# Knowledge Generation in Visual Analytics

**Integrating Human and Machine Intelligence for Exploration of Big Data**

Dissertation zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)

vorgelegt von
**Dominik Sacha**

an der

Universität
Konstanz

Mathematisch-Naturwissenschaftliche Sektion
Fachbereich Informatik und Informationswissenschaft

Tag der mündlichen Prüfung: 6. April 2018

Referenten: Prof. Dr. Daniel A. Keim, University of Konstanz
Prof. Dr. Min Chen, University of Oxford
Prof. Dr. Harald Reiterer, University of Konstanz

*To my beloved wife Janina.*

# Acknowledgements


I never expected such an exciting, inspiring, and informative period of my life when I started to work at the Data Analysis and Visualization Group (DBVIS) at the University of Konstanz in February 2013. Within the last 4,5 years, I had many valuable collaborations and discussions that shaped the work of this dissertation. Therefore, I want to express my deep gratitude to all people I have met and those who supported me during this time.

I want to express my special thanks to Prof. Dr. Daniel Keim as my primary supervisor and mentor. He advised me to have a closer look at "interaction" in visual analytics what rapidly shaped the foundation of all my research. He always provided me with valuable feedback and guided me through all phases of my time as a doctoral student. I also profited a lot from his contacts to other inspiring researchers in the field who became valuable collaborators and co-authors. Without him, my doctoral dissertation would not have been possible.

Further thanks go to Prof. Dr. Min Chen for his constructive feedback and insightful comments that helped me to shape and fine-tune this dissertation. I want also thank Dr. Geoffrey Ellis and Prof. Dr. Tobias Schreck, who helped me in gaining ground within the academic world (e.g., paper writing, asking the right questions, discussions, etc.) and also collaborated with me in many research projects during my entire doctoral student time.

Special thanks go to all my colleagues at the DBVIS group that supported me in any circumstances and issues. Especially in the beginning, Sabine Kuhr helped me a lot with the formal requirements while Dr. Andreas Stoffel, Dr. Sebastian Mittelstädt, Dr. Christian Rohrdantz, Dr. Halldór Janetzko, Dr. Johannes Fuchs, Dr. Michael Behrisch, and Dr. Bum Chul Kwon supported and involved me early on in paper and research projects. Further thanks go to my colleagues Dominik Jäckle, Florian Stoffel, Juri Buchmüller, and Wolfgang Jentner who accompanied me during my whole time at the University of Konstanz. However, I want to thank the whole group for our daily meetings, conversations, and collaborations. I also want to thank the Bachelor and Master students that worked with me in developing research prototypes for this dissertation (Ina Boesecke, Feeras Al-Masoudi, Matthias Kraus).

I also want to thank my external collaborators who became co-authors in many publications that are part of this dissertation: Dr. Leishi Zhang, Dr. Michael Sedlmair, Dr. John Aldo Lee, Prof. Dr. Jaakko Peltonen, Prof. Dr. Daniel Weiskopf, Dr. Stephen North, Dr. Jürgen Bernard, and Prof. Dr. Gennady Andrienko.

Finally, I want to acknowledge that this work was funded by several research projects that I was involved in. I want to thank the German BMBF for funding my research within the project "Visual Analytics for Security Applications" (VASA). I also want to thank the German Research Foundation (DFG) for funding my research in collaboration with the linguistic department at the University of Konstanz (within the Research Initiative "LingVisAnn", the project "Visual Analysis of Language Change and Use Patterns", and the Research Unit "Questions at the Interfaces"). Finally, I want to thank the European Commission for funding my research within the project "Visual Analytics for Sense-Making in Criminal Intelligence Analysis" (VALCRI). I express my gratitude to all the project members for their collaboration, feedback, and discussions within interdisciplinary projects that allowed us to apply and evaluate our research.

# Abstract

Big data poses many facets and challenges when analyzing data, often described with the five big V's of Volume, Variety, Velocity, Veracity, and Value. However, the most important V – Value can only be achieved when knowledge can be derived from the data. The volume of nowadays datasets make a manual investigation of all data records impossible and automated analysis techniques from data mining or machine learning often cannot be applied in a fully automated fashion to solve many real world analysis problems, and hence, need to be manually trained or adapted. Visual analytics aims to solve this problem with a "human-in-the-loop" approach that provides the analyst with a visual interface that tightly integrates automated analysis techniques with human interaction. However, a holistic understanding of these analytic processes is currently an under-explored research area.

