massive Data Sets

The fact the points in high dimensional data become equidistant can be discarded in the field of pattern mining as it does not rely on a distance function but rather on the definition of containment. This poses, however, its own challenge in the sense that all search spaces are exponential (see section 2.2).

On the other hand, also structured data must be considered sparse the larger the alphabets (i.e., number of items or dimensions) grow. At first sight, this might be due to the rather low amounts of transactions in contrast to the vast exponential search spaces which are due to the *combinatorical explosion* or also often called *pattern explosion*. But this is not the main reason. In general, only one transaction is enough to create all possible patterns in the search space. If the alphabet is $\Sigma = \{a, b, c\}$, then *one* transaction with $\{a, b, c\}$ would be enough to create all possible subsets in the search space. For itemsets, the set of all possible patterns of the search space is also called a power set. Luckily in real-world datasets, this is rather uncommon or unlikely. Thinking about the market basket analysis: would a real-world dataset have one or more customers that bought every item in the store in a single transaction?

Similar thinking and real-world constraints also apply to event sequences. An event sequence able to generate the power set must contain every item in every itemset. Since the data is finite, also the event sequence must have a finite length. Using the same alphabet as before and a maximal length of a sequence of 3, the sequence to create all of the possible subsequences would be: $\langle \{a, b, c\}, \{a, b, c\}, \{a, b, c\} \rangle$ If a patient history is modeled as an event sequence and the death of the patient is one event, then there should be no events such as surgeries or doctor visits happening after the event of death. Such (real-world) constraints eliminate many of the possible combinations. Frequent pattern mining builds on top of that by only retaining patterns that occur in at least x amounts of transactions (called *minimum support*).

Equivalence classes are an effect of the curse of high dimensionality since the combinatorial explosion allows for more possible patterns than actual possible clusters in the data. Hence, several patterns describe the same clusters and are part of one equivalence class.

My work described in “Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data” exploits the curse of dimensionality by using a heuristic for a dramatic search space reduction [10].

[10]: Jentner et al. (2022), Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data

Interestingness Measures | 3

3.1 Introduction

Interestingness measures, as the name suggests, are metrics to quantify the interestingness of results obtained through data mining. In structured data mining or pattern mining, this refers to the interestingness of a pattern. Pattern mining is essentially a subspace search technique because every pattern is a cluster representation that matches a subset of the input data. Because of this an exponential amount of patterns can be generated which quickly overwhelm any human being. The type of pattern, e.g., itemset, sequence pattern, association rule, etc., are suitable for different types of structured data and interestingness measures but all of them refer to a subset of the provided data in the database. Interestingness measures, therefore, are used to filter and rank patterns according to the defined measures.

This chapter is not intended to provide a thorough survey of available interestingness measures in data mining or pattern mining. Furthermore, it does not provide an overview of mining algorithms dedicated to various interestingness measures. There exist various surveys and taxonomies [61–66] in the literature that cover that topic. Similarly, there are several surveys and taxonomies available for algorithms such as frequent itemsets [52, 67, 68], association rule mining [69–72], sequential pattern mining [73–76]. This chapter is dedicated to a position on how interestingness measures should be perceived and handled for exploratory analysis using the methodology of visual analytics.

3.1.1 Aspects of interestingness

There exists no formal definition of an interestingness measure since a formal definition of interestingness is quite difficult. Interestingness is always subjective and can quickly change once new information is available. Geng and Hamilton collect the following aspects of interestingness for patterns [64]:

[64]: Geng et al. (2006), Interestingness measures for data mining: A survey

Conciseness There are two types of conciseness. First, the conciseness of a pattern, meaning that fewer items (i.e., lower generation; see Definition 2.1.6), is more optimal since it is easier to understand. This is already debatable since I had several occasions where users were interested in more complex patterns that provided more detail to the underlying data. This underlines that there may be other, more dominant, aspects of interestingness that typically take priority. The second type of conciseness refers to the set of patterns also called the result set. This is generally true since users do not wish to browse through thousands of patterns but on the other hand, a constraint that filters too many patterns may also be undesirable since a potentially interesting pattern could be missed by the user. Because the filtering operation typically relies on thresholds in combination with interestingness measures, the parameter estimation is quite difficult for a user and thus the chance of "missing out" is rather high. However, if the result set is too large and the ranking of patterns is inefficient, the likelihood rises that a user might overlook a potentially interesting pattern.

Generality/Coverage/Frequency This aspect of interestingness refers to how much of the data a pattern describes. Agrawal and Srikant already proposed this in a combination of an efficient algorithm to mine for these patterns [49]. They named the interestingness measure *support* which denotes either the absolute number of data rows a pattern refers to or a relative number (i.e., percentage) where the absolute support is divided by the total number of data rows in the database. With their paper, Agrawal and Srikant started the field of *frequent pattern mining* which inspired others to find efficient algorithms for various types of data structures such as the SPAM [77] and SPADE [78] algorithms for sequential pattern mining. The idea is that a more general pattern that describes a large cluster of the subspace is more interesting to the user as this quickly eliminates any noise.

[49]: Agrawal et al. (1994), Fast Algorithms for Mining Association Rules in Large Databases

[77]: Ayres et al. (2002), Sequential PAttern mining using a bitmap representation

[78]: Zaki et al. (2002), CHARM: An Efficient Algorithm for Closed Itemset Mining

Reliability/Accuracy Accuracy is well-known in the field of classification and information retrieval. For *classification* *rules* it refers to how accurately a pattern predicts a certain outcome. For association rules or sequential rules, the interestingness measure is typically named *confidence* whereas high confidence shows a high prediction accuracy and thus reliability. There exists a large toolkit of measures derived from statistics, probability, and information retrieval that measure the aspect of reliability [64, 79].

Peculiarity This topic was introduced by Zhong et al. [80] and picked up by Hilderman and Hamilton in their book "Knowledge Discovery and Measures of Interest" [81]. A peculiar pattern is dissimilar from the set of other *discovered* patterns. Note that the distance measure to define a similarity is not inherently given and can be determined based on the use case. The intention is that a dissimilar pattern is likely to be more interesting to the user as it prevails more unknown components than a more similar pattern would.

Diversity Similar to conciseness, there are two types of diversity to distinguish: first, the diversity of a pattern refers to the diversity of its elements. The intention behind this is that a more diverse pattern may be of higher interest which is in direct contrast to conciseness. The second type of diversity describes the result set of patterns. A diverse set of patterns shows a greater variance as opposed to any uniform distribution which is likely more interesting to a user assuming that no prior knowledge is available before the mining process. This aspect is theoretical and cannot be directly measured and, thus, quantified.

Novelty A novel pattern is previously unknown to the user and can also not be derived from similar patterns. In general, the entire knowledge of a human being cannot be fully formalized and thus the absence of such knowledge qualifying a novel pattern cannot be quantified either. Novelty can only be determined empirically when the user labels certain patterns. This has, for example, been proposed by Sahar [82] or Klemettinen et al. [83].

Surprisingness/Unexpectedness This aspect may sound overlapping to novelty but it slightly differs as surprisingness does not require a pattern to be novel. A pattern may simply

[64]: Geng et al. (2006), Interestingness measures for data mining: A survey

[79]: Ohsaki et al. (2004), Evaluation of Rule Interestingness Measures with a Clinical Dataset on Hepatitis

[80]: Zhong et al. (1999), Peculiarity Oriented Multi-database Mining

[81]: Hilderman et al. (2001), Knowledge Discovery and Measures of Interest

[82]: Sahar (1999), Interestingness via What is Not Interesting

[83]: Klemettinen et al. (1994), Finding Interesting Rules from Large Sets of Discovered Association Rules

be unexpected or contradict the user's knowledge to be surprising. However, surprisingness suffers from the same challenges as a novelty in the sense that existing knowledge or expectations cannot be completely formalized to be readily available for the mining process. Even if it is possible, is impracticable and provides a high burden to the user. Several papers have been devoted to algorithms and systems that are capable of formalizing the user's expectations to be matched against in the mining process [84–87]. However, the use cases are highly task-dependent and cannot be easily generalized to other domains.

[84]: Silberschatz et al. (1995), On Subjective Measures of Interestingness in Knowledge Discovery
[85]: Silberschatz et al. (1996), What Makes Patterns Interesting in Knowledge Discovery Systems
[86]: Liu et al. (1997), Using General Impressions to Analyze Discovered Classification Rules
[87]: Liu et al. (1999), Finding Interesting Patterns Using User Expectations

Utility The utility aspect refers to specific use cases where it is possible to define a utility function that correlates with interestingness. A popular example is an extension of the market basket analysis where, in addition to the items of the baskets, their profit for the store manager is known. This can then be used to mine for the most profitable combinations of items instead of just the most frequent ones. Utility-based mining sparked its own subfield which has recently been surveyed in the book "High-Utility Pattern Mining" [17].

[17]: Fournier-Viger et al. (2019), High-Utility Pattern Mining

Actionability/Applicability Piatetsky-Shapiro and Matheus were the first to introduce this aspect in an interestingness measure in their KEFIR system without naming it explicitly [88]. Silberschatz and Tuzhilin then referred to this work naming it *actionability* which in combination with *unexpectedness* forms the most important aspects of any subjective interestingness measure (see next section) in their opinion [84]. Actionability refers to the fact that a pattern is interesting if the user can use this information to their advantage (i.e., increase the company's profit). There exists no general measure but several task and domain-specific approaches have been proposed such as by Ling et al. [89] or Wang et al. [90].

[88]: Piatetsky-Shapiro et al. (1994), The interestingness of deviations

[84]: Silberschatz et al. (1995), On Subjective Measures of Interestingness in Knowledge Discovery

[89]: Ling et al. (2002), Mining Optimal Actions for Profitable CRM
[90]: Wang et al. (2002), Profit Mining: From Patterns to Actions

Note that several of these aspects may correlate such as novelty and surprisingness but may also contradict each other such as conciseness and diversity. It is not possible to form general statements of correlations or contradictions as they are task-dependent. Furthermore, it is not likely that all of these aspects can be measured for a given task and use case.

3.1.2 Classification of interestingness measures

Besides the aspects of interestingness, interestingness measures are classified into several categories.

Objective Objective interestingness measures do not include any input from the user and only depend on the raw data. Furthermore, these measures are independent of any user or user group and the specific domain. Objective interestingness measures are typically derived from probability theory and statistics. Popular measures in this class are *support* and *confidence* [49], J-measure [91], and strength [92]. Note that there exist many more measures in this category which have been collected by McGarry [63] and Geng and Hamilton [64]. Piatetsky-Shapiro and Matheus "[...] argue that such objective factors are insufficient and that domain-specific, knowledge-based factors also have to be included." [88]

Subjective Silberschatz and Tuzhilin build upon their work and introduce the term subjective interestingness measures [84]. These measures take into account the user's domain knowledge or derive interestingness through interactions with the data mining system. As previously mentioned, it is quite challenging and time-consuming to formalize extensive domain knowledge. Some of the proposed frameworks [64, 82, 84–86] are capable for domain-specific tasks but cannot be generalized. McGarry provides an extensive survey of subjective interestingness measures [63].

Semantic Yao and Hamilton distinguish further and introduce the class of *semantic-based* interestingness measures [64, 93]. This class is a subclass of subjective interestingness measures and is primarily dedicated to the utility and actionability aspects of interestingness measures. The semantic significance of a pattern is reflected in a utility which is additional data for each item in the database. An example of utility could be the profit of an item. The utility of the pattern could then be reflected as the sum of profits of all items and transactions and can further be combined with an objective interestingness measure such as *support* to additionally reflect the statistical significance.

[49]: Agrawal et al. (1994), Fast Algorithms for Mining Association Rules in Large Databases

[91]: Smyth et al. (1991), Rule Induction Using Information Theory

[92]: Dhar et al. (1993), Abstract-Driven Pattern Discovery in Databases

[63]: McGarry (2005), A survey of interestingness measures for knowledge discovery

[64]: Geng et al. (2006), Interestingness measures for data mining: A survey

[88]: Piatetsky-Shapiro et al. (1994), The interestingness of deviations

[84]: Silberschatz et al. (1995), On Subjective Measures of Interestingness in Knowledge Discovery

[63]: McGarry (2005), A survey of interestingness measures for knowledge discovery

[64]: Geng et al. (2006), Interestingness measures for data mining: A survey

[93]: Yao et al. (2006), Mining itemset utilities from transaction databases

3.1.3 Technical properties

(Anti-) monotonicity property One of the most important properties of an interestingness measure is the *(anti-) monotonicity* also known as *a-priori property* or *downward / upward closure* [49]. This is because if this property is given, an algorithm that is based on this interestingness measure in conjunction with a threshold can be implemented such that patterns are pruned early. In other words, not all possible patterns have to be generated in a first run and then later checked against this threshold.

[49]: Agrawal et al. (1994), Fast Algorithms for Mining Association Rules in Large Databases

In some domains, such as high utility pattern mining, an upper-bound approximation of the interestingness measure is selected which is to be proven a-priori. Then the pattern search space can be efficiently pruned against this approximation and the result set is then checked against the actual interestingness measure and its threshold. Therefore, the better the approximation, the more efficient the algorithm [17].

[17]: Fournier-Viger et al. (2019), High-Utility Pattern Mining

Null-invariance Another important property is the null-invariance property [66]. Null-invariance means that an interestingness measure does not depend on the rows in the database where a specific item is *not* contained. This is important since all occurrence probabilities are likely to be low which is an effect of the curse of dimensionality (see section 2.4). Wu et al. show that if a correlation measure (i.e., interestingness measure) for rule mining is not null-invariant, the measure is less expressive and may associate a low correlation even though two items are highly correlated [66].

[66]: Wu et al. (2010), Re-examination of interestingness measures in pattern mining: a unified framework

[66]: Wu et al. (2010), Re-examination of interestingness measures in pattern mining: a unified framework

3.2 A complementary approach

I find some of the terminologies, that are being used in this field of research, rather misleading. First and foremost, the term *interestingness measure* implies that interestingness can be measured which is unlikely since interestingness is highly subjective and imprecise and even for one person the interestingness towards an object is constantly changing as the person is learning new facts. In the worst-case scenario, no interestingness is being measured at all but in any other common scenario, they should be treated as heuristics since they are only estimates. The various aspects of interestingness measures are overall helpful since they provide a crisper definition of what interestingness may entail. It must be duly noted that, to the best of my knowledge, no interestingness measure is capable of covering all aspects. We cannot even be certain that this list of aspects is exhaustive. Interestingness is influenced by a multitude of factors that may not be quantifiable at all but it is highly unlikely that these multiple factors can be expressed in a scalar value that suffers tremendously from distortion.

The classification of interestingness measures into *objective* and *subjective* is generally useful but the names are a rather poor choice. Eventually, the selection of which measures to use is always subjective. And so is the underlying data that is to be mined. In my opinion, *non-parameterized* and *parameterized* interestingness measures would be more descriptive. Parameterized interestingness measures allow the user to inject their domain knowledge in a restrictive manner whereas non-parameterized measures purely rely on the data. However, the latter does not necessarily mean that the user is not able to provide their domain knowledge since this could also be realized by modifying the data such as the utility values. Another challenge is thresholds which are typically used for any type of measure to filter the patterns. This parameter estimation is typically quite difficult for the user and the complexity only increases the more measures, thus, thresholds are being used.

The process of exploratory analysis is highly iterative. Early work in the field of interestingness measures in data mining assumes a linear process following the traditional KDD process (see Figure 3.1). However, visual analytics assumes

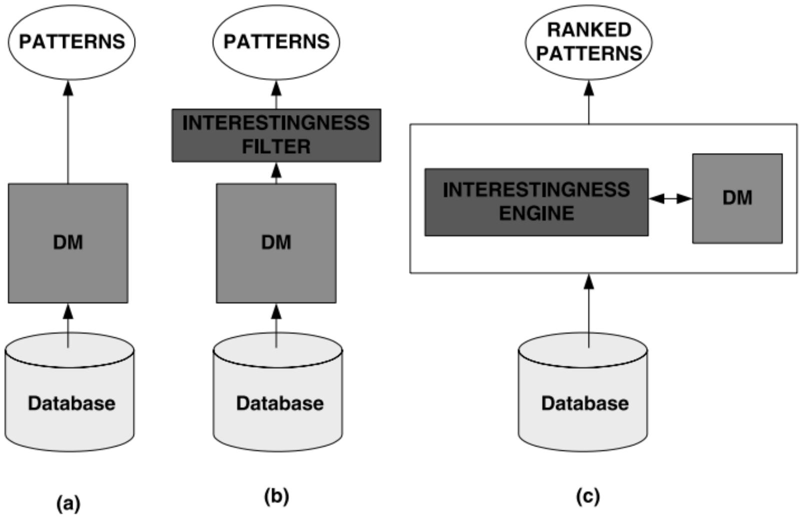


Figure 3.1: A framework for pattern mining where (a) all patterns are mined without any filters; (b) the patterns are filtered in a post-processing step after the mining, and (c) the filtering and ranking of patterns are included in the mining process. This figure is taken from McGarry [63]. A similar figure was first published by Silberschatz and Tuzhilin in 1995 [84].

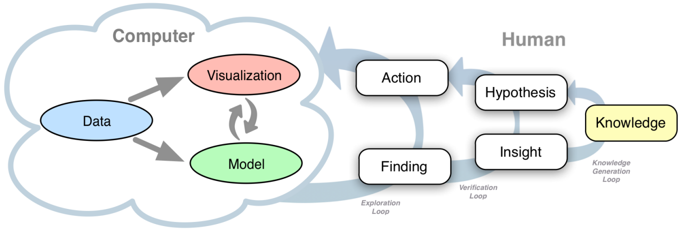


Figure 3.2: The knowledge generation model of visual analytics depicts how knowledge is generated through iterative interaction cycles with a system consisting of data, models, and visualizations. This figure is taken from Sacha et al. [94].

iterative processes such as depicted by the knowledge generation model for visual analytics (Figure 3.2). Knowledge is generated by humans through multiple interaction cycles with a system. This requires feedback loops through interaction as well as visualization to inspect the data, the models, and the models' results. This means that during the exploration the user's knowledge is continuously expanding, however, interestingness measures are quite static as they depend on the algorithm and implemented system. Changes in knowledge can only be input into the system by varying parameters of interestingness measures, their thresholds used for filtering, or modifying the underlying data that generate the patterns. Therefore, it is likely that an interestingness measure correlates quite well with what the user actually finds interesting but may decrease in importance to the user in later exploration stages. It is typically not feasible for a user to design and implement their own interestingness measures and, to the best of my knowledge, there exists no system or framework that supports an abundance of interestingness measures as these measures depend much on the data, the type of pattern mining, and at last the implemented algorithms. The more interestingness measures are being implemented the more the efficiency of the mining decreases and, at the same time, the complexity for the user increases as the parameter estimation becomes more difficult.

At last, we must acknowledge that it is not always possible to quantify interestingness and that aspects of interestingness are embedded into the semantics of patterns and the underlying data in combination with unknown and unformalized (user) knowledge. Researchers led by Edward Feigenbaum argue that a machine must have knowledge in order to act intelligent. He is known as the father of knowledge engineering, a discipline that tries to formalize knowledge such that it is actionable for a system [1]. Researchers realized that it is difficult if not impossible to formalize general knowledge to enable a system to act as a human. This became known as "Feigenbaum's knowledge acquisition bottleneck" [95]. The focus of this area shifted onto machine learning, neural networks, and deep learning which does not require formalized knowledge to be readily available but instead learns such knowledge on its own. This, on the other hand, requires that training data is available which, in the task of clustering, is not the case. However, we still strive to build an intelligent system that is capable of human intelligence but today call it artificial general intelligence (AGI) [96].

[1]: Zhou (2021), Machine Learning

[95]: Hoekstra (2010), The knowledge reengineering bottleneck

[96]: Shevlin et al. (2019), The limits of machine intelligence: Despite progress in machine intelligence, artificial general intelligence is still a major challenge

But if we cannot guarantee to measure interestingness, should we abolish this idea altogether? In other words: if we do not necessarily measure interestingness, **what do we measure?**

Definition 3.2.1 (Interestingness Measure)

$$f : \mathbb{P} \to \mathbb{R}^n$$

At their core, all interestingness measures are simply a function that maps from patterns to real numbers. The definition states n-dimensional vectors to generalize, however, in many cases, the mapping is to scalar values ($n = 1$). This definition follows Kontonasios et al. [97]. But what does that mean? An interestingness measure is essentially a *quantified property* of a pattern. This is also known as a *feature*. Therefore, an interestingness measure is a function generating a feature of a pattern. This is a well-known task in machine learning known as *feature engineering*. The purpose is to find or create

[97]: Kontonasios et al. (2012), Knowledge discovery interestingness measures based on unexpectedness

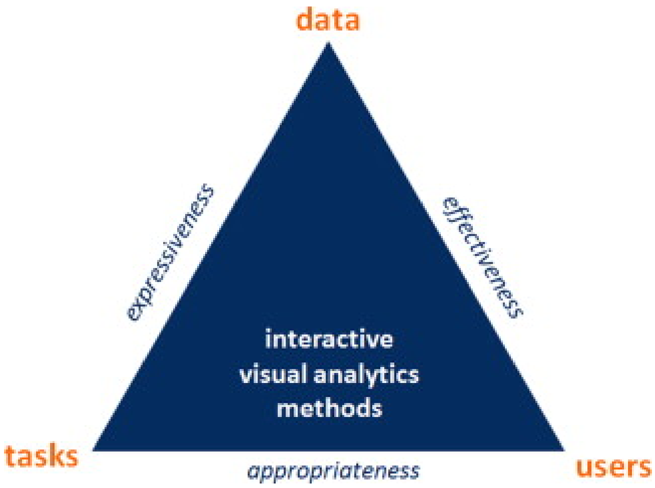


Figure 3.3: The data-users-tasks design triangle of Miksch and Aigner. [98] This figure is taken from their publication.

features that are relevant to the task. This already sheds light that the effectiveness of features depends on the task.

Miksch and Aigner show with their data-users-tasks design triangle [98] (Figure 3.3) how any visual analytics system is mainly influenced by these three factors. This is naturally also true for interestingness measures since these describe properties of a pattern and the clusters they represent which is information that the user can eventually turn into knowledge.

[98]: Miksch et al. (2014), A matter of time: Applying a data-users-tasks design triangle to visual analytics of time-oriented data

Data The data determines what features can possibly be calculated. For instance, calculating the utility of a pattern is only possible if the utility can be derived from the data. Otherwise, it must be provided by the user. The type of structure also influences what measurements can be used. For example, graphs are a rather complex data structure compared to itemsets and provide an extensive amount of properties that can be measured such as the order (number of vertices), size (number of edges), and girth (length of the shortest cycle) [99].

[99]: Diestel (2012), Graph Theory, 4th Edition

Users The users should influence the design of interestingness measures as their (domain) knowledge and expectations ultimately decide what patterns are considered interesting. It is also critical that the user understands how to interpret a certain measure. If they are only told that a number should reflect their interestingness of a pattern they will respond poorly in using the system because it is inevitable that at some point, there will be inconsistencies leading to a decrease of trust in the system by its users [21].

[21]: Jentner et al. (2018), Minions, Sheep, and Fruits: Metaphorical Narratives to Explain Artificial Intelligence and Build Trust

Tasks Tasks ultimately provide purpose to data analysis and the exploration of data. They constrain and guide what of the user's knowledge is necessary to be formalized and captured to design an effective interestingness measure. Moreover, they determine what aspects of the data are necessary to be analyzed.

3.3 Taxonomy

This section is dedicated to an alternative, complementary taxonomy as provided in the literature. Instead of classifying interestingness measures into what aspects they possibly measure, I simply separate them by what properties they are measuring. To recapitulate, patterns are cluster representations of structured data. A dataset contains rows where each row consists of an ID and the structured data itself. A row may contain additional data such as utility values etc. which we consider as metadata. A pattern, therefore, is structured data of the same type that maps to one or multiple rows in the dataset.

3.3.1 Describing what?

Now we must only ask ourselves what property an interestingness measure is describing. The following distinguishes mainly between the properties of the pattern itself and the properties of the cluster the pattern represents. However, the cluster properties are further divided into properties of the structured data, statistical cluster measures, and measures for the cluster metadata.

Pattern

First and foremost, we can express certain properties of the pattern itself. A simple example of this is the *generation* of a pattern (see Definition 2.1.6 & Definition 2.1.7). There are multiple variations of this property imaginable such as the generation of the antecedent (left) or consequent (right) side of the rule which is possible to calculate for association rules, sequential rules etc. Furthermore, it is possible to calculate the number of occurrences for a specific item or itemset. For itemsets and association rules, this number can only be zero or one but for more complex types of structured data such as sequences, trees, and graphs the number of occurrences can be greater than one.

Similarly, for more complex data structures more properties exist as convincingly shown in the field of graph theory [99].

[99]: Diestel (2012), Graph Theory, 4th Edition

We must acknowledge the general limitation of interestingness measure in conjunction with complex data structures such that it is not possible to express all of the structural properties in a single measure but rather certain aspects of it that are possible to calculate given the available data and deemed interesting for the user and their task.

Cluster

Since patterns are cluster representants and the mapping of one pattern to the specific rows in the dataset is maintained, various information can be calculated describing the properties of the cluster. An important distinction to pattern properties is that this information is *aggregated*. We can distinguish what type of property an interestingness measure for cluster information describes.

Structured Data Similar to the patterns, the structured data can be featured in various properties and aggregated in several ways. For example, the generation can be calculated for each data structure in each row in the dataset. Then, the maximum generation of the cluster would indicate the most complex data structure within the cluster.

There are also noteworthy differences between the structure of patterns and the data. For example, sequential patterns such as $\langle\{a\},\{b\}\rangle$ state that *a occurs before b*. It *does not* indicate any distance between a and b which could be measured by how many tokens there are in between. Therefore, the pattern itself only prevails the order but loses some of the information that might be entailed in the original structured data (see Definition 2.1.4). However, a measure can be defined that calculates the distance between a and b for each structured data that the pattern matches. Because a and b can occur multiple times the minimum distance within one event sequence is typically chosen. Then, as usual, the measurements need to be aggregated over the whole cluster whereas typically the maximum is selected. When defined in conjunction with a threshold, the clusters are constrained to this threshold such that there exist no events that are further apart than what the threshold defines. This is known as the *window constraint*.

Statistics (frequency) This class is by far the most common for interestingness measures covering the frequentist approach known as frequent pattern mining. This is typically based on the cluster size that a pattern represents whereas the pattern is said to be frequent for large clusters. The intention behind this is that frequent patterns are statistically more meaningful and important than any infrequent patterns. The most common interestingness measure is called *support* which is defined as:

Definition 3.3.1 (Support)

$$supportP_A = |\{P|P \in D \wedge P_A \sqsubseteq P\}|$$

These are the number of rows in a database D that contain pattern P_A at least once in their structured data. In many cases, the support measure is defined relative to the size of the database estimating the probability of occurrence of pattern P_A.

Definition 3.3.2 (Relative Support)

$$support\_relP_A = \frac{supportP_A}{|D|}$$

Note, that the relative support is not null-invariant whereas the absolute support measure is. For rule mining, such as association rule mining, correlations between patterns can be mined for. One of the first measures, confidence, for a rule $A \rightarrow B$ is defined as the conditional probability [49].

[49]: Agrawal et al. (1994), Fast Algorithms for Mining Association Rules in Large Databases

Definition 3.3.3 (Confidence)

$$confidenceA, B = \frac{supportA \cup B}{supportA} = PB|A$$

Geng and Hamilton list 36 other statistical interestingness measures for association rules that are an extension or variations of the support and confidence measure [64]. Carvalho et al. correlate eleven of the interestingness measures with "real" human interest for eight different datasets and find

[64]: Geng et al. (2006), Interestingness measures for data mining: A survey

that in more than 35% of the cases, the correlation was strong ($> 60\%$) meaning that they are in general a good estimator [100]. However, no interestingness measure was a clear winner across all datasets showing that these measures are highly task, user, and data-dependent.

[100]: Carvalho et al. (2005), Evaluating the Correlation Between Objective Rule Interestingness Measures and Real Human Interest

Statistics of Cluster-Metadata

The last sub-class for clusters is cluster metadata. Here, additional data other than the structured data are being used. For example, the domain of high-utility pattern mining is part of this class. Unlike frequent pattern mining, high-utility pattern mining does not assume that each item in a database is equal (i.e., equally interesting) but uses a utility function to mine for patterns of the highest utility. Furthermore, this allows for quantitative databases meaning that an item can occur multiple times within one transaction. The toy example is an extension of the market basket analysis where transactions are mined of what customers bought at the grocery store. Because of the quantitative nature, this database can also reflect when a customer bought two pieces of bread and one milk instead of just encoding bread and milk in one itemset. Additional metadata is available that assigns a utility value to each of the items such as the profit (e.g., bread=$3 and milk=$2). Then a utility function is defined by multiplying the profit of each item by the quantity of how often it has been bought and summed across all transactions where the pattern is contained.

Naturally, these types of measures are very flexible as they allow the definition of any utility function and even combine multiple utility values and input domain knowledge by specifying weights for these utilities.

3.3.2 Using How?

An interestingness measure needs to be interpretable by the user to be useful. This will be more elaborated in section 5.6. This section sheds some light on how interestingness measures can be used and also what their limits are.

Typically, interestingness measures are used in conjunction with filters (thresholds) or as a means to rank patterns. The latter can also be combined with a threshold which is then referred to as top-k mining [101–103]. This frees the user from specifying a specific threshold and instead lets the user choose how many patterns should be returned according to a combined interestingness measure. Therefore, the user may choose to receive the top 20 patterns of the highest support instead of specifying that the support should be greater than 50% which may return more or fewer patterns.

[101]: Han et al. (2002), Mining Top-K Frequent Closed Patterns without Minimum Support
[102]: Tzvetkov et al. (2005), TSP: Mining top-*k* closed sequential patterns
[103]: Ryang et al. (2015), Top-k high utility pattern mining with effective threshold raising strategies

In visualization, interestingness measures are often used to provide an overview of the data and to make interesting patterns visibly stand out. This releases some restrictions as, for example, instead of showing a list of ranked patterns to present them based on their similarity while the saturation of the pattern represents the support [8]. The next chapter surveys visualization approaches for patterns and shows different designs and tradeoffs that have to be considered.

[8]: Jentner et al. (2018), Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool

3.4 Use cases

This section shows three use cases of how interestingness measures can be applied in certain domains. This shall provide the reader with some intuition behind the theory of the previous sections.

3.4.1 Comparative Case Analysis

Task Description

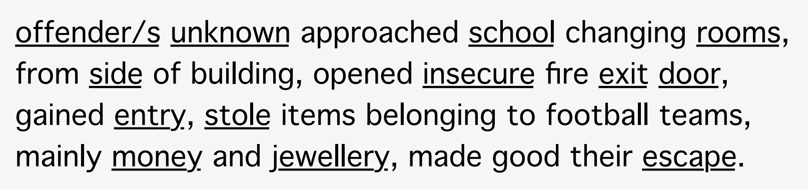
[1] Comparative Cases Analysis (CCA) or Similar Fact Analysis (SFA) [104] is an important task of Criminal Intelligence Analysis [105]. In the VACLRI-project (Visual Analytics for Sense-making in CRiminal Intelligence analysis), [2] the available data of crime reports focuses on burglaries. A crime report is a form that police officers fill out and try to capture as many details about the incident as possible. For burglaries, form elements try to capture the incident's location and the possible time in a time range (earliest possible time and last possible time). Such information can be used to find similar cases using geospatial and temporal information, but this typically does not capture any details, such as if the burglar

[1]: This section is based on my publication "Making Machine Intelligence Less Scary for Criminal Analysts: Reflections on Designing a Visual Comparative Case Analysis Tool" (Section 1: Introduction) [8]. I have been the main author of this publication and have written major parts of the content. The paper was co-authored by my co-authors Dominik Sacha and Florian Stoffel and edited by Geoffrey Ellis, Leishi Zhang, and Daniel Keim.

[104]: Prowse et al. (2000), Working Manual of Criminal Law

[105]: NPIA (2008), National Policing Improvement Agency: Professional Practice on Analysis

[2]: https://cordis.europa.eu/project/id/608142, accessed March 25, 2022



offender/s unknown approached school changing rooms, from side of building, opened insecure fire exit door, gained entry, stole items belonging to football teams, mainly money and jewellery, made good their escape.

Figure 3.4: The Modus Operandi (MO), a free-text-field of a crime report. The police officer describes in bullet points how the burglar entered the building, how the search was performed, what was stolen, and how the burglar exited the building. The figure is taken from the original publication [8].

used specific tools to break into a property.

There are two types of CCA. Firstly, a criminal investigator uses CCA to find information about an unsolved crime. Other similar crimes may have already been solved, or a known suspect who could make this suspect also interesting for the case at hand [106]. Secondly, a tactical analyst periodically analyzes crime reports to find new trends and patterns. This can help the police to act preventative such as sending patrols to the location deemed the highest risk for a crime.

[106]: Cope (2004), Intelligence Led Policing or Policing Led Intelligence?: Integrating Volume Crime Analysis into Policing

The crime reports entail a free text form which is called *Modus Operandi (MO)* (see Figure 3.4). In the MO field, the police officer describes in a short text how the burglar entered the building, how the search was conducted, what was stolen, and how the burglar exited the building. If there were any tools used, the MO also states this. From a natural language process (NLP) perspective, the text quality of these short texts is rather poor. The reasons for this are manifold, the texts are often written at the scene, and the field in the form is pretty small. Therefore, the texts contain spelling errors, abbreviations, incorrect grammar, etc. Sometimes, the handwritten text had to be digitized, introducing additional errors. In other cases, the police officers did not write down the text themselves but called the police station and dictated what to write to a person over the phone.

The task is to compare crimes using the MO text, which provides the highest level of detail regarding the incident. To do that, concepts from the text reflect a similar meaning. For example, screwdrivers, hammers, etc., are being added to the concept of "tools". Criminal investigators perform a similar technique. However, it is conducted manually using a spreadsheet. The table of crime reports is then sorted and filtered according to the extracted concepts that are deemed useful. Our contribution to VALCRI intends to automate that process as much as possible and provide various interactive visualization techniques in addition to the crime table.

Limitations of existing work

[3] Our analysis approach combines many different analytical techniques, such as textual feature extraction, sequential pattern mining, high-dimensional data analysis, and visual interactive clustering applied to criminal intelligence analysis. We illustrate these with examples within each area.

Comparative Case Analysis CCA is based on the notion of comparison, which is a fundamental technique used by many social sciences and scientific domains [107]. CCA starts with processing the text to extract key features, followed by reasoning and sense-making based on similarity comparison. One challenge of CCA is feature extraction - most of the feature extraction reported in the literature is manual. For example, Bennell et al. [108] manually extracted features from MO of 86 solved commercial burglaries committed by 43 serial offenders to compare the similarity between burglary cases. The findings were used to examine if a high degree of similarity between them enables different cases to be validly linked to a common offender. This requires a significant amount of work even with this relatively small amount of data. Another challenge is the comparison. Given a set of crimes, what to compare and how to compare has to be decided by the analyst [104]. Work carried out by Canter et al. [109] used the *Jaccard coefficient* to measure the proportion of co-occurring features in crimes. The work also applied *multidimensional scaling* on the data to investigate the consistency of features across organized and disorganized cases. The research revealed that disorganized features were either easy to identify or more common, probably due to their vast number compared to organized features. To the best of our knowledge, no work has been reported on automatic feature extraction, feature selection and weighting for CCA.

Automated Feature Extraction for CCA For the feature generation, we use a custom framework based on components

[3]: This section is taken from my publication “Making Machine Intelligence Less Scary for Criminal Analysts: Reflections on Designing a Visual Comparative Case Analysis Tool” (Section 2: Related Work) [8]. I have been the main author of this publication and have written major parts of the content. The paper was co-authored by my co-authors Dominik Sacha and Florian Stoffel and edited by Geoffrey Ellis, Leishi Zhang, and Daniel Keim.

[107]: Collier (1993), The Comparative Method

[108]: Bennell et al. (2002), Linking commercial burglaries by modus operandi: tests using regression and ROC analysis

[104]: Prowse et al. (2000), Working Manual of Criminal Law

[109]: Canter et al. (2004), The Organized/Disorganized Typology of Serial Murder: Myth or Model?

[110]: Manning et al. (2014), The Stanford CoreNLP Natural Language Processing Toolkit

[111]: (), Apache OpenNLP

from Stanford CoreNLP [110] and Apache OpenNLP [111]. For the characterization of concepts and automated class assignments, two different resources, Wordnet [112] and Framenet [113] are used. Besides customized retrieval and classification methods, the analytic parts are based upon state-of-the-art as described by Manning et al. [114] or Jurafsky and Martin [115].

For our system, we use a sequential pattern mining algorithm to mine for frequent sequences of terms occurring in the MO of the crime reports. The problem was formally defined by Agrawal et al. [116]. To avoid redundant patterns, we mine for a set of closed sequential patterns [117, 118]. We use a DR on the mined frequent patterns and visualize them in a feature similarity space. Similarity measures for sequential patterns exist [119], however, to be consistent with the data similarity space, we use a binary feature vector containing the crime reports where a bit is set to one if the sequence occurs in that crime report.

Visual Analytics for CCA Automatic analysis methods such as feature extraction, pattern mining, clustering and dimensionality reduction provide effective means of analyzing a large amount of crime data and extracting patterns from it. However, visual analytics tools for supporting CCA are scarce. Software systems such as IBM I2 [120] and Jigsaw [121] were developed for the general purpose of Criminal Intelligence Analysis but little work has been carried out to improve on the manual CCA process. Jäckle et al. proposed a projection-based approach [122] for analyzing similarity between textual data items but the approach does not allow police officers to form the customary structured tables. The Spherical Similarity Explorer system developed by Zhang et al. [123] allows the analyst to project crime data onto a spherical surface for similarity analysis - the tool focuses on one DR algorithm with limited interaction possibilities.

Interactive Visual Machine Learning As Sacha et al. stated: for a VA system to be effective, it is essential to allow the user to interact with the data and the models at different stages of the analysis to iteratively improve, adapt, and combine analysis methods to solve the analysis task [124]. Recent work by Sacha et al. [125] surveyed existing visual Dimensionality Reduction (DR) tools and highlighted interaction possibilities

[112]: Miller (1995), WordNet: A Lexical Database for English

[113]: Baker et al. (1998), The Berkeley FrameNet Project

[114]: Manning et al. (2008), Introduction to information retrieval

[115]: Jurafsky et al. (2009), Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition

[116]: Agrawal et al. (1995), Mining Sequential Patterns

[117]: Gomariz et al. (2013), ClaSP: An Efficient Algorithm for Mining Frequent Closed Sequences

[118]: Yan et al. (2003), CloSpan: Mining Closed Sequential Patterns in Large Datasets

[119]: Saneifar et al. (2008), S2MP: Similarity Measure for Sequential Patterns

[120]: IBM (), IBM i2 Intelligence Analysis Platform

[121]: Stasko et al. (2008), Jigsaw: supporting investigative analysis through interactive visualization

[122]: Jäckle et al. (2017), Interpretation of Dimensionally-reduced Crime Data: A Study with Untrained Domain Experts

[123]: Zhang et al. (2016), Spherical Similarity Explorer for Comparative Case Analysis

[124]: Sacha et al. (2017), What you see is what you can change: Human-centered machine learning by interactive visualization

[125]: Sacha et al. (2017), Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis

to improve the effectiveness of the tools. The interpretability of results and the usability of interactive DR systems, especially for domain expert users (without technical and data analysis background) is a major area for improvement.

Existing visual text analytics approaches such as IN-SPIRE [126] (and its predecessors [127, 128]), or recent works described by Ruppert et al. [129], shed light on the possibility of automatically processing textual documents to obtain and explore document clusters. These systems adopt different Dimensionality Reduction (DR) and/or clustering techniques to generate visual embeddings of the high-dimensional data to enable the analyst to compare the similarity between data items and examine interesting patterns in the data. Given that DR and clustering are complex processes that involve a series of selection, computation and validation, input from the human analyst is often beneficial and largely unavoidable. Wenskovitch et al. [130] provide a good overview of how to combine DR and clustering and also recommend design decisions that need to be considered.

Hybrid Views Hybrid views, also often referred to as dual views aim to provide simultaneous access to the data and feature space. Van der Corput and Van Wijk [131] are using I^F-F^I tables to support access to both spaces. Turkay et al. [132] and Yuan et al. [133] use two tightly coupled scatter plots. We follow this strategy by creating these scatter plots through DR. However, additionally, we use one table where both, data and features, are combined and the clusters generated in the data space can be interpreted. Demiralp [134] uses a heatmap-matrix diagram in combination with a scatter plot to interpret clustering results. We follow this approach, however, we utilize bar charts in a table to enable the user to perform a cluster comparison.

From Modus Operandi to Sequential Patterns

[4] Figure 3.5 shows how the MOs are being transferred to sequential patterns and, eventually, how the sequential patterns are being used as feature vectors to measure the similarity of the crime reports. Florian Stoffel contributed concept extraction and entails an extensive NLP pipeline performing spelling correction, various heuristics for abbreviations and

[126]: Wise (1999), The Ecological Approach to Text Visualization

[127]: Endert et al. (2012), Semantic Interaction for Sensemaking: Inferring Analytical Reasoning for Model Steering

[128]: Bradel et al. (2014), Multimodel semantic interaction for text analytics

[129]: Ruppert et al. (2017), Visual Interactive Creation and Validation of Text Clustering Workflows to Explore Document Collections

[130]: Wenskovitch et al. (2017), Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics

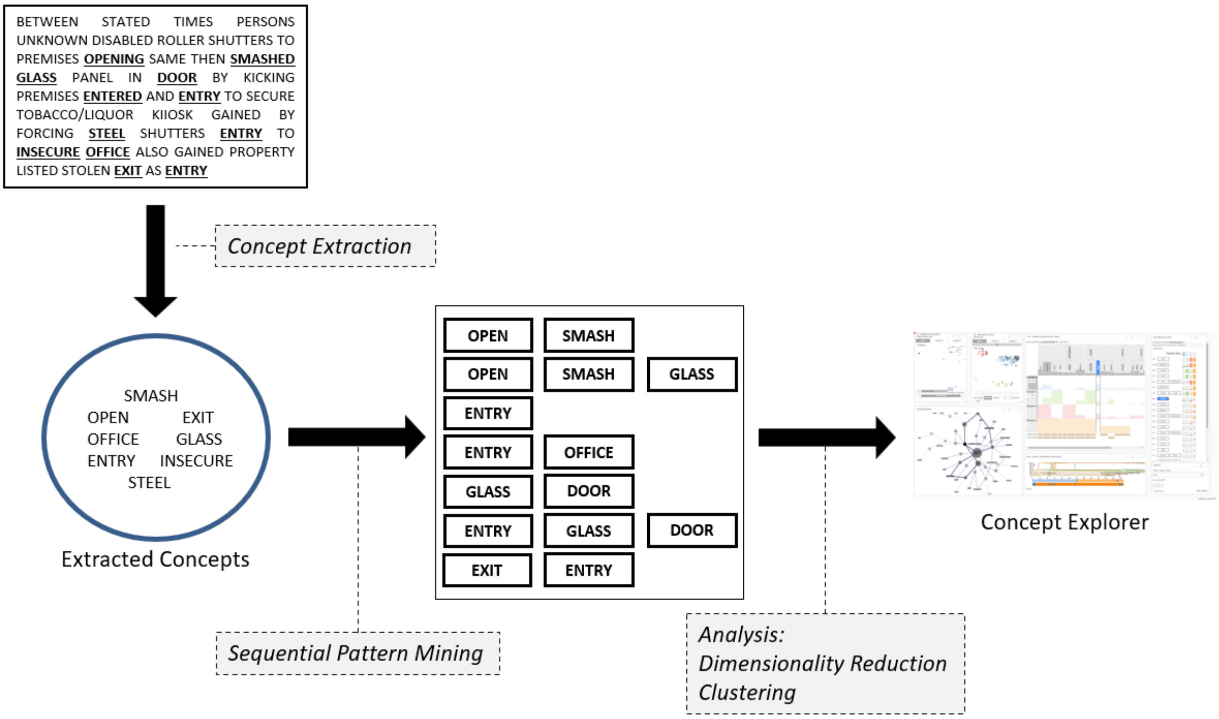
[131]: Corput et al. (2016), Exploring Items and Features with I^F, F^I-Tables

[132]: Turkay et al. (2011), Brushing Dimensions - A Dual Visual Analysis Model for High-Dimensional Data

[133]: Yuan et al. (2013), Dimension Projection Matrix/Tree: Interactive Subspace Visual Exploration and Analysis of High Dimensional Data

[134]: Demiralp (2017), Clustrophile: A Tool for Visual Clustering Analysis

4: This section is based on my publication "Making Machine Intelligence Less Scary for Criminal Analysts: Reflections on Designing a Visual Comparative Case Analysis Tool" [8] (Section 3.1). I have been the main author of this publication and have written major parts of the content. The paper was co-authored by my co-authors Dominik Sacha and Florian Stoffel and edited by Geoffrey Ellis, Leishi Zhang, and Daniel Keim.



BETWEEN STATED TIMES PERSONS UNKNOWN DISABLED ROLLER SHUTTERS TO PREMISES **OPENING** SAME THEN **SMASHED** **GLASS** PANEL IN **DOOR** BY KICKING PREMISES **ENTERED** AND **ENTRY** TO SECURE TOBACCO/LIQUOR KIIOSK GAINED BY FORCING **STEEL** SHUTTERS **ENTRY** TO **INSECURE** **OFFICE** ALSO GAINED PROPERTY LISTED STOLEN **EXIT** AS **ENTRY**

Figure 3.5: The processing pipeline for modus operandi texts. The concept extraction normalizes and extracts concepts from the MOs. Sequential pattern mining then generates frequent patterns that maintain the order of the events. The patterns are then used as feature vectors for dimensionality reduction of crime reports and clustering. The figure is taken from internal project presentations and has been slightly adapted.

data cleaning, tokenization, part-of-speech- (POS-) tagging, lemmatization, stemming, and ontologies to extract the concepts. The exported concepts and their token position (and character position additionally) are used for pattern mining. Initially, we used a bag of word model in combination to use the tokens as a feature vector similar to the term frequency [135] of the popular TF-IDF measure [136] to measure the distance used by dimensionality projection algorithms and clustering. This idea stems from Dominik Sacha, Geoffrey Ellis & Florian Stoffel. Throughout the project and user feedback, it became clear that this is too inaccurate as the order of events is meant to measure similarity. The assumption here is that, despite the poor quality of the MOs, the MOs are written in a way that maintains the order of events such that the beginning always states how the suspect entered the building and what tools were being used. Sequential patterns are capable of modeling this. However, the feature vectors for each crime report grow much larger since all permutations (honoring the order) are enclosed. In the beginning, we considered whether a sequential pattern is contained in an MO and how often. This is, however, a rare occurrence due to the sparseness of the data (see section 2.4). Also, the

[135]: Luhn (1957), A Statistical Approach to Mechanized Encoding and Searching of Literary Information

[136]: Jones (2004), A statistical interpretation of term specificity and its application in retrieval

Table 3.1: Each crime report is annotated with a vector of sequential patterns. A 1 denotes if the pattern occurs in the crime report and a 0 if the pattern does not occur. This is similar to term frequencies [135].

Crime Report ID	$\langle\{\text{smash}\}\rangle$	$\langle\{\text{open}\},\{\text{smash}\}\rangle$	$\langle\{\text{entry}\}\rangle$	$\langle\{\text{entry}\},\{\text{office}\}\rangle$
1	1	1	0	0
2	1	0	0	0
3	0	0	1	0
4	0	0	1	1

domain experts consisting of police officers, criminal investigators, and tactical analysts expressed their opinions that the occurrence amount is irrelevant to the distance. Table 3.1 shows the vectors exemplary.

Description of Interestingness Measures

[5] Initially, we used a sequential pattern mining approach that used the *minimum support* threshold (see Definition 3.3.1) and *window constraint* parameter (see section 3.3.1) [18]. The window constraint was a max-threshold that defined how many tokens may occur between two tokens of a pattern. For example, a max-gap of 2 would allow two tokens in between two concepts that have been extracted. Considering Figure 3.4, this would mean that a pattern $\langle\{\text{school}\},\{\text{rooms}\}\rangle$ is valid because there is only one token in between but $\langle\{\text{school}\},\{\text{side}\}\rangle$ would be discarded since the token distance is three. The minimum support was fixed to 10%. Initially, the users were happy with the tool as the automation aspect of the concept extraction and modeling as sequential patterns were meaningful and important to them. However, upon using the tool more intensely, they stated that estimating the window constraint parameter proved difficult as the MO texts varied a lot. It was easy to miss an important crime report because the window constraint was set too tight. Furthermore, the users were overwhelmed by the number of patterns leading to the popular exchange between a criminal investigator and me (see Chapter 1, Figure 1.1).

In a trial reported in neither publication, we experimented with utility mining. Since the extracted tokens have no inherent utility, we let the users assign values (ranging from 1-10) to each token. Then, patterns were mined using a high-utility mining approach. The parameter estimation for the threshold and the utility assignment proved difficult for the users.

[5] This section is based on two of my publications "A Visual Analytics Approach for Crime Signature Generation and Exploration" [18] (Section 4: Visual Analytics Approach) and "Making Machine Intelligence Less Scary for Criminal Analysts: Reflections on Designing a Visual Comparative Case Analysis Tool" [8] (Section 3 & 4). The former publication was the first prototype of the VALCRI project, which was further developed and eventually ended with the project prototype described in the second publication.

[18]: Jentner et al. (2016), A visual analytics approach for crime signature generation and exploration

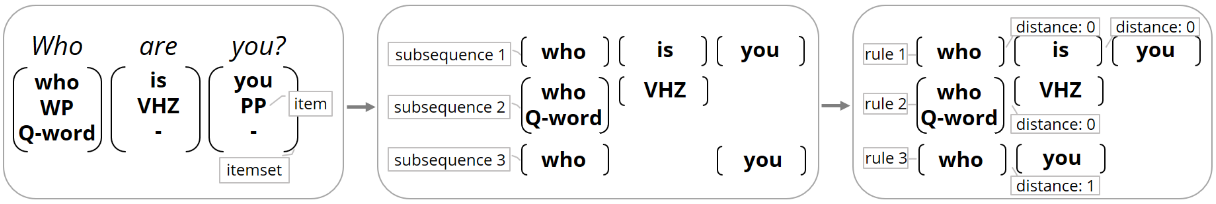
Concepts such as *screwdriver* are, by themselves, not very important but only in combination with a *window* or *door* indicating that the burglar used said tool to break into the property. On the other hand, if a *screwdriver* has been stolen from a property, this is not of particular interest. Because of this, we quickly discarded going forward with utility mining.

In the later development [8], we, therefore, switched our approach and discarded the window constraint entirely (threshold is unlimited). We also reduced the minimum support from 10% to 5% since even a low amount of similar crime reports may prove to be a useful resource. These two steps increased the number of patterns, so we introduced a generation threshold (see Definition 2.1.7). Because each itemset in the sequence only consists of one item (the token itself), the generation is identical to the sequence length (i.e., the number of itemsets in the sequence). Furthermore, we allowed the user to mine selected patterns of a higher generation (see section 5.5). We also introduced a distance measure based on the same strategy as measuring the similarity of crime reports (see Table 3.1). To calculate the distance of two patterns, we transpose the table and use the vectors of what crime reports support the pattern. Even though the minimum support for mining is set to 5%, the user can further filter the patterns by increasing the minimum support and lowering the maximum support. Note that for these interactions, the mining does not have to be repeated, and the operations are only conducted on the pattern result set, improving the load times of the application. Maximum support patterns cannot be calculated efficiently during the mining as the support property is anti-monotone (i.e., decreases when pattern generations increase). In VALCRI, typically, the patterns *door* and *window* have the highest supports close to 50%, which matches the expectations of the criminal investigators since windows and doors are the most common entry point used by burglars. Therefore, these patterns, by themselves, are not meaningful as they are not specific enough to identify a common behavior of a suspect. A combination, e.g., *screwdriver* and *window*, can be more interesting for the users.

[8]: Jentner et al. (2018), Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool

Finally, the users can directly weigh the crime-pattern vectors (see Table 3.1). This influences the distance measure and, eventually, the dimensionality reduction algorithms. If, for example, the user increases the weight of the pattern *door*.

The projection algorithm will place all crime reports that have the pattern *door* from the ones that do not have the pattern *door*. If additionally, the weight of *window* is increased as well, four groups will become visible: (i) crime reports with *window* and *door*; (ii) crime reports only with *window*; (iii) crime reports only with *door*; (iv) crime reports that neither contain *door* or *window*.



(a) Sequence (b) Subsequences (c) Rules

Figure 3.6: Questions are modeled as event sequences (a) allowing to carry the token in each itemset as well as various token-based annotations (i.e., labels). The event sequences can then be mined as sequential patterns (b) which allow for containing only partial itemsets or items. The constrained sequential rules (c) then serve as a classifier to distinguish the question type for each pattern. A similar figure can be found in the original publication (Figure 2) [9]. Image credit: Rita Sevastjanova

3.4.2 QuestionComb

Task Description

[6] Linguistic researchers are interested in generating and finding rules to discover relationships between syntax, semantics, and pragmatics. Specifically, the relationship between syntax and semantics is interesting as it allows generating models where machines can derive semantics based on syntax. The rules, on the other hand, are a tool that helps humans to better understand these relationships as the nowadays large language models are quite powerful but understanding their inner workings is rather difficult. One of these tasks of generating rules is to distinguish *information-seeking questions (ISQ)* from *non-information-seeking questions (NISQ)* or also *rethorical questions* [137]. This task is challenging since the syntax itself typically does not necessarily reflect the type of question but rather the context of how the question has been posed [138, 139].

QuestionComb is an approach that uses a visual explanation of linguistic phenomena through interactive labeling. The interactive labeling is done because available labeled data resources are often scarce. Creating such datasets is a time-consuming process that requires sufficient domain expertise. QuestionComb, with its gamification approach, can help users generate such data while learning more about how question types can be distinguished using rules.

Data Modeling

[7] As with the comparative case analysis use case (see subsec-

6: This section is based on the publication “QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling” (Section 1: Introduction) [9]. I have co-authored this publication and contributed to the data modeling as well as the pattern mining components. The paper is a joint effort of Rita Sevastjanova, Fabian Sperrle, Rebecca Kehlbeck, Jürgen Bernard, and Mennatallah El-Assady.

[137]: Kalouli et al. (2018), A Multilingual Approach to Question Classification

[138]: Ranganath et al. (2016), Identifying Rhetorical Questions in Social Media

[139]: Ranganath et al. (2018), Understanding and Identifying Rhetorical Questions in Social Media

7: This section is based on the publication “QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling” (Section 4.1: Data as Sequences of Words) [9]. I have co-authored this publication and contributed to the data modeling as well as the pattern mining components. The paper is a joint effort of Rita Sevastjanova, Fabian Sperrle, Rebecca Kehlbeck, Jürgen Bernard, Mennatallah El-Assady, and myself.

tion 3.4.1), the questions are modeled as event sequences to preserve the order of the words. However, in this use case, the itemsets do not only consist of the token itself but can be extended by any type of attribute that can be mapped to a specific token. A popular example is Part-of-Speech (POS) tags. Our lingvis.io framework is capable of generating a multitude of annotations including token-based annotations [39]. Our data modeling is flexible and allows for arbitrary annotations such as WH-question annotations (i.e., if a token is what, when, who, etc.), discourse particles, and speech acts.

[39]: El-Assady et al. (2019), lingvis.io - A Linguistic Visual Analytics Framework

The event sequences also allow annotating context such as a speaker label for the person that posed the question. Another example might be a label for whether this question was a follow-up question or not. Such annotations are added as the first itemset of the event sequence. They allow finding rules such as *Speaker A always asks rhetorical questions*.

Because this task is essentially a classification task, the ISQ and NISQ labels are also added as itemsets at the end of the sequence. A constraint sequential rule approach then allows us to generate the respective rules.

Description of Interestingness Measures

[8] Sequential pattern mining allows us to use the classic *support* measure (see Definition 3.3.1). The minimum support is hardcoded and set to 1% which has been determined through multiple experiments and discussions with linguistic experts. This only prunes random extremely infrequent patterns but still considers the majority of the search space causing a so-called *pattern explosion*.

We further model the patterns as *constrained sequential rules*. They are constrained such that the right-hand side of the rule (i.e., the consequent side) only allows for the ISQ and NISQ labels. The otherwise unconstraint rule mining would increase the search space even more. To further prune the result set of rules, the *confidence* measure (see Definition 3.3.3) is being used with a threshold for min-confidence of 95%. The confidence measure is an estimator for the conditional probability that the sequential pattern matches the ISQ or NISQ

8: This section is based on the publication "QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling" (Section 4.2: Sequential Pattern Mining for an Explainable Classifier) [9]. I have co-authored this publication and contributed to the data modeling as well as the pattern mining components. The paper is a joint effort of Rita Sevastjanova, Fabian Sperrle, Rebecca Kehlbeck, Jürgen Bernard, Mennatallah El-Assady, and myself.

label. This threshold has been, again, determined through various experiments and discussions with linguistic experts.

Another interestingness measure used in this use case is the *window constraint*. According to linguistic experts, any rule with more than five tokens in between the itemsets is not descriptive enough to be useful. Therefore, the maximum gap between two itemsets in any rule must never be greater than five.

We also only consider closed patterns (see section 2.3) in this use case. This reduces the result set even more and shows fewer overlapping rules. Because the linguistic experts favor larger rules (with more items), we favor closed patterns over generator patterns in this approach.

3.4.3 Multidimensional Pattern Mining

Task Description

[9] 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 person, such as gender, age, and diabetes type. The researcher is interested in finding significant patterns for all the patients and within specific groups of patients (called cohorts), for example, only female patients older than 80. 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 many attributes are involved since the possible amount of filter settings is exponential. Furthermore, comparing 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 analyzes customers and their market baskets in combination with attributes describing the customers. A pharmaceutical researcher also analyzes 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 patient's 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 [51], 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 [52] can be mined, and association rules [140]. For sequences, sub-sequences can be mined, better known as sequential patterns, but it is possible to mine for episodes or sequential rules [73]. While many pattern mining algorithms for various structured data types are available, a standard pattern mining algorithm cannot identify significant patterns in subspaces [141].

9: This section is taken from my publication "Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data" (Section 1: Introduction) [10]. I have been the main author of this publication and section and have written all the contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.

[51]: (2014), Frequent Pattern Mining

[52]: Fournier-Viger et al. (2017), A survey of itemset mining

[140]: Zhang et al. (2010), Survey on association rules mining algorithms

[73]: Fournier-Viger et al. (2017), A survey of sequential pattern mining

[141]: Pinto et al. (2001), Multi-Dimensional Sequential Pattern Mining

The use case of the medical research 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 [142]. 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 [143]. 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.

[142]: Kriegel et al. (2012), Subspace clustering

[143]: Findlater et al. (2003), Iceberg-cube algorithms: An empirical evaluation on synthetic and real data

Subspace clustering algorithms focus mainly on numerical data where distances between the data points are calculated. Some algorithms also focus on categorical data. However, they suffer from scalability issues [144]. Itemset mining can also be considered a form of subspace clustering where the itemsets form the attributes and 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 interestingness measures if the algorithm supports them. Such a parameter estimation is difficult, if not impossible, in an exploratory data analysis scenario.

[144]: Gan et al. (2004), Subspace clustering for high dimensional categorical data

Limitations of existing work

[10] 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 numerical dimensions instead of only categorical ones.

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-

[10] This section is taken from my publication “Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data” (Section 3: Related Work) [10]. I have been the main author of this publication and section and have written all the contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.

defined subspaces as multidimensional sequential pattern mining [141]. 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 [145]. Pinto et al. [141] 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).

[141]: Pinto et al. (2001), Multi-Dimensional Sequential Pattern Mining

[145]: Grahne et al. (2001), On Dual Mining: From Patterns to Circumstances, and Back

[141]: Pinto et al. (2001), Multi-Dimensional Sequential Pattern Mining

UNISEQ Approach UNISEQ is not an acronym but merely a name that stands for the embedding of multidimensional discrete attributes in 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 [146] 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.

[146]: Pei et al. (2001), PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth

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 [143] and a sequential pattern mining algorithm. For this approach, the attribute characteristics are mined with the Bottom-up cube (BUC) algorithm [147] 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 3.4.3. However, the automated BUC algorithm is exchanged with the manual labor defining the various filter settings for the attributes.

[143]: Findlater et al. (2003), Iceberg-cube algorithms: An empirical evaluation on synthetic and real data

[147]: Beyer et al. (1999), Bottom-Up Computation of Sparse and Iceberg CUBEs

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 [148]. Closed pattern mining removes redundant patterns that hold the same information [51]. We do not apply this concept, as we exploit this redundancy to highlight subspaces.

[148]: Songram et al. (2006), Closed Multidimensional Sequential Pattern Mining

[51]: (2014), Frequent Pattern Mining

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 [149]. This effectively shows

[149]: Ankerst et al. (2008), DataJewel: Integrating Visualization with Temporal Data Mining

the distribution of events for multiple days but is limited to a single attribute. Tools, such as *Lifelines2* [150], *Outflow* [151], *EventFlow* [152], *DecisionFlow* [153], and *EventPad* [154] 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 [153]. *UpSet* by Lex et al. [155], 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 [156], higher scalability can be achieved with matrix-like visualizations which is exactly the approach we are following.

[150]: Wang et al. (2009), Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison

[151]: Wongsuphasawat et al. (2012), Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization

[152]: Monroe et al. (2013), Temporal Event Sequence Simplification

[153]: Gotz et al. (2014), DecisionFlow: Visual Analytics for High-Dimensional Temporal Event Sequence Data

[154]: Cappers et al. (2018), Exploring Multivariate Event Sequences Using Rules, Aggregations, and Selections

[153]: Gotz et al. (2014), DecisionFlow: Visual Analytics for High-Dimensional Temporal Event Sequence Data

[155]: Lex et al. (2014), UpSet: Visualization of Intersecting Sets

[156]: Lex et al. (2014), Points of view: Sets and intersections

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 [157]. 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 [158]. Stolper et al. extend this approach in their *Progressive Insights* system by allowing the user to interactively prioritize and prune the search space

[157]: Vrotsou et al. (2009), ActiviTree: Interactive Visual Exploration of Sequences in Event-Based Data Using Graph Similarity

[158]: Perer et al. (2014), Frequence: interactive mining and visualization of temporal frequent event sequences

of the underlying mining algorithm [159]. 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 [8]. Di Bartolomeo et al. recently published an intuitive and elegant design allowing the combination and exploration of event sequences with *one* attribute [160]. However, the design lacks scalability with larger alphabets as it uses the shortest common subsequence and cannot be easily applied to other types of structured data.

[159]: Stolper et al. (2014), Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics

[8]: Jentner et al. (2018), Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool

[160]: Bartolomeo et al. (2021), Sequence Braiding: Visual Overviews of Temporal Event Sequences and Attributes

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.

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 [161] are capable of providing relevant subspaces while removing redundancy but do not consider the domain knowledge of the user. The dissertation of Stephan Günnemann [162] 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 [163] 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 increases. 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 [164]

[161]: Parsons et al. (2004), Subspace clustering for high dimensional data: a review

[162]: Günnemann (2012), Subspace clustering for complex data

[163]: Baumgartner et al. (2004), Subspace Selection for Clustering High-Dimensional Data

[164]: Inselberg et al. (1990), Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry

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 [165]. 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. [166] 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 that 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. [167] and EvoSets [168] are tools to track subspaces and their effect onto 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 [169] 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 [170]. 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"[11] to describe such an abundance of filter options in dashboards. Such cross-filter approaches follow the “common approach" (see section 3.4.3), which implies an exponential amount of filter settings to reveal underlying commonalities in the data. A user likely 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

[165]: Jäckle et al. (2017), Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces

[166]: Tatu et al. (2012), Subspace search and visualization to make sense of alternative clusterings in high-dimensional data

[167]: Lehmann et al. (2016), Optimal Sets of Projections of High-Dimensional Data

[168]: Sun et al. (2022), EvoSets: Tracking the Sensitivity of Dimensionality Reduction Results Across Subspaces

[169]: Blumenschein et al. (2018), SMARTexplore: Simplifying High-Dimensional Data Analysis through a Table-Based Visual Analytics Approach

[170]: Behrisch et al. (2019), Commercial Visual Analytics Systems-Advances in the Big Data Analytics Field

[11] 11: https://medium.com/visualizing-the-field/there-be-dragons-dataviz-in-the-industry-652e712394a0

Table 3.2: The input comprised of structured data and discrete attributes. The data represents customers and their transactions modeled as itemsets. The attributes provide additional information for each customer.

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

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-visualizations where a limited number of perspectives and some additional metrics allow the user to explore all subspaces to be found in the structured data.

Description of Interestingness Measures

[12] This section describes what type of data we expect as input and how it is 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 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 chose this example due to its simplicity and because itemsets have the smallest search space compared to other structured data types. All of the following can be generalized to any 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 show the a-priori property of co-occurrences, which is why our approach works. The section closes by describing the achieved search-space reduction.

12: This section is taken from my publication "Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data" (Section 4: Multi-dimensional Pattern Exploration Approach) [10]. I have been the main author of this publication and section and have written all the contents. My co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim reviewed the paper internally.

Table 3.3: The input data transformed into explicit encoding. 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.

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

Input The input is shown in Table 3.2. 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 customer information, such as gender, age group, and country. Each row refers to a *transaction* of a customer referred to by the ID.

The input must generally 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 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 ($|\Sigma| = |\{bread, candy, juice, milk, soda, vegetables\}| = 6$).

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 3.3 displays the transformed data. The transformation affects only the attribute information. The structured data remains untouched. Each attribute characteristics 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 that the values are dependent and cannot be independently true simultaneously. 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 not to 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 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 as it is commonly used in an $Attribute \rightarrow Struct$ approach, this equals 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 of explicitly encoding null values, such as missing ones occurring in the attributes. A missing value can be added as its 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 is 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

Table 3.4: The data is aggregated by rows where the structured data is equal, yielding a table of distinct structured data entities. The co-occurrences are added. The IDs are propagated, allowing a back-reference to the original data table.

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

could be modeled as probabilities across multiple attribute characteristics.

Process The previous transformation and treatment of the values as co-occurrences is the foundation of the MDPE-approach allowing the data's row- and column-wise aggregations. We opt for row-based aggregations only, as detailed later in Section 12. 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. 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 structured data types are typically an extension and consist of multiple itemsets. For example, sequences are itemsets that occur in a sequence. Table 3.4 shows the resulting aggregated information. Only the itemset $\{\text{bread}, \text{candy}, \text{soda}\}$ occurs twice in the input data (see Table 3.2) in rows 2 and 4. The respective co-occurrences are added. The table, for example, now clearly shows that all customers are buying $\{\text{bread}, \text{candy}, \text{soda}\}$ exclusively (without any other item) in the *AgeGroup* of “≤ 18". This form of aggregation yields a table that enlists all distinct structured data entities in the input data. All structured data entities of the input may be equal, resulting 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 a little or no reduction of rows. We discuss this further in Section 12.

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 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 3.7 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 [171], ECIaT [172]) is being employed without any additional termination criteria. The resulting figure is comparable to a Hasse diagram [48]. 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 3.2). 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 3.4). Using Figure 3.7, several observations can be made:

[171]: Han et al. (2000), Mining Frequent Patterns without Candidate Generation

[172]: Zaki (2000), Scalable Algorithms for Association Mining

[48]: authors (2021), Hasse diagram

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

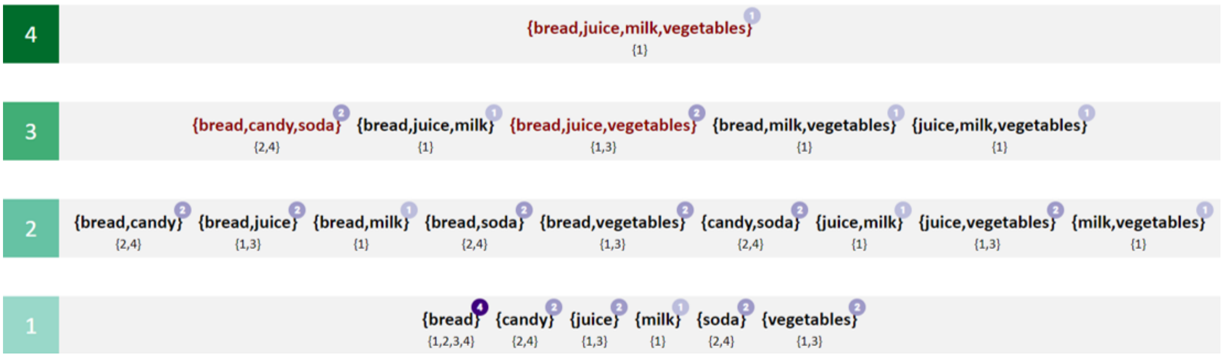


Figure 3.7: The search space of an itemset mining algorithm visualized as a Hasse diagram [48]. The length (i.e., cardinality) is encoded in the vertical position. The support is denoted in purple. The red itemsets occur in the input data.

data (Table 3.2) 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 12.

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 $\{\text{candy}\}$, $\{\text{soda}\}$, $\{\text{bread}, \text{candy}\}$, $\{\text{bread}, \text{soda}\}$, $\{\text{candy}, \text{soda}\}$, and $\{\text{bread}, \text{candy}, \text{soda}\}$ 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 3.2) and $sup_T I$ be the function for the support. The a priori property states that:

$$\forall I : \forall J \sqsupseteq I : sup_T J \leq sup_T I \tag{3.1}$$

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:

$$\forall I : \forall J \sqsupset I : |J| > |I| \tag{3.2}$$

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 3.4). The rows, specifically the transactions, of this table match the itemsets depicted in red in Figure 3.7. 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 [173]. Observation 4 states that the itemsets occurring of the transaction data (depicted in red in Figure 3.7) occur mostly at the top of this search space and are already covered by the low aggregation table (Table 3.4). 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.

[173]: Agrawal et al. (1996), Fast Discovery of Association Rules

Action 2: Search Space Reduction by length This follows the first action and results from 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 occurs for higher supports. Observation 1 concludes that the highest number of itemsets are typical 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

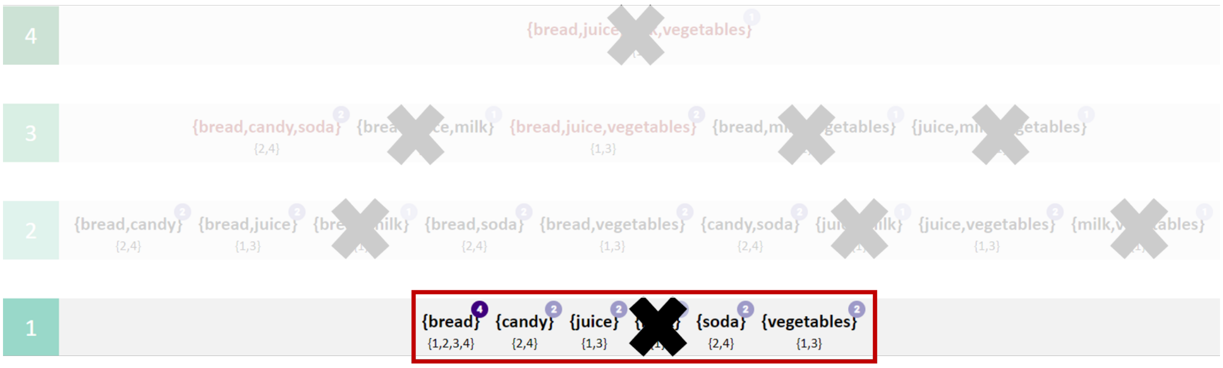


Figure 3.8: The search space is pruned with a minimum support of *two* (Action 1) depicted by the crossed out itemsets. A second pruning step by length (cardinality) with a parameter setting of the *Initial Mining Depth* of *one* drastically reduces the size of the search space (Action 2). The remaining sub-itemsets are highlighted in the red box.

edge cases where a deviation of this default value is necessary.

Action 3: Interactive Mining Figure 3.8 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 12. To further mitigate this, an interactive mining technique can be used by using a user-defined selection of the already mined patterns and mining for the patterns of the next, higher generation obeying the partial order. For example, the user might select the patterns $\{\text{candy}\}$ and $\{\text{soda}\}$ 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 $\{\text{bread}, \text{candy}\}$, $\{\text{bread}, \text{soda}\}$, and $\{\text{candy}, \text{soda}\}$ (see Figure 3.7 or Figure 3.8, 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 5.5.3.

Table 3.5: The data is aggregated by rows where the original transaction contains a pattern. The co-occurrences are added. The IDs are propagated, allowing a back-reference to the original data table (Table 3.2).

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 remaining patterns are used to generate a table analog to the low aggregation table (Table 3.4). We call this table high aggregation table (see Table 3.5). 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 (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 3.3).

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 regarded as aggregations of the original transactions (aggregations of rows). The tables are derived from the initial transformation of the data into an explicit encoding (see Table 3.3 and Section 12, respectively). We name the first generated table *low aggregation table* (Table 3.4), 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 3.7).

The second table is called *high aggregation table* (Table 3.5) 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 12). The high aggregation table covers the bottom of the search space (see Figure 3.8).

Table 3.6: This output is optional and represents a vector of co-occurrences which are aggregated using all of the data. This vector can then be subtracted from the other vectors, which is used by some normalizations (see Section 12).

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

Table 3.7: The input data of Table 3.2 modeled by the *UniStruct* approach. Every attribute characteristics is treated as an item and merged with the itemset of the structured data.

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}

A third optional output is generated, aggregating the co-occurrences of all data into a single vector. This is represented in Table 3.6. This vector can then be subtracted from other output vectors to calculate a divergence because, unlike 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 12.

Co-Occurrences are A-Priori We have previously described that the IMs support and *length* are a-priori. The same is true for the co-occurrences and is 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 show 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 3.7). This can be done for all of the transactions, and a standard pattern mining algorithm can be applied. The problem is that the algorithm now has two combined, exponential search spaces, 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 3.7), and c be a defined attribute characteristic. Let further be $cooc_S I, c$ 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 3.5). Let $sup_S I$ be the function evaluating the support of a pattern. Using the *UniStruct* approach, we can show that:

$$sup_S I \cup c = cooc_S I, c \tag{3.3}$$

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 $I \cup c$: $\{\text{candy}, \text{AgeGroup:}{\leq}18\}$. This itemset is a subset of the transactions 2 and 4, thus, $sup_S\{\text{candy}, \text{AgeGroup:}{\leq}18\} = 2$. Because of Equation 3.1 and Equation 3.3 we can conclude that:

$$\forall I : \forall J \sqsupseteq I : cooc_S J, c \leq cooc_S I, c \tag{3.4}$$

This means that a co-occurrence value of a pattern in the high aggregation table (Table 3.5) 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 3.4). Because these two tables cover the boundaries of the search space (see Figure 3.7 & Figure 3.8), 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. Let I be a distinct transaction of the low aggregation table (Table 3.4), K be a pattern of the high aggregation table (Table 3.5), J be any possible pattern occurring in the search space, and c be a fixed attribute characteristic. It follows that:

$$\forall J : \forall I \sqsubseteq J \sqsubseteq \forall K : cooc_S I, c \leq cooc_S J, c \leq cooc_S K, c \tag{3.5}$$

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 3.2) and n' be the number of distinct transactions (Table 3.4). 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 gx and $g_k x$ where gx 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 structured data types, such as sequences and graphs, allow more combinations of structured entities due to their structure. As discussed in Section 12, 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:

$$g|\Sigma| * 2^m = 2^{|\Sigma| m} \tag{3.6}$$

which can also be 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 3.4). The second reason is the partial order that occurs when various attribute characteristics are combined. The partial order is visible in Figure 3.7, 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:

$$g|\Sigma| * m = 2^{|\Sigma|} * m \tag{3.7}$$

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 3.4 & 3.5). As described in Section 12, actions 1 and 2 reduce the search space of the structured mining, which is equivalent to the rows of the tables. Therefore, it is now required to determine the maximum number of rows possible for each table. The maximum number of rows for the low aggregation table is 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 3.2):

$$\mathcal{O}n‘ = minn, g|\Sigma| \tag{3.8}$$

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 unlimited 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 amount 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:

$$\mathcal{O}z = \mathcal{O}\sum_{x=1}^{d} g_x|\Sigma| = \mathcal{O}\sum_{x=1}^{d} \binom{|\Sigma|}{x} \tag{3.9}$$

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.

Additional variations based on the Interestingness Measures

[13] The co-occurrence values can be normalized in various ways to highlight different aspects of the data. Hence, from

[13] This section is taken from my publication “Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data” (Section 5.2: Normalization) [10]. I have been the main author of this publication and section and have written all the contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.

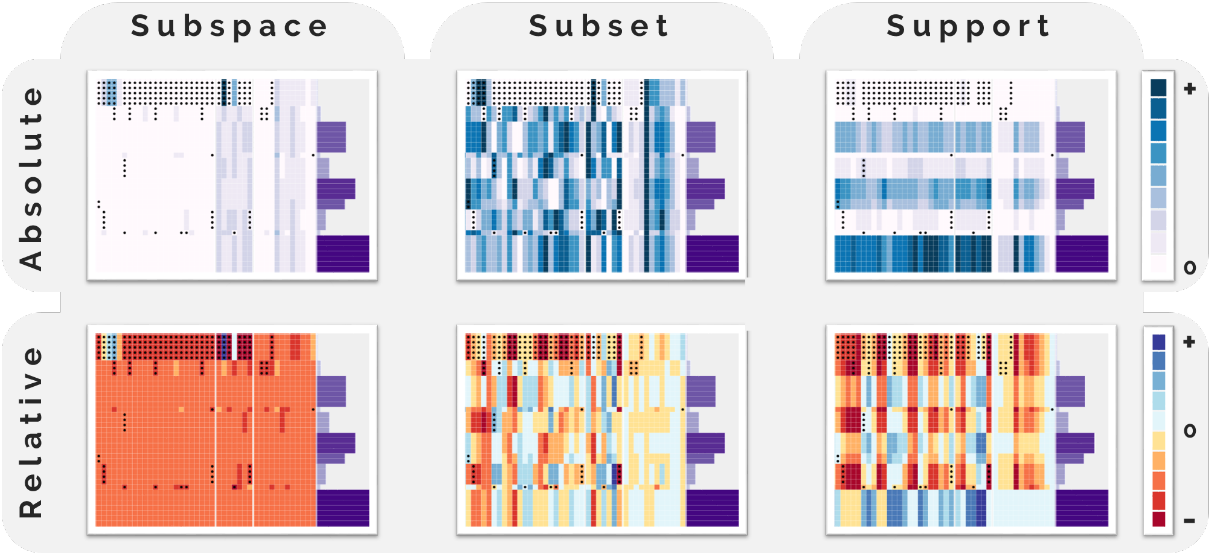


Figure 3.9: Six distinct perspectives on the same data. The subspace view highlights attribute characteristics (columns). The subset view highlights the visible blocks. The support view correlates with the support of the aggregation, which is noticeable by the corresponding purple bar chart. The relative views show the difference in comparison to the population.

a user's point of view, these normalizations offer various perspectives on the data. Our system supports six different perspectives as shown in Figure 3.9. 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 in the populations' distributions compared to the subsets (rows) which is done by subtracting the vectors of the output tables (Table 3.4 & Table 3.5) by the global vector (Table 3.6). 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 [174]. 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 with a small

[174]: Harrower et al. (2003), ColorBrewer. org: an online tool for selecting colour schemes for maps

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 a no co-occurrences (co-occurrence = 0). If the value is around 0, the values are independent. And for 1, they are correlated.

3.5 Conclusions

The point of this chapter is not to argue for abolishing the idea of interestingness measures altogether but to acknowledge their limitations in that an interestingness measure and its value highly depends on the data, user, and task. I, therefore, offer a complementary perspective for the existing surveys and taxonomies by arguing that an interestingness measure is, at its core, a function to generate a feature that represents some properties of a pattern in a quantified form. We can then further distinguish what type of property the measure describes such as properties of the pattern itself or the cluster it represents. The cluster properties can be further divided into structure-based properties, statistical properties, and cluster-metadata-based properties. This categorization is independent of any user assessment of interestingness.

The use cases highlight that applications typically rely on multiple interestingness measures. This, on the other hand, makes it more difficult for the user to choose the optimal thresholds (parameter estimation). The fact that the user does not have exact knowledge about the correct parameters must be assumed in an EDA context. This chapter advocates for understanding that interestingness measures describe certain properties that are, ideally, correlating with the interestingness of a user for a certain task. This requires that the user has at least a conceptual understanding of what the interestingness measure represents. This will be further elaborated in section 5.6.

Visualization Techniques for Structured Data Patterns 4

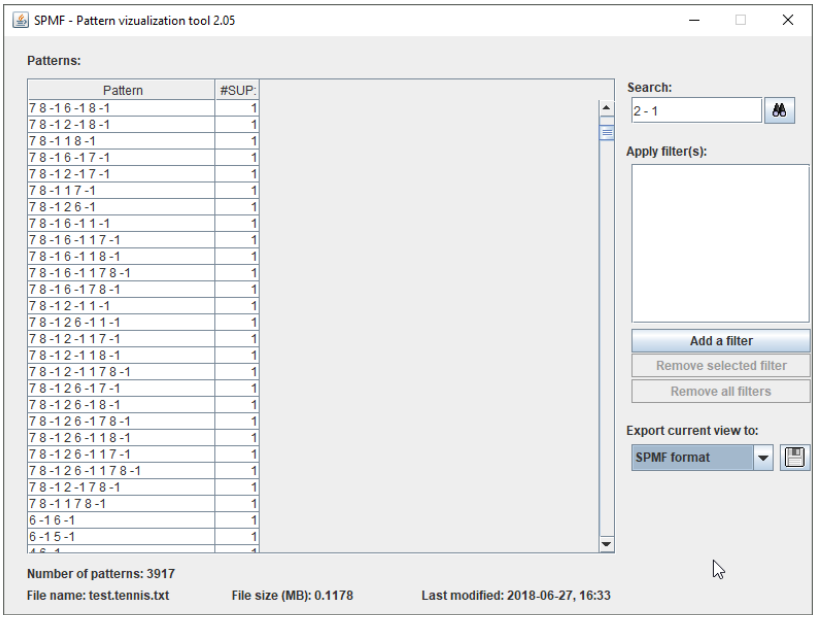
4.1 Introduction

Frequent pattern mining is an important concept in data mining and exploratory data analysis [51]. 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 [175]. It is of special interest how users can integrate their knowledge into the data analysis process and eventually generate more knowledge [94][1].

As frequent pattern mining tends to produce many patterns, a lot of research has been devoted to finding interestingness measures filtering and ranking useful and interesting patterns for the user [51, 64]. Multiple surveys cover the algorithmic side of the mining process for itemsets [52, 68, 176], association rules [70, 177], and sequential patterns [73, 75, 178]. Implementations of these algorithms can be found in libraries such as WEKA [179] or SPMF [180] or for example in the FIMI repository [181, 182]. 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 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 4.1). The viewer supports the interactive exploration of patterns by filtering the patterns by a given string (positive and negative templating), by their interestingness measures (e.g., greater than a

[1] This entire chapter is taken from my publication "Visualization and Visual Analytics Techniques for Patterns". I have been the main author of this publication and have written all the contents. The paper was internally reviewed by Florian Stoffel, Mennatallah El-Assady, and my co-author Daniel Keim.



SPMF - Pattern vizualization tool 2.05

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

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

Figure 4.1: The *Pattern Viewer* of the Sequential Pattern Mining Framework (SPMF).

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 view 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 [183] or works on time-oriented data visualization [184–186] which is related to visualizing event sequences and sequential patterns. Other papers feature an extensive related work section, e.g., [187]. Hahsler and Chelluboina provide a survey of visualization for association rules in their paper for the R-package arulesViz [188]. In this chapter, we contribute a survey that thoroughly surveys visualization techniques for each of the specializations and systematically analyzes their strengths and weaknesses.

[183]: Alsallakh et al. (2016), The State-of-the-Art of Set Visualization

[184]: Aigner et al. (2011), Visualization of Time-Oriented Data

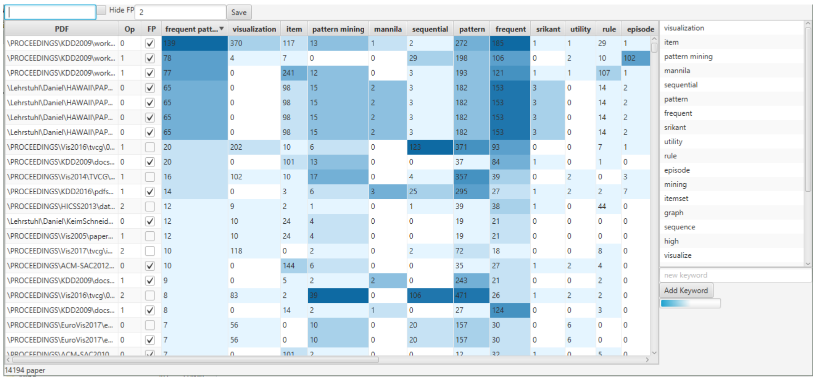
[185]: Silva et al. (2000), Visualization of Linear Time-Oriented Data: A Survey

[186]: Aigner et al. (2007), Visualizing time-oriented data - A systematic view

[187]: Hofmann et al. (2000), Visualizing association rules with interactive mosaic plots

[188]: Hahsler et al. (2011), Visualizing association rules: Introduction to the R-extension package arulesViz

The next Section explains our used methodology. Section 4.3 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 4.6 where we bridge the gap to utility mining and identify further research challenges.