A major contribution of this dissertation is a conceptual model-driven approach to visual analytics that focuses on the human-machine interplay during knowledge generation. At its core, it presents the knowledge generation model which is subsequently specialized for human analytic behavior, visual interactive machine learning, and dimensionality reduction. These conceptual processes extend and combine existing conceptual works that aim to establish a theoretical foundation for visual analytics. In addition, this dissertation contributes novel methods to investigate and support human knowledge generation processes, such as semi-automation and recommendation, analytic behavior and trust building, or visual interaction with machine learning. These methods are investigated in close collaboration with real experts from different application domains (such as soccer analysis, linguistic intonation research, and criminal intelligence analysis) and hence, different data characteristics (geo-spatial movement, time series, and high-dimensional). The results demonstrate that this conceptual approach leads to novel, more tightly integrated, methods that support the analyst in knowledge generation. In a final broader discussion, this dissertation reflects the conceptual and methodological contributions and enumerates research areas at the intersection of data mining, machine learning, visualization, and human-computer interaction research, with the ultimate goal to make big data exploration more effective, efficient, and transparent.

# Zusammenfassung

Die Analyse von großen Datenmengen (Big Data) birgt eine Vielzahl an Fassetten und Herausforderungen, welche oft durch die fünf Vs (engl.) Volume, Variety, Velocity, Veracity und Value beschrieben werden. Der wichtigste Aspekt dieser Vs – Nutzen (Value) kann jedoch nur gewonnen werden, wenn Wissen aus den Daten generiert werden kann. Das Volumen heutiger Datensätze macht eine manuelle Inspektion der Daten unmöglich und automatisierte Analyseverfahren aus dem Data Mining oder maschinellen Lernen können meist nicht vollautomatisiert auf reale Probleme angewandt werden und müssen daher manuell angepasst oder trainiert werden. Um dieses Problem zu lösen, bindet Visual Analytics den Menschen in den Analyseprozess ein (engl. “human-in-the-loop”). Dies wird durch eine visuelle Benutzerschnittstelle realisiert, welche automatisierte Analyseverfahren eng mit menschlicher Benutzerinteraktion koppelt. Ein ganzheitliches Verständnis solcher Analyseprozesse wurde jedoch noch nicht ausreichend erforscht und bildet somit das Problemfeld dieser Arbeit.

Diese Dissertation wählt eine konzeptionelle, modellgetriebene Herangehensweise an Visual Analytics, welche die Zusammenarbeit von Mensch und Maschine während des Wissenserzeugungsprozesses fokussiert. Als Grundlage wird das Wissenserzeugungs-Prozessmodell vorgestellt, welches dann schrittweise für das menschliche analytische Verhalten, visuell-interaktives maschinelles Lernen und visuell-interaktive Dimensionsreduktion spezialisiert wird. Die konzeptionellen Prozesse vereinen und kombinieren die bereits bestehenden Arbeiten, die zum Ziel haben, eine theoretische Grundlage in der Visual Analytics Forschung zu etablieren. Zusätzlich trägt diese Dissertation neuartige Verfahren zur Untersuchung und Unterstützung von menschlichen Wissenserzeugungsprozessen bei. Es handelt sich um Themen wie Halb-Automatisierung und Empfehlungssysteme, analytisches Verhalten und Vertrauensbildung, oder visuelle Interaktion mit maschinellem Lernen. Diese Verfahren werden in enger Zusammenarbeit mit echten Experten aus verschiedenen Anwendungsgebieten untersucht (Analyse von Bewegungen im Fußball, linguistische Intonationsforschung, Analyse von Kriminalfällen) und somit unterschiedlichen Daten-Eigenschaften (räumliche Bewegungsdaten, Zeitserien und hochdimensionale Daten). Die Ergebnisse zeigen, dass die konzeptionelle Herangehensweise zu neuartigen, eng integrierten Verfahren führt, welche die Analysten bei der Wissenserzeugung unterstützen. In einer abschließenden breiteren Diskussion reflektiert diese Dissertation die konzeptionellen und methodischen Beiträge und zeigt weitere Forschungsgebiete an den Schnittstellen von Data Mining, maschinellem Lernen, Visualisierung und Mensch-Maschine Interaktion auf. Diese haben das gemeinsame Ziel die Exploration großer Datenmengen effektiver, effizienter und transparenter zu machen.