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

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

Figure 4.2: Our prototype to scan a large set of academic work.

4.2 Methodology

We use our internal database which includes conference and workshop proceedings (e.g., Vis[2], EuroVis[3], KDD[4]), journals (e.g., TVCG[5], Information Visualization[6]), 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 4.2). 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, the ACM digital library, IEEE Xplore digital library, EG digital library, and DBLP computer science bibliography were used for keyword searches.

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 visually closely related episodes.

[2] http://ieeevis.org

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

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

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

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

4.3 Visualizations and Visual Analytics Techniques

Mining for itemsets, association rules, and sequential patterns has commonalities. This fact also holds for visualizing such patterns. This section will explain visualization techniques and further elaborate on how visual analytics is applied to 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.

4.3.1 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 the k-itemset. Borgelt provides a good introduction and overview of algorithms, data structures, and extensions [68].

[68]: Borgelt (2012), Frequent item set mining

Visualizing sets is a heavily researched topic. For a comprehensive state-of-the-art survey we refer to Alsallahk et al. [183] and their companion website SetViz[7]. In the following, we focus on work that is specifically developed for frequent itemsets.

[183]: Alsallakh et al. (2016), The State-of-the-Art of Set Visualization

7: http://setviz.net, accessed Feb. 2018

Lattice Representations

A representation of a lattice, or also concept hierarchy [189], 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 4.3a). 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 [78] or maximal [190].

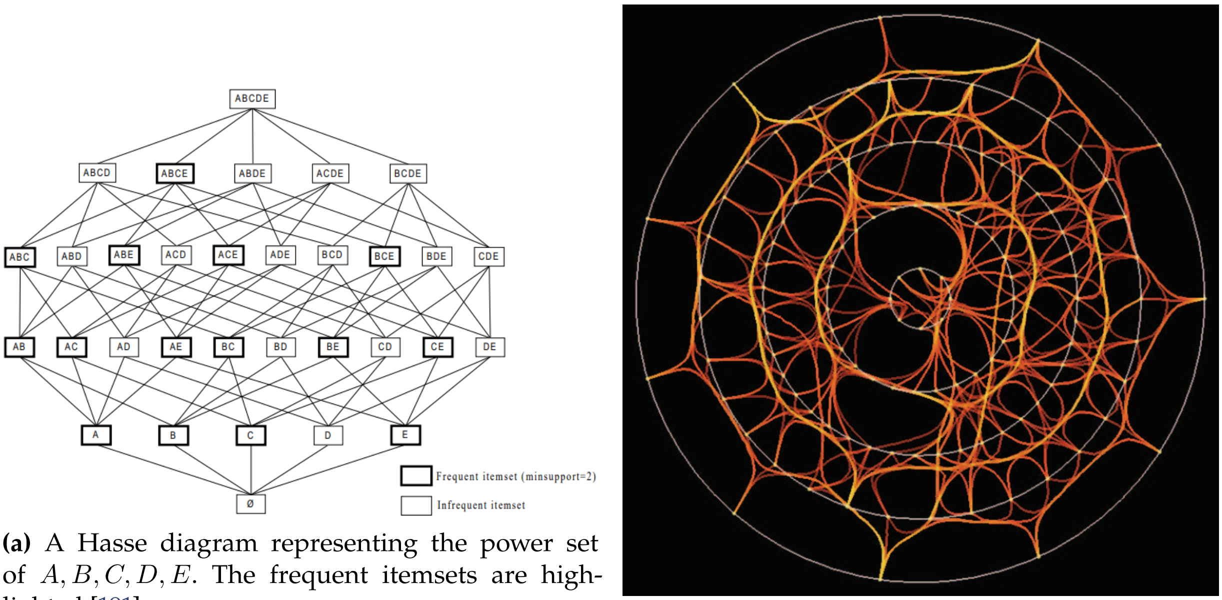
[189]: Han (1995), Mining Knowledge at Multiple Concept Levels

[78]: Zaki et al. (2002), CHARM: An Efficient Algorithm for Closed Itemset Mining

[190]: Burdick et al. (2001), MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases

Klemettinen et al. already discuss the problem of clutter in directed graphs (see Section 4.3.2) due to too many edges [83].

[83]: Klemettinen et al. (1994), Finding Interesting Rules from Large Sets of Discovered Association Rules



(a) A Hasse diagram representing the power set of A, B, C, D, E. The frequent itemsets are highlighted [191].

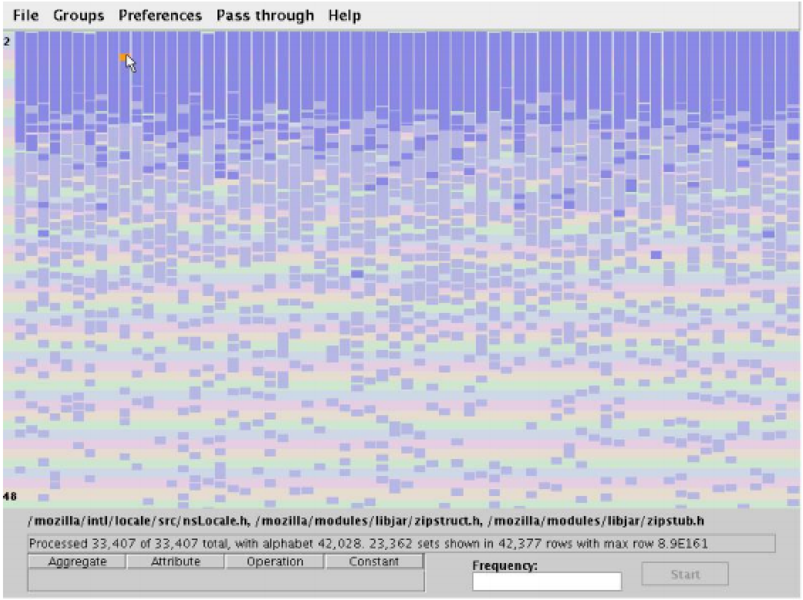
(b) A radial graph layout by Bothorel et al. [192]. Each frequent itemset is represented as a separate node. Infrequent itemsets are not shown in the graph.

Figure 4.3: Two lattice representations showing frequent itemsets and their relations.

The authors propose the Spiders technique [193] where multiple instances of one node are allowed. Bothorel et al. follow this idea and they display, similar to the *PowerSetViewer*, every frequent itemset distinctly [192] (Figure 4.3b). The graph does not show the power set but 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.

[193]: Collier (1987), Thoth-II: Hypertext with Explicit Semantics

[192]: Bothorel et al. (2013), Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm



/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

Figure 4.4: PowerSetViewer (PSV) by Munzner et al. [194]. Each cell represents one or more itemsets (indicated by their saturation). The itemsets are ordered vertically by their size and lexicographically in a horizontal direction.

Pixel Based Visualizations

Munzner et al. use the accordion drawing technique that features guaranteed visibility and a rubber sheet navigation [194] (Figure 4.4). 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 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 [194].

[194]: Munzner et al. (2005), Visual mining of power sets with large alphabets

[194]: Munzner et al. (2005), Visual mining of power sets with large alphabets

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 one or more k-itemsets. A total order of the items can be applied without the loss of generality. Thus, a prefix tree can be generated which is also known as FP-tree [196].

[196]: Han et al. (2004), Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach

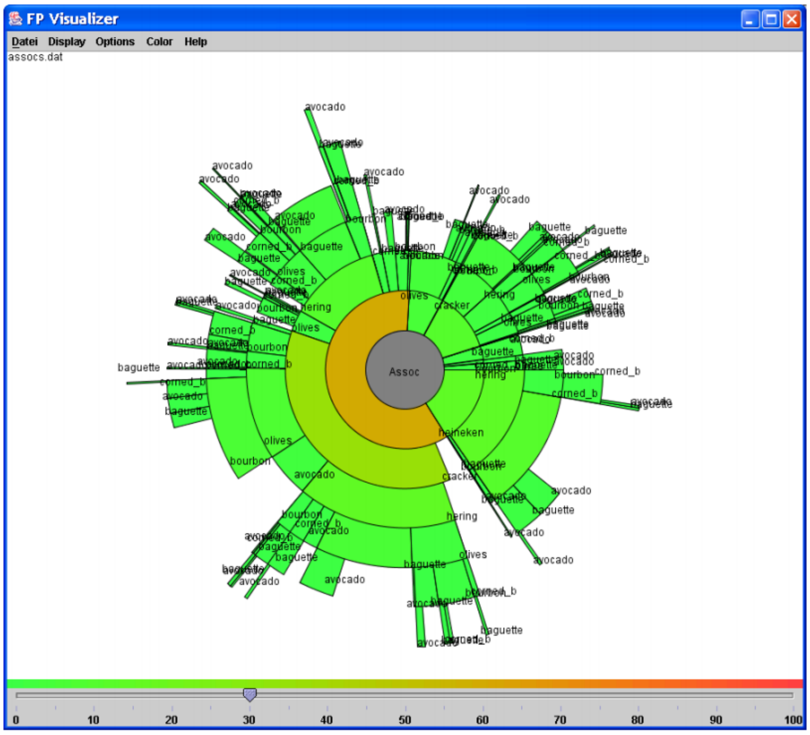


Figure 4.5: FP-Viz [195] by Keim et al. showing the FP-tree in a hierarchical visualization known as sunburst. Each ring represents a tree hierarchy and each segment displays one item. Itemsets can be read by neighboring segments from in- to outside.

The FP-Viz tool by Keim et al. [195] uses a sunburst [197] and interring [198] visualization technique to display the FP-tree [196]. Each circle segment represents a node of the FP-tree. The segments are ordered according to the tree hierarchy from the 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 [199]. In contrast to FP-Viz, the color is mapped onto the cardinality.

Leung et al. use FPMapViz [200] to visualize the hierarchy in a tree-map [201]. 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 [202] (Figure 4.6) 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 either as a trapezoidal block or triangle block depending on whether this item can be extended by another item or not. This impression

[195]: Keim et al. (2005), Fp-viz: Visual frequent pattern mining

[197]: Stasko et al. (2000), Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations

[198]: Yang et al. (2003), InterRing: a visual interface for navigating and manipulating hierarchies

[196]: Han et al. (2004), Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach

[199]: Leung et al. (2012), RadialViz: An Orientation-Free Frequent Pattern Visualizer

[200]: Leung et al. (2011), FpMapViz: A Space-Filling Visualization for Frequent Patterns

[201]: Shneiderman (1992), Tree Visualization with Tree-Maps: 2-d Space-Filling Approach

[202]: Leung et al. (2016), PyramidViz: Visual Analytics and Big Data Visualization for Frequent Patterns

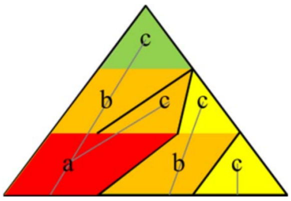


Figure 4.6: The PyramidViz [202] by Leung et al. shows items as blocks of a pyramid. Connecting the blocks as shown by the thin gray lines forms the itemsets. The color references the support.

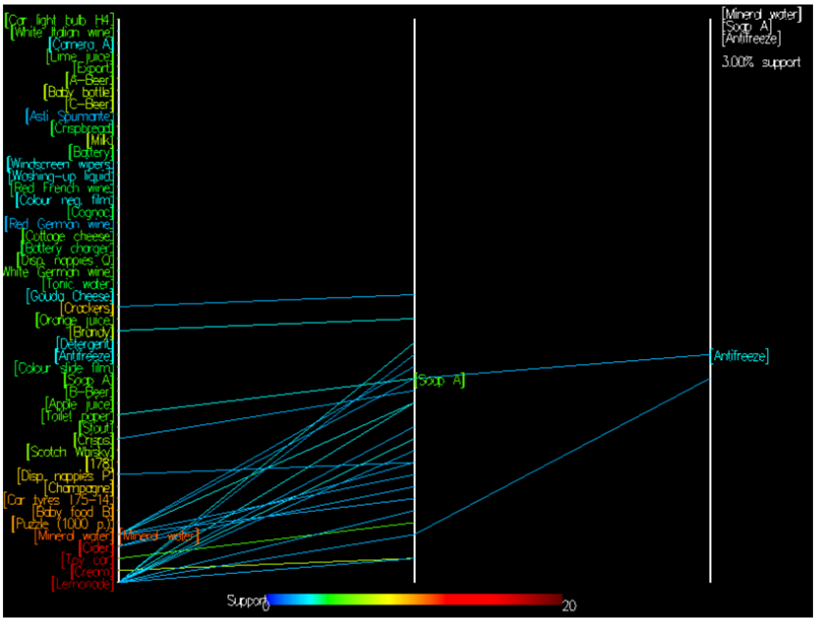


Figure 4.7: Frequent itemsets displayed as parallel coordinates by Yang [203]. The support is mapped to color.

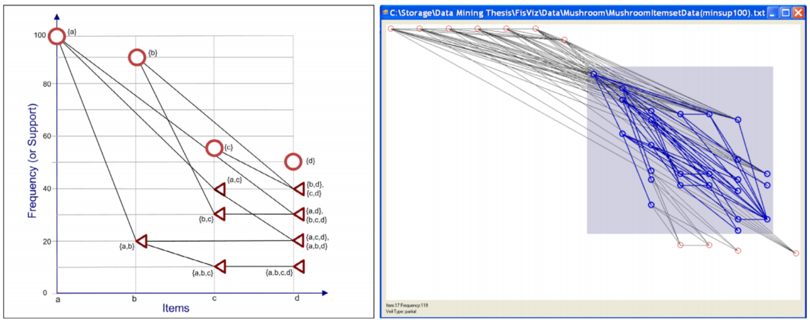
is enforced by more vertical black lines to separate the blocks. Additionally, the items are connected by grey lines. The color hue of the blocks is used to display the frequency information. The visualization technique allows insights into the decreasing support of supersets of itemsets which is known as the Apriori property [49].

[49]: Agrawal et al. (1994), Fast Algorithms for Mining Association Rules in Large Databases

Linear Visualizations

Yang visualizes frequent itemsets as parallel coordinates [203] (Figure 4.7). 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 that 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, and items within a group

[203]: Yang (2003), Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates



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

Figure 4.8: FIsViz [205] 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.

are sorted according to their frequency in descending order. This ensures positive slopes and reduces clutter. The support can either be mapped to the 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 [204].

[204]: Yang (2005), Pruning and Visualizing Generalized Association Rules in Parallel Coordinates

[205]: Leung et al. (2008), FIsViz: A Frequent Itemset Visualizer

Leung et al. use a different mapping of the axis in their tool FIsViz [205] (Figure 4.8). 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 indicating 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.

[206]: Leung et al. (2008), WiFIsViz: Effective Visualization of Frequent Itemsets

As a countermeasure Leung et al. propose WiFIsViz [206] 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 4.9, 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

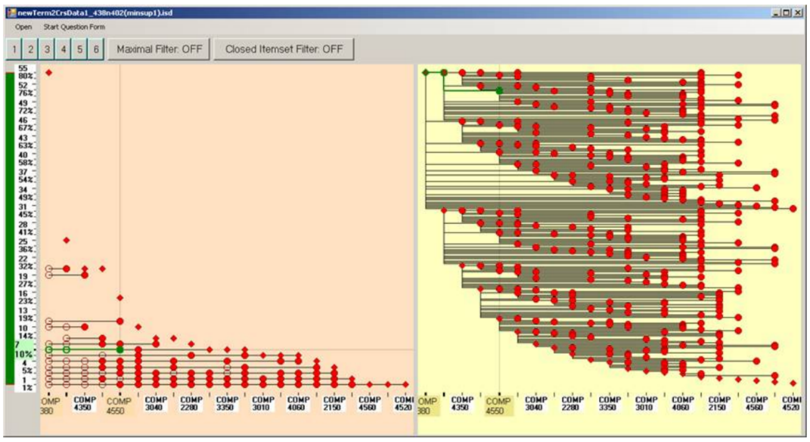
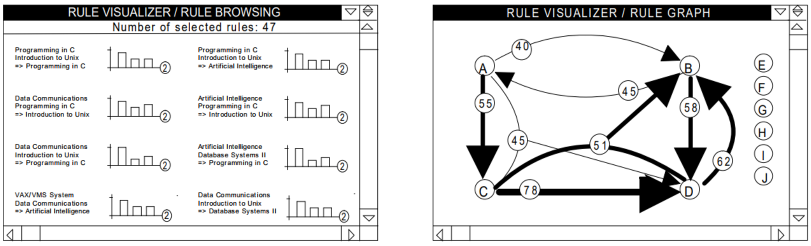


Figure 4.9: WiFIsViz collapses multiple itemsets into horizontal lines to reduce clutter [206].


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

Figure 4.10: Association Rule Visualizer by Klemettinen et al. [83]. The left shows a browsing view where every bar chart represents one rule. The right shows a directed graph where multiple association rules are displayed.

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 4.9. The wiring-type technique is also used in FpVAT [207] where it is combined with a raw data visualization module.

[207]: Leung et al. (2009), FpVAT: a visual analytic tool for supporting frequent pattern mining

4.3.2 Association Rules

Association rules [208] 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) 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.

Individual Representations

Klemettinen et al. developed *Rule Visualizer* a tool to visualize and explore association rules [83]. Figure 4.10 shows two views of the tool: a browsing view on the left and a rule graph on the right (see Section 4.3.2). In the browsing view, every association rule is represented as one bar chart. The left bar represents confidence, and the right bar the support. The central bar displays the commonness. The number of attributes on 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.

[83]: Klemettinen et al. (1994), Finding Interesting Rules from Large Sets of Discovered Association Rules

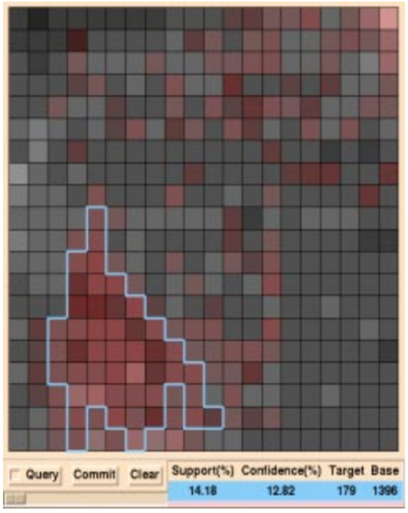
Directed Graph

The rule graph (Figure 4.10, right) is a *directed graph* and visualizes several association rules simultaneously [83]. 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) exclusion of association rules by removing items (as shown by the nodes E - J in Figure 4.10); (ii) inclusion of items using templates; (iii) by letting the user set a maximal rule size for the itemset of 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.

[83]: Klemettinen et al. (1994), Finding Interesting Rules from Large Sets of Discovered Association Rules

Han et al. [209] (DBMiner) and Rainsford et al. [210] 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

[210]: Rainsford et al. (2000), Visualisation of Temporal Interval Association Rules



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

Figure 4.11: A matrix-based visualization for two-dimensional numerical association rules by Fukuda et al. [211].

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.

Matrix Views

Fukuda et al. are visualizing two-dimensional numerical association rules (e.g., $XY \rightarrow Z$) using a matrix-based visualization [211]. In this case, the left-hand side of the association rule is not based on binary features but on numerical attributes (e.g., age). With an equal-sized binning, the numerical features are discretized and can be mapped on a two-dimensional grid (see Figure 4.11). 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.

[211]: Fukuda et al. (1996), Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization

AViz by Han et al. [212] extends the two-dimensional model to a three-dimensional space.

[212]: Han et al. (2000), AViz: A Visualization System for Discovering Numeric Association Rules

Commercial tools like IBM QUEST [213][8] 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

[213]: Agrawal et al. (1996), The Quest Data Mining System

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

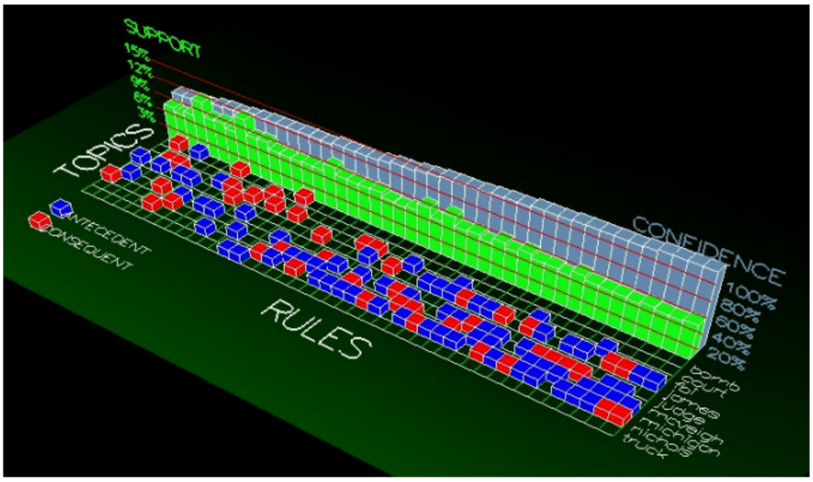


Figure 4.12: A 3D table visualization by Wong et al. [215] representing each association rule as a separate column. The alphabet of items is shown as rows. Additionally, confidence and support for each association rule are provided.

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 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[9] [214] 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 confidence. The color shows 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.

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

[214]: Brunk et al. (1997), MineSet: An Integrated System for Data Mining

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 [215]. Another general problem with 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

[215]: Wong et al. (1999), Visualizing Association Rules for Text Mining

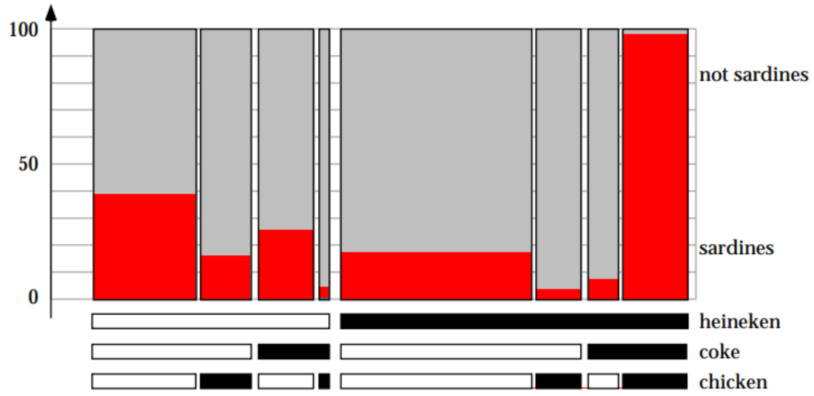


Figure 4.13: Mosaic Plot of multiple association rules by Hofmann et al. [187].

rows represent one single item (Figure 4.12). In the back, two bar charts provide the support and confidence properties for each rule. Figure 4.12 shows, for example, the rule $james$ & $michigan \to nichols$ on the rightmost column (next to the labels of the items) with the confidence of 100% and 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 highlighting the respective columns and rows. Lee et al. use the same technique to visualize association rules based on geo-tagged photos [216].

[216]: Lee et al. (2013), Mining Points-of-Interest Association Rules from Geo-tagged Photos

Mosaic Plots

Hofmann et al. build interactive Mosaic plots [187, 217], called Double Decker plots, based on contingency tables created from association rules enabling the user understanding the underlying structure [187]. Figure 4.13 shows such a plot for the rule

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

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

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

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 parts combined). The red highlighting in each bar shows the confidence measure. The

[187]: Hofmann et al. (2000), Visualizing association rules with interactive mosaic plots

[217]: Hartigan et al. (1981), Mosaics for contingency tables

[187]: Hofmann et al. (2000), Visualizing association rules with interactive mosaic plots

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.

Linear Visualizations

Yang's idea to visualize itemsets in a parallel coordinate plot works as well for association rules [203]. 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 still held true (see Section 7). 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.

[203]: Yang (2003), Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates

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 [218]. 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 4.14) separates four different types of identified rules: conforming rules, unexpected condition rules, unexpected consequent

[218]: Liu et al. (2000), Analyzing the Subjective Interestingness of Association Rules

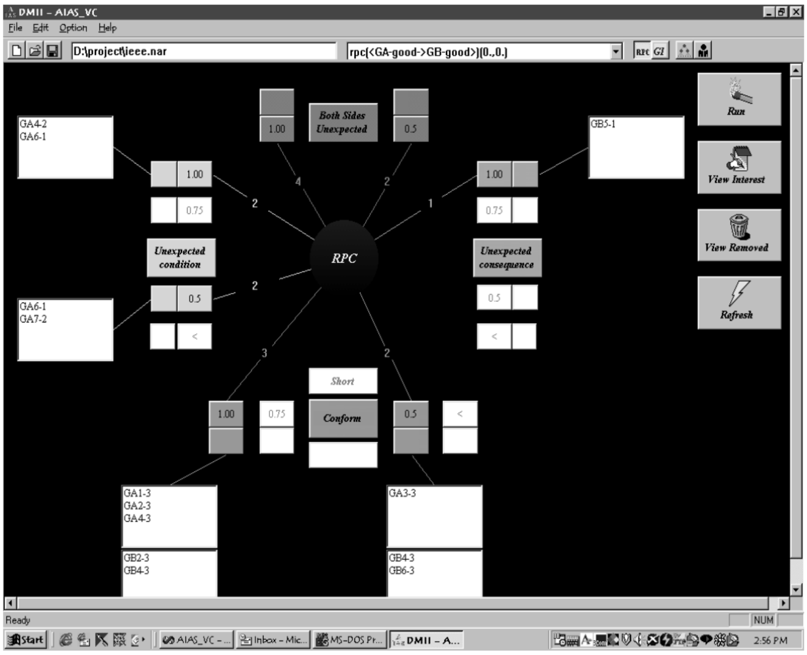


Figure 4.14: The interestingness analysis system separates four different types of association rules based on their potential interestingness [218].

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.

4.3.3 Sequential Patterns

[116]: Agrawal et al. (1995), Mining Sequential Patterns

[51]: (2014), Frequent Pattern Mining

Mining sequential patterns [116] describes an extension to frequent itemset mining where subsequences of a given sequence database are discovered [51]. In contrast to frequent itemsets and association rules, the order is defined in the data which puts limitations on 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 that defines their order and occurrence in time. In application areas such as DNA sequences, weblogs, and click streams [178] there are most of the time only 1-itemsets involved which simplifies visualizations. Sequential patterns are also useful for mining trajectories [219] and find common patterns in movement [220]. In general, a pattern can be viewed as a prototype of multiple event sequences and, therefore, be visualized in such a manner. In a geospatial context, this allows, for example, to identify common travel routes in traffic.

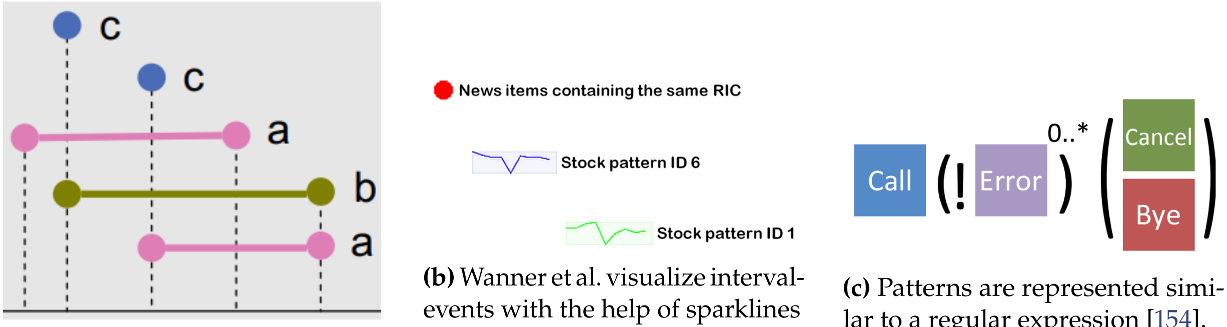
[178]: Mabroukeh et al. (2010), A taxonomy of sequential pattern mining algorithms

[219]: Giannotti et al. (2007), Trajectory pattern mining

[220]: Andrienko et al. (2013), Visual Analytics of Movement

This section focuses on visualization techniques specifically for sequential patterns. For a broader view of time-oriented data, we refer to the book of Aigner et al. [184] and other

[184]: Aigner et al. (2011), Visualization of Time-Oriented Data



(a) A representation for interval and point event patterns by Shin et al. [222].

(b) Wanner et al. visualize interval-events with the help of sparklines generated by a SOM [25].

(c) Patterns are represented similar to a regular expression [154].

Figure 4.15: 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 the creation of queries and rules (c).

surveys [185, 186].

[185]: Silva et al. (2000), Visualization of Linear Time-Oriented Data: A Survey
[186]: Aigner et al. (2007), Visualizing time-oriented data - A systematic view

Individual Representations

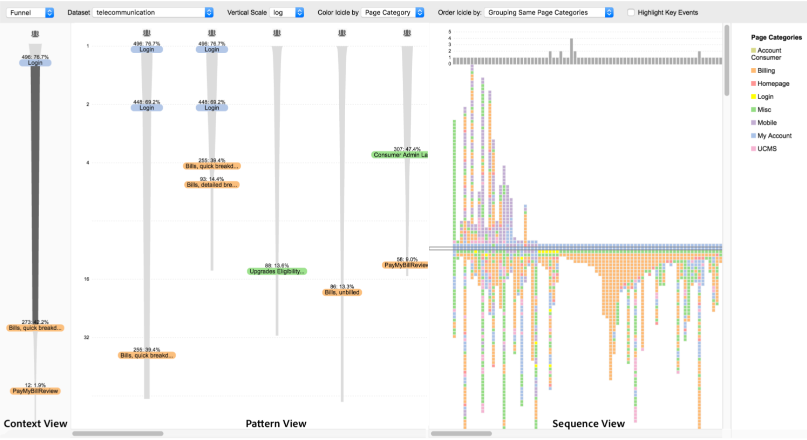
Patterns can be represented in the form of text, for example, $\langle\{a\},\{b,c\}\rangle$ [221] where an item a occurs before items b and c which occur at the same time. In case of interval events that do not only occur at a specific point in time but rather over a given period with a defined start and ending point, Shin et al. [222] 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.

[221]: Fournier-Viger et al. (2014), Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information

[222]: Wu et al. (2009), Discovering hybrid temporal patterns from sequences consisting of point- and interval-based events

Shin et al. also present another representation that is more intuitively comprehensible (Figure 4.15a). 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 allow visualizing additional information for an event [25] (Figure 4.15b). In the case of a larger alphabet, the visual

[25]: Wanner et al. (2016), Integrated visual analysis of patterns in time series and text data - Workflow and application to financial data analysis



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

Figure 4.16: Maximal sequential patterns from clickstream data are visualized on the left. The right side shows an aligned sequence view based on a selected event [223].

[29]: Jentner et al. (2017), Feature Alignment for the Analysis of Verbatim Text Transcripts

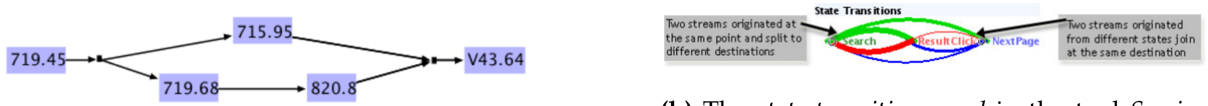
representation does not scale well regarding understandability [29]. Furthermore, as a single pattern representation, it is not desirable to visualize many patterns at the same time. In none of these representations, additional properties such as the support are mapped. This can be added for example 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 they might distort colors and hinder the user from comparing different patterns.

[154]: Cappers et al. (2018), Exploring Multivariate Event Sequences Using Rules, Aggregations, and Selections

Cappers et al. use a representation that is very similar to regular expressions [154] (Figure 4.15c). This is especially powerful for creating rules and queries to simplify and filter a large set of rules. It also allows for simplifying 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.

[223]: Liu et al. (2017), Patterns and Sequences: Interactive Exploration of Clickstreams to Understand Common Visitor Paths

Liu et al. mine for maximal sequential patterns in clickstream data [223]. As shown in Figure 4.16, 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 for each event. The view on the right visualizes each sequence individually. Clicking


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. [230] 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 [225].

Figure 4.17: Two node-link visualizations that use a state transition analogy to display and aggregate multiple sequential patterns.

on an event of a pattern will align all sequences by that event. The 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 [224] resembles. It is noteworthy to mention that icicle plots are quite popular for visualizing event sequences [225–228] and that aligning sequences leads to a better overview for comparison [152, 154, 223, 225]

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. Transitions or connections 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 [229]. 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 [230]. The pattern display is laid out horizontally and can split up into different events that might converge later on (Figure 4.17a).

Lam et al. also use the idea of a state transition representation [225] in their tool *Session Viewer* (Figure 4.17b). For logs, such state transition representations are well known

[224]: Kruskal et al. (1983), Icicle plots: Better displays for hierarchical clustering

[225]: Lam et al. (2007), Session Viewer: Visual Exploratory Analysis of Web Session Logs

[226]: Wongsuphasawat et al. (2011), LifeFlow: visualizing an overview of event sequences

[227]: Wongsuphasawat et al. (2014), Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at Twitter

[228]: Bernard et al. (2015), A Visual-Interactive System for Prostate Cancer Cohort Analysis

[152]: Monroe et al. (2013), Temporal Event Sequence Simplification

[154]: Cappers et al. (2018), Exploring Multivariate Event Sequences Using Rules, Aggregations, and Selections

[223]: Liu et al. (2017), Patterns and Sequences: Interactive Exploration of Clickstreams to Understand Common Visitor Paths

[225]: Lam et al. (2007), Session Viewer: Visual Exploratory Analysis of Web Session Logs

[229]: Mannila et al. (2000), Global partial orders from sequential data

[230]: Patnaik et al. (2011), Experiences with mining temporal event sequences from electronic medical records: initial successes and some challenges

[225]: Lam et al. (2007), Session Viewer: Visual Exploratory Analysis of Web Session Logs

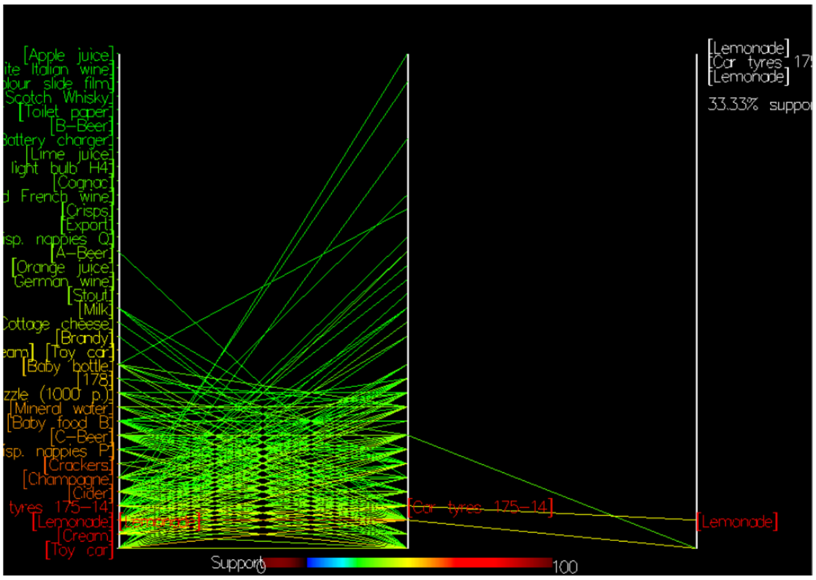


Figure 4.18: A parallel coordinates plot to visualize sequential patterns [203].

especially when Markov models are being used [231]. The width of the arcs represents the frequency of the respective transition.

[231]: Guzdial et al. (1993), Characterizing process change using log file data

Hu et al. mine sequential patterns based on words from tweets and visualize the words in a node-link diagram [232]. In their tool *SentenTree*, the nodes (words) have different sizes and colors representing their frequencies (double-encoded).

[232]: Hu et al. (2017), Visualizing Social Media Content with SentenTree

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 [233].

[233]: Bernard et al. (2013), MotionExplorer: Exploratory Search in Human Motion Capture Data Based on Hierarchical Aggregation

Parallel coordinates are also suitable for sequential patterns as their axes promote a natural ordering. In contrast to visualizing itemsets (Section 7) and association rules (Section 9) 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 [203]. The fact that itemsets are mapped to distinct elements on the axes as well as the nonuniform slopes increases the clutter.

[203]: Yang (2003), Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates

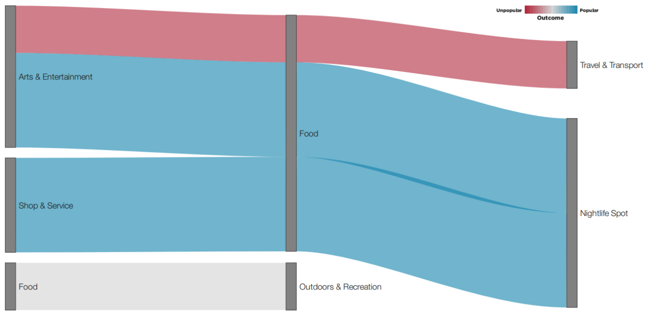


Figure 4.19: Sequential patterns visualized using a Sankey diagram [158].

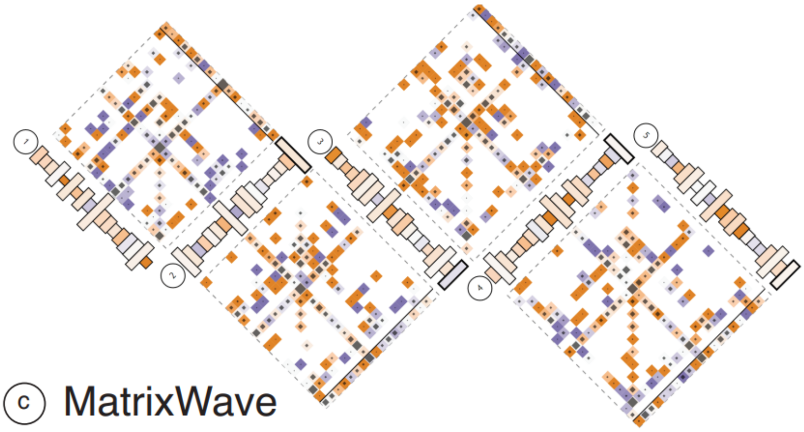


Figure 4.20: *MatrixWave* [236], a zig-zag layout of multiple matrices which can be derived from the Sankey-diagram visualization.

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 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 [158] (Figure 4.19). This type of visualization can aggregate multiple patterns and implicitly show subpatterns. The support of each pattern is mapped onto the width of the line. In the 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 data [234]. Perer et al. combine their Sankey diagram representation with a bubble chart which shows the most frequent 1-event patterns [235].

[158]: Perer et al. (2014), Frequence: interactive mining and visualization of temporal frequent event sequences

[234]: Chou et al. (2016), Privacy preserving event sequence data visualization using a Sankey diagram-like representation

[235]: Perer et al. (2015), Mining and exploring care pathways from electronic medical records with visual analytics

Zhao et al. point out that Sankey diagrams, like other node-link diagrams, tend to produce clutter due to many overlapping edges [236]. 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 is replaced by bars on

[236]: Zhao et al. (2015), MatrixWave: Visual Comparison of Event Sequence Data

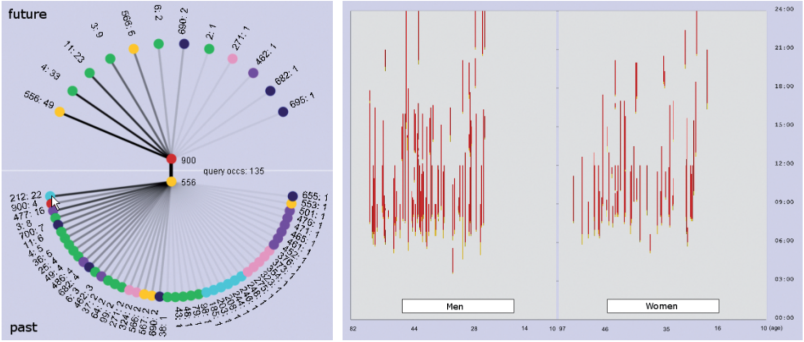


Figure 4.21: The *ActiviTree* interface [157]. The left shows a graph where the central part is the current pattern in focus. Preceding and succeeding events are placed below and above the central part whereas the support is mapped onto the opacity of the lines. The right part visualizes the sequences where the yellow parts represent the selected pattern.

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 color can be 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 [157] (Figure 4.21, 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 4.21) 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.

[157]: Vrotsou et al. (2009), ActiviTree: Interactive Visual Exploration of Sequences in Event-Based Data Using Graph Similarity

The earlier mentioned icicle plot visualization could also be used for *branching patterns* [237]. Liu et al. can alternatively display a node-link diagram or a combination of both as shown in Figure 4.22. 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

[237]: Liu et al. (2017), CoreFlow: Extracting and Visualizing Branching Patterns from Event Sequences

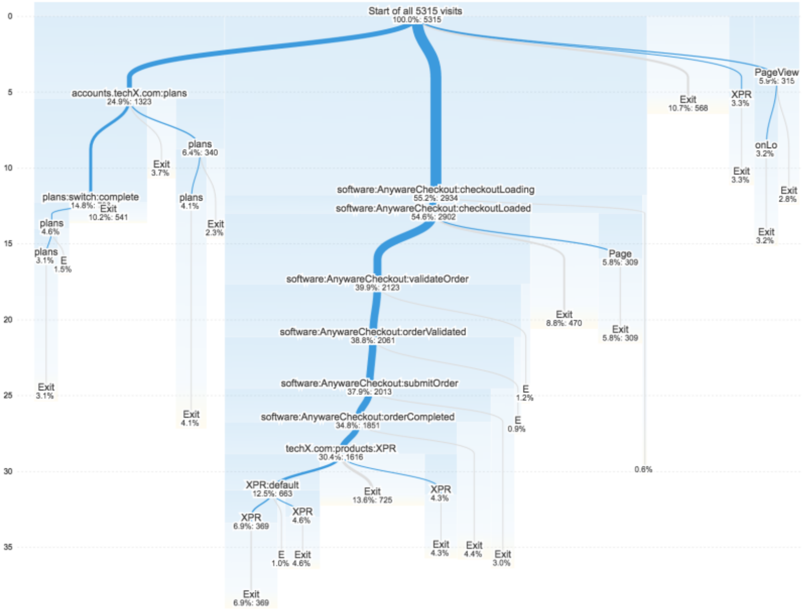


Figure 4.22: A combination of an icicle plot visualization and node-link diagram for branching patterns [237].

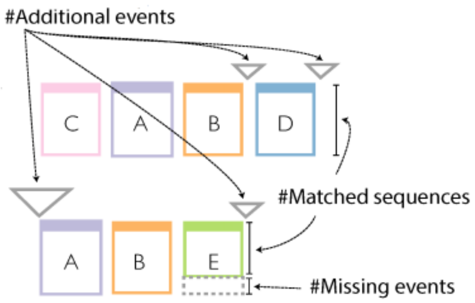


Figure 4.23: Two aggregated patterns represent two disjoint sets of similar patterns [15].

color to make them distinctive.

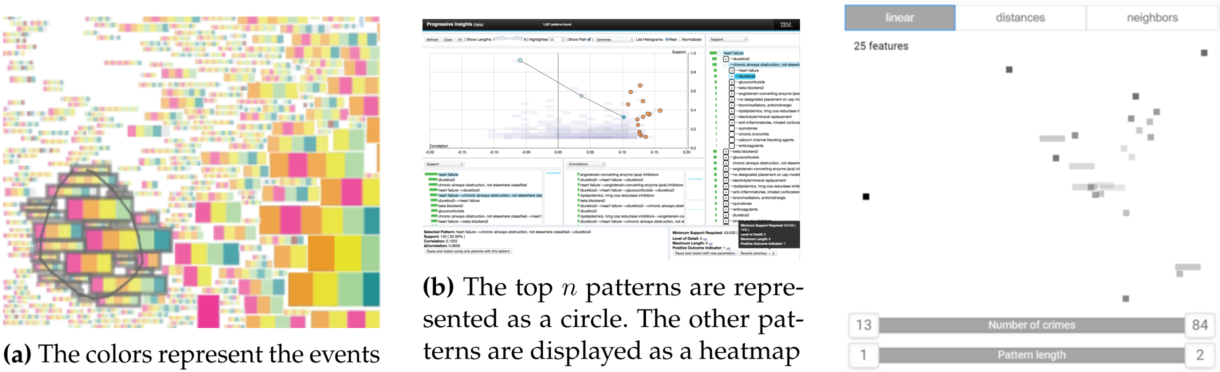
Aggregated Patterns

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.

Chen et al. summarize multiple patterns according to the minimum description length [238] and visualize the so-gained prototype including the lost information [15]. This is achieved using corrections that insert or delete events. The visualization, as shown in Figure 4.23, 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.

[238]: Grünwald (2004), A tutorial introduction to the minimum description length principle

[15]: Chen et al. (2018), Sequence Synopsis: Optimize Visual Summary of Temporal Event Data



(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 [239].

(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 [159].

(c) Patterns are represented through rectangles which are placed on a 2D plane using projection methods. Similar patterns are closer together [8].

Figure 4.24: Patterns can be clustered and placed on a 2D plane reflecting their similarity or different metrics can be used to layout patterns.

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 find interesting patterns by placing them according to metrics.

[239]: Wei et al. (2012), Visual cluster exploration of web clickstream data

Wei et al. use differently colored, horizontally placed rectangles to represent web click stream patterns [239]. 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 [240]. The significance of a pattern is mapped onto the size.

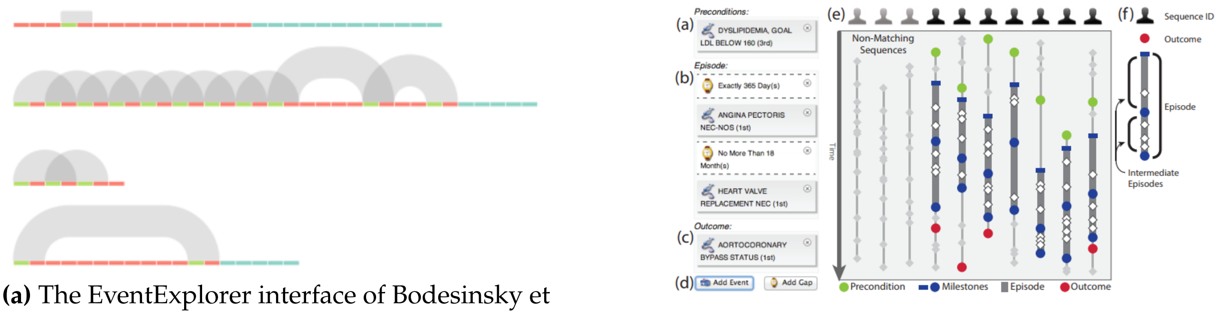
[240]: Viégas et al. (2009), Participatory Visualization with Wordle

[159]: Stolper et al. (2014), Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics

Stolper et al. use a scatterplot to plot patterns [159]. 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.

[241]: Gotz et al. (2014), A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data

Gotz et al. use a scatterplot where the support is mapped to the x- and the y-axis [241] of different patients' outcomes.



(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 [243].

(b) *DecisionFlow* shows episodes using glyphs and highlights the respective space between the beginning and end of the episode [153].

Figure 4.25: Two techniques to visualize serial episodes in event sequences.

The size of the circles (patterns) is mapped to the correlation to the outcome.

In previous work, we used a scatterplot to depict sequential patterns on a 2D space [8]. 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 in 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.

[8]: Jentner et al. (2018), Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool

Episode Visualization

In episode mining [242], 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 the 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.

[242]: Mannila et al. (1997), Discovery of Frequent Episodes in Event Sequences

Arc diagrams [244] 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

[244]: Wattenberg (2002), Arc Diagrams: Visualizing Structure in Strings

et al. make use of a variation of arc diagrams to visualize recurrent patterns of serial episodes in event sequences [243] (Figure 4.25a).

[243]: Bodesinsky et al. (2015), Exploration and Assessment of Event Data

In *DecisionFlow*, Gotz et al. use a glyph representation to visualize episodes in event sequences [153] (Figure 4.25b). 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 the circular shapes of the following milestones. The serial episode is additionally highlighted.

[153]: Gotz et al. (2014), DecisionFlow: Visual Analytics for High-Dimensional Temporal Event Sequence Data

In a previous work [29], we use horizontal lines above or below 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.

[29]: Jentner et al. (2017), Feature Alignment for the Analysis of Verbatim Text Transcripts

4.4 Comparison

This section compares the different visualization techniques within their specific domain. We hereby use visual variables that Bertin identified [245] and their ranking [246, 247]. We use this ranking to depict for example the task of comparing the support for different patterns. Additionally, we provide an analysis of 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 of completely identifying 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. [10]

[245]: Bertin (1983), Semiology of graphics: diagrams, networks, maps

[246]: Mackinlay (1986), Automating the Design of Graphical Presentations of Relational Information

[247]: Munzner (2014), Visualization Analysis and Design

4.4.1 Frequent Itemsets

Table 4.1 shows that the pixel-based visualization features the best scalability. However, it does not provide insight into

[10] 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 [248]. We consider this as future work.

Table 4.1: A comparison of visualization techniques for frequent itemsets categorized as pixel-based visualizations, tree visualizations, and linear visualizations.

		Year	Σ-Scalability	Itemset-Scalability	Support Comparison	Identification
lattice	[191]	1999	-	-	/	+
	[192]	2013	+	+	-	-
pix	[194]	2005	++	++	/	/
tree	[195]	2005	- -	+	+	-
	[200]	2011	-	+	-	-
	[199]	2012	- -	+	+	-
	[202]	2016	+	-	+	++
linear	[203]	2003	+	+	++	-
	[205]	2008	+	-	++	+
	[206]	2008	+	++	+	+

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 [192, 195, 200, 202] as the visual variable. Linear visualizations provide the best support comparison as they map this feature to the position.

[192]: Bothorel et al. (2013), Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm

[195]: Keim et al. (2005), Fp-viz: Visual frequent pattern mining

[200]: Leung et al. (2011), FpMapViz: A Space-Filling Visualization for Frequent Patterns

[202]: Leung et al. (2016), PyramidViz: Visual Analytics and Big Data Visualization for Frequent Patterns

4.4.2 Association Rules

Association rules typically contain two interestingness measures. Table 4.2 shows that visualizing both simultaneously is difficult while preserving their structure and being scalable at the same time. Graph-based systems can aggregate multiple patterns but quickly get cluttered [83]. A solution to this is to generate a node for each itemset instead of each item [209, 210]. 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 biases [211]. 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 [215]. The double-decker plots [187] aggregate similar rules featuring a good comparison for both interestingness measures and enabling the user to compare different rules.

[83]: Klemettinen et al. (1994), Finding Interesting Rules from Large Sets of Discovered Association Rules

[209]: Han et al. (1996), DBMiner: A System for Mining Knowledge in Large Relational Databases

[210]: Rainsford et al. (2000), Visualisation of Temporal Interval Association Rules

[211]: Fukuda et al. (1996), Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization

[215]: Wong et al. (1999), Visualizing Association Rules for Text Mining

[187]: Hofmann et al. (2000), Visualizing association rules with interactive mosaic plots

Table 4.2: Visualization techniques for association rules that are categorized as individual representations, graph visualizations, matrix- and table-based visualizations, mosaic plots, and linear visualizations.

					Comparison			
		Year	Σ-Scalability	AR-Scalability	Support	Confidence	Both	Identification
ind	[83]	1994	- -	- -	++	++	++	++
graph	[83]	1994	- -	-	+	+	-	-
	[209]	1997	- -	-	+	+	/	+
	[210]	2000	-	+	/	/	/	+
matrix	[211]	1996	+	+	+	+	+	-
	[213]	1996	-	+	+	-	-	-
	[214]	1997	-	+	-	+	-	-
	[212]	2000	+	+	+	+	-	- -
tab	[215]	1999	+	++	+	+	+	+
mos	[187]	2000	-	- -	+	++	+	+
lin	[203]	2003	+	+	-	-	/	+

Hahsler et al. provide a similar comparison of visualization techniques for association rules [188].

[188]: Hahsler et al. (2011), Visualizing association rules: Introduction to the R-extension package arulesViz

4.4.3 Sequential Patterns

Compared to frequent itemsets and association rules we could identify the most visualization techniques for sequential patterns (Table 4.3). 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.

Table 4.3: A comparison of visualization techniques for sequential patterns using individual representations, flow diagrams, aggregated pattern visualizations, placement strategies, and episode visualizations.

		Year	Σ-Scalability	Pattern-Scalability	Support Comparison	Identification
ind	[222]	2009	- -	- -	/	++
	[25]	2016	- -	- -	/	++
	[154]	2018	+	+	/	+
flow	[229]	2000	-	-	/	+
	[203]	2003	-	+	+	-
	[225]	2007	-	-	-	-
	[157]	2009	+	-	+	-
	[230]	2011	-	-	/	-
	[158]	2014	+	-	+	+
	[236]	2015	+	+	+	-
	[237]	2017	-	-	+	+
	[232]	2017	-	-	+	+
agg	[15]	2018	+	++	+	+
place	[239]	2012	-	++	+	+
	[159]	2014	++	++	++	/
	[241]	2014	++	++	++	/
	[8]	2018	++	++	+	/
episode	[153]	2014	-	–	/	++
	[243]	2015	+	+	- -	+
	[29]	2017	+	++	/	/

4.5 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 that must be retained to not lose 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 [249] on which information can be mapped.

[249]: Bertin (1973), Sémiologie graphique: Les diagrammes-Les réseaux-Les cartes

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 [246, 247] 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.

[246]: Mackinlay (1986), Automating the Design of Graphical Presentations of Relational Information
[247]: Munzner (2014), Visualization Analysis and Design

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 allow 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 not only reduces 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 to include her domain knowledge in the mining process. However, it does not guarantee that a pattern mining algorithm only reveals the interesting patterns that impose 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 [250]. 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.

[250]: Shneiderman (1996), The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations

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 toward 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 one to view the data from multiple perspectives. This has to be carefully designed, for example through linking and brushing [251] to provide a benefit to the user. Such systems often raise the complexity and steepen the learning curve.

[251]: Keim (2002), Information Visualization and Visual Data Mining

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 identify 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 4.1, software libraries are available to provide 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 [170]. The arulesViz R-package [188] features different types of visualizations for association rules. This might be because no

[170]: Behrisch et al. (2019), Commercial Visual Analytics Systems-Advances in the Big Data Analytics Field

[188]: Hahsler et al. (2011), Visualizing association rules: Introduction to the R-extension package arulesViz

best-practice visualizations are established that are superior to the other available techniques.

4.6 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 reflect 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 providing details on demand.

Visual Pattern Analytics 5

This chapter strives to summarize different approaches and techniques known from the fields of visual analytics and interactive visualization and transfers them for the use of exploratory data analysis in pattern mining. The term *visual pattern analytics* is a variation of visual text analytics which Risch et al. describe as:

> "[...] a class of information analysis techniques and processes that enable knowledge discovery via the use interactive graphical representations of textual data." [252]

In the previous chapters, we have seen that interestingness measures, or more generally, features reflect certain properties of patterns or the cluster they represent (see Chapter 3). The visualization chapter (see Chapter 4) shows that any visualization technique that strives to show the structural parts of the patterns such that specific items are visible does not scale well beyond 100 patterns. This is in contrast to the so-called pattern explosions that are occurring due to the combinatorial explosion of pattern mining (see section 2.2).

Jarke J. van Wijk mentions:

> "The most important case is simply when the amount of data to be shown does not fit on the screen or is too large to be understood from a single image. In this case, navigation and selection of the data has to be supported to enable the user to interactively explore the data." [6]

Everyday techniques such as scroll bars and pagination come to mind that allow visualizing beyond the boundaries of the screen but at the same time, the user may lose context. However, with regards to the exponential search spaces of pattern mining the linear, quadratic, or even cubic (considering 3D), screen space may never be sufficient to effectively solve this problem. But we must not only think about the technical visual aspects when it comes to scalability. Ultimately the user must evaluate and decide which patterns are interesting and it is unlikely that a user wants to spend the effort to explore thousands or millions of patterns. Additional measures are

[6]: Wijk (2006), Views on Visualization

required.

Note that this chapter does not have a dedicated related work section because I summarize different techniques and classify my work as well as related work.

5.1 Interactive Visualization

Pattern mining can be well embedded into interactive tools. The simple approach is to let the user select data and thresholds for the pattern mining algorithm, run the algorithm, and present the output to the user (see Figure 5.1). Additionally, the user may interact with the output depending on the task and domain. We neglect this last aspect here. Then the user can tune the parameters or transform the input data to alter the results. This section also does not cover the transformation of data into structured data. This also heavily depends on the data at hand and the task. Domains for event sequences may be time series analysis, text analysis, click stream sequences, server logs, patient histories, etc.

5.1.1 Structured Data Selection & Transformation

Selection and filtering are technically typically trivial. For pattern mining, we must distinguish between two types of selection: transaction selection and event transformation.

Selection of Transactions

The former is typically performed to find various subspaces in the data that the user wants to compare or is specifically interested in. Note that this functionality is provided in almost every interactive tool. Exemplarily I want to mention the ODIX tool here which makes heavy use of transaction selection and pattern mining to solve the task at hand.

The ODIX application is a solution to the VAST Challenge 2017 Mini Challenge 1 [20]. The data consists of cars that drive to the park and are observed at certain points in the park at a certain time. We model this data as event sequences

[20]: Whiting et al. (2017), VAST Challenge 2017 Mini Challenge 1

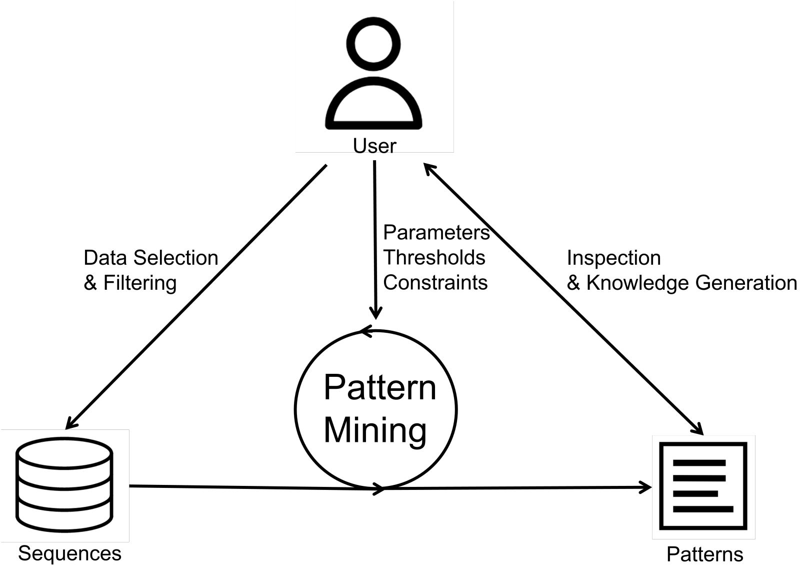


Figure 5.1: A simplified version of Fayad's KDD pipeline [253]. The separate selection, preprocessing, and transformation steps are collapsed. The sequence database is then mined using various parameters, thresholds, and constraints leading to patterns that are in turn inspected and, potentially, knowledge is generated.

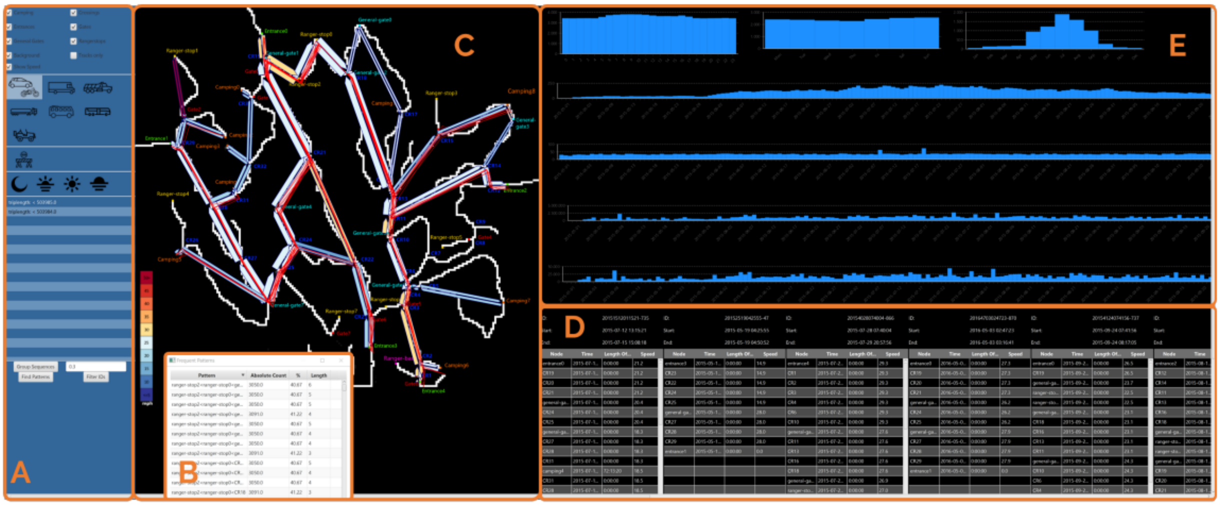


Figure 5.2: The ODIX interface: Filtering and Sequence Analysis options are available in the left sidebar (A) and detailed information about extracted sequences is in the sequence detail window (B). The main map view (C) shows the amount of traffic and average speed for street segments as well as individual trips. Bar charts on the right (E) provide temporal distribution in semantic aggregation levels as well as the trip amount, speed, duration and length statistics. Single trips can be analyzed in the detail view in (D). This image is taken from the original publication [19].

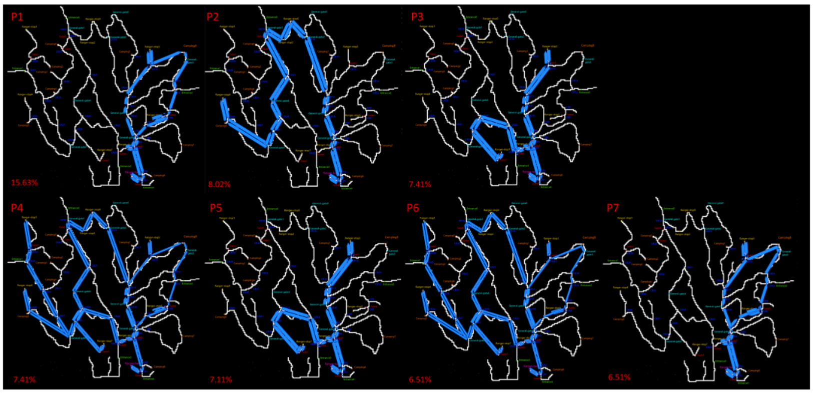


Figure 5.3: Typical patterns of ranger patrols derived using frequent pattern extraction, ordered by their share amongst all ranger patrols. The first and the last as well as the third and fifth patterns feature almost the same route, but patrolled in opposite directions. This image is taken from the original publication [19].

such that every car trip corresponds to one sequence whereas the positions of the car at a given time correspond to the itemsets. [1] We chose to implement ODIX, a novel prototype helping to explore origin-destination data both in detail and in aggregated form. The ODIX interface consists of three main categories: The left side (Figure 5.2, A) shows general settings and categorical filters, e.g. car types or violations. The center (C) gives the user spatial access to the data where the underlying graph structure is mapped onto the map of the park. The right side (E) provides temporal information. The upper three bar charts aggregate the number of trips per hour, weekdays, and months. The bar chart below shows the number of trips for each date. The user can set a date-range filter by clicking on this chart. The lower three bar charts use statistical data: speed, stop time, and the length of a trip. For each, the user can select to display the maximum or the average per day. Additionally, the user can add filters based on these statistics. For example, the user can filter trips with higher or lower speeds than a given value. By clicking on one of the segments in the map, a spatial filter is applied to filter trips that pass through this segment also considering the direction. Finally, it is possible to group sequences. This presents the user with all distinct trips in the current filter selection. Several trips can be selected to further filter down the data. We used this strategy to find and generate the typical ranger patterns as depicted in Figure 5.3. Similarly, the user can search for frequent sequential patterns. These are also displayed in a table and can be used as a filter. All filters that are currently in use are listed on the left side. Each of the filters can be negated or removed individually. Not shown is a table, which is located below, displaying each trip. Selecting trips there highlights their route in red in the graph visualization as well as opens a detail pane for each trip (D). This helps the user to investigate a few selected trips in detail and was used, for example, for detailed comparisons of cars involved in a race.

Note that transaction filtering does not necessarily impact the search space of the pattern mining problem which would reduce the execution time of the algorithm as well as the result set.

[1] This part is taken from the publication “ODIX: A Rapid Hypotheses Testing System for Origin-Destination Data” (Section 3: Application) [19]. I have been the main author of this section and have written all the contents. The paper was internally reviewed and edited by my co-authors Juri Buchmüller, Dirk Streeb, and Daniel Keim.

Event Transformation

This section summarizes how event sequences, or structured data in general, may be transformed. Note that these interactions with the structured data may greatly impact the search space of pattern mining.

Item & Itemset Deletion This step is trivial but can greatly reduce the search space and result set of patterns. The reason to delete items or even whole itemsets of an event sequence is simply if a domain expert considers them irrelevant to the task at hand. This is also an option to reduce noise in the data.

Item & Itemset Addition Adding items or itemsets to the data may be less common but can be useful if a user annotates certain events in an event sequence.

Item & Itemset Modification Modifying (i.e., renaming) an item can be useful to provide a more semantically meaningful name. Modifying itemsets typically refers to switching the order. This can be useful for erroneous data or uncertain data.

Merging Items Merging items is a great way to reduce the search space. The reason for this can be a high correlation between the two items. This is one possibility to aggregate low-level data into more semantically meaningful data.

Merging Itemsets Although similar to the previous interaction, merging itemsets is semantically different as itemsets reflect the order in a sequence. An example might be a low-level click stream sequence where the user first clicks on the save icon and then confirms to save the document in the following dialog. Such patterns can be summarized as an event that represents the user saving the document. The user ultimately has to decide how to merge the two timestamps that are associated with the underlying events. Because sequential pattern mining is sensitive to the order of events, uncertainties in time cannot be easily mapped. A user may therefore choose to combine two items that are alternating in

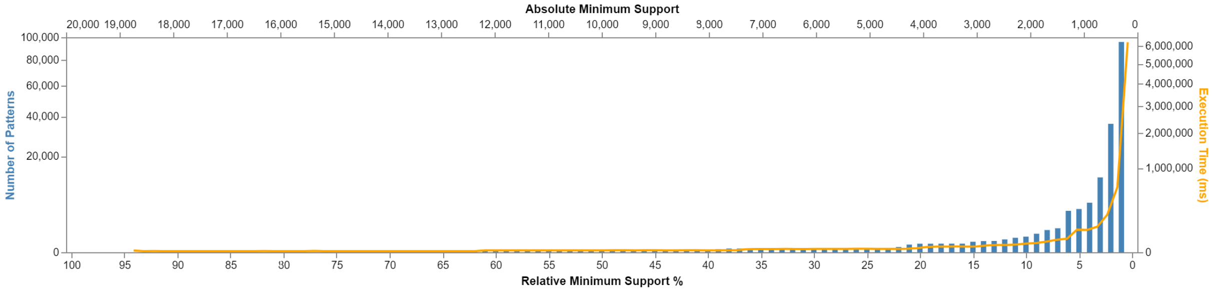


Figure 5.4: The number of patterns and execution time in milliseconds calculated from the VAST Challenge Mini Challenge 1 [20] dataset for varying numbers of the minimum support threshold. Note that the y-axes are square root scaled. The chart underlines how the number of patterns and execution time grows exponentially.

one itemset such that they are being modeled as occurring simultaneously.

Specifically for the merging parts of itemsets and items, Cappers et al. have created a powerful tool, called Event-Pad, where the user is capable of generating rules, based on a visual regular expression syntax, to merge events [154, 254].

[154]: Cappers et al. (2018), Exploring Multivariate Event Sequences Using Rules, Aggregations, and Selections
[254]: Cappers et al. (2018), Eventpad: Rapid Malware Analysis and Reverse Engineering using Visual Analytics

Note that parameter estimation is difficult for a user, specifically in exploratory data analysis where no specific questions (i.e., parameter settings) can be assumed but rather trying out various hypotheses. The underlying problem is that the interestingness measures, such as the relative minimum support, are linear (i.e., from 0% to 100%), however, the result set that this threshold influences the number of patterns in the result, as well as computation time and memory/space complexity, grow exponentially. On the other hand, if the parameter is too restrictive, no patterns will be returned at all. Figure 5.4 shows this as a chart. Note that the y-axes are square root scaled. A logarithmic scale is not possible because the number of patterns is zero for minimum supports 100% - 66% which would result in a negative infinity value. Further note that for minimum support of 1%, the absolute minimum support is 187 since there are 18739 transactions in the dataset. This number might be too high for certain analysis tasks where even more infrequent patterns are sought. But already for this parameter setting, the execution time of the algorithm is well over one hour. For even lower minimum supports it is likely that the algorithm fails with an out-of-resources exception. It is possible to mine for rare patterns specifically [255, 256] but the search space depends on the distribution of the data (items) and cannot be well predicted to perform an

[255]: Koh et al. (2016), Unsupervised Rare Pattern Mining: A Survey
[256]: Borah et al. (2019), Rare pattern mining: challenges and future perspectives

automated switch between algorithms. In EDA, it can be assumed that a user does not follow a linear or another specific strategy to estimate the parameters. Moreover, it will most likely follow some sort of binary search where first a very high minimum support threshold is being used resulting in an empty output, followed by a very low threshold resulting in thousands of patterns with a long execution time. In the worst-case scenario, the algorithm fails with an exception because the system is out of resources. Later the thresholds are calibrated somewhere in between. While adding more parameters allows the user to steer the result set more fine-grained, it also adds a lot of complexity to the system and the results can quickly be similar as described before. This also assumes that the user has, at least, a conceptual understanding of what the parameters influence besides the number of patterns in the result set (see section 5.6 for more details). Without any understanding, in a black-box scenario, the user will undoubtedly end up with an out-of-resource exception eventually.

5.2 Linking and Brushing

Linking and brushing is a well-known interactive visualization technique, to combine several charts using visual linking such as colors, shapes etc. A simple example is multiple scatterplots that show various dimensions of a dataset or different projections. Clusters of the dataset could be indicated by color, whereas the selection of one or multiple points would highlight the respective points in all charts. According to Daniel Keim:

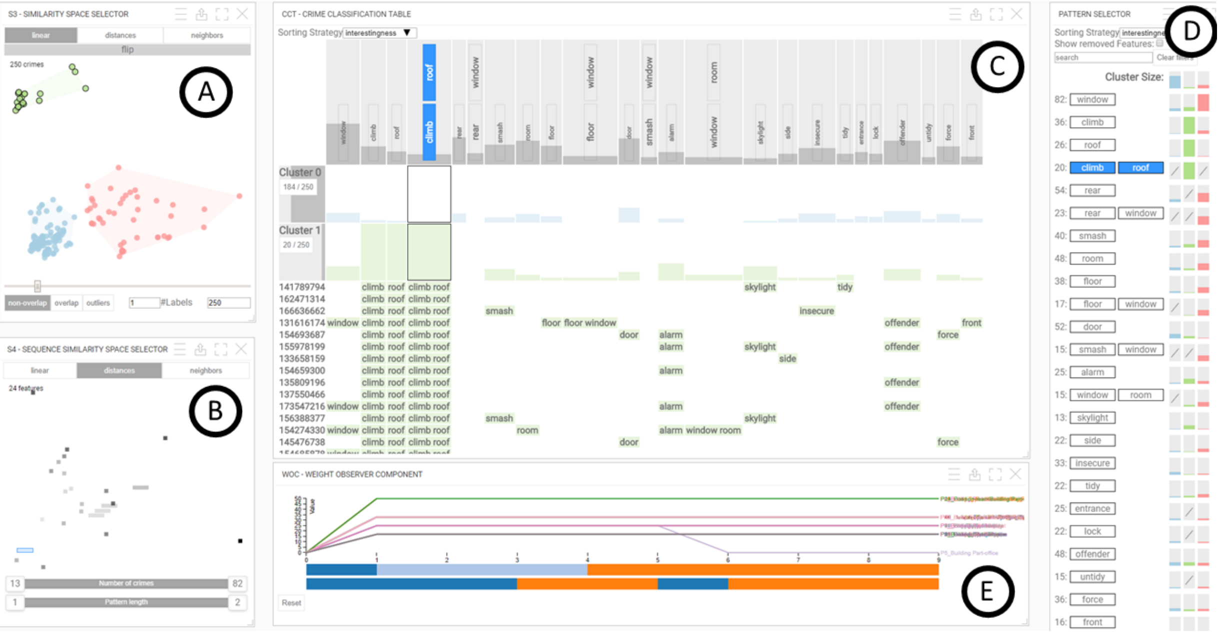
> "The idea of linking and brushing is to combine different visualization methods to overcome the shortcomings of single techniques." [251]

It further allows us to inspect the same data from various perspectives (such as dimensions/projections) but also allows us to combine multiple data sources visually.

The *Concept Explorer* of the VALCRI prototype makes heavy use of linking and brushing [8]. It allows us to overcome scalability problems that occur when mining patterns. For example, the *Pattern Selector* component shows sequential patterns in a list view and displays each item. Therefore, its scalability is quite poor by showing a max of 10 patterns at the same time. The *Sequence Similarity Space Selector (S^4)* does not show individual items but rather interestingness measures of the sequential patterns such as their length (i.e., generation) and support while also displaying their similarities (see section 2.3 and subsection 3.4.1). Therefore, these two components overcome their individual shortcomings. When the user hovers over one pattern, the pattern is highlighted in all of the other components.

5.2.1 VALCRI Concept Explorer

The following describes the *Concept Explorer*, its components, and also presents a use case. For a task description, the modeling of the data to event sequences, and the description of interestingness measures see subsection 3.4.1.



Sorting Strategy interestingness

1 #Labels 250

13 Number of crimes 82
1 Pattern length 2

Cluster 0 184 / 250
Cluster 1 20 / 250

141789794 climb roof climb roof skylight tidy
162471314 climb roof climb roof
166636662 climb roof climb roof smash insecure
131616174 window climb roof climb roof floor floor window offender front
154693687 climb roof climb roof door alarm force
155978199 climb roof climb roof alarm skylight offender
133658159 climb roof climb roof side
154659300 climb roof climb roof alarm
135809196 climb roof climb roof offender
137550466 climb roof climb roof
173547216 window climb roof climb roof alarm offender
156388377 climb roof climb roof smash skylight
154274330 window climb roof climb roof room alarm window room
145476738 climb roof climb roof door force

Sorting Strategy interestingness
Show removed Features: □
Cluster Size:

82: window
36: climb
26: roof
20: climb roof
54: rear
23: rear window
40: smash
48: room
38: floor
17: floor window
52: door
15: smash window
25: alarm
15: window room
13: skylight
22: side
33: insecure
22: tidy
25: entrance
22: lock
48: offender
15: untidy
36: force
16: front

Figure 5.5: The visual interface of our interactive CCA system. Crime cases and clusters are shown at the center within the crime cluster table (C – CCT) to support the actual CCA task. We provide the analyst with a hybrid analysis perspective on the data and feature space on the left-hand side: A two-dimensional embedding of the crime similarities and the clustering is shown in the Similarity Space Selector (S^3 – A). Another two-dimensional embedding of the crime pattern similarity (features) based on the shared crimes is shown in the Sequence Similarity Space Selector (S^4 – B). The respective crime patterns are also shown in the pattern selector on the right-hand side (D). Tracked interactions and configurations are shown in the Weight Observer Component (WOC – E). All views are linked and allow criminal analysts to develop and verify alternative clusterings from different tightly integrated perspectives. The figure is taken from the original publication [8].

Concept Explorer Components

[2] This section briefly introduces all components of the concept explorer.

Similarity Space Selector - S^3 The Similarity Space Selector (S^3) provides a simple interface for crime investigators to understand the relations and similarities among multiple crimes across different dimensionality reduction and clustering results. It represents the two-dimensional data space as crimes are arranged according to feature similarities (i.e. if they contain similar crime patterns).

Crime Cluster Table - CCT The comparative case analysis (CCA) table is a central component of our user interface. As mentioned, crime investigators manually maintain such a spreadsheet where crimes are listed and manually identified characteristics of a crime are placed column-wise. The

2: This section takes short excerpts from my publication "Making Machine Intelligence Less Scary for Criminal Analysts: Reflections on Designing a Visual Comparative Case Analysis Tool" (Section 3: Design Study Methodology) [8]. I have been the main author of this publication and have written major parts of the content. The paper was co-authored by my co-authors Dominik Sacha and Florian Stoffel and edited by Geoffrey Ellis, Leishi Zhang, and Daniel Keim.

respective cells of crimes that contain such a characteristic are color-coded and contain annotations. Manual filtering and sorting operations in the spreadsheet table provide some sense-making capabilities.

Sequence Similarity Space Selector - S^4 S^4 is a feature projection view that offers an important perspective on the feature space, supporting the feature selection and emphasis task to improve the data clusters of the projections in the data projection view S^3. The visual clusters are only as good and useful as the features, therefore, the user needs to understand feature characteristics of the analyzed dataset including overlaps, redundancies, correlations and outliers.

Pattern Selector The Pattern Selector allows the user to browse and explore multiple feature patterns. It shows all sequential patterns in a list-based form whereas their items remain visible.

Weight Observer Component - WOC The Weight Observer Component (WOC) provides analytical provenance [257] and captures user interactions [258]. It was initially designed as a tool for the developers to track and understand how the Concept Explorer was being used. It tracks and visualizes the feature weights in a multi-line chart and the DR and clustering configuration in state-history charts (Figure 5.7). The end-users did not find it particularly useful but suggested that it could be part of a reporting feature, outlining their exploration of the data. The Security, Ethics, Privacy & Legal (SEPL) board highlighted its crucial role in court cases when analysts have to justify their decision-making. We also observed that the component can be useful as a bookmarking feature to save and load configurations and feature weights for specific analytical tasks.

[257]: Xu et al. (2015), Analytic Provenance for Sensemaking: A Research Agenda

[258]: Endert (2016), Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering

Concept Explorer

[3] The components introduced in the previous section are part of the Concept Explorer and embedded into the VAL-CRI framework (Figure 5.5). This framework provides a web-based dashboard design in the front end and a Java-based back end to perform more complex operations such

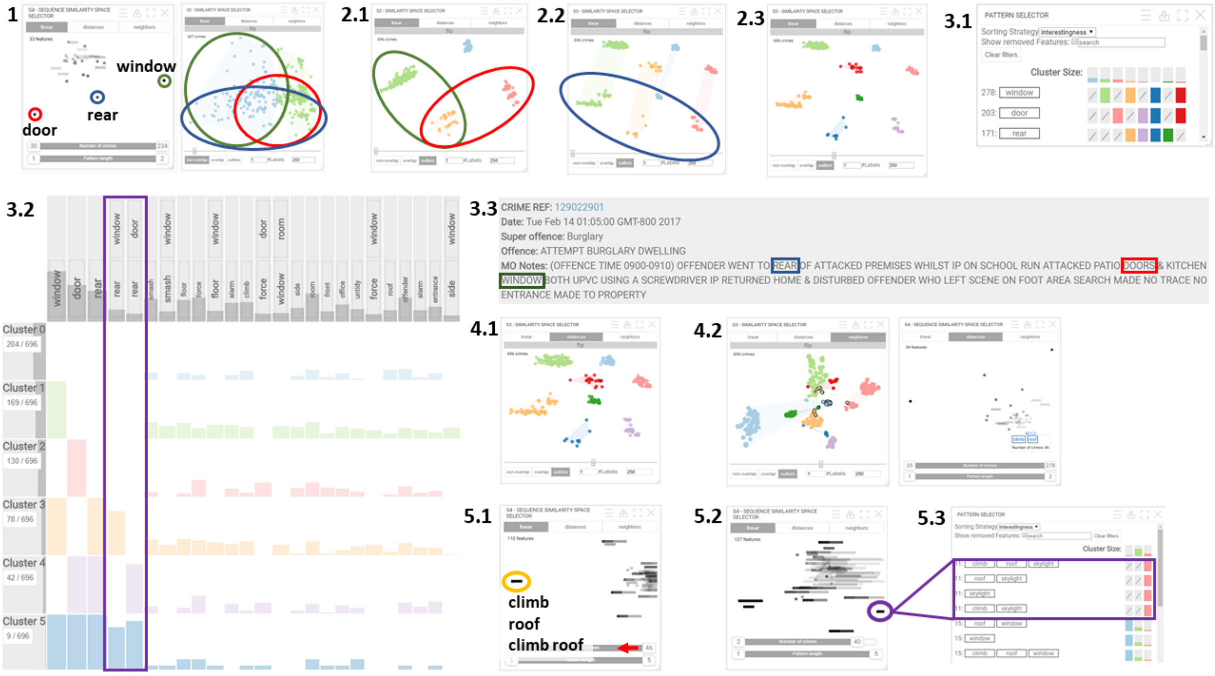
3: This section is taken from my publication "Making Machine Intelligence Less Scary for Criminal Analysts: Reflections on Designing a Visual Comparative Case Analysis Tool" (Section 4: Concept Explorer) [8]. I have been the main author of this publication and have written major parts of the content. The paper was co-authored by my co-authors Dominik Sacha and Florian Stoffel and edited by Geoffrey Ellis, Leishi Zhang, and Daniel Keim.

as dimensional reduction (DR) and clustering. The VALCRI workstation consists of two stacked 27-inch touch screens. The user can open a canvas on each screen, each with multiple components that can be arranged and resized freely. The components are tightly coupled to provide a better analytical understanding of the data and its features. In general, hovering over a feature (e.g. in S^4) will highlight the feature in other components (e.g. Pattern Selector), with all crime reports, described by that feature, highlighted as well. Similarly, hovering over crime reports highlights features within the reports. This linking and brushing capability are important to understanding the influences of features in the data similarity space (see Section 3). The Similarity Space Selector (S^3) is in charge of creating the clusters, the cluster information is broadcast to the other components, for example, the CCT. Filters can be applied by all components to reduce the crime report data set and enable users to drill down for a specific set of crime reports containing a user-defined set of features. We present a use case demonstrating a workflow that we could observe during user evaluations and then report expert feedback obtained for the current system.

Use Case

The crime set that is being investigated is normally specific to a region and a time range and this can be obtained with the respective components available in the VALCRI framework (timeline and map). Additionally, the set is filtered by search terms to receive similar types of crimes. In the present use case, the user is interested in burglaries in *schools*.

After opening the S^3 and S^4 components, the user is presented with a view as can be seen in Figure 5.6 step 1. S^4 shows three exposed dark quadratic rectangles referring to three feature sequences containing a single term that occurs frequently. These three terms are *door* (red), *rear* (blue), and *window* (green). The fact that these features are exposed and are highly saturated, suggests to the user that the data similarity space visible in S^3 is mainly separated by these features. We have annotated the regions where the crime reports are located in the same colors as in S^4. The linking and brushing features of the components are used to obtain this insight.



CRIME REF: 129022901
Date: Tue Feb 14 01:05:00 GMT-800 2017
Super offence: Burglary
Offence: ATTEMPT BURGLARY DWELLING
MO Notes: (OFFENCE TIME 0900-0910) OFFENDER WENT TO REAR OF ATTACKED PREMISES WHILST IP ON SCHOOL RUN ATTACKED PATIO DOORS & KITCHEN WINDOW BOTH UPVC USING A SCREWDRIVER IP RETURNED HOME & DISTURBED OFFENDER WHO LEFT SCENE ON FOOT AREA SEARCH MADE NO TRACE NO ENTRANCE MADE TO PROPERTY

Figure 5.6: A frequently observable use case starting with the initial data. The user identifies the main features separating the data space (1) and increases the weights (importance) of interesting features and defines a new clustering (2). A CCA analysis with detail on demand follows (3). The cluster robustness across different DRs is tested afterward (4). The use-case ends with a drill-down operation including the pruning of the feature space (5). The figure is taken from the original publication [8].

The user is further interested in these features and increases the weights for the features *door* and *window* and applies a new clustering to better distinguish the crime reports. The results are visible in step 2.1 where S^3 shows four clusters. The green and the yellow cluster circled in green, contain crime reports with the feature *window*. The yellow and the red cluster circled in red, contain *door*. The blue cluster, on top, does not contain any of both features. All clusters contain crime reports with the feature *rear* meaning that the similarity space is currently not separated by this. The user is also interested in the feature *rear* and therefore increases its weight. S^3 updates immediately resulting in the view given in step 2.2 - note that all clusters are rather distorted. The lower part of the clusters circled in blue, consists of crime reports containing the feature *rear*. The user manually triggers a re-clustering and also increases the number of clusters using the lower slider in S^3. The result can be seen in step 2.3. This sub-workflow presented in step 2 can be frequently observed - we call it “cluster-mitosis".

The clusters can be interpreted using the Pattern Selector (step 3.1) and the CCT (step 3.2). The dark-blue cluster (third column from the right) contains only crime reports that have all three features. This cluster is located in the bottom center location in S^3 (step 2.3). With the CCT (step 3.2) the user can now perform typical CCA tasks, such as comparing the features of the clusters to spot interesting co-occurring features. The feature sequences *rear window* and *rear door*, circled in purple, are only present in clusters where the single-term sequences are present (orange, purple and dark-blue clusters). The bars displaying the frequency of the feature in the cluster are not full, showing that some crimes contain for example the feature sequence *door rear*. But this sequence is too infrequent to be in the feature result set and therefore is not visible as a column in the table. Furthermore, the gray bars in the header, show that the feature sequence *rear window* is more frequent than *rear door*. The user expands the dark-blue cluster (bottom) in the CCT to inspect the individual crime reports. A similar view is visible in Figure 5.5 (C). By clicking on one crime report, the "crime card" component opens showing more details of that crime including the Modus Operandi (step 3.3). As this cluster only contains crime reports holding all three features, the user can find these features in the text. Due to the order of the terms, the crime also contains the sequences *rear door* and *rear window*.

The user checks the other projection methods such as "distance" (MDS) and "neighbors" (t-SNE). Whilst the "distances" only show that the clusters expanded a little (4.1), the "neighbors" projection shows a different picture (4.2). This projection favors neighborhoods and therefore shows identical crimes in non-overlapping rings. These "crime rings" can be important in the users' analysis. The user learns that several crime rings contain the feature sequence *climb roof*. The feature is highlighted in step 4.2 (right side) and the crime reports are highlighted with a black border in step 4.2 (left side). The user is further interested in these crimes and filters the crime data set on the feature *climb roof* (drill down).

The remaining dataset contains 46 crime reports. However, the set of features has increased to 110 because a pattern must only occur in 5% of the crime reports (i.e. 2) to remain in the result set. These longer and more specific sequences can be interpreted by the experts without the need to read the MO. S^4 shows an outlier circled in yellow in step 5.1. At this location three features *climb*, *roof*, and *climb roof* are overplotted. These features are outliers since they describe all crimes in the result set. Thus, these features are uninter-

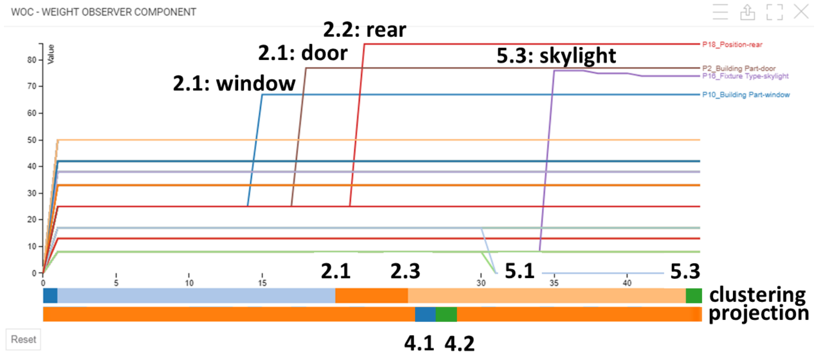


Figure 5.7: The use case as it is recorded in the WOC component. The line chart represents the weights of the features. The upper state-history chart represents the clustering. The different colors represent different clustering algorithms or changes in the parameters. The lower state-history chart shows changes in the dimensionality reduction algorithms. The figure is taken from the original publication [8].

esting and do not influence the DR in the data space. The user removes these features by changing the range slider as indicated by the red arrow in step 5.1. The user sets the range of the support for a feature to *2 - 40*. Features that have fewer or more occurrences are removed by setting their weights to 0. This change only affects the outlier features. The remaining feature set contains 107 features and their similarity space is newly laid out (step 5.2). Note that this did not change the data similarity space in S^3.

Whilst browsing the features in the Pattern Selector, the user spots one feature *climb roof skylight* and repeats the cluster-mitosis step to obtain a cluster for this feature. Now, these features are described by the red cluster (step 5.3) - as they are redundant they are overplotted in S^4 (step 5.2, purple circle).

This use case was captured by the Weight Observer Component (WOC) as shown in Figure 5.7. Going from left to right, it shows the weights (importance) were increased for the features *window* and *door*. The upper state history chart then shows that a new clustering was triggered manually (change from light blue to orange). Afterward, the weight for feature *rear* was increased in step 2.2. A re-clustering was executed in step 2.3 which is visible in the change of color from orange to light orange in the upper state-history chart. The user experimented with the projections as shown in the lower state-history chart (steps 4.1 and 4.2). The tactical analyst proceeded with a drill-down for *climb roof* and then removed uninteresting features (their weight was changed to 0 in step 5.1). The cluster-mitosis step was repeated with the feature *skylight* for step 5.3.

5.3 Querying

Querying for patterns is powerful since it allows the user to validate expectations in the form of "is pattern X contained in the results". More sophisticated querying engines also allow searching for similar patterns. For sequential patterns, a regular expression-based engine is predestined, however, typically difficult for a user to understand. Therefore, visual approaches that hide this complexity have been introduced [154]. Depending on the domain, query interfaces may involve sketch-based approaches [259]. Klemettinen et al. already used a similar approach in 1994 but called it templating instead of querying [83]. They distinguished between positive/inclusive and negative/restrictive templates. For the former, the mining algorithm would specifically include association rules that match the template, for the latter it would exclude the specified rules. Note that this worked in combination with other thresholds of interestingness measures such as support and confidence.

[154]: Cappers et al. (2018), Exploring Multivariate Event Sequences Using Rules, Aggregations, and Selections

[259]: Seebacher et al. (2021), Investigating the Sketchplan: A Novel Way of Identifying Tactical Behavior in Massive Soccer Datasets

[83]: Klemettinen et al. (1994), Finding Interesting Rules from Large Sets of Discovered Association Rules

A pure query interface is somewhat contradictory to our used definition of EDA which mentions that EDA forcefully reveals patterns (see Chapter 1). A query interface that relies on querying only is not sufficient according to that definition as the user has to specifically search for certain things which may satisfy rapid hypothesis testing. However, querying capabilities in an exploratory system are tremendously helpful as they allow the user to verify expectations. Verification is an important aspect of knowledge generation [94] (see Figure 3.2 in section 3.2). section 5.6 will also discuss querying in the light of explainable artificial intelligence and trust building.

[94]: Sacha et al. (2014), Knowledge Generation Model for Visual Analytics

5.4 Visual Information Seeking Mantra

The popular visual information-seeking mantra by Ben Shneiderman states:

> "Overview first, zoom and filter, then details-on-demand." [250]

[250]: Shneiderman (1996), The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations

This mantra applies to the exploration of patterns as well. Arguably, the most challenging part when it comes to exploring patterns is the overview part. A human may never be capable of overviewing the sheer amount of patterns that can be generated in pattern mining. To worsen this, the previous chapter shows that if the structure in its detail is being visualized the scalability is vastly reduced to hundreds or fewer patterns. But how can we provide an overview of a pattern result set that scales better? I argue that the best way to do this is by exploiting interestingness measures, aka features. Even though they only show aspects, they can provide an insightful picture to the human if carefully selected. A proper selection, of course, largely depends on the data, user, and task (see section 3.2). However, even then we cannot scale indefinitely which is why filters such as thresholds for interestingness measures must be applied a-priori. Another approach is to use progressive visual pattern analytics (see section 5.5) to not immediately overwhelm the user with the amount of information. Note that I intentionally do not survey design aspects as this would be out of scope for this thesis but the previous chapter may provide some inspiration about different designs.

5.4.1 Distant reading, close reading

A term that aligns well with the visual information-seeking mantra is *distant reading, close reading*. The term distant reading was coined by Franco Moretti in 2000 [261]. There exists no crisp definition for the term but it broadly refers to the use of computational methods on documents and text corpora to gain insights. This is to free humans from the burden of reading hundreds or thousands of documents to gain relevant insight. Moretti provides one such example by measuring the length of the title of novels between 1740 and 1850 and showing that the average and median length decreases over time (Figure 5.8). Moretti correlates this with the expansion

[261]: Moretti (2000), Conjectures on world literature

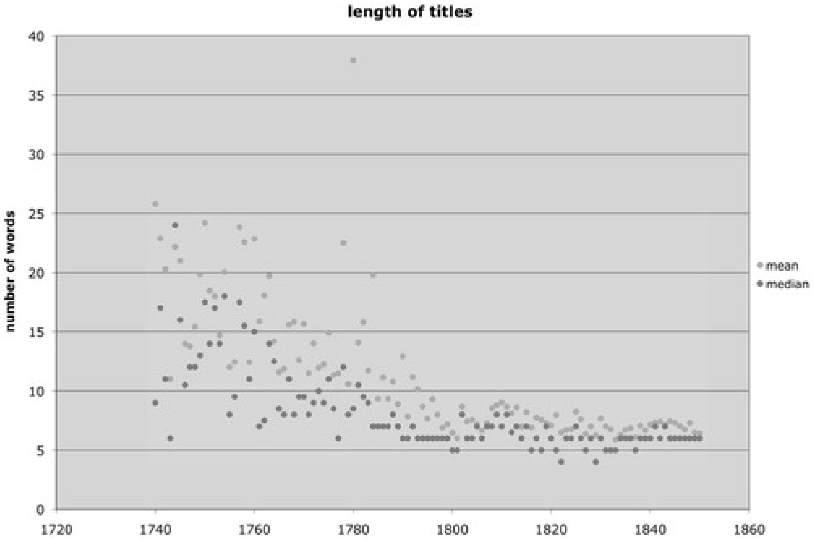


Figure 5.8: An example of distant reading by F. Moretti. The figure shows how the number of words in the title of novels between 1740 to 1850 became shorter on average [260].

of the markets which made publishing novels more competitive and "[...] they learn to compress meaning; and as they do that, they develop special "signals" to place books in the right market niche." [260]

[260]: Moretti (2009), Style, Inc. Reflections on Seven Thousand Titles (British Novels, 1740–1850)

These computational methods extract (numerical) features from the documents to analyze them and eventually gain insight and generate knowledge. Close reading on the other hand refers to the careful study of a text. This does not mean the opposite of distant reading in general but rather suggests that both techniques can be combined such that a user gains insights through distant reading and then can inspect a certain text or text passage in detail. The assumption is that the text (passage) is representative of the other documents belonging to the corpus or cluster.

This translates, of course, well to pattern mining since the interestingness measures of patterns are features that represent clusters. Distant reading thus refers to the analysis of these measures whereas close reading corresponds to the analysis of the pattern itself and the cluster it represents.

5.4.2 Example: Multi-Dimensional Pattern Exploration

For a task description and description of interestingness measures see subsection 3.4.3.

Requirement Analysis

[4] 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 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 allow us 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 has to 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.

[4] This section is taken from my publication "Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data" (Section 2.1: Requirement Analysis) [10]. I have been the main author of this publication and section and have written all the contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.

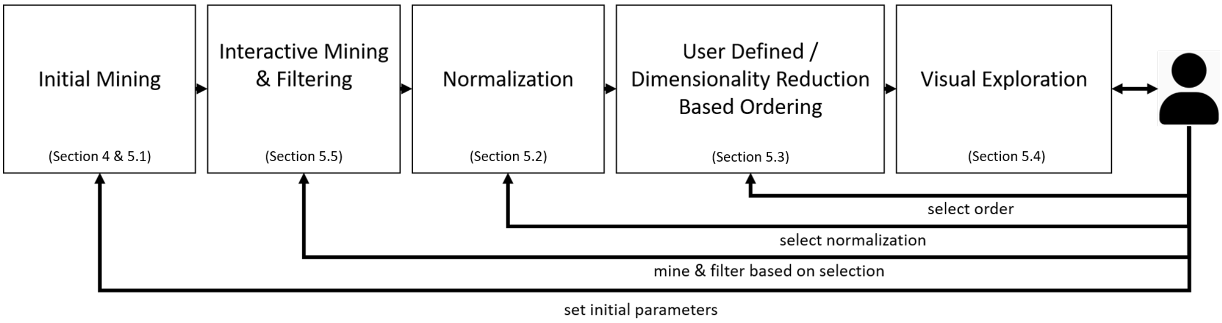


Figure 5.9: The processing pipeline of our implemented approach. The user can parameterize and influence every step. The results are propagated to the MDPE-vis.

Tool

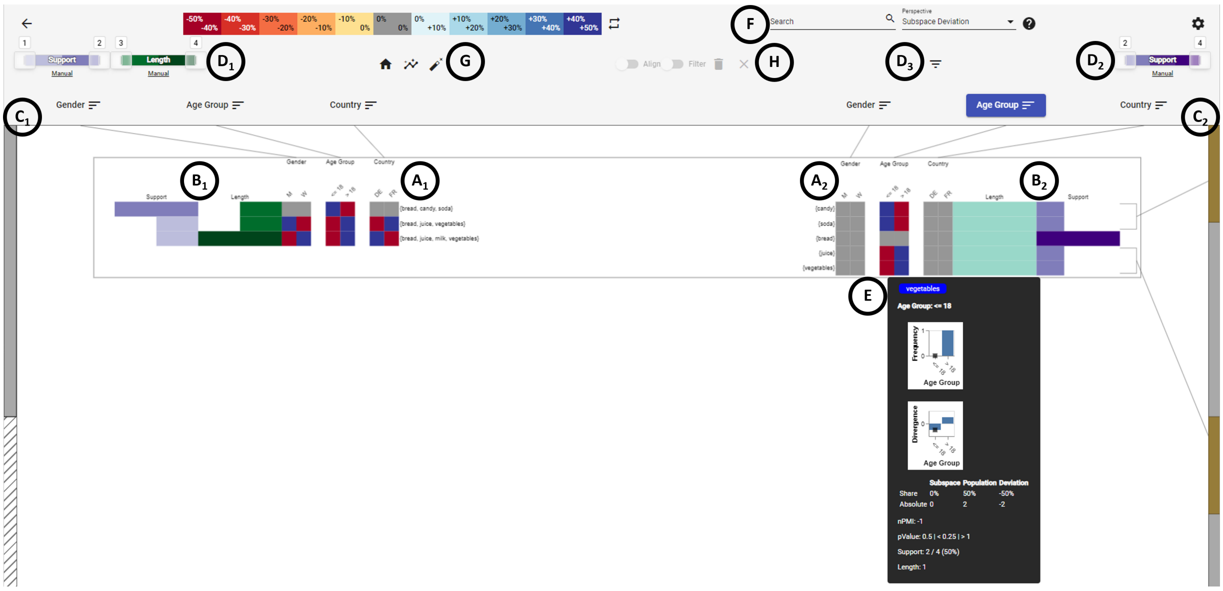
[5] Our interactive visual interface (MDPE-vis) implements the MDPE-approach described in the previous section. Figure 5.9 shows how the output from the *initial (pattern) mining*, in the form of two tables, is further processed before being visualized to the user. The *interactive mining & filtering* step is optional and will be described at the end of this section to improve the readability and clarity of the process.

Initial Mining

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 5.10 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 using 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

[5] This section is taken from my publication "Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data" (Section 5: Multi-Dimensional Pattern Exploration Visualization) [10]. I have been the main author of this publication and section and have written all the contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.



Age Group: <= 18

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

nPMI: -1

pValue: 0.5 | < 0.25 | > 1

Support: 2 / 4 (50%)

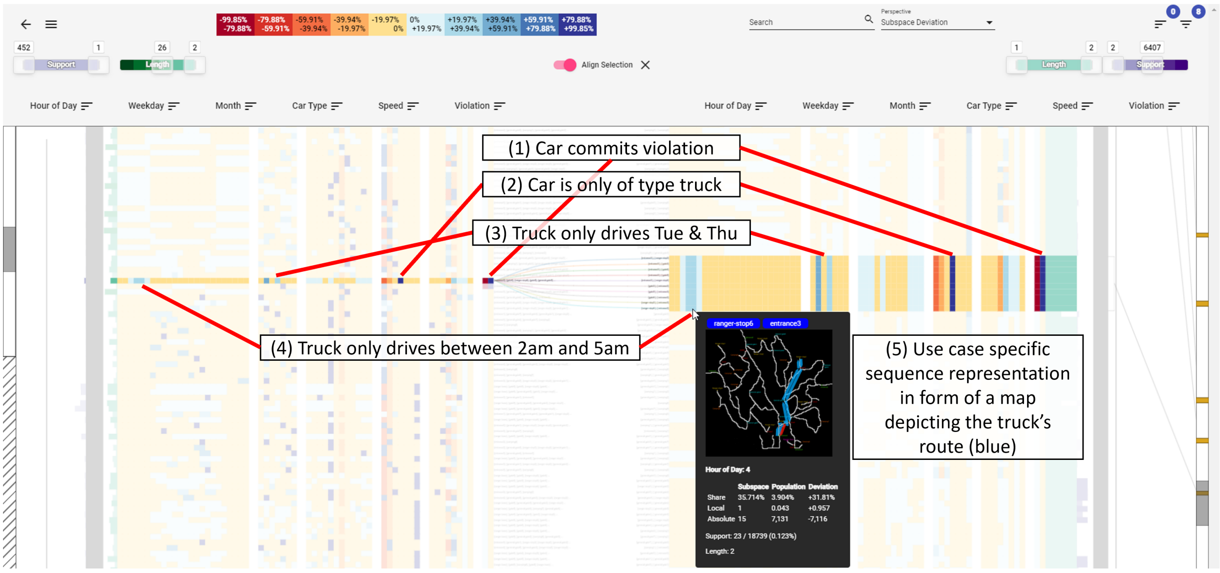
Length: 1

Figure 5.10: The overview of MDPE-vis displaying the output Table 3.4 (A_1) and Table 3.5 (B_1) of our approach. 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).

through the interactive mining capabilities (subsection 5.5.3) and because of the a-priori-property of the co-occurrences (subsection 3.4.3). 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.

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 3). The MDPE-approach also



(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

Figure 5.11: 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 in 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.

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) [262]. 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 an 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 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 3.9).

[262]: Van Der Maaten et al. (2009), Dimensionality reduction: A comparative review

Visual Exploration

The visual exploration is enabled by MDPE-vis (Figure 5.10). The figure shows the example data from subsection 3.4.3, more specifically Table 3.4 (A_1) and Table 3.5 (A_2). Note, that the order of the rows is different because of the applied dimensionality reduction as discussed in Section 5. 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 5.10.

(A) Pixel-based Tables The pixel-based tables represent the co-occurrence values of the tables based on our MDPE-approach in Section 3.4.3. 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 [263]. 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 5.10) refers to the output Table 3.4 where the distinct structured entities and their co-occurrences are listed. The right table (A_2) represents the data of Table 3.5. 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). The pixels are colored using a diverging colormap because of the relative subspace perspective. In the example case (Table 3.2), the co-occurrence vector for the entire population would contain 2 everywhere (Table 3.6). 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 $\frac{2}{22} = 50\%$, where the numerator is the co-occurrence value of the cell and the denominator is the sum of all co-occurrence values

[263]: Yang et al. (2022), The Pattern is in the Details: An Evaluation of Interaction Techniques for Locating, Searching, and Contextualizing Details in Multivariate Matrix Visualizations

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 3.5 in the attribute *Age Group*, the co-occurrence value for the attribute characteristic *Age Group:*≤ 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 5.11 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 shows all entities where the currently selected sub-entity is contained. 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 are represented as bar charts. This design is agnostic to any specific interestingness measure (**R2**). In the example, as shown in Figure 5.10, 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 Figure 3.7 and 3.8 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 5.10 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 3.7). 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 3.8) 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 5.11 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 5.11, 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 5.10, 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 occurs because the co-occurrence-vectors of these rows are equal (see Observation 2 in Section 12) and placed together because of the dimensionality reduction step (see Section 5). A *visible block* is formed if two or more rows have an equal co-occurrence vector. These blocks do not always stand 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 5.10 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 5.10, D). The user can also filter for specific values (D_3), or visually speaking, 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 preserves 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 [159]. 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 5.10 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).

[159]: Stolper et al. (2014), Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics

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 [83, 218]. This is also described as a design goal by Stolper et al. [159].

[83]: Klemettinen et al. (1994), Finding Interesting Rules from Large Sets of Discovered Association Rules
[218]: Liu et al. (2000), Analyzing the Subjective Interestingness of Association Rules
[159]: Stolper et al. (2014), Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics

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 follow 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 itemsets themselves are aligned horizontally. For the VAST Challenge 2017 use case (see Section 3), 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 5.11). 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 12) 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 5.10, the support for the sub-entity *{candy}* is 2, which is 50% relative support because the input data consists of 4 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 the users to verify preexisting knowledge about the data and better learn and understand how the application works and how visual patterns can be interpreted.

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.

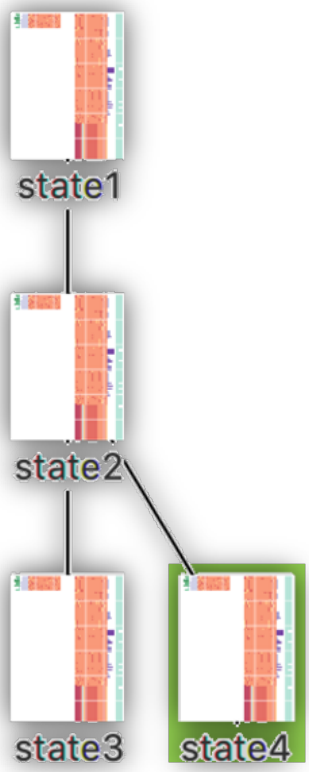


Figure 5.12: The provenance view allows the user to backtrack to various states (i.e., filter and order settings).

On the right side are two buttons 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.

[264]: North et al. (2011), Analytic provenance: process+interaction+insight

MDPE-vis supports analytic provenance [264]. This is not only useful for analytic reasoning, justification, and reporting but also practical to the user as it provides an "undo" functionality. Such a feature is useful as the user can filter the data and may want to revert to an earlier state in the analysis to choose a different path in the exploration. We support this with a state metaphor. A state is a combination of filters and sorting preferences, plus settings such as the currently selected perspective. At any point in time, the user can open a state panel and save the current state. A snapshot of the current view is automatically generated and combined with a user-defined label. The state representation is added to the panel in the form of an icon plus the label. New states are added below the last state and form a sequence. If the user reverts to an older state and saves this again as a new state, a branching will occur forming a tree as it is visible on the right. This allows the user to branch out and try different paths in the exploration. The active state is highlighted.

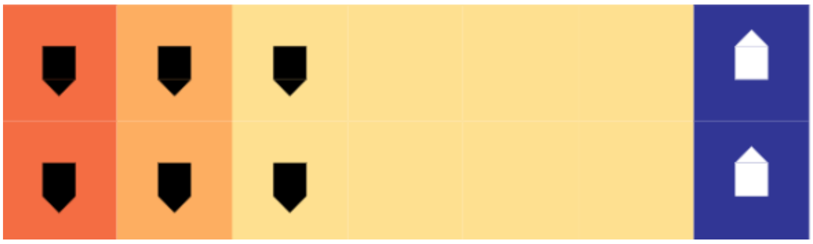
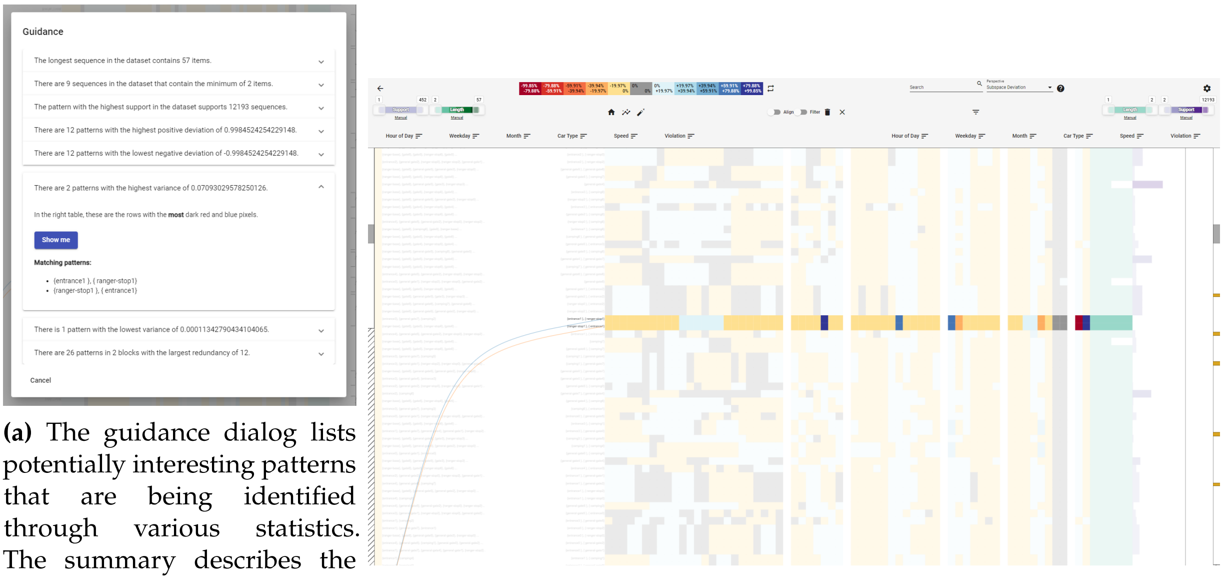


Figure 5.13: Glyphs on top of the pixels indicate whether the deviation is statistically significant.



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.

Figure 5.14: 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.

(*G*) Statistical Overlay and Guidance The leftmost 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 5.10 (G)), opens a dialog window (Figure 5.14a) which entails a list of possibly interesting patterns. Their interestingness 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 most blue and red 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 4.5, 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.

5.4.3 Example: QuestionComb

[6] QuestionComb is an application that allows the user to label questions as information-seeking and non-information-seeking (see subsection 3.4.2). The interface consists of several views where the user can select, label, and group instances (see Figure 5.15). An instance is a question plus its context consisting of the utterance before and after the question plus the speaker’s information. In the instance selection view, the user can select the next question deemed appropriate to label. The selected instance is shown in the instance labeling

[6]: This section is based on the publication “QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling” (Section 5: QuestionComb: The Interface) [9]. I have co-authored this publication and contributed to the data modeling as well as the pattern mining components. The paper is a joint effort of Rita Sevastjanova, Fabian Sperrle, Rebecca Kehlbeck, Jürgen Bernard, and Mennatallah El-Assady.

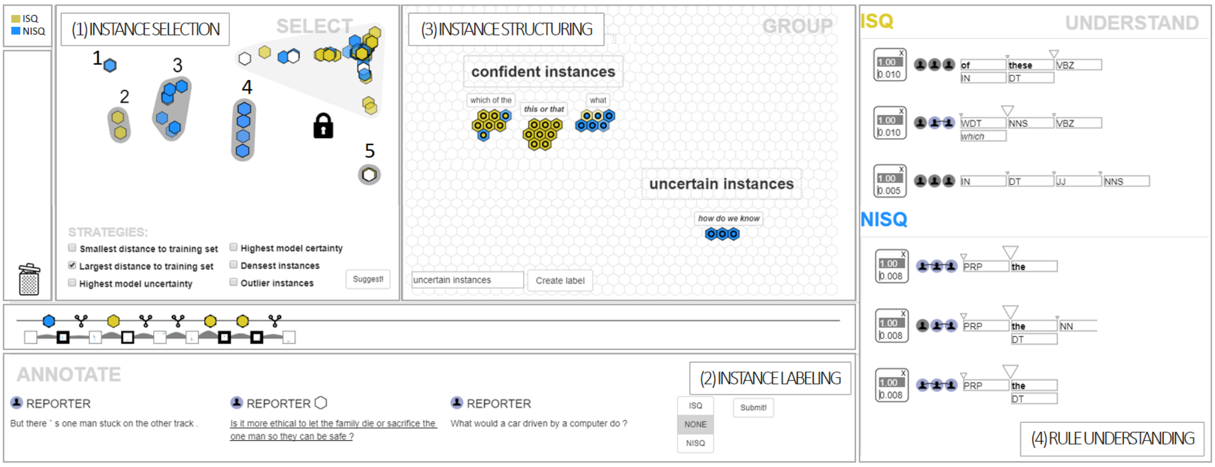


Figure 5.15: The QuestionComb interface consists of instances (i.e., questions with context) that are presented in the instance selection view (1). The annotation view (2) shows one instance (i.e., the question with the utterance before and after plus speaker information) and lets the user assign a label. The instance structuring view lets the user group the questions into custom clusters and assign the clusters with custom labels (3). The rule view consists of two panels that divide the rules into information-seeking questions and non-information-seeking questions (4). The figure is taken from the original publication [9].

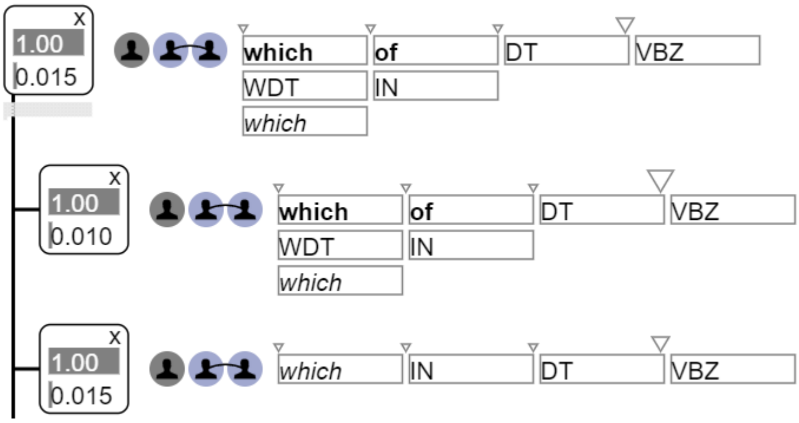


Figure 5.16: The rules are being visualized in hierarchical form. Maximal rules are visualized on the top-level hierarchy and the user can visualize all contained rules. The figure is taken from the original publication [9].

view where the utterance before and after are displayed and the relevant question is in between. The user can label the question which changes the instance's color in all views. In the instance structuring view, the user can move instances and group them. Then the groups can be labeled individually. The rule view is divided into two panels representing the respective labels. The rules in these labels are sequential rules.

The rules in the panels are maximal rules (see section 2.3) to reduce the number of rules being displayed and not overwhelm the user [265]. Figure 5.16 shows three rules whereas the top rule is a maximal rule and the user has opened up the rules that are contained in this rule. The

[265]: Ham et al. (2009), "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest

maximal rule could be formalized as

$$\langle\{\text{which}, \text{WDT}, \text{which}\}, \{\text{of}, \text{IN}\}, \{\text{DT}\}, \{\text{VBZ}\}\rangle \rightarrow \langle\{ISQ\}\rangle \tag{5.1}$$

The visual representation shows the items in one itemset vertically and the itemsets themselves horizontally. The font style of the items indicates the type such as bold font for the lemmatized tokens, question words are italic, and POS tags have a normal font. This design is inspired by Brath et al. [266]. Between the itemsets, the size of the triangles indicates the maximum distance between the itemsets of all sequences the rule represents. Here, the minimum is 1 and the maximum is 5 as this is a threshold set when mining the sequences (see subsection 3.4.2). This design is inspired by Chen et al. [15] (see Figure 4.23 and subsection 4.3.3). The numbers in the box represent the interestingness measures *confidence* (see Definition 3.3.3) and *support* (see Definition 3.3.2). The number with the gray background represents the confidence value. The confidence ranges from $0.95, 1$ since the minimum confidence is set to 95% (see subsection 3.4.2). The support ranges from $0.01, 1$ since the minimum support is set to 1%.

[266]: Brath et al. (2014), Using font attributes in knowledge maps and information retrieval

[15]: Chen et al. (2018), Sequence Synopsis: Optimize Visual Summary of Temporal Event Data

Note that the support value is static whereas the confidence value is continuously updated whenever the user labels a question. Therefore, the list of maximal rules changes whenever the user assigns a label to one instance. The lists are sorted by the rules with the highest confidence and the highest support such that the user has immediate feedback about the model that is being created.

5.4.4 Conclusions

Ben Shneiderman's information visualization mantra is very general and does not detail any specific designs in order to reach that goal. The examples from the multi-dimensional pattern exploration (MDPE) and QuestionComb show possible designs on how to achieve overview first, zoom and filter, and details on demand. The former example leans more towards distant and close reading by not visualizing the structure in the overview at all but only interestingness measures in the form of co-occurrences (or other correlation measures) and common interestingness measures such as length and support. Visualizing only interestingness measures empowers the tool on its scalability which could not be achieved by visualizing structures only or in combination with interestingness measures. The approach also leverages the information from the lattice by ordering the rows according to their co-occurrence vectors effectively placing patterns of one equivalence class (see section 2.3) below each other creating visual blocks. These blocks provide visual cues and become more apparent the greater the equivalence class is. The zooming is quite literally supported as the whole visualization is drawn on a zoomable and pannable canvas. The canvas performs semantic zooming where the structure of the patterns becomes visible at high zoom levels. Filtering is directly supported using filters on the interestingness measures (sliders) which do not require a recalculation of the mining algorithm and are thus fast to compute and provide immediate feedback to the user. Filtering can be performed using minimal and maximal thresholds. A search box allows the user to query for specific patterns and is important for rapid hypothesis testing as well as verification. The structure itself provided by the semantic zooming can be interpreted as details on demand, however, a tooltip provides even more detailed information on structure and co-occurrence metrics.

The QuestionComb tool follows another approach to provide an overview exploiting equivalence classes and the lattice itself using maximal patterns. This helps to reduce the overall number of patterns and leaves only a few patterns that need to be visualized in the tool allowing us to visualize the structure of the rules themselves. This is important in this instance, as the users are linguistic experts and they need to understand the model they are creating while labeling the instances (questions). We, therefore, find it difficult to abstract from this information. The details on demand are in this instance the rules themselves that are contained in a maximal rule.

The VALCRI concept explorer (see subsection 5.2.1) uses yet another approach as overview- and detail-visualizations are placed side-by-side and connected through linking and brushing techniques. Views such as S^4 provide an overview while, again, only visualizing interestingness measures (support as opacity and length as the width of the rectangle).

Also, this tool exploits the lattice and places the patterns according to their distances on the scatterplot. Patterns of one equivalence class are therefore on top of each other (depending on the dimensionality reduction algorithm used). As in the MDPE-approach, the concept explorer also allows the user to apply minimum and maximum thresholds hiding the respective patterns in all visualizations.

5.5 Progressive Visual Pattern Analytics

Progressive Visual Analytics (PVA) is a subfield of visual analytics that is dedicated to showing the user intermediate results and allowing the user to steer running processes to generate knowledge more efficiently. The term has been coined by Stolper et al. [159], however, similar approaches have been seen before such as by Williams and Munzner in their MDSteer tool [267]. Jean-Daniel Fekete is an active and well-known researcher in this field with many significant publications [268–271]. A doctoral thesis was published in 2020 by Vincent Raveneau on the topic of progressive visual analytics for sequential pattern mining [272]. The scope of my thesis goes beyond the topic of PVA for sequential pattern mining but at the same time acknowledges its importance for the exploration of patterns.

The scope of this section is specifically targeted to approaches that involve the human in the mining process itself (see Figure 5.17).

[159]: Stolper et al. (2014), Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics

[267]: Williams et al. (2004), Steerable, Progressive Multidimensional Scaling

[268]: Zgraggen et al. (2017), How Progressive Visualizations Affect Exploratory Analysis

[269]: Fekete et al. (2016), Progressive Analytics: A Computation Paradigm for Exploratory Data Analysis

[270]: Fekete et al. (2018), Progressive Data Analysis and Visualization (Dagstuhl Seminar 18411)

[271]: Turkay et al. (2018), Progressive Data Science: Potential and Challenges

[272]: Raveneau (2020), Interaction in Progressive Visual Analytics. An application to progressive sequential pattern mining. (Interaction en Analyse Visuelle Progressive. Une application à la fouille progressive de motifs séquentiels)

5.5.1 Incremental, generation-based mining

Pattern mining can be easily adapted to be incremental. The major action is to change the algorithms such as SPAM [77] or SPADE [273] (for sequential pattern mining) from a depth-first-search to a breadth-first-search [159]. These algorithms automatically produce new generations (see Definition 2.1.7), therefore, the algorithms must only be halted after an entire

[77]: Ayres et al. (2002), Sequential PAttern mining using a bitmap representation

[273]: Zaki (2001), SPADE: An Efficient Algorithm for Mining Frequent Sequences

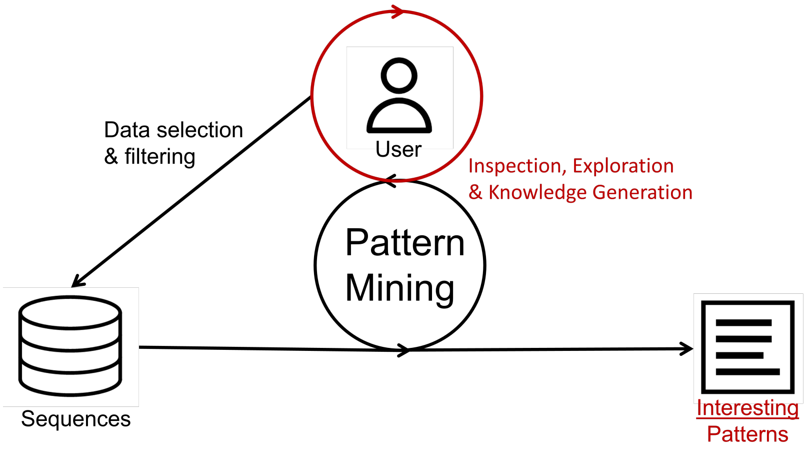


Figure 5.17: Progressive Visual Pattern Analytics. The user is taken into the loop of pattern mining to explore the patterns while they are being generated. This allows the user to influence the mining process early on to, optimally, only receive interesting patterns.

generation has been calculated. However, an incremental algorithm is not sufficient to be classified as progressive as it does not allow the user to include their feedback [272]. Furthermore, it does not prevent any type of pattern explosion but only delays it since eventually generations with many patterns will be mined (see section 2.2).

[159]: Stolper et al. (2014), Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics

[272]: Raveneau (2020), Interaction in Progressive Visual Analytics. An application to progressive sequential pattern mining. (Interaction en Analyse Visuelle Progressive. Une application à la fouille progressive de motifs séquentiels)

5.5.2 Prefix and Suffix Mining

This type of mining is not restricted to sequences as a total order of items in sets can be defined without loss of generality. This is also known as pattern growth and is used in mining algorithms such as SPAM [77]. Vrotsou et al. use this interactive mining technique by letting the user extend event sequences (sequential patterns) while linking other information of the data to the currently created sequences[274]. Their graph representation lets the user append another event (itemset) as a prefix or suffix to a sequence eventually creating a graph (see Figure 4.21). Users typically have a good understanding of such extensions which are more intuitive than the general containment definition used in pattern mining (further discussed in section 5.6). This type of interaction involves the user maximally but there are two limitations to this method: Firstly, this approach is not directly usable for sequences that contain multiple items in one itemset as the representation only represents linear sequences. Secondly, the user might not discover all important patterns since a pattern $\langle\{a\},\{b\},\{c\}\rangle$ can only be created with the patterns $\langle\{a\},\{b\}\rangle$ and extending $\{c\}$ as a suffix or $\langle\{b\},\{c\}\rangle$ or by appending $\{a\}$ as a prefix. It cannot be created if the current pattern is $\langle\{a\},\{c\}\rangle$.

[77]: Ayres et al. (2002), Sequential PAttern mining using a bitmap representation

[274]: Vrotsou et al. (2014), Interactive visual sequence mining based on pattern-growth

5.5.3 Multi-selection-based Mining

This technique is a combination of (multi-) selection-based mining and incremental, generation-based mining. It allows the user to select one or more patterns from a current generation and mine for the next higher generation that contains the selected patterns. There are some technical peculiarities that I describe in the following. The task and interestingness measures description of this technique is in subsection 3.4.3 and the tool is described in section 4.

[7] The user can drill down into the search space using interactive mining, which is the implementation of the described Action 3 in Section 3.4.3. 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 Definition 2.1.7) should be mined. The default and recommended value is to mine only the next higher generation of the selection.

7: This section is taken from my publication "Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data" (Section 5.3: Interactive Mining & Filtering) [10]. I have been the main author of this publication and section and have written all the contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.

Our algorithmic basis is the CM-SPAM algorithm by Fournier et al. as it shows good performance with dense datasets [221]. The CMAP approach of Fournier et al. is an extension to the SPAM algorithm [77] 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.

[221]: Fournier-Viger et al. (2014), Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information

[77]: Ayres et al. (2002), Sequential PAttern mining using a bitmap representation

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. [158] which allows us to effectively mine structured sub-entities by generation (see section 2.2). 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

[158]: Perer et al. (2014), Frequence: interactive mining and visualization of temporal frequent event sequences

of structured sub-entities of the current highest generation: $p \in P_k \subset P | lengthp = k = l_{MAX}$. 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_{kx}. 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 3.4, Figure 5.10 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: $minSup = \min_{p \in P_k} supportp$. 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 3.5, Figure 5.10 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 $\{p_l \in P_l \mid support p \geq s_{MIN} \wedge length p = l\}$. 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 $\mathcal{O}m$ 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 5.10 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.

5.5.4 Representation Learning

Van Leeuwen provides an overview and more details about these approaches [275]. The general idea is to present a subset of patterns to the user and let the user label these patterns (i.e., interesting and not interesting) [276] or let the user provide a user-defined ranking of these patterns [277]. Then the question is how to proceed with the information the user supplied. One possibility is to adapt weights that are in turn used to specify a sampling distribution [276]. Another possibility is to use the lattice itself and define a distance measure where either similar patterns or dissimilar patterns are proposed [277]. Also, a domain-specific distance measure could be used to achieve the same task [278]. Another

[275]: Leeuwen (2014), Interactive Data Exploration Using Pattern Mining

[276]: Bhuiyan et al. (2012), Interactive pattern mining on hidden data: a sampling-based solution

[277]: Dzyuba et al. (2013), Active Preference Learning for Ranking Patterns

[278]: Galbrun et al. (2012), A case of visual and interactive data analysis: Geospatial redescription mining

possibility is to use various types of preference learning to derive an interestingness measure from the user's feedback. A simple approach would be to correlate a user-supplied ranking with various interestingness measures. Boley et al. use a combined strategy by learning which algorithm produces the best results and by learning a utility function over the feature representation [279]. The best-deemed algorithm then efficiently generates the best-tailored result set, thus, saving computation time. The learned utility function then provides a ranking of the most promising patterns that the user might be interested in. Dzyuba et al. assume that a user can prefer one pattern over another but cannot express this preference for any pattern pairs [277]. The user is asked to rank a small subset of the patterns and the utility function over the feature representation and then tries to derive a total order from that. The main problem of these approaches is the utility function and available features as they need to be defined a-priori and thus introduce a bias [275].

[279]: Boley et al. (2013), One click mining: interactive local pattern discovery through implicit preference and performance learning

[277]: Dzyuba et al. (2013), Active Preference Learning for Ranking Patterns

[275]: Leeuwen (2014), Interactive Data Exploration Using Pattern Mining

5.5.5 Belief system

Subjective interestingness measures (see Chapter 3) try to formalize existing knowledge to use it to find either pattern that the user is looking for (i.e., similar to the input) or try to avoid such similar patterns to "surprise" the user. The main challenge is how existing knowledge can be formalized as this heavily depends on the data and task. Lastly, the user must be able to understand how they can formalize their knowledge and this is typically a time-consuming endeavor. In data mining, such formalized domain knowledge is typically referred to as a belief system. De Bie conceptualized a framework that is based on maximum entropy where a prior probability distribution can be created through a belief system [280]. The patterns that are interesting according to this distribution can be presented to the user and re-evaluated. The input of the user is then used to update the maximum entropy-based probability distribution. While such an approach lifts the burden of defining a custom utility function, it does require certain normalizations and data, and task-specific adjustments [281, 282].

[280]: Bie (2011), An information theoretic framework for data mining

[281]: Bie (2011), Maximum entropy models and subjective interestingness: an application to tiles in binary databases

[282]: Spyropoulou et al. (2014), Interesting pattern mining in multi-relational data

5.5.6 Conclusions

The most challenging problem in progressive and interactive pattern mining is the initial sample of patterns that the user must evaluate. This sample must be representative and yet small enough to not discourage the user. The next problem is that the mining based on this sample may still produce a large number of patterns if thresholds or utility functions are set wrong. In my experience, the multi-selection approach combined with generation-based mining works well as users do not tend to select too many patterns at once and typically leave the default of only mining the next generation. A simple interaction to delete uninteresting patterns (i.e., hiding them from the interface) also prevents the user from selecting these patterns to mine deeper into the search space. The selection-based approach also does not require any utility function or belief system and the user can freely choose to perform the mining based on the semantics of the pattern (or the underlying data) which may not be captured in any interestingness measure.

5.6 Explainable Artificial Intelligence

[283]: Gunning et al. (2019), XAI - Explainable artificial intelligence

One might wonder how the popular topic of explainable artificial intelligence (xAI) [283] relates to pattern mining. After all, pattern mining is typically a deterministic approach of incremental algorithms where every step and every solution can be traced back to the original data. While this is true, a person outside of the fields of computer science or machine learning may not have the relevant knowledge about how these algorithms work and thus perceive such systems as a black box.

5.6.1 Understanding Patterns

This already starts with the interpretation of a pattern itself. For example for a sequential pattern $\langle\{a\},\{b\}\rangle$ I try to avoid saying "*b* is followed by *a*" or "*a* before *b*" as many users imply that *b* immediately follows *a*. This is, however, not true as the pattern without any additional constraints (e.g., window constraint) allows for any number of events in between *a* and *b*. I, therefore, try saying "*a* occurs anywhere before *b*". It may not be immediately clear but it at least sparks questions from the user referring to that topic which allows me to explain it in more detail.

5.6.2 Understanding Interestingness Measures

Interestingness measures are typically used to filter and rank patterns. Therefore, the user typically has to estimate parameters, such as thresholds, to tune a system. Since interestingness measures are essentially quantified properties of patterns or the cluster they represent a user must have at least a *conceptual* understanding of what this measure represents. Not only this, but furthermore, the user must understand what certain thresholds and the combination of them, imply.

5.6.3 Understanding the Lattice

The lattice (see section 2.2) and the concept of containment often seem complex to the users. This is mostly because of the

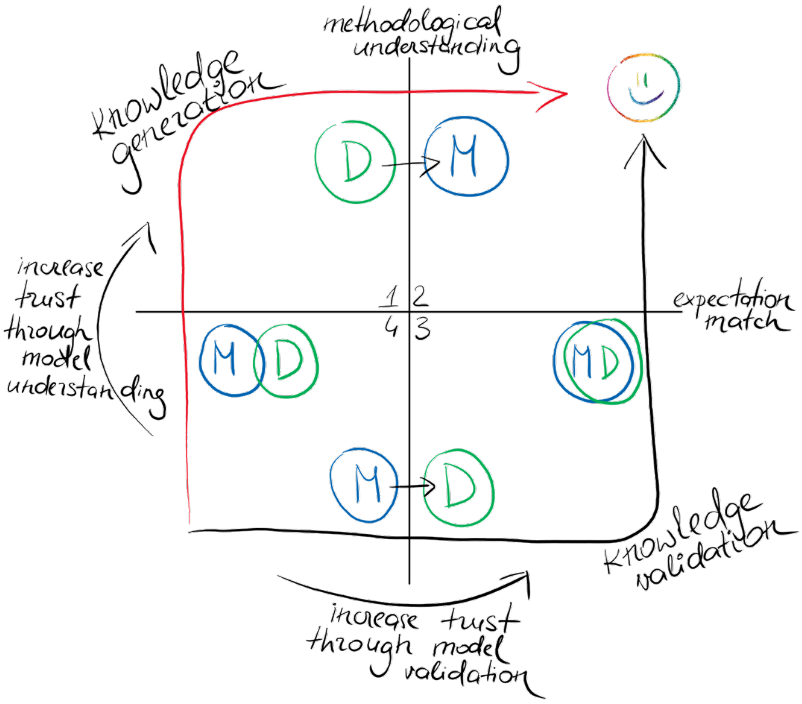

methodological understanding
knowledge generation
D
M
increase trust through model understanding
1
2
4
3
expectation match
M
D
M D
M
D
knowledge validation
increase trust through model validation

Figure 5.18: Our proposed trust-building model illustrates the trust-building process using two orthogonal concepts: *methodological understanding* and *expectation match*. The figure is taken from the original publication [21].

partial order and that one pattern can have multiple parents (contained patterns) which is contrary to a much easier concept of prefix or suffix extensions. However, the lattice contains important information about similarities of patterns (see section 2.3). I find it useful to represent this information in a continuous space such as scatterplots (subsection 5.2.1) or reduced to one dimension (see subsection 5.4.2) to represent this similarity. It is a much more natural understanding for the user of how these patterns are related and lifts the complexity of the partial order and containment-specific relationships of the patterns.

5.6.4 Understanding Concepts & Metaphorical Narratives

In the following, we argue that a conceptual understanding also called methodological understanding is key for any system.

[8] The application of metaphorical narratives positively influences the trust-building process of the domain expert. We argue that the trust-building of a domain expert in an AI model can be decomposed into two major dimensions. We hereby assume that the user has a general motivation to work with the application as it promises to ease her daily routines

8: This section is taken from my publication 'Minions, Sheep, and Fruits: Metaphorical Narratives to Explain Artificial Intelligence and Build Trust" (Section 3.2 Trust-Building Model) [21]. I have been the main author of this publication and section and have written all the contents. The paper was co-authored by Rita Sevastjanova, Florian Stoffel, Daniel Keim, Jürgen Bernard, and Mennatallah El-Assady.

and provide more insights into some available data. Our proposed trust-building model is depicted in Figure 5.18. The dimension, shown on the x-axis, is called expectation match. Typically, domain experts have a good understanding of what to expect as the outcome of some given system according to their expertise. We denote the expectation match as two intersecting sets whereas set M represents the output of the system and D is the output as it is expected by the domain expert. The expectations arise from the respective domain knowledge of the task and the data that are provided by the domain expert. An increasing expectation match is visualized in the chart from left to right. Quantifying this dimension is not trivial due to the facts that: (i) it is often difficult for domain experts to fully formalize their expectations and (ii) the output of any system is typically not consumed directly but through interpreting different (interactive) visualizations whereas the interpretation is affected by many occurring biases [284]. Advancement to the right of the chart can be performed in two ways. The first one is to modify the model such that the output of the model changes. We denote this as $M \rightarrow D$. The second way is an adaption of the user's expectations which we refer to as $D \rightarrow M$. Both are not exclusive and may happen simultaneously in practice. The dimension depicted on the y-axis represents methodological understanding. We hereby explicitly refer to the complete system including all used AI models plus the visualizations and interaction possibilities. Furthermore, methodological understanding refers to an understanding of what the system as a whole is performing, how the data is transformed during the process, and how this is associated with the given task(s). A position at the bottom thus depicts a user with no methodological understanding of something which is also typically referred to as a black box. The other extreme in the upper region of the chart is a user that has a full understanding of how the data is transformed and how the results are being generated and can be explained. The resulting four quadrants describe the states of the user concerning the system and can explain how the user possibly reacts. Quadrant 4 describes a user with no methodological understanding and no expectation match. Through interviews and observations of the domain experts, a typical reaction in such a case is the repetition of the analysis process to see whether the output of the system is changing or not. This may also include various, random parameter settings. However, if the output does not

[284]: (2018), Cognitive Biases in Visualizations

increase the expectation match, the users discontinue using the system (or the respective part of the system) and explore alternative ways to receive the expected output. This might be even to an extent where the data is processed manually. We, therefore, consider this a state where a user has no trust in the system. Quadrant 3 refers to a domain expert who does not have any or little methodological understanding but the output of the system matches the expectations. While the user might have trust in this system, it gives the modeling expert great powers - and responsibilities. From a pessimistic perspective also the great ability to manipulate the user. This situation is, however, not uncommon as we can experience this in many commercial products of our everyday lives, for example, in recommender systems of online shops, search engines, and social networks. Such systems try to continuously adapt their output towards the user's expectations which imposes a high risk of including the user's biases and not producing objectively correct results. The consequence of this phenomenon is also called a "filter bubble." Quadrant 2 is the desired state as only here the user can effectively use the system as the underlying methods are understood, and the output of the system is valid from the user's point of view. We consider this as the quadrant with the highest trust in the system and where it is likely that the best conditions exist to generate more knowledge and validate existing knowledge. This is possible by using different data where the expected output is little or unknown and by varying parameter settings. Ultimately, the user should have understood the limitations of the system and the underlying methods. Lee et al. name this state a calibrated trust [285]. We consider quadrant 1 as an intermediate state where the user has a high methodological understanding but the output of the system does not match the user's expectations. However, trust in the system is likely to be high. The user is therefore possibly motivated to validate the used models and processes or even check the implementation for errors. We refer to this process as debugging the system. In the case of finding an error on the concept-, implementation-, or even data level, the user is adapting the model ($M \rightarrow D$) and thus progressing towards quadrant 2. If no errors can be discovered the user might be even willing to adapt her expectations towards the output ($D \rightarrow M$). This is mainly due to the higher trust in the system as compared to the bottom of the chart. An advancement from quadrant 4 to quadrant 3 is possible but probably not as

[285]: Lee et al. (2004), Trust in Automation: Designing for Appropriate Reliance

efficient. In this case, the model might randomly change the output due to the random parameter settings set by the user or the model applies an active learning methodology which typically only gradually changes the output. As the trust is missing the user will not be as persistent in using the system. In general, we consider this transition to be slower than from quadrants 1 to 2. We propose metaphorical narratives as a method to elevate the domain expert in her methodological understanding. In Figure 5.18 this would result in transitions from quadrant 4 to 1 or 3 to 2, respectively. We further argue that a movement as depicted by the red arrow (Figure 5.18) is ideal for two reasons. First, the domain expert can validate the methods and may discover that some applied AI models are not suitable for the given task. This is especially important in the earlier stages of the design study and helps to prevent the time-consuming development of systems that turn out to be ineffective in supporting the domain expert in her tasks. Second, the user might be willing to adapt her expectations ($D \rightarrow M$). We consider the second effect as an essential part of the knowledge-generation process. While a state depicted by quadrant 3 is not desirable for the analysis of data in a scientific manner, the metaphorical narratives can be used to transition to quadrant 2 (black arrow).

We propose metaphorical narratives as a method to elevate the domain expert in her methodological understanding. In Figure 5.18 this would result in transitions from quadrant 4 to 1 or 3 to 2, respectively. We further argue that a movement as depicted by the red arrow (Figure 5.18) is ideal for two reasons. First, the domain expert can validate the methods and may discover that some applied AI models are not suitable for the given task. This is especially important in the earlier stages of the design study and helps to prevent the time-consuming development of systems that turn out to be ineffective in supporting the domain expert in her tasks. Second, the user might be willing to adapt her expectations ($D \rightarrow M$). We consider the second effect as an essential part of the knowledge-generation process. While a state depicted by quadrant 3 is not desirable for the analysis of data in a scientific manner, the metaphorical narratives can be used to transition to quadrant 2 (black arrow).

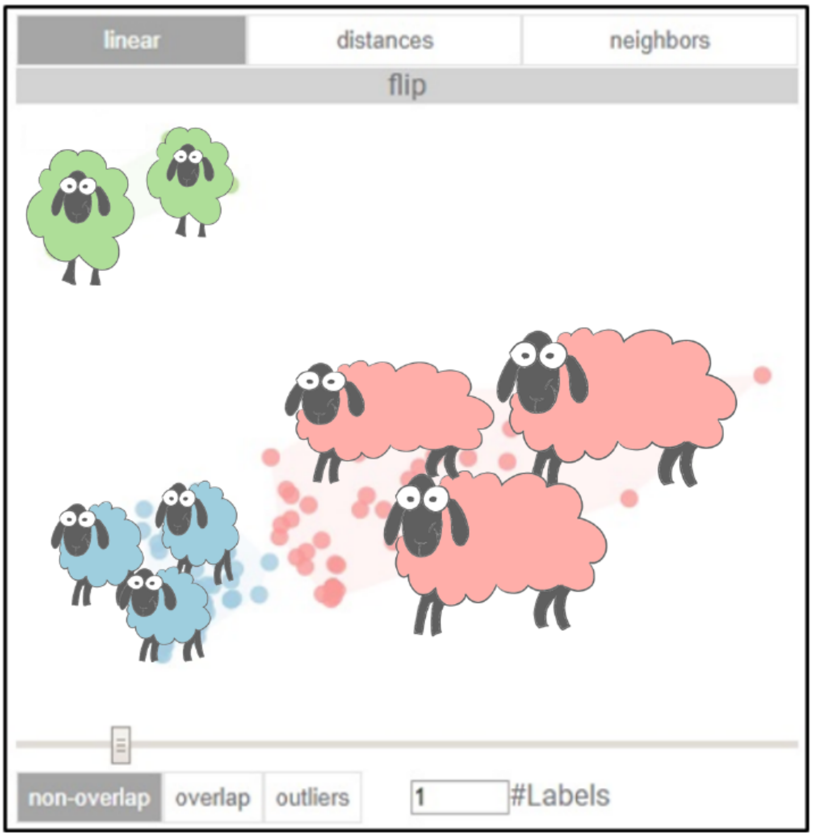


Figure 5.19: The concept of a projection plot in combination with visual clustering is explained by the metaphorical narrative of a sheep run. The figure is taken from the original publication [21]. Image credit: Rita Sevastjanova

Example Metaphorical Narrative This example is based on our VALCRI prototype (see subsection 3.4.1 & subsection 5.2.1). We used this metaphorical narrative after one user stated "your clustering scares me to death!". We observed that users were very hesitant to use the system as they feared breaking something or getting into a state in the system they could not get out of. The metaphorical narrative helped the users to gain a conceptual understanding of how central parts of the concept explorer are functioning and improved their trust in the system significantly.

[9] The Concept Explorer combines multiple complex AI models to support the criminal investigator in its Comparative Case Analysis task [8]. Two central AI models are dimensionality reduction techniques with weighted feature vectors and visual clustering techniques that operate on the low-dimensional output of the dimensionality reduction model. To explain the difference between both methods and the general concepts behind them, we chose the metaphorical narrative of a flock of sheep (Figure 5.19). Sheep have different attributes such as size, length, and height. We explained that the domain expert can tell the shepherd what attributes she

9: This section is taken from my publication 'Minions, Sheep, and Fruits: Metaphorical Narratives to Explain Artificial Intelligence and Build Trust" (Section 3.3: Exemplary Metaphorical Narratives) [21]. I have been the main author of this publication and section and have written all the contents. The paper was co-authored by Rita Sevastjanova, Florian Stoffel, Daniel Keim, Jürgen Bernard, and Mennatallah El-Assady.

[8]: Jentner et al. (2018), Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool

considers more or less important. The shepherd tries to place the sheep onto the sheep run based on how similar the sheep are according to their user-weighted attributes. Afterward, the user provides the shepherd with a set of colors. The shepherd tries to find groups of sheep on the sheep run without looking at their attributes and assigns each group one of the colors. The user can investigate and explore the groups, look at the distinctive attributes or find attributes that are shared among different groups. After teaching the basic concepts of dimensionality reduction and clustering techniques, the domain experts started to use the tool with much more confidence. The evaluations after establishing the metaphorical narrative showed that the users ceased their wishes for more guidance from the tool and observations confirmed the now more exploratory data analysis with the support of the system.

Conclusions

Although this is not a popular research area for xAI, it is crucial for pattern mining as well. In the end, the user is the ultimate decision maker judging the interestingness of a pattern and therefore must have an idea of what a pattern represents and what, in turn, the interestingness measures (i.e., properties) of a pattern represent. It is not required that a user understands the details of pattern mining and interestingness measures in detail but rather has an idea of the most crucial concepts. Another crucial part is the expectation match. A simple example would be a black box system with two dials and a display showing a number. The task of the user is to turn both dials to get a specific value showing on the display. If the dials are variables for a linear model, the user will soon understand the concept but not necessarily the underlying formula and should be able to reach the number eventually. For non-linear models, this may be much harder if not impossible and the user will likely find no underlying pattern to these dials and the output. After a while, the user will simply be discouraged from using the system any longer as they deem it not working properly. Translated to pattern mining this might be simply a pattern that a user expects to be in the result set. This is for example in contrast to the aspect of surprisingness or unexpectedness of interestingness measures (see subsection 3.1.1).

We suggest metaphorical narratives as one tool to convey concepts of a more complex system. However, these are likely not the only possibility. In another work, we describe the building blocks of xAI [42].

[42]: El-Assady et al. (2019), Towards XAI: structuring the processes of explanations

5.7 Evaluating Visual Pattern Analytics

The following presents one user study conducted for the multi-dimensional pattern exploration tool. See subsection 3.4.3 for a task description and subsection 5.4.2 for a tool description.

5.7.1 User Study: What We Eat In America Dataset

[10] We 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 would 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 wrote 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

[10] This section is taken from my publication "Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data" (Section 6.4: User Study: What We Eat In America Dataset) [10]. I have been the main author of this publication and section and have written all contents. The paper was internally reviewed by my co-authors Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel Keim.

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 were 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 to 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 amount of combinations using 60 attribute characteristics. The second question was to estimate the theoretically possible amount 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 amount 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 an 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 114540 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)" [286]. For the study, the dataset from 2017-2018 had been prepared and consisted of 7640 participants logging their intake of one day [287]. 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)" [288] 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)" [289] 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+)

[286]: Agriculture (2022), What We Eat In America (WWEIA) Database

[287]: Disease Control et al. (2022), National Health and Nutrition Examination Survey; 2017-2018 Data Documentation, Codebook, and Frequencies; Individual Foods, First Day (DR1IFF_J)

[288]: Disease Control et al. (2022), National Health and Nutrition Examination Survey; 2017-2018 Data Documentation, Codebook, and Frequencies; Body Measures (BMX_J)

[289]: Disease Control et al. (2022), National Health and Nutrition Examination Survey; 2017-2018 Data Documentation, Codebook, and Frequencies; Demographic Variables and Sample Weights (DEMO_J)

Table 5.1: The WWEIA dataset consists of 7640 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$\}, \ldots >$	m	20-25	normal	Mo	{8,10,...}	No
2	$< \{$o:Cena, t:Meat$\}, \{$o:Botana, t:Ice cream$\}, \ldots >$	f	40-25	obese I	Fr	{18,22,...}	No
3	$< \{$o:Breakfast, t:Cereal$\}, \{$o:Lunch, t:Salad$\}, \ldots >$	f	15-20	normal	We	{6,12,...}	No

- 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 5.1 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 to how to tackle the dataset, none of the participants know of a readily

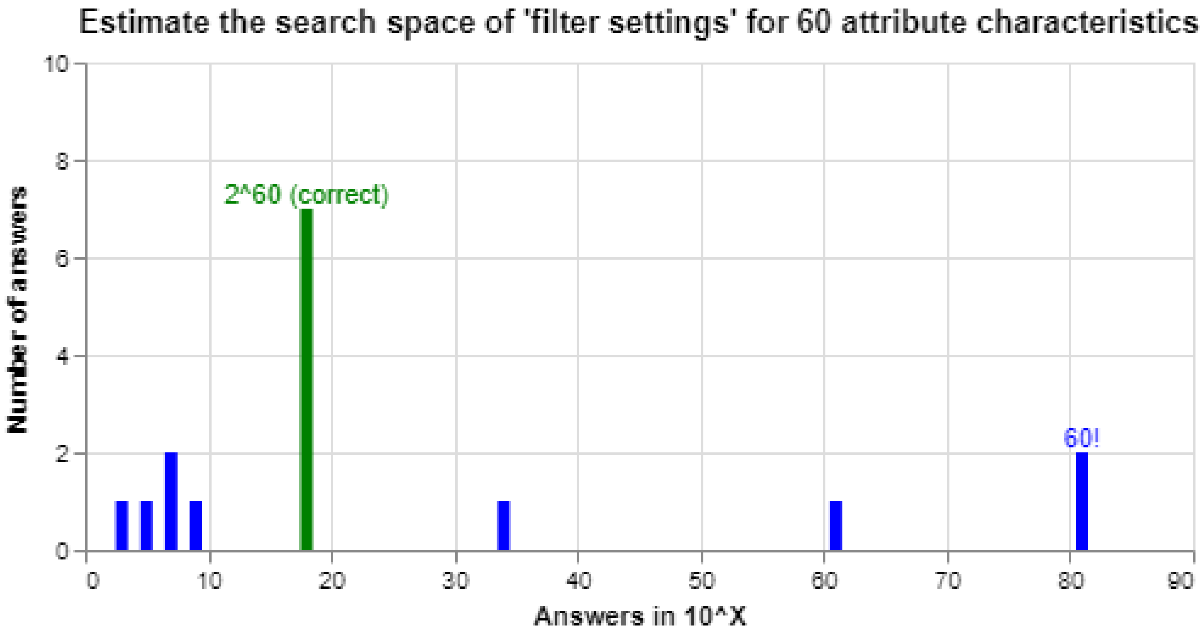


Figure 5.20: Participants' estimates about the search space of attribute characteristics. The correct answer is marked in green. Note that the x-axis is logarithmic.

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 stated 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 5.20). 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 par-

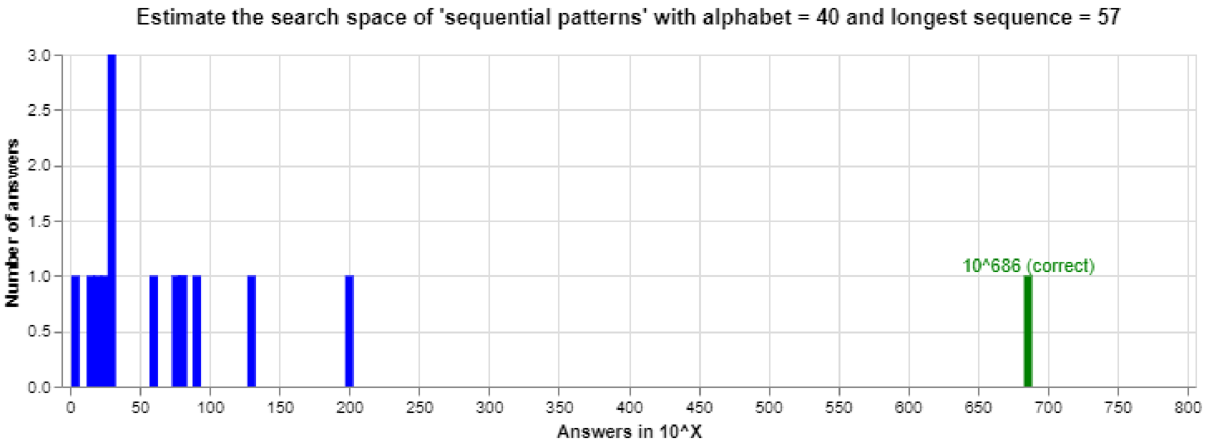


Figure 5.21: Participants' estimates about the search space of sequential pattern mining. The correct answer is marked in green. Note that the x-axis is logarithmic. Only one participant estimated the correct amount by deriving the correct formulas. All other participants greatly underestimated the size of the search space.

ticipants answered in ranges from 10^3 to 10^{200}, whereas the correct answer is 10^{686} (Figure 5.21). The actual amount of subsequences in the VAST Challenge dataset was guessed more correctly, however, eight of the participants responded with 10^2 to 10^5 and five participants with 10^7 to 10^10 whereas only one participant estimated correctly with 10^6 (Figure 5.22).

During the paired analytics phase, the participants spent the most time with the view, as shown in Figure 5.23. 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 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 that 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 the data. Using browsing, filtering, and sorting, they analyzed the data in a hypothesis-driven manner.

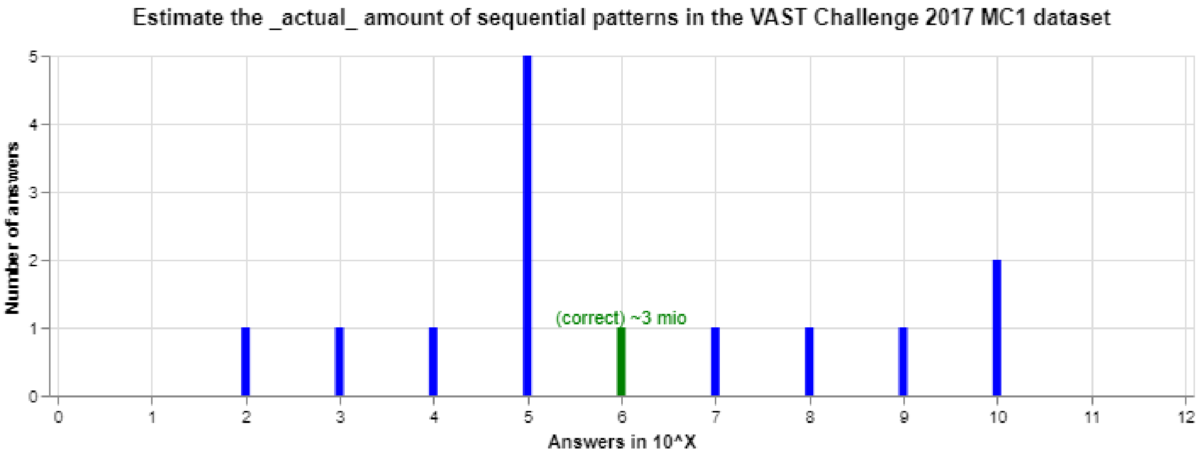
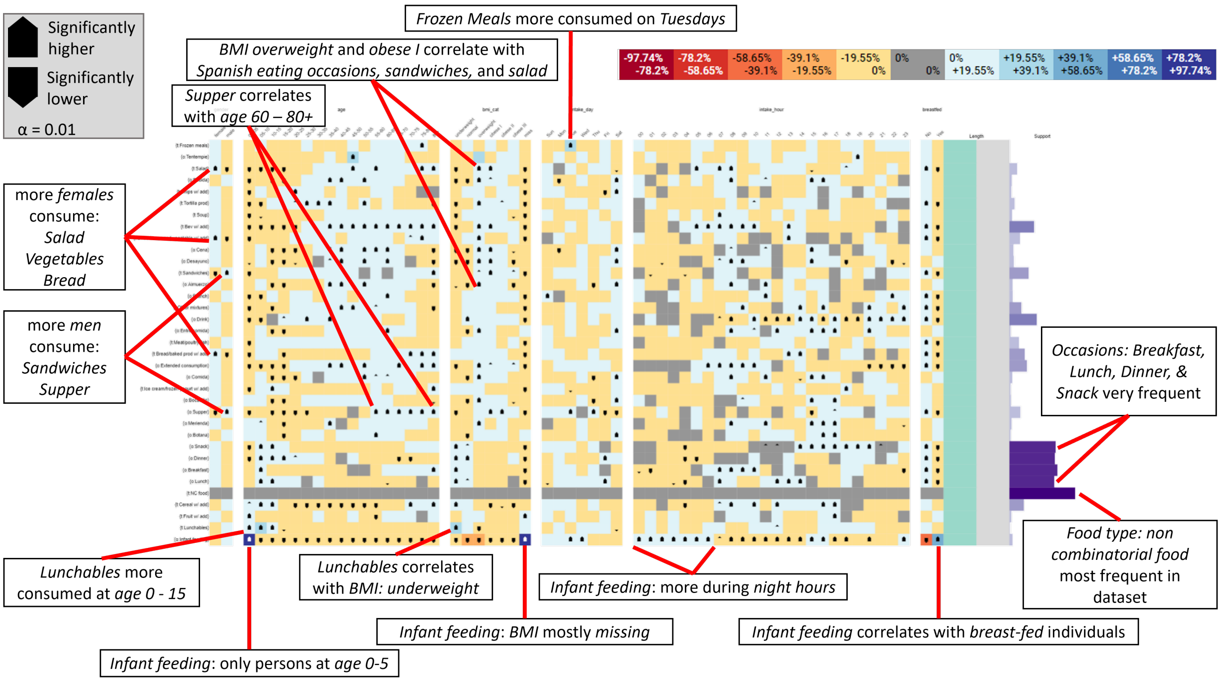


Figure 5.22: Participants' estimates about the search space of sequential patterns that are actually in the data. The correct answer is marked in green. Note that the x-axis is logarithmic. Note that the participants may have been biased, as the correct answer to the previous question was given to them before answering it.



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*

Figure 5.23: 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 breastfeeding. 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 can be derived only based on this view.

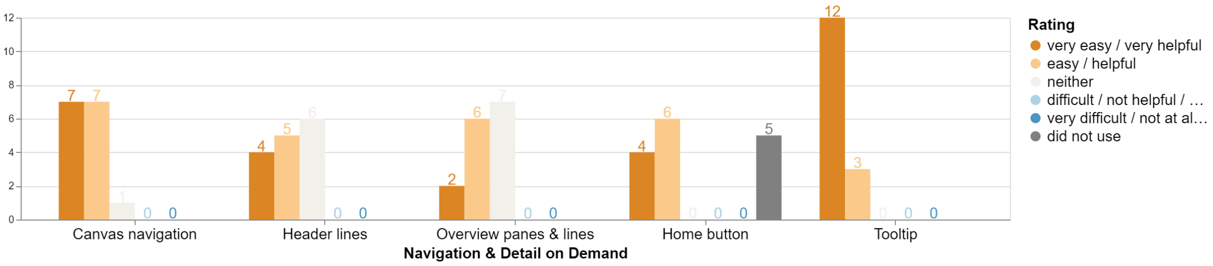


Figure 5.24: The various ratings of the canvas navigation and features to support navigating the canvas. Note the slightly different answer type for each question and note that the did not use option was only available for the home button.

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 and the knowledge of what task to use what perspective. 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.

The canvas navigation was deemed as very easy or easy by 14 of the participants (see Figure 5.24). The study revealed that panning and zooming in non-chromium browsers is 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 did not use 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 12) varied significantly. 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 had 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 5.26a). 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 who 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 AND in the current state, whereas only the co-occurrence filters are ORed. Multiple participants expressed the wish to search for multiple

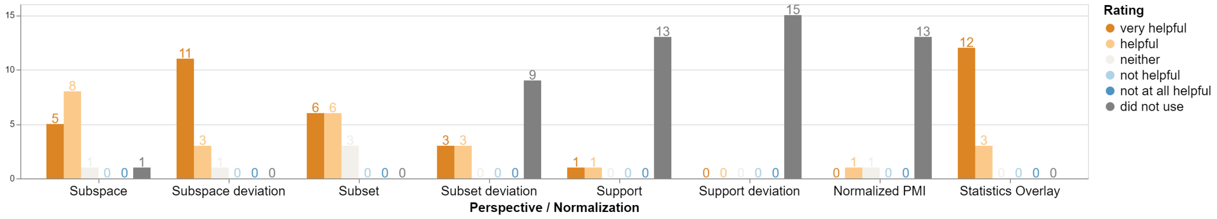
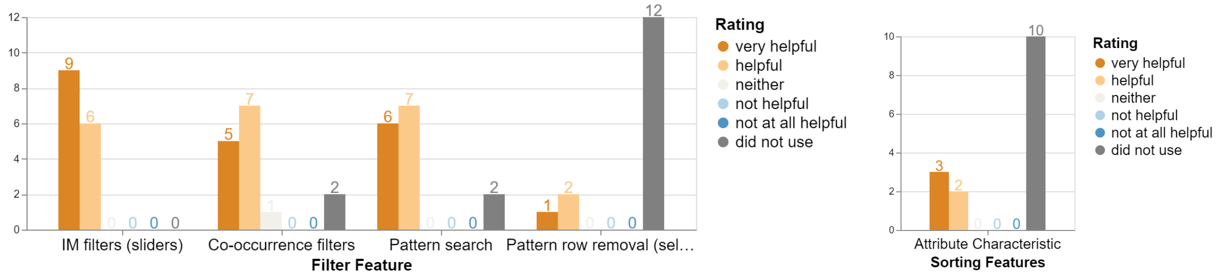


Figure 5.25: The various ratings for the different perspectives/normalizations and the statistics overlay



(a) Filter Features

(b) Sorting Feature

Figure 5.26: The ratings for filter and sorting features.

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 5.27). 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.

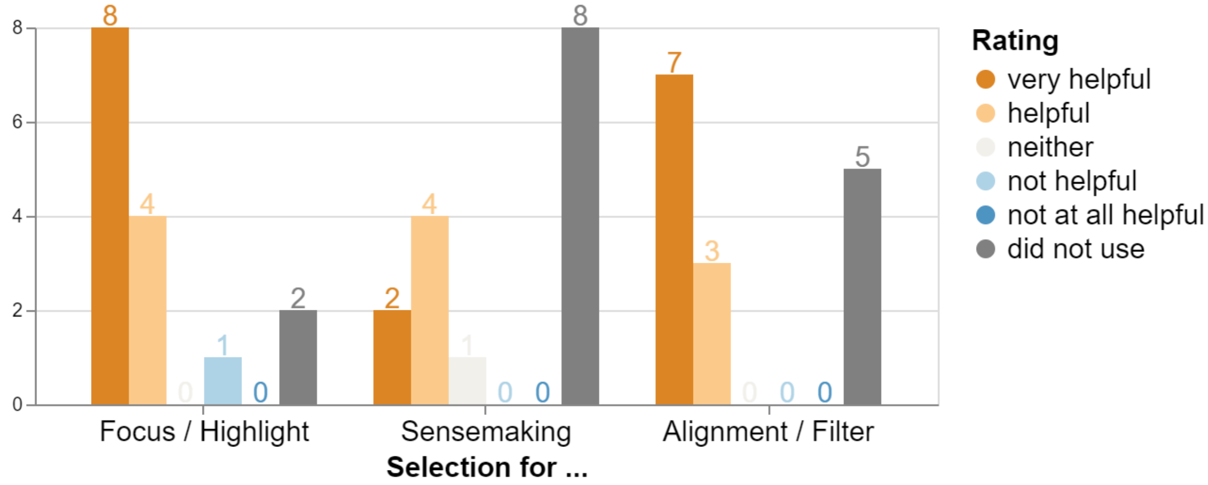


Figure 5.27: The various selection features and use cases.

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 was 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.

5.7.2 Conclusions

Evaluating VPA, like any system, is crucial. However, there are some limitations. When it comes to exploration and sense-making, it is ultimately the user doing the verdict. This implies that useful data and a useful task have to be used. This is because we derive interestingness from semantics. If a user is asked to rank the two patterns $\langle\{a\},\{b\}\rangle$ and $\langle\{b\},\{a\}\rangle$ most users would indecisive because the items a and b have no semantic meaning. This would not necessarily change if we supply interestingness measures such as the support that could be 50% for the first pattern and 70% for the second one. Most users would argue that this information is only some small perspective on these patterns whereas the full picture is not evident. Of course, one could simply argue that the task is to find the pattern with the highest support but this task is arbitrary and does not necessarily reflect the meaning of exploration which is vaguer (see Chapter 1). Therefore, a VPA system is always bound to a specific dataset and tasks and cannot be easily transferred to a different domain and task.

Another limitation of evaluations is that they only show that a user can find certain interesting patterns in a dataset within a certain timeframe. It is much more difficult, if not impossible to show that *all* interesting patterns were found. If all interesting patterns are known a-priori then the task is not truly an exploration task as the assumption is that the user has little knowledge about the dataset and therefore cannot estimate if all interesting patterns were found. On the other hand, the interestingness is subjective, therefore it is difficult to assess what the user should find interesting.

An important aspect of the example evaluation is guidance. Exploration tools are typically complex and thus guidance is appreciated. This is important for the trust-building phase as a guidance tool shows the user why a certain pattern is deemed interesting and how this is conveyed in the visualization(s). Even though the user does not necessarily agree that a specific pattern is of interest it provides comparison capabilities and an understanding of visual cues.

Discussions and Conclusions 6

We are now at a point to reflect on this dissertation.

6.1 Discussion

We are faced with two major challenges: Firstly, how can a user properly define thresholds and parameters with little or no prior knowledge about a dataset? Secondly, trial and error of the aforementioned parameters will inevitably end up in a pattern explosion. Are we able to prevent that?

Scalability Starting with the second challenge we have to humbly admit that our measures are not eliminating the exponential search spaces. This is an impossible endeavor as the root cause of the search spaces lies in the definition of *containment* itself - the deep core of structure mining allowing us to neglect the need for a distance measure. But we have identified several approaches to mitigate the problem starting by constraining the pattern mining algorithm using thresholds and templates. These constraints can be very effective, specifically for verification purposes and rapid hypothesis testing. However, for exploratory data analysis, they impose danger as too relaxed constraints may cause a pattern explosion potentially overwhelming the user or even breaking the application. Nevertheless, it is rewarding to spend effort on designing specific (task, user, & data) interestingness measures (feature engineering) as these cannot only be used for pruning but furthermore be visualized with far better scalability than if the structures need to be visualized themselves. One should not neglect the explainable artificial intelligence aspects in this design as the better the users understand these measures, the more effectively they can make use of them. This does not mean that they have to be taught in every detail but merely a conceptual understanding may be sufficient [21].

Other actions such as equivalence classes are another possibility to eliminate duplicates. However, I do not argue for always using closed pattern mining even though it is deemed

[21]: Jentner et al. (2018), Minions, Sheep, and Fruits: Metaphorical Narratives to Explain Artificial Intelligence and Build Trust

to be lossless regarding information. I find it difficult to create a general rule on whether generator patterns or closed patterns are better suited to providing an overview. This remains data-, user-, and task-dependent and should be decided case to case *thoroughly involving* the user in this discussion. For my work of finding subspaces in structured data [10], I even find the redundancy of equivalence classes helpful as these create visual blocks of equal pixels that additionally highlight the subspaces (see Figure 5.11).

[10]: Jentner et al. (2022), Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data

Various techniques such as linking and brushing, or the general information visualization mantra are helpful to combine the visual scalability of interestingness measures with the semantic, high-dimensional details that lay in the structure of the data. However, the greatest potential lies in progressive visual pattern analytics (VPA) as it allows us to always start with a relatively small set of patterns and gradually expand it while navigating through the search space. This is so effective as the search spaces visually have a diamond-type shape (see section 2.2) where relatively few patterns are on top and the bottom. Progressive VPA sparks two questions: (i) can a user derive potentially interesting patterns from their sub-patterns? And: (ii) does a user need to see every pattern to determine its interestingness for the task at hand? While both questions cannot be universally answered, this thesis, my research and the research of other scientists show that it is possible for certain areas and can be successfully applied. This, however, does not mean that it is generally possible to rely on this assumption. For the second question, we can quite heretically argue that a user will never be able nor willing to assess all patterns in large search spaces anyway. The answer to the question "Have I seen enough (patterns)?" will depend very much on the user and from case to case as this is an entirely subjective decision.

Chicken and Egg Situation of Constraints Back to the first challenge: did we gain anything? The previously mentioned techniques and approaches allow us to tame the pattern explosion. This gives the user ultimately the possibility to relax certain constraints without fearing breaking the tool or being presented with a ridiculous amount of results. This effort in "making machine learning less scary" ultimately encourages the users to fully exploit an application and its functionalities which in return allows them to explore more of the data [8, 21]. It further enables the machine learning expert to define sensible default thresholds and constraints to provide the user with a good starting point. Guidance features in the application are a welcomed tool for users to initially learn and understand how a potentially interesting pattern may be represented in a visual analytics tool (see section 5.7). However, understanding the concepts of how the data is modeled, pattern-mining, and interestingness measures still remain relevant.

[8]: Jentner et al. (2018), Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool

[21]: Jentner et al. (2018), Minions, Sheep, and Fruits: Metaphorical Narratives to Explain Artificial Intelligence and Build Trust

Is Pattern Mining Still Relevant? A most daring question concerning the recent impressive advances in deep neural networks, specifically, image generation, latent diffusion models, and reinforcement learning from human feedback (RHEL) with their prototypes like Dall-E2 [290], Stable Diffusion [291], and ChatGPT [292]. Paaßen et al. underline that structured data does not have a vectorial and survey various methods of vectorial representations of structured data [293]. It comes to mind to exploit machine learning to automatically learn which patterns are interesting. However, this requires training data to be available and as I show in Chapter 3, the measures will likely depend on data, user, and the task, and therefore, cannot be easily transferred across domains. There are, however, approaches to leverage neural networks to mine patterns improving on the scalability [294]. The verdict is *yes*, pattern mining is still relevant and is being actively researched even after almost 30 years.

[290]: Ramesh et al. (2022), Hierarchical Text-Conditional Image Generation with CLIP Latents

[291]: Rombach et al. (2022), High-Resolution Image Synthesis with Latent Diffusion Models

[292]: OpenAI (2022), ChatGPT: Optimizing Language Models for Dialogue

[293]: Paaßen et al. (2019), Embeddings and Representation Learning for Structured Data

[294]: Jamshed et al. (2020), Deep learning-based sequential pattern mining for progressive database

6.2 Future Work

This thesis lays out a foundation for the exploration of mined patterns in structured data. There is a plethora of future opportunities for research. On the one hand, improving the performance of pattern mining algorithms is being actively researched [295], on the other hand, the design space of interestingness measures, algorithms, (interactive) visualization-, and visual analytics techniques is vast such that there are many possible combinations and extensions imaginable on how to enable a user exploring patterns for a certain domain and task [296].

[295]: Gan et al. (2018), A Survey of Parallel Sequential Pattern Mining

[296]: Plaisant et al. (2016), The diversity of data and tasks in event analytics

Uncertainty Mining Because structured data is discrete, pattern mining is not robust to uncertainty [297]. There are different types of uncertainties such as did event a occurred before b or at the same time, or the other way round. Another type of uncertainty is if an event occurred at all. Mining on uncertain databases, also called fuzzy mining, is tackled by some data mining approaches [298, 299]. The major challenge is that fuzzy mining increases the search space even more as it extends the number of possible combinations. However, uncertainty is not only algorithmically complex but also complex to visualize as it requires the usage of more visual encodings and must be quantifiable [300].

[297]: Gotz (2016), Soft patterns: Moving beyond explicit sequential patterns during visual analysis of longitudinal event datasets

[298]: Aggarwal et al. (2009), Frequent pattern mining with uncertain data

[299]: Leung et al. (2008), A Tree-Based Approach for Frequent Pattern Mining from Uncertain Data

[300]: Hullman (2020), Why Authors Don't Visualize Uncertainty

Comparison of Pattern Sets Visualizing and exploring a single set of patterns is already challenging. Comparing result sets is even more difficult because the comparison can be made via interestingness measures, structure, and semantics. It is, however, interesting to do for certain applications such as to compare two datasets using pattern mining or to compare different thresholds, utility functions, etc. Giuliana Lindholz (Dehn), whom I supervised for her thesis, created such an approach to compare sequential rules in her master thesis [301].

[301]: Dehn (2020), Visual Analytics for the Exploration of Sequential Rules

Guidance, Active Learning, and Speculative Execution These are core topics of progressive visual pattern analytics and there are many valuable research possibilities. For example, Sperrle et al. proposed a framework for guidance [302]. There are existing (task-dependent) approaches to using belief systems, and utility functions over feature representations (see section 5.5) but visual analytics tools that combine guidance with these techniques are scarce. I see great potential in such approaches if a tool is capable of explaining why a certain pattern is potentially interesting to the user. This would also allow the user to interact better with the system and accept or deny the conclusion of the model *actively* influencing an underlying utility function. Speculative execution can be useful to mine for certain types of patterns, however, the pattern explosion aspect will be quite challenging [303].

[302]: Sperrle et al. (2022), Lotse: A Practical Framework for Guidance in Visual Analytics

[303]: Sperrle et al. (2019), Speculative Execution for Guided Visual Analytics

Note that these are only three selected areas that I find interesting for further research but there are many more! This thesis talked so often about the problem of a *pattern explosion* that here I can joyfully state that the circumstances are present for a *design explosion* as there are so many combinations of algorithmic-, interaction-, and visualization techniques possible that almost every new endeavor will have a unique aspect to it.

6.3 Conclusions

This thesis presents various techniques and approaches to how patterns from structured data mining can be explored. After introducing interestingness measures, it assesses their properties and provides a complementary perspective and taxonomy of what interestingness measures are and how they can be successfully used. This ultimately gives the user more responsibility in the need to understand what these interestingness measures represent but in return allows for more effective use. Several use cases underline these arguments. A survey of visualization techniques follows showing how structure and interestingness measures can be visualized. It highlights the creativity of the research community in finding unique and elegant solutions. The comparison of the techniques reveals that tradeoffs are necessary when it comes to scalability vs. visualizing structures (in detail). The interactive visualization and visual analytics techniques are presented and transferred to the pattern mining domain demonstrating how individual shortcomings of interestingness measures and visual designs can be mitigated. Specifically, progressive visual pattern analytics is a resourceful area for further research. I have been thoroughly inspired by many works and I hope that my research and this thesis serve as an inspiration for others. I am very excited to see how this field continues to develop and grow in the future.

Bibliography

[1] Zhi-Hua Zhou. Machine Learning. Springer, 2021 (cit. on pp. 1, 39).

[2] Christopher Westphal and Teresa Blaxton. Data mining solutions: methods and tools for solving real-world problems. John Wiley & Sons, Inc., 1998 (cit. on p. 1).

[3] Bing Liu. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Second Edition. Data-Centric Systems and Applications. Springer, 2011 (cit. on p. 1).

[4] David Caster Hoaglin, Frederick Mosteller, and John Wilder Tukey. Understanding robust and exploratory data analysis. Vol. 3. Wiley New York, 1983 (cit. on p. 1).

[5] Eytan Adar. Banning exploration in my infovis class. `https://medium.com/eytanadar/banning-exploration-in-my-infovis-class-9578676a4705`. 2017 (cit. on pp. 1, 2).

[6] Jarke J. van Wijk. Views on Visualization. In: *IEEE Trans. Vis. Comput. Graph.* 12 (2006), pp. 421–433. doi: `10.1109/TVCG.2006.80` (cit. on pp. 3, 113).

[7] Wolfgang Jentner, Geoffrey Ellis, Florian Stoffel, Dominik Sacha, and Daniel A. Keim. A Visual Analytics Approach for Crime Signature Generation and Exploration. In: *The Event Event: Temporal & Sequential Event Analysis, IEEE VIS 2016 Workshop*. 2016 (cit. on pp. 4, 6).

[8] Wolfgang Jentner, Dominik Sacha, Florian Stoffel, Geoffrey P. Ellis, Leishi Zhang, and Daniel A. Keim. Making machine intelligence less scary for criminal analysts: reflections on designing a visual comparative case analysis tool. In: *The Visual Computer* 34 (2018), pp. 1225–1241 (cit. on pp. 4, 6, 46–48, 50, 52, 53, 63, 104, 105, 109, 120–122, 124, 126, 159, 177).

[9] Rita Sevastjanova, Wolfgang Jentner, Fabian Sperrle, Rebecca Kehlbeck, Jürgen Bernard, and Mennatallah El-Assady. QuestionComb: A Gamification Approach for the Visual Explanation of Linguistic Phenomena through Interactive Labeling. In: *ACM Trans. Interact. Intell. Syst.* 11 (2021), 19:1–19:38. doi: `10.1145/3429448` (cit. on pp. 4, 7, 55, 56, 142, 143).

[10] Wolfgang Jentner, Giuliana Lindholz, Hanna Hauptmann, Mennatallah El-Assady, Kwan-Liu Ma, and Daniel A. Keim. Visual Analytics of Co-Occurrences to Discover Subspaces in Structured Data. In: *Accepted at Association for Computing Machinery, Transactions on Interactive Intelligent Systems (ACM TIIS)* (2022) (cit. on pp. 4, 8, 22, 23, 29, 58, 59, 65, 77, 130, 131, 149, 162, 176).

[11] Daniel Seebacher. Visual Analytics of Spatial Events: Methods for the Interactive Analysis of Spatio-Temporal Data Abstractions. PhD thesis. University of Konstanz, Germany, 2021 (cit. on p. 5).

[12] Visual Analytics for Sense-making and Criminal Intelligence Analysis, http://www.valcri.org/, last retrieved 14th Sep.,2017 (cit. on pp. 6, 9).

[13] Dominik Sacha, Wolfgang Jentner, Leishi Zhang, Florian Stoffel, and Geoffrey P. Ellis. Visual Comparative Case Analytics. In: *8th International EuroVis Workshop on Visual Analytics, EuroVA@EuroVis 2017, Barcelona, Spain, 12-13 June 2017*. 2017, pp. 49–53. DOI: 10.2312/eurova.20171119 (cit. on pp. 6, 11).

[14] Dominik Sacha, Wolfgang Jentner, Leishi Zhang, Florian Stoffel, Geoffrey Ellis, and Daniel A. Keim. Applying Visual Interactive Dimensionality Reduction to Criminal Intelligence Analysis. Tech. rep. VALCRI, 2017 (cit. on pp. 6, 15).

[15] Yuanzhe Chen, Panpan Xu, and Liu Ren. Sequence Synopsis: Optimize Visual Summary of Temporal Event Data. In: *IEEE Trans. Vis. Comput. Graph.* 24 (2018), pp. 45–55. DOI: 10.1109/TVCG.2017.2745083 (cit. on pp. 7, 103, 109, 144).

[16] Wolfgang Jentner and Daniel A. Keim. Visualization and Visual Analytic Techniques for Patterns. In: Springer International Publishing, 2019. Chap. 12, pp. 303–337. DOI: 10.1007/978-3-030-04921-8 (cit. on p. 9).

[17] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Roger Nkambou, Bay Vo, and Vincent S Tseng. High-Utility Pattern Mining. Springer, 2019 (cit. on pp. 9, 34, 36).

[18] Wolfgang Jentner, Geoffrey Ellis, Florian Stoffel, Dominik Sacha, and Daniel Keim. A visual analytics approach for crime signature generation and exploration. In: *IEEE VIS2016 Workshop on Temporal & Sequential Event Analysis*. 2016 (cit. on pp. 9, 52).

[19] Juri Buchmüller, Wolfgang Jentner, Dirk Streeb, and Daniel A. Keim. ODIX: A Rapid Hypotheses Testing System for Origin-Destination Data IEEE VAST Challenge Award for Excellence in Spatio-temporal Graph Analytics. In: *12th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2017, Phoenix, AZ, USA, October 3-6, 2017*. 2017, pp. 197–198. DOI: 10.1109/VAST.2017.8585686 (cit. on pp. 9, 115, 116).

[20] 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. VAST Challenge 2017 Mini Challenge 1. http://www.vacommunity.org/VAST+Challenge+2017+MC1. Accessed: 2018-12-12. 2017 (cit. on pp. 9, 23, 114, 118).

[21] Wolfgang Jentner, Rita Sevastjanova, Florian Stoffel, Daniel A. Keim, Jürgen Bernard, and Mennatallah El-Assady. Minions, Sheep, and Fruits: Metaphorical Narratives to Explain Artificial Intelligence and Build Trust. In: *Workshop on Visualization for AI Explainability*. 2018 (cit. on pp. 10, 40, 155, 159, 175, 177).

[22] Mennatallah El Assady, Daniel Hafner, Michael Hund, Wolfgang Jentner, Christian Rohrdantz, Fabian Fischer, Svenja Simon, Tobias Schreck, and Daniel Keim. Visual analytics for the prediction of movie rating and box office performance. In: 2014 (cit. on p. 10).

[23] Franz Wanner, Tobias Schreck, Wolfgang Jentner, Lyubka Sharalieva, and Daniel A. Keim. Relating interesting quantitative time series patterns with text events and text features. In: *Visualization and Data Analysis 2014, San Francisco, CA, USA, February 3-5, 2014*. Vol. 9017. 2014, 90170G. DOI: 10.1117/12.2039639 (cit. on p. 11).

[24] Mennatallah El-Assady, Wolfgang Jentner, Manuel Stein, Fabian Fischer, Tobias Schreck, and Daniel Keim. Predictive visual analytics: Approaches for movie ratings and discussion of open research challenges. In: *An IEEE VIS 2014 Workshop: Visualization for Predictive Analytics*. 2014 (cit. on p. 11).

[25] Franz Wanner, Wolfgang Jentner, Tobias Schreck, Andreas Stoffel, Lyubka Sharalieva, and Daniel A. Keim. Integrated visual analysis of patterns in time series and text data - Workflow and application to financial data analysis. In: *Inf. Vis.* 15 (2016), pp. 75–90. DOI: 10.1177/1473871615576925 (cit. on pp. 11, 97, 109).

[26] Wolfgang Jentner, Mennatallah El-Assady, Dominik Sacha, Dominik Jäckle, and Florian Stoffel. Dynamite: Dynamic Monitoring Interface for Task Ensembles. In: *IEEE Conf. on Visual Analytics Science and Technology (VAST Challenge MC1)*. 2016 (cit. on p. 11).

[27] Mennatallah El-Assady, Valentin Gold, Annette Hautli-Janisz, Wolfgang Jentner, Miriam Butt, Katharina Holzinger, and Daniel A Keim. VisArgue: A visual text analytics framework for the study of deliberative communication. In: *PolText 2016-The International Conference on the Advancesin Computational Analysis of Political Text*. 2016, pp. 31–36 (cit. on p. 11).

[28] Florian Stoffel, Wolfgang Jentner, Michael Behrisch, Johannes Fuchs, and Daniel A. Keim. Interactive Ambiguity Resolution of Named Entities in Fictional Literature. In: *Comput. Graph. Forum* 36 (2017), pp. 189–200. DOI: 10.1111/cgf.13179 (cit. on p. 11).

[29] Wolfgang Jentner, Mennatallah El-Assady, Bela Gipp, and Daniel A. Keim. Feature Alignment for the Analysis of Verbatim Text Transcripts. In: *8th International EuroVis Workshop on Visual Analytics, EuroVA@EuroVis 2017, Barcelona, Spain, 12-13 June 2017*. 2017, pp. 13–17. DOI: 10.2312/eurova.20171113 (cit. on pp. 11, 98, 106, 109).

[30] Dirk Streeb, Juri Buchmüller, Udo Schlegel, Wolfgang Jentner, Michael Behrisch, Bruno Schneider, and Daniel Seebacher. Uncovering the Mistford Toxic Conspiracy. In: *12th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2017, Phoenix, AZ, USA, October 3-6, 2017*. 2017, pp. 243–244. DOI: 10.1109/VAST.2017.8585661 (cit. on p. 11).

[31] Wolfgang Jentner, Dominik Jäckle, Ulrich Engelke, Daniel A. Keim, and Tobias Schreck. A Concept for Consensus-based Ordering of Views. In: *9th International EuroVis Workshop on Visual Analytics, EuroVA@EuroVis 2018, Brno, Czech Republic, June 4, 2018*. 2018, pp. 61–65. DOI: 10.2312/eurova.20181114 (cit. on p. 11).

[32] Wolfgang Jentner, Florian Stoffel, Dominik Jäckle, Alexander Gärtner, and Daniel A Keim. DeepClouds: Stereoscopic 3D Wordle based on Conical Spirals. In: *Workshop on Visualization as Added Value in the Development, Use and Evaluation of Language Resources (VisLR III)@ LREC*. 2018 (cit. on p. 12).

[33] Eren Cakmak, Giuliano Castiglia, Wolfgang Jentner, Juri Buchmüller, and Daniel A Keim. Visualization For Train Management: Improving Overviews in Safety-critical Control Room Environments. In: *BDVA 2018: 4th International Symposium on Big Data Visual and Immersive Analytics*. 2018 (cit. on p. 12).

[34] Niklas Weiler, Matthias Kraus, Timon Kilian, Wolfgang Jentner, and Daniel A Keim. Visual Analytics for Semi-Automatic 4D Crime Scene Reconstruction. In: (2018) (cit. on p. 12).

[35] Isabel Piljek, Giuliana Dehn, Jannik Frauendorf, Ziad Salem, Yerzhan Niyazbayev, Juri Buchmüller, Eren Cakmak, Wolfgang Jentner, Florian Stoffel, and Daniel A. Keim. Identifying Patterns and Anomalies within Spatiotemporal Water Sampling Data : VAST Challenge 2018: Award for Elegant Design of an Interactive Display. In: *13th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2018, Berlin, Germany, October 21-26, 2018*. 2018, pp. 98–99. doi: 10.1109/VAST.2018.8802466 (cit. on p. 12).

[36] Benedikt Bäumle, Ina Boesecke, Raphael Buchmüller, Yannick Metz, Juri Buchmüller, Eren Cakmak, Wolfgang Jentner, and Daniel A. Keim. Interactive Webtool for Tempospatial Data and Visual Audio Analysis : VAST Challenge 2018: Honorable Mention for Interactive Analytic Tool. In: *13th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2018, Berlin, Germany, October 21-26, 2018*. 2018, pp. 96–97. doi: 10.1109/VAST.2018.8802517 (cit. on p. 12).

[37] Eren Cakmak, Udo Schlegel, Matthias Miller, Juri Buchmüller, Wolfgang Jentner, and Daniel A. Keim. Interactive Classification Using Spectrograms and Audio Glyphs. In: *13th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2018, Berlin, Germany, October 21-26, 2018*. 2018, pp. 110–111. doi: 10.1109/VAST.2018.8802500 (cit. on p. 12).

[38] Udo Schlegel, Wolfgang Jentner, Juri Buchmüller, Eren Cakmak, Giuliano Castiglia, Renzo Canepa, Simone Petralli, Luca Oneto, Daniel A. Keim, and Davide Anguita. Visual Analytics for Supporting Conflict Resolution in Large Railway Networks. In: *Recent Advances in Big Data and Deep Learning, Proceedings of the INNS Big Data and Deep Learning Conference INNSBDDL 2019, held at Sestri Levante, Genova, Italy 16-18 April 2019*. 2019, pp. 206–215. doi: 10.1007/978-3-030-16841-4_22 (cit. on p. 12).

[39] Mennatallah El-Assady, Wolfgang Jentner, Fabian Sperrle, Rita Sevastjanova, Annette Hautli-Janisz, Miriam Butt, and Daniel A. Keim. lingvis.io - A Linguistic Visual Analytics Framework. In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*. 2019, pp. 13–18. doi: 10.18653/v1/p19-3003 (cit. on pp. 12, 56).

[40] Michael Dose, Norbert Wendt, Markus Mühling, Thomas Pollok, Wolfgang Jentner, Stephan Schindler, Anna Louise Tilling, Ross King, Florian Fest, Thomas Philipp, and Markus Kastelitz. FLORIDA: Analyse von Videomassendaten im Kontext terroristischer Anschläge. In: *Crisis Prevention* (2019) (cit. on p. 12).

[41] Wolfgang Jentner, Juri Buchmüller, Fabian Sperrle, Rita Sevastjanova, Thilo Spinner, Udo Schlegel, Dirk Streeb, and Hanna Schäfer. N.E.A.T. - Novel Emergency Analysis Tool. In: *14th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2019, Vancouver, BC, Canada, October 20-25, 2019*. 2019, pp. 104–105. doi: 10.1109/VAST47406.2019.8986900 (cit. on p. 13).

[42] Mennatallah El-Assady, Wolfgang Jentner, Rebecca Kehlbeck, Udo Schlegel, Rita Sevastjanova, Fabian Sperrle, Thilo Spinner, and Daniel Keim. Towards XAI: structuring the processes of explanations. In: *Proceedings of the ACM Workshop on Human-Centered Machine Learning, Glasgow, UK*. Vol. 4. 2019 (cit. on pp. 13, 161).

[43] Thomas Pollok, Matthias Kraus, Chengchao Qu, Matthias Miller, Tobias Moritz, Timon Urs Kilian, Daniel A. Keim, and Wolfgang Jentner. Computer vision meets visual analytics: Enabling 4D crime scene investigation from image and video data. In: *9th International Conference on Imaging for Crime Detection and Prevention, ICDP 2019, London, UK, December 16-18, 2019*. 2019, pp. 44–49. doi: 10.1049/cp.2019.1166 (cit. on p. 13).

[44] Matthias Kraus, Thomas Pollok, Matthias Miller, Timon Urs Kilian, Tobias Moritz, Daniel Schweitzer, Jürgen Beyerer, Daniel A. Keim, Chengchao Qu, and Wolfgang Jentner. Toward Mass Video Data Analysis: Interactive and Immersive 4D Scene Reconstruction. In: *Sensors* 20 (2020), p. 5426. doi: 10.3390/s20185426 (cit. on p. 13).

[45] Luis Marti-Bonmati, Ángel Alberich-Bayarri, Ruth Ladenstein, Ignacio Blanquer, J Damian Segrelles, Leonor Cerdá-Alberich, Polyxeni Gkontra, Barbara Hero, JM Garcia-Aznar, Daniel Keim, et al. PRIMAGE project: predictive in silico multiscale analytics to support childhood cancer personalised evaluation empowered by imaging biomarkers. In: *European radiology experimental* 4 (2020), pp. 1–11 (cit. on p. 13).

[46] Maximilian T. Fischer, Simon David Hirsbrunner, Wolfgang Jentner, Matthias Miller, Daniel A. Keim, and Paula Helm. Promoting Ethical Awareness in Communication Analysis: Investigating Potentials and Limits of Visual Analytics for Intelligence Applications. In: *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*. 2022, pp. 877–889. doi: 10.1145/3531146.3533151 (cit. on p. 13).

[47] Wolfgang Jentner, Fabian Sperrle, Daniel Seebacher, Matthias Kraus, Rita Sevastjanova, Maximilian T Fischer, Udo Schlegel, Dirk Streeb, Matthias Miller, Thilo Spinner, et al. Visualisierung der COVID-19-Inzidenzen und Behandlungskapazitäten mit CoronaVis. 2022 (cit. on p. 14).

[48] Wikipedia authors. Hasse diagram. https://en.wikipedia.org/wiki/Hasse_diagram. 2021 (cit. on pp. 21, 24, 69).

[49] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules in Large Databases. In: *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*. 1994, pp. 487–499 (cit. on pp. 23, 32, 35, 36, 44, 88).

[50] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Discovering Frequent Closed Itemsets for Association Rules. In: *Database Theory - ICDT '99, 7th International Conference, Jerusalem, Israel, January 10-12, 1999, Proceedings*. Vol. 1540. 1999, pp. 398–416. doi: 10.1007/3-540-49257-7_25 (cit. on p. 24).

[51] Frequent Pattern Mining. Springer, 2014 (cit. on pp. 25, 58, 61, 81, 96).

[52] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Bay Vo, Tin Chi Truong, Ji Zhang, and Hoai Bac Le. A survey of itemset mining. In: *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 7 (2017). DOI: 10.1002/widm.1207 (cit. on pp. 26, 31, 58, 81).

[53] Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In: *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*. 1998, pp. 604–613. DOI: 10.1145/276698.276876 (cit. on p. 28).

[54] Mario Köppen. The curse of dimensionality. In: *5th online world conference on soft computing in industrial applications (WSC5)*. Vol. 1. 2000, pp. 4–8 (cit. on p. 28).

[55] Frances Y Kuo and Ian H Sloan. Lifting the curse of dimensionality. In: *Notices of the AMS* 52 (2005), pp. 1320–1328 (cit. on p. 28).

[56] Michel Verleysen and Damien François. The Curse of Dimensionality in Data Mining and Time Series Prediction. In: *Computational Intelligence and Bioinspired Systems, 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10, 2005, Proceedings*. Vol. 3512. 2005, pp. 758–770. DOI: 10.1007/11494669_93 (cit. on p. 28).

[57] Michael E. Houle, Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Can Shared-Neighbor Distances Defeat the Curse of Dimensionality? In: *Scientific and Statistical Database Management, 22nd International Conference, SSDBM 2010, Heidelberg, Germany, June 30 - July 2, 2010. Proceedings*. Vol. 6187. 2010, pp. 482–500. DOI: 10.1007/978-3-642-13818-8_34 (cit. on p. 28).

[58] Arthur Zimek, Ira Assent, and Jilles Vreeken. Frequent Pattern Mining Algorithms for Data Clustering. In: *Frequent Pattern Mining*. Springer, 2014, pp. 403–423. DOI: 10.1007/978-3-319-07821-2_16 (cit. on p. 28).

[59] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, and Prabhakar Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: *SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 2-4, 1998, Seattle, Washington, USA*. 1998, pp. 94–105. DOI: 10.1145/276304.276314 (cit. on p. 28).

[60] Harsha S. Nagesh, Sanjay Goil, and Alok N. Choudhary. Adaptive Grids for Clustering Massive Data Sets. In: *Proceedings of the First SIAM International Conference on Data Mining, SDM 2001, Chicago, IL, USA, April 5-7, 2001*. 2001, pp. 1–17. DOI: 10.1137/1.9781611972719.7 (cit. on p. 28).

[61] Robert James Hilderman and Howard John Hamilton. Knowledge discovery and interestingness measures: A survey. Department of Computer Science, University of Regina Regina, 1999 (cit. on p. 31).

[62] Pang-Ning Tan, Vipin Kumar, and Jaideep Srivastava. Selecting the right interestingness measure for association patterns. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, July 23-26, 2002, Edmonton, Alberta, Canada*. 2002, pp. 32–41. DOI: 10.1145/775047.775053 (cit. on p. 31).

[63] Kenneth McGarry. A survey of interestingness measures for knowledge discovery. In: *Knowledge Eng. Review* 20 (2005), pp. 39–61. DOI: 10.1017/S0269888905000408 (cit. on pp. 31, 35, 38).

[64] Liqiang Geng and Howard J. Hamilton. Interestingness measures for data mining: A survey. In: *ACM Comput. Surv.* 38 (2006), p. 9 (cit. on pp. 31–33, 35, 44, 81).

[65] Yuejin Zhang, Lingling Zhang, Guangli Nie, and Yong Shi. A Survey of Interestingness Measures for Association Rules. In: *BIFE*. 2009, pp. 460–463 (cit. on p. 31).

[66] Tianyi Wu, Yuguo Chen, and Jiawei Han. Re-examination of interestingness measures in pattern mining: a unified framework. In: *Data Min. Knowl. Discov.* 21 (2010), pp. 371–397. DOI: 10.1007/s10618-009-0161-2 (cit. on pp. 31, 36).

[67] José Maria Luna, Philippe Fournier-Viger, and Sebastián Ventura. Frequent itemset mining: A 25 years review. In: *WIREs Data Mining Knowl. Discov.* 9 (2019). DOI: 10.1002/widm.1329 (cit. on p. 31).

[68] Christian Borgelt. Frequent item set mining. In: *WIREs Data Mining Knowl. Discov.* 2 (2012), pp. 437–456. DOI: 10.1002/widm.1074 (cit. on pp. 31, 81, 84).

[69] Jeffrey D. Ullman. A Survey of Association-Rule Mining. In: *Discovery Science, Third International Conference, DS 2000, Kyoto, Japan, December 4-6, 2000, Proceedings*. Vol. 1967. 2000, pp. 1–14. DOI: 10.1007/3-540-44418-1_1 (cit. on p. 31).

[70] Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for Association Rule Mining - A General Survey and Comparison. In: *SIGKDD Explor.* 2 (2000), pp. 58–64. DOI: 10.1145/360402.360421 (cit. on pp. 31, 81).

[71] Akbar Telikani, Amir H. Gandomi, and Asadollah Shahbahrami. A survey of evolutionary computation for association rule mining. In: *Inf. Sci.* 524 (2020), pp. 318–352. DOI: 10.1016/j.ins.2020.02.073 (cit. on p. 31).

[72] Qiankun Zhao and Sourav S Bhowmick. Association rule mining: A survey. In: *Nanyang Technological University, Singapore* 135 (2003) (cit. on p. 31).

[73] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. A survey of sequential pattern mining. In: *Data Science and Pattern Recognition* 1 (2017), pp. 54–77 (cit. on pp. 31, 58, 81).

[74] Chetna Chand, Amit Thakkar, and Amit Ganatra. Sequential pattern mining: Survey and current research challenges. In: *International Journal of Soft Computing and Engineering* 2 (2012), pp. 185–193 (cit. on p. 31).

[75] Qiankun Zhao and Sourav S Bhowmick. Sequential pattern mining: A survey. In: *ITechnical Report CAIS Nayang Technological University Singapore* 1 (2003), p. 135 (cit. on pp. 31, 81).

[76] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Philip S. Yu. A Survey of Parallel Sequential Pattern Mining. In: *ACM Trans. Knowl. Discov. Data* 13 (2019), 25:1–25:34. DOI: 10.1145/3314107 (cit. on p. 31).

[77] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential PAttern mining using a bitmap representation. In: *KDD*. 2002, pp. 429–435 (cit. on pp. 32, 147–149).

[78] Mohammed Javeed Zaki and Ching-Jiu Hsiao. CHARM: An Efficient Algorithm for Closed Itemset Mining. In: *SDM*. 2002, pp. 457–473 (cit. on pp. 32, 84).

[79] Miho Ohsaki, Shinya Kitaguchi, Kazuya Okamoto, Hideto Yokoi, and Takahira Yamaguchi. Evaluation of Rule Interestingness Measures with a Clinical Dataset on Hepatitis. In: *PKDD*. Vol. 3202. 2004, pp. 362–373 (cit. on p. 33).

[80] Ning Zhong, Yiyu Yao, and Setsuo Ohsuga. Peculiarity Oriented Multi-database Mining. In: *Principles of Data Mining and Knowledge Discovery, Third European Conference, PKDD '99, Prague, Czech Republic, September 15-18, 1999, Proceedings*. Vol. 1704. 1999, pp. 136–146. DOI: 10.1007/978-3-540-48247-5_15 (cit. on p. 33).

[81] Robert J Hilderman and Howard J Hamilton. Knowledge Discovery and Measures of Interest. Kluwer Academic Publishers, 2001 (cit. on p. 33).

[82] Sigal Sahar. Interestingness via What is Not Interesting. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 15-18, 1999*. 1999, pp. 332–336. DOI: 10.1145/312129.312272 (cit. on pp. 33, 35).

[83] Mika Klemettinen, Heikki Mannila, Pirjo Ronkainen, Hannu Toivonen, and A. Inkeri Verkamo. Finding Interesting Rules from Large Sets of Discovered Association Rules. In: *Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, Maryland, USA, November 29 - December 2, 1994*. 1994, pp. 401–407. DOI: 10.1145/191246.191314 (cit. on pp. 33, 84, 90, 91, 107, 108, 127, 138).

[84] Abraham Silberschatz and Alexander Tuzhilin. On Subjective Measures of Interestingness in Knowledge Discovery. In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, August 20-21, 1995*. 1995, pp. 275–281 (cit. on pp. 34, 35, 38).

[85] Abraham Silberschatz and Alexander Tuzhilin. What Makes Patterns Interesting in Knowledge Discovery Systems. In: *IEEE Trans. Knowl. Data Eng.* 8 (1996), pp. 970–974. DOI: 10.1109/69.553165 (cit. on pp. 34, 35).

[86] Bing Liu, Wynne Hsu, and Shu Chen. Using General Impressions to Analyze Discovered Classification Rules. In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, USA, August 14-17, 1997*. 1997, pp. 31–36 (cit. on pp. 34, 35).

[87] Bing Liu, Wynne Hsu, Lai-Fun Mun, and Hing-Yan Lee. Finding Interesting Patterns Using User Expectations. In: *IEEE Trans. Knowl. Data Eng.* 11 (1999), pp. 817–832. DOI: 10.1109/69.824588 (cit. on p. 34).

[88] Gregory Piatetsky-Shapiro and Christopher J Matheus. The interestingness of deviations. In: *Proceedings of the AAAI-94 workshop on Knowledge Discovery in Databases*. Vol. 1. 1994, pp. 25–36 (cit. on pp. 34, 35).

[89] Charles X. Ling, Tielin Chen, Qiang Yang, and Jie Cheng. Mining Optimal Actions for Profitable CRM. In: *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan*. 2002, pp. 767–770. DOI: 10.1109/ICDM.2002.1184049 (cit. on p. 34).

[90] Ke Wang, Senqiang Zhou, and Jiawei Han. Profit Mining: From Patterns to Actions. In: *Advances in Database Technology - EDBT 2002, 8th International Conference on Extending Database Technology, Prague, Czech Republic, March 25-27, Proceedings*. Vol. 2287. 2002, pp. 70–87. DOI: 10.1007/3-540-45876-X_7 (cit. on p. 34).

[91] Padhraic Smyth and Rodney M. Goodman. Rule Induction Using Information Theory. In: *Knowledge Discovery in Databases*. AAAI/MIT Press, 1991, pp. 159–176 (cit. on p. 35).

[92] Vasant Dhar and Alexander Tuzhilin. Abstract-Driven Pattern Discovery in Databases. In: *IEEE Trans. Knowl. Data Eng.* 5 (1993), pp. 926–938. DOI: 10.1109/69.250075 (cit. on p. 35).

[93] Hong Yao and Howard J. Hamilton. Mining itemset utilities from transaction databases. In: *Data Knowl. Eng.* 59 (2006), pp. 603–626. DOI: 10.1016/j.datak.2005.10.004 (cit. on p. 35).

[94] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey P. Ellis, and Daniel A. Keim. Knowledge Generation Model for Visual Analytics. In: *IEEE Trans. Vis. Comput. Graph.* 20 (2014), pp. 1604–1613. DOI: 10.1109/TVCG.2014.2346481 (cit. on pp. 38, 81, 127).

[95] Rinke Hoekstra. The knowledge reengineering bottleneck. In: *Semantic Web* 1 (2010), pp. 111–115. DOI: 10.3233/SW-2010-0004 (cit. on p. 39).

[96] Henry Shevlin, Karina Vold, Matthew Crosby, and Marta Halina. The limits of machine intelligence: Despite progress in machine intelligence, artificial general intelligence is still a major challenge. In: *EMBO reports* 20 (2019), e49177 (cit. on p. 39).

[97] Kleanthis-Nikolaos Kontonasios, Eirini Spyropoulou, and Tijl De Bie. Knowledge discovery interestingness measures based on unexpectedness. In: *WIREs Data Mining Knowl. Discov.* 2 (2012), pp. 386–399. DOI: 10.1002/widm.1063 (cit. on p. 39).

[98] Silvia Miksch and Wolfgang Aigner. A matter of time: Applying a data-users-tasks design triangle to visual analytics of time-oriented data. In: *Comput. Graph.* 38 (2014), pp. 286–290. DOI: 10.1016/j.cag.2013.11.002 (cit. on p. 40).

[99] Reinhard Diestel. Graph Theory, 4th Edition. Vol. 173. Graduate texts in mathematics. Springer, 2012 (cit. on pp. 40, 42).

[100] Deborah R. Carvalho, Alex Alves Freitas, and Nelson F. F. Ebecken. Evaluating the Correlation Between Objective Rule Interestingness Measures and Real Human Interest. In: *Knowledge Discovery in Databases: PKDD 2005, 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3-7, 2005, Proceedings*. Vol. 3721. 2005, pp. 453–461. DOI: 10.1007/11564126_45 (cit. on p. 45).

[101] Jiawei Han, Jianyong Wang, Ying Lu, and Petre Tzvetkov. Mining Top-K Frequent Closed Patterns without Minimum Support. In: *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan*. 2002, pp. 211–218. DOI: 10.1109/ICDM.2002.1183905 (cit. on p. 46).

[102] Petre Tzvetkov, Xifeng Yan, and Jiawei Han. TSP: Mining top-*k* closed sequential patterns. In: *Knowl. Inf. Syst.* 7 (2005), pp. 438–457. DOI: 10.1007/s10115-004-0175-4 (cit. on p. 46).

[103] Heungmo Ryang and Unil Yun. Top-k high utility pattern mining with effective threshold raising strategies. In: *Knowl. Based Syst.* 76 (2015), pp. 109–126. DOI: 10.1016/j.knosys.2014.12.010 (cit. on p. 46).

[104] Janet Prowse and Elizabeth Bennett. Working Manual of Criminal Law. Carswell Legal Pubns, 2000 (cit. on pp. 46, 48).

[105] NPIA. National Policing Improvement Agency: Professional Practice on Analysis. 2008 (cit. on p. 46).

[106] Nina Cope. Intelligence Led Policing or Policing Led Intelligence?: Integrating Volume Crime Analysis into Policing. In: *Br. J. Criminol.* 44 (2004), pp. 188–203 (cit. on p. 47).

[107] David Collier. The Comparative Method. In: *POLITICAL SCIENCE: THE STATE OF DISCIPLINE II*. 1993, pp. 105–118 (cit. on p. 48).

[108] C. Bennell and D. V. Canter. Linking commercial burglaries by modus operandi: tests using regression and ROC analysis. In: *Science & Justice* 42 (2002) (cit. on p. 48).

[109] David V. Canter, Laurence J. Alison, Emily Alison, and Natalia Wentink. The Organized/Disorganized Typology of Serial Murder: Myth or Model? In: *Psychology, Public Policy, and Law* 10 (2004), pp. 293–320 (cit. on p. 48).

[110] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, System Demonstrations*. 2014, pp. 55–60 (cit. on pp. 48, 49).

[111] Apache OpenNLP. https://opennlp.apache.org/. Accessed: 2017-09-13 (cit. on pp. 48, 49).

[112] George A. Miller. WordNet: A Lexical Database for English. In: *Commun. ACM* 38 (1995), pp. 39–41. DOI: 10.1145/219717.219748 (cit. on p. 49).

[113] Collin F. Baker, Charles J. Fillmore, and John B. Lowe. The Berkeley FrameNet Project. In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference.* 1998, pp. 86–90 (cit. on p. 49).

[114] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. Introduction to information retrieval. Cambridge University Press, 2008 (cit. on p. 49).

[115] Dan Jurafsky and James H. Martin. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition, 2nd Edition. Prentice Hall series in artificial intelligence. Prentice Hall, Pearson Education International, 2009 (cit. on p. 49).

[116] Rakesh Agrawal and Ramakrishnan Srikant. Mining Sequential Patterns. In: *Proceedings of the Eleventh International Conference on Data Engineering, March 6-10, 1995, Taipei, Taiwan*. 1995, pp. 3–14. DOI: 10.1109/ICDE.1995.380415 (cit. on pp. 49, 96).

[117] Antonio Gomariz, Manuel Campos, Roque MarıU0301XXXXXXXXXXXn, and Bart Goethals. ClaSP: An Efficient Algorithm for Mining Frequent Closed Sequences. In: *Advances in Knowledge Discovery and Data Mining, 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part I*. 2013, pp. 50–61. DOI: 10.1007/978-3-642-37453-1_5 (cit. on p. 49).

[118] Xifeng Yan, Jiawei Han, and Ramin Afshar. CloSpan: Mining Closed Sequential Patterns in Large Datasets. In: *Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May 1-3, 2003*. 2003, pp. 166–177. DOI: 10.1137/1.9781611972733.15 (cit. on p. 49).

[119] Hassan Saneifar, Sandra Bringay, Anne Laurent, and Maguelonne Teisseire. S2MP: Similarity Measure for Sequential Patterns. In: *Data Mining and Analytics 2008, Proceedings of the Seventh Australasian Data Mining Conference (AusDM 2008). Glenelg/Adelaide, SA, Australia, 27-28 November 2008, Proceedings*. 2008, pp. 95–104 (cit. on p. 49).

[120] IBM. IBM i2 Intelligence Analysis Platform (cit. on p. 49).

[121] John T. Stasko, Carsten Görg, and Zhicheng Liu. Jigsaw: supporting investigative analysis through interactive visualization. In: *Information Visualization* 7 (2008), pp. 118–132. DOI: 10.1057/palgrave.ivs.9500180 (cit. on p. 49).

[122] Dominik Jäckle, Florian Stoffel, Sebastian Mittelstädt, Daniel A. Keim, and Harald Reiterer. Interpretation of Dimensionally-reduced Crime Data: A Study with Untrained Domain Experts. In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017) - Volume 3: IVAPP, Porto, Portugal, February 27 - March 1, 2017.* 2017, pp. 164–175. DOI: 10.5220/0006265101640175 (cit. on p. 49).

[123] Leishi Zhang, Chris Rooney, Lev Nachmanson, B. L. William Wong, Bum Chul Kwon, Florian Stoffel, Michael Hund, Nadeem Qazi, Uchit Singh, and Daniel A. Keim. Spherical Similarity Explorer for Comparative Case Analysis. In: *Visualization and Data Analysis 2016, San Francisco, California, USA, February 14-18, 2016*. 2016, pp. 1–10 (cit. on p. 49).

[124] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John Aldo Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. What you see is what you can change: Human-centered machine learning by interactive visualization. In: *Neurocomputing* 268 (2017), pp. 164–175. DOI: 10.1016/j.neucom.2017.01.105 (cit. on p. 49).

[125] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John Aldo Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. In: *IEEE Trans. Vis. Comput. Graph.* 23 (2017), pp. 241–250. doi: 10.1109/TVCG.2016.2598495 (cit. on p. 49).

[126] James A. Wise. The Ecological Approach to Text Visualization. In: *J. Am. Soc. Inf. Sci.* 50 (1999), pp. 1224–1233. doi: 10.1002/(SICI)1097-4571(1999)50:13\<1224::AID-ASI8\>3.0.CO;2-4 (cit. on p. 50).

[127] Alex Endert, Patrick Fiaux, and Chris North. Semantic Interaction for Sensemaking: Inferring Analytical Reasoning for Model Steering. In: *IEEE Trans. Vis. Comput. Graph.* 18 (2012), pp. 2879–2888. doi: 10.1109/TVCG.2012.260 (cit. on p. 50).

[128] Lauren Bradel, Chris North, Leanna House, and Scotland Leman. Multi-model semantic interaction for text analytics. In: *9th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2014, Paris, France, October 25-31, 2014*. 2014, pp. 163–172. doi: 10.1109/VAST.2014.7042492 (cit. on p. 50).

[129] Tobias Ruppert, Michael Staab, Andreas Bannach, Hendrik Lücke-Tieke, Jürgen Bernard, Arjan Kuijper, and Jörn Kohlhammer. Visual Interactive Creation and Validation of Text Clustering Workflows to Explore Document Collections. In: *Visualization and Data Analysis 2017, Burlingame, CA, USA, 29 January 2017 - 2 February 2017*. 2017, pp. 46–57. doi: 10.2352/ISSN.2470-1173.2017.1.VDA-388 (cit. on p. 50).

[130] John Wenskovitch, Ian Crandell, Naren Ramakrishnan, Leanna House, and Chris North. Towards a Systematic Combination of Dimension Reduction and Clustering in Visual Analytics. In: *IEEE Transactions on Visualization and Computer Graphics* (2017) (cit. on p. 50).

[131] Paul van der Corput and Jarke J. van Wijk. Exploring Items and Features with I^{F}, F^{I}-Tables. In: *Comput. Graph. Forum* 35 (2016), pp. 31–40. doi: 10.1111/cgf.12879 (cit. on p. 50).

[132] Cagatay Turkay, Peter Filzmoser, and Helwig Hauser. Brushing Dimensions - A Dual Visual Analysis Model for High-Dimensional Data. In: *IEEE Trans. Vis. Comput. Graph.* 17 (2011), pp. 2591–2599. doi: 10.1109/TVCG.2011.178 (cit. on p. 50).

[133] Xiaoru Yuan, Donghao Ren, Zuchao Wang, and Cong Guo. Dimension Projection Matrix/Tree: Interactive Subspace Visual Exploration and Analysis of High Dimensional Data. In: *IEEE Trans. Vis. Comput. Graph.* 19 (2013), pp. 2625–2633. doi: 10.1109/TVCG.2013.150 (cit. on p. 50).

[134] Çagatay Demiralp. Clustrophile: A Tool for Visual Clustering Analysis. In: *CoRR* abs/1710.02173 (2017) (cit. on p. 50).

[135] Hans Peter Luhn. A Statistical Approach to Mechanized Encoding and Searching of Literary Information. In: *IBM J. Res. Dev.* 1 (1957), pp. 309–317. doi: 10.1147/rd.14.0309 (cit. on pp. 51, 52).

[136] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. In: *J. Documentation* 60 (2004), pp. 493–502. DOI: 10.1108/00220410410560573 (cit. on p. 51).

[137] Aikaterini-Lida Kalouli, Katharina Kaiser, Annette Hautli-Janisz, Georg A. Kaiser, and Miriam Butt. A Multilingual Approach to Question Classification. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018.* 2018 (cit. on p. 55).

[138] Suhas Ranganath, Xia Hu, Jiliang Tang, Suhang Wang, and Huan Liu. Identifying Rhetorical Questions in Social Media. In: *Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016.* 2016, pp. 667–670 (cit. on p. 55).

[139] Suhas Ranganath, Xia Hu, Jiliang Tang, Suhang Wang, and Huan Liu. Understanding and Identifying Rhetorical Questions in Social Media. In: *ACM Trans. Intell. Syst. Technol.* 9 (2018), 17:1–17:22. DOI: 10.1145/3108364 (cit. on p. 55).

[140] Mingzhu Zhang and Changzheng He. Survey on association rules mining algorithms. In: *Advancing Computing, Communication, Control and Management*. Springer, 2010, pp. 111–118 (cit. on p. 58).

[141] Helen Pinto, Jiawei Han, Jian Pei, Ke Wang, Qiming Chen, and Umeshwar Dayal. Multi-Dimensional Sequential Pattern Mining. In: *CIKM*. 2001, pp. 81–88 (cit. on pp. 58, 60).

[142] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Subspace clustering. In: *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 2 (2012), pp. 351–364 (cit. on p. 59).

[143] Leah Findlater and Howard J. Hamilton. Iceberg-cube algorithms: An empirical evaluation on synthetic and real data. In: *Intell. Data Anal.* 7 (2003), pp. 77–97 (cit. on pp. 59, 60).

[144] Guojun Gan and Jianhong Wu. Subspace clustering for high dimensional categorical data. In: *SIGKDD Explor.* 6 (2004), pp. 87–94. DOI: 10.1145/1046456.1046468 (cit. on p. 59).

[145] Gösta Grahne, Laks V. S. Lakshmanan, Xiaohong Wang, and Ming Hao Xie. On Dual Mining: From Patterns to Circumstances, and Back. In: *ICDE*. 2001, pp. 195–204 (cit. on p. 60).

[146] Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, Qiming Chen, Umeshwar Dayal, and Meichun Hsu. PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. In: *ICDE*. 2001, pp. 215–224 (cit. on p. 60).

[147] Kevin S. Beyer and Raghu Ramakrishnan. Bottom-Up Computation of Sparse and Iceberg CUBEs. In: *SIGMOD Conference*. 1999, pp. 359–370 (cit. on p. 60).

[148] Panida Songram, Veera Boonjing, and Sarun Intakosum. Closed Multidimensional Sequential Pattern Mining. In: *ITNG*. 2006, pp. 512–517 (cit. on p. 61).

[149] Mihael Ankerst, Anne Kao, Rodney Tjoelker, and Changzhou Wang. DataJewel: Integrating Visualization with Temporal Data Mining. In: *Visual Data Mining*. Vol. 4404. Lecture Notes in Computer Science. Springer, 2008, pp. 312–330 (cit. on p. 61).

[150] Taowei David Wang, Catherine Plaisant, Ben Shneiderman, Neil Spring, David Roseman, Greg Marchand, Vikramjit Mukherjee, and Mark S. Smith. Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison. In: *IEEE Trans. Vis. Comput. Graph.* 15 (2009), pp. 1049–1056 (cit. on p. 62).

[151] Krist Wongsuphasawat and David Gotz. Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization. In: *IEEE Trans. Vis. Comput. Graph.* 18 (2012), pp. 2659–2668. DOI: 10.1109/TVCG.2012.225 (cit. on p. 62).

[152] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. Temporal Event Sequence Simplification. In: *IEEE Trans. Vis. Comput. Graph.* 19 (2013), pp. 2227–2236. DOI: 10.1109/TVCG.2013.200 (cit. on pp. 62, 99).

[153] David Gotz and Harry Stavropoulos. DecisionFlow: Visual Analytics for High-Dimensional Temporal Event Sequence Data. In: *IEEE Trans. Vis. Comput. Graph.* 20 (2014), pp. 1783–1792. DOI: 10.1109/TVCG.2014.2346682 (cit. on pp. 62, 105, 106, 109).

[154] Bram C. M. Cappers and Jarke J. van Wijk. Exploring Multivariate Event Sequences Using Rules, Aggregations, and Selections. In: *IEEE Trans. Vis. Comput. Graph.* 24 (2018), pp. 532–541. DOI: 10.1109/TVCG.2017.2745278 (cit. on pp. 62, 97–99, 109, 118, 127).

[155] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. UpSet: Visualization of Intersecting Sets. In: *IEEE Trans. Vis. Comput. Graph.* 20 (2014), pp. 1983–1992 (cit. on p. 62).

[156] Alexander Lex and Nils Gehlenborg. Points of view: Sets and intersections. In: *nature methods* 11 (2014), p. 779 (cit. on p. 62).

[157] Katerina Vrotsou, Jimmy Johansson, and Matthew Cooper. ActiviTree: Interactive Visual Exploration of Sequences in Event-Based Data Using Graph Similarity. In: *IEEE Trans. Vis. Comput. Graph.* 15 (2009), pp. 945–952. DOI: 10.1109/TVCG.2009.117 (cit. on pp. 62, 102, 109).

[158] Adam Perer and Fei Wang. Frequence: interactive mining and visualization of temporal frequent event sequences. In: *19th International Conference on Intelligent User Interfaces, IUI 2014, Haifa, Israel, February 24-27, 2014*. 2014, pp. 153–162. DOI: 10.1145/2557500.2557508 (cit. on pp. 62, 101, 109, 149).

[159] Charles D. Stolper, Adam Perer, and David Gotz. Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. In: *IEEE Trans. Vis. Comput. Graph.* 20 (2014), pp. 1653–1662. DOI: 10.1109/TVCG.2014.2346574 (cit. on pp. 63, 104, 109, 138, 147, 148).

[160] Sara Di Bartolomeo, Yixuan Zhang, Fangfang Sheng, and Cody Dunne. Sequence Braiding: Visual Overviews of Temporal Event Sequences and Attributes. In: *IEEE Trans. Vis. Comput. Graph.* 27 (2021), pp. 1353–1363. DOI: 10.1109/TVCG.2020.3030442 (cit. on p. 63).

[161] Lance Parsons, Ehtesham Haque, and Huan Liu. Subspace clustering for high dimensional data: a review. In: *SIGKDD Explorations* 6 (2004), pp. 90–105 (cit. on p. 63).

[162] Stephan Günnemann. Subspace clustering for complex data. PhD thesis. RWTH Aachen University, 2012 (cit. on p. 63).

[163] Christian Baumgartner, Claudia Plant, Karin Kailing, Hans-Peter Kriegel, and Peer Kröger. 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*. 2004, pp. 11–18. DOI: 10.1109/ICDM.2004.10112 (cit. on p. 63).

[164] Alfred Inselberg and Bernard Dimsdale. Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry. In: *IEEE Visualization*. 1990, pp. 361–378 (cit. on p. 63).

[165] Dominik Jäckle, Michael Hund, Michael Behrisch, Daniel A. Keim, and Tobias Schreck. Pattern Trails: Visual Analysis of Pattern Transitions in Subspaces. In: *12th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2017, Phoenix, AZ, USA, October 3-6, 2017*. 2017, pp. 1–12. DOI: 10.1109/VAST.2017.8585613 (cit. on p. 64).

[166] Andrada Tatu, Fabian Maass, Ines Färber, Enrico Bertini, Tobias Schreck, Thomas Seidl, and Daniel A. Keim. 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*. 2012, pp. 63–72. DOI: 10.1109/VAST.2012.6400488 (cit. on p. 64).

[167] Dirk J. Lehmann and Holger Theisel. Optimal Sets of Projections of High-Dimensional Data. In: *IEEE Trans. Vis. Comput. Graph.* 22 (2016), pp. 609–618. DOI: 10.1109/TVCG.2015.2467132 (cit. on p. 64).

[168] Guodao Sun, Sujia Zhu, Qi Jiang, Wang Xia, and Ronghua Liang. EvoSets: Tracking the Sensitivity of Dimensionality Reduction Results Across Subspaces. In: *IEEE Trans. Big Data* 8 (2022), pp. 1566–1579. DOI: 10.1109/TBDATA.2021.3079200 (cit. on p. 64).

[169] Michael Blumenschein, Michael Behrisch, Stefanie Schmid, Simon Butscher, Deborah R. Wahl, Karoline Villinger, Britta Renner, Harald Reiterer, and Daniel A. Keim. SMARTexplore: Simplifying High-Dimensional Data Analysis through a Table-Based Visual Analytics Approach. In: *13th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2018, Berlin, Germany, October 21-26, 2018*. 2018, pp. 36–47. DOI: 10.1109/VAST.2018.8802486 (cit. on p. 64).

[170] Michael Behrisch, Dirk Streeb, Florian Stoffel, Daniel Seebacher, Brian Matejek, Stefan Hagen Weber, Sebastian Mittelstädt, Hanspeter Pfister, and Daniel A. Keim. Commercial Visual Analytics Systems-Advances in the Big Data Analytics Field. In: *IEEE Trans. Vis. Comput. Graph.* 25 (2019), pp. 3011–3031. DOI: 10.1109/TVCG.2018.2859973 (cit. on pp. 64, 111).

[171] Jiawei Han, Jian Pei, and Yiwen Yin. 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*. 2000, pp. 1–12. DOI: 10.1145/342009.335372 (cit. on p. 69).

[172] Mohammed Javeed Zaki. Scalable Algorithms for Association Mining. In: *IEEE Trans. Knowl. Data Eng.* 12 (2000), pp. 372–390. DOI: 10.1109/69.846291 (cit. on p. 69).

[173] Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo. Fast Discovery of Association Rules. In: *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996, pp. 307–328 (cit. on p. 71).

[174] Mark Harrower and Cynthia A Brewer. ColorBrewer. org: an online tool for selecting colour schemes for maps. In: *The Cartographic Journal* 40 (2003), pp. 27–37 (cit. on p. 78).

[175] Daniel A. Keim, Jörn Kohlhammer, Geoffrey P. Ellis, and Florian Mansmann. Mastering the Information Age - Solving Problems with Visual Analytics. Eurographics Association, 2010 (cit. on p. 81).

[176] Bart Goethals. Survey on frequent pattern mining. In: *Univ. of Helsinki* 19 (2003), pp. 840–852 (cit. on p. 81).

[177] Sotiris Kotsiantis and Dimitris Kanellopoulos. Association rules mining: A recent overview. In: *GESTS International Transactions on Computer Science and Engineering* 32 (2006), pp. 71–82 (cit. on p. 81).

[178] Nizar R. Mabroukeh and Christie I. Ezeife. A taxonomy of sequential pattern mining algorithms. In: *ACM Comput. Surv.* 43 (2010), 3:1–3:41 (cit. on pp. 81, 96).

[179] Ian H. Witten, Eibe Frank, and Mark A. Hall. Data mining: practical machine learning tools and techniques, 3rd Edition. Morgan Kaufmann, Elsevier, 2011 (cit. on p. 81).

[180] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Zhihong Deng, and Hoang Thanh Lam. The SPMF Open-Source Data Mining Library Version 2. In: *ECML/PKDD (3)*. Vol. 9853. 2016, pp. 36–40 (cit. on p. 81).

[181] FIMI '03, Frequent Itemset Mining Implementations, Proceedings of the ICDM 2003 Workshop on Frequent Itemset Mining Implementations, 19 December 2003, Melbourne, Florida, USA. Vol. 90. CEUR Workshop Proceedings. CEUR-WS.org, 2003 (cit. on p. 81).

[182] FIMI ’04, Proceedings of the IEEE ICDM Workshop on Frequent Itemset Mining Implementations, Brighton, UK, November 1, 2004. Vol. 126. CEUR Workshop Proceedings. CEUR-WS.org, 2005 (cit. on p. 81).

[183] Bilal Alsallakh, Luana Micallef, Wolfgang Aigner, Helwig Hauser, Silvia Miksch, and Peter J. Rodgers. The State-of-the-Art of Set Visualization. In: *Comput. Graph. Forum* 35 (2016), pp. 234–260. DOI: 10.1111/cgf.12722 (cit. on pp. 82, 84).

[184] Wolfgang Aigner, Silvia Miksch, Heidrun Schumann, and Christian Tominski. Visualization of Time-Oriented Data. Human-Computer Interaction Series. Springer, 2011 (cit. on pp. 82, 96).

[185] Sônia Fernandes Silva and Tiziana Catarci. Visualization of Linear Time-Oriented Data: A Survey. In: *WISE*. 2000, pp. 310–319 (cit. on pp. 82, 97).

[186] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visualizing time-oriented data - A systematic view. In: *Computers & Graphics* 31 (2007), pp. 401–409 (cit. on pp. 82, 97).

[187] Heike Hofmann, Arno Siebes, and Adalbert F. X. Wilhelm. Visualizing association rules with interactive mosaic plots. In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*. 2000, pp. 227–235. DOI: 10.1145/347090.347133 (cit. on pp. 82, 94, 107, 108).

[188] Michael Hahsler and Sudheer Chelluboina. Visualizing association rules: Introduction to the R-extension package arulesViz. In: *R project module* (2011), pp. 223–238 (cit. on pp. 82, 108, 111).

[189] Jiawei Han. Mining Knowledge at Multiple Concept Levels. In: *CIKM*. 1995, pp. 19–24 (cit. on p. 84).

[190] Douglas Burdick, Manuel Calimlim, and Johannes Gehrke. MAFIA: A Maximal Frequent Itemset Algorithm for Transactional Databases. In: *ICDE*. 2001, pp. 443–452 (cit. on p. 84).

[191] Nicolas Pasquier, Yves Bastide, Rafik Taouil, and Lotfi Lakhal. Efficient Mining of Association Rules Using Closed Itemset Lattices. In: *Inf. Syst.* 24 (1999), pp. 25–46 (cit. on pp. 85, 107).

[192] Gwenael Bothorel, Mathieu Serrurier, and Christophe Hurter. Visualization of Frequent Itemsets with Nested Circular Layout and Bundling Algorithm. In: *Advances in Visual Computing - 9th International Symposium, ISVC 2013, Rethymnon, Crete, Greece, July 29-31, 2013. Proceedings, Part II*. Vol. 8034. 2013, pp. 396–405. DOI: 10.1007/978-3-642-41939-3_38 (cit. on pp. 85, 107).

[193] George H. Collier. Thoth-II: Hypertext with Explicit Semantics. In: *Hypertext’87 Proceedings, November 13-15, 1987, Chapel Hill, North Carolina, USA*. 1987, pp. 269–289. DOI: 10.1145/317426.317446 (cit. on p. 85).

[194] Tamara Munzner, Qiang Kong, Raymond T Ng, Jordan Lee, Janek Klawe, Dragana Radulovic, and Carson K Leung. Visual mining of power sets with large alphabets. In: *Department of Computer Science, The University of British Columbia* (2005) (cit. on pp. 86, 107).

[195] Daniel A Keim, Jörn Schneidewind, and Mike Sips. Fp-viz: Visual frequent pattern mining. In: *InfoVis*. 2005 (cit. on pp. 87, 107).

[196] Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. In: *Data Min. Knowl. Discov.* 8 (2004), pp. 53–87. doi: 10.1023/B:DAMI.0000005258.31418.83 (cit. on pp. 86, 87).

[197] John T. Stasko and Eugene Zhang. Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations. In: *IEEE Symposium on Information Visualization 2000 (INFOVIS'00), Salt Lake City, Utah, USA, October 9-10, 2000*. 2000, pp. 57–65. doi: 10.1109/INFVIS.2000.885091 (cit. on p. 87).

[198] Jing Yang, Matthew O. Ward, Elke A. Rundensteiner, and Anilkumar Patro. InterRing: a visual interface for navigating and manipulating hierarchies. In: *Information Visualization* 2 (2003), pp. 16–30. doi: 10.1057/palgrave.ivs.9500035 (cit. on p. 87).

[199] Carson Kai-Sang Leung and Fan Jiang. RadialViz: An Orientation-Free Frequent Pattern Visualizer. In: *Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29 - June 1, 2012, Proceedings, Part II*. Vol. 7302. 2012, pp. 322–334. doi: 10.1007/978-3-642-30220-6_27 (cit. on pp. 87, 107).

[200] Carson Kai-Sang Leung, Fan Jiang, and Pourang P. Irani. FpMapViz: A Space-Filling Visualization for Frequent Patterns. In: *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011*. 2011, pp. 804–811. doi: 10.1109/ICDMW.2011.86 (cit. on pp. 87, 107).

[201] Ben Shneiderman. Tree Visualization with Tree-Maps: 2-d Space-Filling Approach. In: *ACM Trans. Graph.* 11 (1992), pp. 92–99. doi: 10.1145/102377.115768 (cit. on p. 87).

[202] Carson K. Leung, Vadim V. Kononov, Adam G. M. Pazdor, and Fan 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*. 2016, pp. 913–916. doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.158 (cit. on pp. 87, 88, 107).

[203] Li Yang. Visualizing Frequent Itemsets, Association Rules, and Sequential Patterns in Parallel Coordinates. In: *Computational Science and Its Applications - ICCSA 2003, International Conference, Montreal, Canada, May 18-21, 2003, Proceedings, Part I*. Vol. 2667. 2003, pp. 21–30. DOI: 10.1007/3-540-44839-X_3 (cit. on pp. 88, 95, 100, 107–109).

[204] Li Yang. Pruning and Visualizing Generalized Association Rules in Parallel Coordinates. In: *IEEE Trans. Knowl. Data Eng.* 17 (2005), pp. 60–70. DOI: 10.1109/TKDE.2005.14 (cit. on p. 89).

[205] Carson Kai-Sang Leung, Pourang Irani, and Christopher L. Carmichael. FIsViz: A Frequent Itemset Visualizer. In: *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008, Osaka, Japan, May 20-23, 2008 Proceedings*. Vol. 5012. 2008, pp. 644–652. DOI: 10.1007/978-3-540-68125-0_60 (cit. on pp. 89, 107).

[206] Carson Kai-Sang Leung, Pourang Irani, and Christopher 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*. 2008, pp. 875–880. DOI: 10.1109/ICDM.2008.93 (cit. on pp. 89, 90, 107).

[207] Carson Kai-Sang Leung and Christopher L. Carmichael. FpVAT: a visual analytic tool for supporting frequent pattern mining. In: *SIGKDD Explor.* 11 (2009), pp. 39–48. DOI: 10.1145/1809400.1809407 (cit. on p. 90).

[208] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. Mining Association Rules between Sets of Items in Large Databases. In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, May 26-28, 1993*. 1993, pp. 207–216. DOI: 10.1145/170035.170072 (cit. on p. 90).

[209] Jiawei Han, Yongjian Fu, Wei Wang, Jenny Chiang, Wan Gong, Krzysztof Koperski, Deyi Li, Yijun Lu, Amynmohamed Rajan, Nebojsa Stefanovic, Betty Xia, and Osmar R. ZaıU0308XXXXXXXXXXXane. DBMiner: A System for Mining Knowledge in Large Relational Databases. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA*. 1996, pp. 250–255 (cit. on pp. 91, 107, 108).

[210] Chris P. Rainsford and John F. Roddick. Visualisation of Temporal Interval Association Rules. In: *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*. Vol. 1983. 2000, pp. 91–96. DOI: 10.1007/3-540-44491-2_14 (cit. on pp. 91, 107, 108).

[211] Takeshi Fukuda, Yasukiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization. In: *ACM SIGMOD Record* 25 (1996), pp. 13–23 (cit. on pp. 92, 107, 108).

[212] Jianchao Han and Nick Cercone. AViz: A Visualization System for Discovering Numeric Association Rules. In: *Knowledge Discovery and Data Mining, Current Issues and New Applications, 4th Pacific-Asia Conference, PADKK 2000, Kyoto, Japan, April 18-20, 2000, Proceedings*. Vol. 1805. 2000, pp. 269–280. DOI: 10.1007/3-540-45571-X_33 (cit. on pp. 92, 108).

[213] Rakesh Agrawal, Manish Mehta, John C. Shafer, Ramakrishnan Srikant, Andreas Arning, and Toni Bollinger. The Quest Data Mining System. In: *KDD*. 1996, pp. 244–249 (cit. on pp. 92, 108).

[214] Clifford Brunk, James Kelly, and Ron Kohavi. MineSet: An Integrated System for Data Mining. In: *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, USA, August 14-17, 1997*. 1997, pp. 135–138 (cit. on pp. 93, 108).

[215] Pak Chung Wong, Paul Whitney, and James 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*. 1999, pp. 120–123. DOI: 10.1109/INFVIS.1999.801866 (cit. on pp. 93, 107, 108).

[216] Ickjai Lee, Guochen Cai, and Kyungmi 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*. 2013, pp. 1580–1588. DOI: 10.1109/HICSS.2013.401 (cit. on p. 94).

[217] John A Hartigan and Beat Kleiner. Mosaics for contingency tables. In: *Computer science and statistics: Proceedings of the 13th symposium on the interface*. Springer. 1981, pp. 268–273 (cit. on p. 94).

[218] Bing Liu, Wynne Hsu, Shu Chen, and Yiming Ma. Analyzing the Subjective Interestingness of Association Rules. In: *IEEE Intell. Syst.* 15 (2000), pp. 47–55. DOI: 10.1109/5254.889106 (cit. on pp. 95, 96, 138).

[219] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In: *KDD*. 2007, pp. 330–339 (cit. on p. 96).

[220] Gennady L. Andrienko, Natalia V. Andrienko, Peter Bak, Daniel A. Keim, and Stefan Wrobel. Visual Analytics of Movement. Springer, 2013 (cit. on p. 96).

[221] Philippe Fournier-Viger, Antonio Gomariz, Manuel Campos, and Rincy Thomas. Fast Vertical Mining of Sequential Patterns Using Co-occurrence Information. In: *PAKDD (1)*. Vol. 8443. 2014, pp. 40–52 (cit. on pp. 97, 149).

[222] Shin-yi Wu and Yen-Liang Chen. Discovering hybrid temporal patterns from sequences consisting of point- and interval-based events. In: *Data Knowl. Eng.* 68 (2009), pp. 1309–1330 (cit. on pp. 97, 109).

[223] Zhicheng Liu, Yang Wang, Mira Dontcheva, Matthew Hoffman, Seth Walker, and Alan Wilson. Patterns and Sequences: Interactive Exploration of Clickstreams to Understand Common Visitor Paths. In: *IEEE Trans. Vis. Comput. Graph.* 23 (2017), pp. 321–330. DOI: 10.1109/TVCG.2016.2598797 (cit. on pp. 98, 99).

[224] Joseph B Kruskal and James M Landwehr. Icicle plots: Better displays for hierarchical clustering. In: *The American Statistician* 37 (1983), pp. 162–168 (cit. on p. 99).

[225] Heidi Lam, Daniel M. Russell, Diane Tang, and Tamara Munzner. Session Viewer: Visual Exploratory Analysis of Web Session Logs. In: *2nd IEEE Symposium on Visual Analytics Science and Technology, IEEE VAST 2007, Sacramento, CA,USA, October 30 - November 1, 2007*. 2007, pp. 147–154. DOI: 10.1109/VAST.2007.4389008 (cit. on pp. 99, 109).

[226] Krist Wongsuphasawat, John Alexis Guerra Gómez, Catherine Plaisant, Taowei David Wang, Meirav Taieb-Maimon, and Ben Shneiderman. LifeFlow: visualizing an overview of event sequences. In: *CHI*. 2011, pp. 1747–1756 (cit. on p. 99).

[227] Krist Wongsuphasawat and Jimmy Lin. Using visualizations to monitor changes and harvest insights from a global-scale logging infrastructure at Twitter. In: *9th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2014, Paris, France, October 25-31, 2014*. 2014, pp. 113–122. DOI: 10.1109/VAST.2014.7042487 (cit. on p. 99).

[228] Jürgen Bernard, David Sessler, Thorsten May, Thorsten Schlomm, Dirk Pehrke, and Jörn Kohlhammer. A Visual-Interactive System for Prostate Cancer Cohort Analysis. In: *IEEE Computer Graphics and Applications* 35 (2015), pp. 44–55 (cit. on p. 99).

[229] Heikki Mannila and Christopher Meek. Global partial orders from sequential data. In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, MA, USA, August 20-23, 2000*. 2000, pp. 161–168. DOI: 10.1145/347090.347122 (cit. on pp. 99, 109).

[230] Debprakash Patnaik, Patrick Butler, Naren Ramakrishnan, Laxmi Parida, Benjamin J. Keller, and David A. Hanauer. Experiences with mining temporal event sequences from electronic medical records: initial successes and some challenges. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*. 2011, pp. 360–368. DOI: 10.1145/2020408.2020468 (cit. on pp. 99, 109).

[231] Mark Guzdial, Chris Walton, Michael Konemann, and Elliot Soloway. Characterizing process change using log file data. Tech. rep. Georgia Institute of Technology, 1993 (cit. on p. 100).

[232] Mengdie Hu, Krist Wongsuphasawat, and John T. Stasko. Visualizing Social Media Content with SentenTree. In: *IEEE Trans. Vis. Comput. Graph.* 23 (2017), pp. 621–630. DOI: 10.1109/TVCG.2016.2598590 (cit. on pp. 100, 109).

[233] Jürgen Bernard, Nils Wilhelm, Björn Krüger, Thorsten May, Tobias Schreck, and Jörn Kohlhammer. MotionExplorer: Exploratory Search in Human Motion Capture Data Based on Hierarchical Aggregation. In: *IEEE Trans. Vis. Comput. Graph.* 19 (2013), pp. 2257–2266 (cit. on p. 100).

[234] Jia-Kai Chou, Yang Wang, and Kwan-Liu Ma. Privacy preserving event sequence data visualization using a Sankey diagram-like representation. In: *SIGGRAPH Asia Symposium on Visualization*. 2016, 1:1–1:8 (cit. on p. 101).

[235] Adam Perer, Fei Wang, and Jianying Hu. Mining and exploring care pathways from electronic medical records with visual analytics. In: *Journal of Biomedical Informatics* 56 (2015), pp. 369–378 (cit. on p. 101).

[236] Jian Zhao, Zhicheng Liu, Mira Dontcheva, Aaron Hertzmann, and Alan Wilson. MatrixWave: Visual Comparison of Event Sequence Data. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI 2015, Seoul, Republic of Korea, April 18-23, 2015*. 2015, pp. 259–268. doi: 10.1145/2702123.2702419 (cit. on pp. 101, 109).

[237] Zhicheng Liu, Bernard Kerr, Mira Dontcheva, Justin Grover, Matthew Hoffman, and Alan Wilson. CoreFlow: Extracting and Visualizing Branching Patterns from Event Sequences. In: *Comput. Graph. Forum* 36 (2017), pp. 527–538 (cit. on pp. 102, 103, 109).

[238] Peter Grünwald. A tutorial introduction to the minimum description length principle. In: *CoRR* math.ST/0406077 (2004) (cit. on p. 103).

[239] Jishang Wei, Zeqian Shen, Neel Sundaresan, and Kwan-Liu Ma. Visual cluster exploration of web clickstream data. In: *IEEE VAST*. 2012, pp. 3–12 (cit. on pp. 104, 109).

[240] Fernanda B. Viégas, Martin Wattenberg, and Jonathan Feinberg. Participatory Visualization with Wordle. In: *IEEE Trans. Vis. Comput. Graph.* 15 (2009), pp. 1137–1144 (cit. on p. 104).

[241] David Gotz, Fei Wang, and Adam Perer. A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data. In: *Journal of Biomedical Informatics* 48 (2014), pp. 148–159 (cit. on pp. 104, 109).

[242] Heikki Mannila, Hannu Toivonen, and A. Inkeri Verkamo. Discovery of Frequent Episodes in Event Sequences. In: *Data Min. Knowl. Discov.* 1 (1997), pp. 259–289 (cit. on p. 105).

[243] Peter Bodesinsky, Bilal Alsallakh, Theresia Gschwandtner, and Silvia Miksch. Exploration and Assessment of Event Data. In: *6th International EuroVis Workshop on Visual Analytics, EuroVA@EuroVis 2015, Cagliari, Sardinia, Italy, May 25-26, 2015*. 2015, pp. 67–71. doi: 10.2312/eurova.20151106 (cit. on pp. 105, 106, 109).

[244] Martin Wattenberg. Arc Diagrams: Visualizing Structure in Strings. In: *INFOVIS*. 2002, pp. 110–116 (cit. on p. 105).

[245] Jacques Bertin. Semiology of graphics: diagrams, networks, maps. In: (1983) (cit. on p. 106).

[246] Jock D. Mackinlay. Automating the Design of Graphical Presentations of Relational Information. In: *ACM Trans. Graph.* 5 (1986), pp. 110–141 (cit. on pp. 106, 110).

[247] Tamara Munzner. Visualization Analysis and Design. A.K. Peters visualization series. A K Peters, 2014 (cit. on pp. 106, 110).

[248] Geoffrey P. Ellis and Alan J. Dix. The plot, the clutter, the sampling and its lens: occlusion measures for automatic clutter reduction. In: *AVI*. 2006, pp. 266–269 (cit. on p. 106).

[249] Jacques Bertin. Sémiologie graphique: Les diagrammes-Les réseaux-Les cartes. In: (1973) (cit. on p. 109).

[250] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: *Proceedings of the 1996 IEEE Symposium on Visual Languages, Boulder, Colorado, USA, September 3-6, 1996*. 1996, pp. 336–343. DOI: 10.1109/VL.1996.545307 (cit. on pp. 110, 128).

[251] Daniel A. Keim. Information Visualization and Visual Data Mining. In: *IEEE Trans. Vis. Comput. Graph.* 8 (2002), pp. 1–8. DOI: 10.1109/2945.981847 (cit. on pp. 111, 120).

[252] John Risch, Anne Kao, Steve Poteet, and Yuan-Jye Jason Wu. Text Visualization for Visual Text Analytics. In: *Visual Data Mining - Theory, Techniques and Tools for Visual Analytics*. Vol. 4404. Lecture Notes in Computer Science. Springer, 2008, pp. 154–171. DOI: 10.1007/978-3-540-71080-6_11 (cit. on p. 113).

[253] Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. In: *AI Mag.* 17 (1996), pp. 37–54. DOI: 10.1609/aimag.v17i3.1230 (cit. on p. 115).

[254] Bram C. M. Cappers, Paulus N. Meessen, Sandro Etalle, and Jarke J. van Wijk. Eventpad: Rapid Malware Analysis and Reverse Engineering using Visual Analytics. In: *15th IEEE Symposium on Visualization for Cyber Security, VizSec 2018, Berlin, Germany, October 22, 2018*. 2018, pp. 1–8. DOI: 10.1109/VIZSEC.2018.8709230 (cit. on p. 118).

[255] Yun Sing Koh and Sri Devi Ravana. Unsupervised Rare Pattern Mining: A Survey. In: *ACM Trans. Knowl. Discov. Data* 10 (2016), 45:1–45:29. DOI: 10.1145/2898359 (cit. on p. 118).

[256] Anindita Borah and Bhabesh Nath. Rare pattern mining: challenges and future perspectives. In: *Complex & Intelligent Systems* 5 (2019), pp. 1–23 (cit. on p. 118).

[257] Kai Xu, Simon Attfield, T. J. Jankun-Kelly, Ashley Wheat, Phong H. Nguyen, and Nallini Selvaraj. Analytic Provenance for Sensemaking: A Research Agenda. In: *IEEE Computer Graphics and Applications* 35 (2015), pp. 56–64. DOI: 10.1109/MCG.2015.50 (cit. on p. 122).

[258] Alex Endert. Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering. Synthesis Lectures on Visualization. Morgan & Claypool Publishers, 2016 (cit. on p. 122).

[259] Daniel Seebacher, Thomas Polk, Halldor Janetzko, Daniel Keim, Tobias Schreck, and Manuel Stein. Investigating the Sketchplan: A Novel Way of Identifying Tactical Behavior in Massive Soccer Datasets. In: *IEEE Transactions on Visualization and Computer Graphics* (2021) (cit. on p. 127).

[260] Franco Moretti. Style, Inc. Reflections on Seven Thousand Titles (British Novels, 1740–1850). In: *Critical Inquiry* 36 (2009), pp. 134–158 (cit. on p. 129).

[261] Franco Moretti. Conjectures on world literature. In: *New left review* 1 (2000), p. 54 (cit. on p. 128).

[262] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: A comparative review. In: *J Mach Learn Res* 10 (2009), pp. 66–71 (cit. on p. 133).

[263] Yalong Yang, Wenyu Xia, Fritz Lekschas, Carolina Nobre, Robert Krüger, and Hanspeter Pfister. The Pattern is in the Details: An Evaluation of Interaction Techniques for Locating, Searching, and Contextualizing Details in Multivariate Matrix Visualizations. In: *CoRR* abs/2203.05109 (2022) (cit. on p. 134).

[264] Chris North, Remco Chang, Alex Endert, Wenwen Dou, Richard May, Bill Pike, and Glenn A. Fink. Analytic provenance: process+interaction+insight. In: *CHI Extended Abstracts*. 2011, pp. 33–36 (cit. on p. 140).

[265] Frank van Ham and Adam Perer. "Search, Show Context, Expand on Demand": Supporting Large Graph Exploration with Degree-of-Interest. In: *IEEE Trans. Vis. Comput. Graph.* 15 (2009), pp. 953–960. DOI: 10.1109/TVCG.2009.108 (cit. on p. 143).

[266] Richard Brath and Ebad Banissi. Using font attributes in knowledge maps and information retrieval. In: *CEUR Workshop Proceedings*. Vol. 1311. CEUR Workshop Proceedings. 2014, pp. 23–30 (cit. on p. 144).

[267] Matt Williams and Tamara Munzner. Steerable, Progressive Multidimensional Scaling. In: *10th IEEE Symposium on Information Visualization (InfoVis 2004), 10-12 October 2004, Austin, TX, USA*. 2004, pp. 57–64. DOI: 10.1109/INFVIS.2004.60 (cit. on p. 147).

[268] Emanuel Zgraggen, Alex Galakatos, Andrew Crotty, Jean-Daniel Fekete, and Tim Kraska. How Progressive Visualizations Affect Exploratory Analysis. In: *IEEE Trans. Vis. Comput. Graph.* 23 (2017), pp. 1977–1987. DOI: 10.1109/TVCG.2016.2607714 (cit. on p. 147).

[269] Jean-Daniel Fekete and Romain Primet. Progressive Analytics: A Computation Paradigm for Exploratory Data Analysis. In: *CoRR* abs/1607.05162 (2016) (cit. on p. 147).

[270] Jean-Daniel Fekete, Danyel Fisher, Arnab Nandi, and Michael Sedlmair. Progressive Data Analysis and Visualization (Dagstuhl Seminar 18411). In: *Dagstuhl Reports* 8 (2018), pp. 1–40. DOI: 10.4230/DagRep.8.10.1 (cit. on p. 147).

[271] Cagatay Turkay, Nicola Pezzotti, Carsten Binnig, Hendrik Strobelt, Barbara Hammer, Daniel A. Keim, Jean-Daniel Fekete, Themis Palpanas, Yunhai Wang, and Florin Rusu. Progressive Data Science: Potential and Challenges. In: *CoRR* abs/1812.08032 (2018) (cit. on p. 147).

[272] Vincent Raveneau. Interaction in Progressive Visual Analytics. An application to progressive sequential pattern mining. (Interaction en Analyse Visuelle Progressive. Une application à la fouille progressive de motifs séquentiels). PhD thesis. University of Nantes, France, 2020 (cit. on pp. 147, 148).

[273] Mohammed Javeed Zaki. SPADE: An Efficient Algorithm for Mining Frequent Sequences. In: *Mach. Learn.* 42 (2001), pp. 31–60 (cit. on p. 147).

[274] Katerina Vrotsou and Aida Nordman. Interactive visual sequence mining based on pattern-growth. In: *9th IEEE Conference on Visual Analytics Science and Technology, IEEE VAST 2014, Paris, France, October 25-31, 2014*. 2014, pp. 285–286. doi: 10.1109/VAST.2014.7042532 (cit. on p. 148).

[275] Matthijs van Leeuwen. Interactive Data Exploration Using Pattern Mining. In: *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics - State-of-the-Art and Future Challenges*. Vol. 8401. Lecture Notes in Computer Science. Springer, 2014, pp. 169–182. doi: 10.1007/978-3-662-43968-5_9 (cit. on pp. 151, 152).

[276] Mansurul Bhuiyan, Snehasis Mukhopadhyay, and Mohammad Al Hasan. Interactive pattern mining on hidden data: a sampling-based solution. In: *21st ACM International Conference on Information and Knowledge Management, CIKM'12, Maui, HI, USA, October 29 - November 02, 2012*. 2012, pp. 95–104. doi: 10.1145/2396761.2396777 (cit. on p. 151).

[277] Vladimir Dzyuba, Matthijs van Leeuwen, Siegfried Nijssen, and Luc De Raedt. Active Preference Learning for Ranking Patterns. In: *25th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2013, Herndon, VA, USA, November 4-6, 2013*. 2013, pp. 532–539. doi: 10.1109/ICTAI.2013.85 (cit. on pp. 151, 152).

[278] Esther Galbrun and Pauli Miettinen. A case of visual and interactive data analysis: Geospatial redescription mining. In: *Proceedings of the Instant Interactive Data Mining Workshop at ECML/PKDD 2012, IID'12*. 2012, pp. 1–12 (cit. on p. 151).

[279] Mario Boley, Michael Mampaey, Bo Kang, Pavel Tokmakov, and Stefan Wrobel. One click mining: interactive local pattern discovery through implicit preference and performance learning. In: *Proceedings of the ACM SIGKDD Workshop on Interactive Data Exploration and Analytics, IDEA@KDD 2013, Chicago, Illinois, USA, August 11, 2013*. 2013, pp. 27–35. doi: 10.1145/2501511.2501517 (cit. on p. 152).

[280] Tijl De Bie. An information theoretic framework for data mining. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011*. 2011, pp. 564–572. doi: 10.1145/2020408.2020497 (cit. on p. 152).

[281] Tijl De Bie. Maximum entropy models and subjective interestingness: an application to tiles in binary databases. In: *Data Min. Knowl. Discov.* 23 (2011), pp. 407–446. doi: 10.1007/s10618-010-0209-3 (cit. on p. 152).

[282] Eirini Spyropoulou, Tijl De Bie, and Mario Boley. Interesting pattern mining in multi-relational data. In: *Data Mining and Knowledge Discovery* 28 (2014), pp. 808–849 (cit. on p. 152).

[283] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. XAI - Explainable artificial intelligence. In: *Sci. Robotics* 4 (2019). doi: 10.1126/scirobotics.aay7120 (cit. on p. 154).

[284] Cognitive Biases in Visualizations. Springer, 2018 (cit. on p. 156).

[285] John D. Lee and Katrina A. See. Trust in Automation: Designing for Appropriate Reliance. In: *Human Factors* 46 (2004), pp. 50–80. DOI: 10.1518/hfes.46.1.50.30392 (cit. on p. 157).

[286] U.S. Department of Agriculture. What We Eat In America (WWEIA) Database. 2022. URL: https://data.nal.usda.gov/dataset/what-we-eat-america-wweia-database (cit. on p. 164).

[287] U.S. Centers for Disease Control and Prevention. National Health and Nutrition Examination Survey; 2017-2018 Data Documentation, Codebook, and Frequencies; Individual Foods, First Day (DR1IFF_J). 2022. URL: https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DR1IFF_J.htm (cit. on p. 164).

[288] U.S. Centers for Disease Control and Prevention. National Health and Nutrition Examination Survey; 2017-2018 Data Documentation, Codebook, and Frequencies; Body Measures (BMX_J). 2022. URL: https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/BMX_J.htm (cit. on p. 164).

[289] U.S. Centers for Disease Control and Prevention. National Health and Nutrition Examination Survey; 2017-2018 Data Documentation, Codebook, and Frequencies; Demographic Variables and Sample Weights (DEMO_J). 2022. URL: https://wwwn.cdc.gov/Nchs/Nhanes/2017-2018/DEMO_J.htm (cit. on p. 164).

[290] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents. In: *CoRR* abs/2204.06125 (2022). DOI: 10.48550/arXiv.2204.06125 (cit. on p. 177).

[291] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*. 2022, pp. 10674–10685. DOI: 10.1109/CVPR52688.2022.01042 (cit. on p. 177).

[292] OpenAI. ChatGPT: Optimizing Language Models for Dialogue. https://openai.com/blog/chatgpt/. 2022 (cit. on p. 177).

[293] Benjamin Paaßen, Claudio Gallicchio, Alessio Micheli, and Alessandro Sperduti. Embeddings and Representation Learning for Structured Data. In: *ESANN*. 2019 (cit. on p. 177).

[294] Aatif Jamshed, Bhawna Mallick, and Pramod Kumar. Deep learning-based sequential pattern mining for progressive database. In: *Soft Comput.* 24 (2020), pp. 17233–17246. DOI: 10.1007/s00500-020-05015-2 (cit. on p. 177).

[295] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, Han-Chieh Chao, and Philip S. Yu. A Survey of Parallel Sequential Pattern Mining. In: *CoRR* abs/1805.10515 (2018) (cit. on p. 177).

[296] Catherine Plaisant and Ben Shneiderman. The diversity of data and tasks in event analytics. In: *Proceedings of the IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis*. 2016 (cit. on p. 177).

[297] David Gotz. Soft patterns: Moving beyond explicit sequential patterns during visual analysis of longitudinal event datasets. In: *Proceedings of the IEEE VIS 2016 Workshop on Temporal & Sequential Event Analysis*. 2016 (cit. on p. 178).

[298] Charu C. Aggarwal, Yan Li, Jianyong Wang, and Jing Wang. Frequent pattern mining with uncertain data. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*. 2009, pp. 29–38. DOI: 10.1145/1557019.1557030 (cit. on p. 178).

[299] Carson Kai-Sang Leung, Mark Anthony F. Mateo, and Dale A. Brajczuk. A Tree-Based Approach for Frequent Pattern Mining from Uncertain Data. In: *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008, Osaka, Japan, May 20-23, 2008 Proceedings*. Vol. 5012. 2008, pp. 653–661. DOI: 10.1007/978-3-540-68125-0_61 (cit. on p. 178).

[300] Jessica Hullman. Why Authors Don't Visualize Uncertainty. In: *IEEE Trans. Vis. Comput. Graph.* 26 (2020), pp. 130–139. DOI: 10.1109/TVCG.2019.2934287 (cit. on p. 178).

[301] Giuliana Dehn. Visual Analytics for the Exploration of Sequential Rules. MA thesis. University of Konstanz, 2020 (cit. on p. 178).

[302] Fabian Sperrle, Davide Ceneda, and Mennatallah El-Assady. Lotse: A Practical Framework for Guidance in Visual Analytics. In: *CoRR* abs/2208.04434 (2022). DOI: 10.48550/arXiv.2208.04434 (cit. on p. 178).

[303] Fabian Sperrle, Jürgen Bernard, Michael Sedlmair, Daniel A. Keim, and Mennatallah El-Assady. Speculative Execution for Guided Visual Analytics. In: *CoRR* abs/1908.02627 (2019) (cit. on p. 178).