# Contents

# 1 Introduction

*"The greatest value of a picture is when it forces us to notice what we never expected to see."* – John Tukey

**Contents**



## 1.1 Motivation

Many data records of different types are created every second and stored in large databases, such as time-series of share-prices, business transactions, captured experiments in research, tracked movement data, or text documents in social media and humanity is subject to massive information overload [150]. It remains a challenge to analyze and extract valuable information that is hidden in often noisy data sources. Purely automatic approaches and conventional statistics are often insufficient for domain experts to tackle real world analysis problems. They often require a more exploratory fashioned data analysis [289] to iteratively shape and refine hypothesis that can be confirmed or rejected. Information visualization [48] transforms data attributes into visual structures and enables the analyst to explore and interact with the results visually. However, noisy data and information overload often preclude the analyst from spotting patterns in the visualizations and the massive amount of big data poses a major challenge to visualize the raw data. Hence, there is an essential need for automated analysis techniques (e.g., from data mining or machine learning) to process, aggregate, and "prepare" the data to reveal patterns in the visualizations. Computers are able to perform enormous amounts of computations and are able to leverage memory that is only limited by hardware. Humans are able to leverage their knowledge that composes of many facets (e.g., experiences, tactical-, domain-knowledge, skills) – information that is often unavailable to the machine – allowing the analysts to reason and think about their problems. Visual Analytics (VA) provides a visual interface between automated techniques and the human analyst with the aim to effectively combine these human and machine strengths [150]. This duality results in a conversation between human and machine, where computational (intermediate) results are communicated by visualizations and human feedback is expressed by user interaction. Many analysis systems lack the ability to support the human effectively during the analysis process in generating knowledge from data [248]. Humans are restricted to remember a limited amount of information chunks [195] and our attention during the analysis process is framed by response time limits [196]. Furthermore, analysts are often overwhelmed by information and a lot of configuration options. Visual interfaces need to be tailored to effectively leverage human strengths, such as pre-attentive processing of the human's visual system [309] in order to spot patterns and allowing the analyst intuitively to induce their domain knowledge and expectations into the computations by

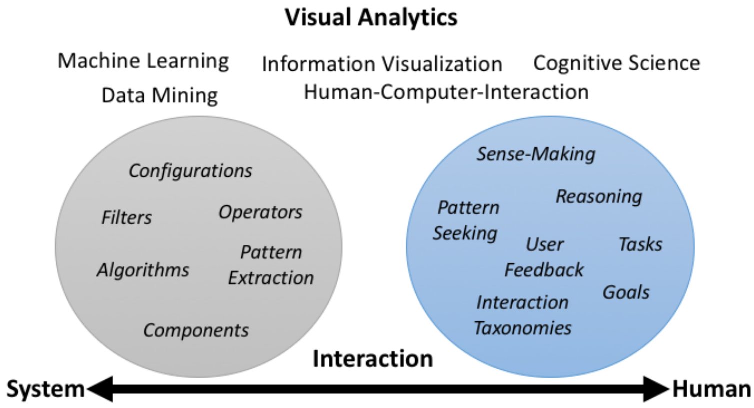



Figure 1.1: Related conceptual works and research areas.

interacting with the system. To advance in VA "better and more usable solutions" are needed to extract knowledge from data with a specific challenge to meet the needs of the users [150].

A theoretical foundation of analytical reasoning and a "science of interaction" that supports reasoning needs to be established to improve the design of future VA systems [286]. Hence, an interdisciplinary perspective and combination of existing works from several domains that interfere with VA are needed [150]. Figure 1.1 shows that VA involves data mining, machine learning (ML), information visualization (IV), human-computer interaction (HCI) as well as the cognitive sciences. The focus of the conceptual works within these areas varies between system and human aspects. For example, pipeline based approaches such as the knowledge discovery in databases process [91], the reference model for IV [48], or the VA process model [150] describe data analysis systems as a set of connected components, algorithms, or operations that can be configured and manipulated. Another set of works focuses on interactions between systems and humans. Interaction taxonomies describe analysis tasks (e.g., [38]) and interaction types on different levels (e.g., [104, 220]) while other models from HCI describe several stages of actions ranging from goals, via executions of actions, to evaluations of results (e.g., [207]). Other, more human-centered approaches focus on human sense-making or reasoning activities (e.g., [221]). Recently, a panel at the IEEE VIS Conference in Baltimore, Maryland in October 2016, discussed several aspects of pursuing theoretical research in visualization. The resulting paper by Chen et. al [57] covers different "pathways for theoretical advances in visualization" discussing major aspects or stages of an evolving theoretical foundation (also compared to other scientific fields, such as physics) that influence each other (shown in Figure 1.2): *Taxonomies and Ontologies* describe a set of concepts and their relations, *Principles and Guidelines* postulate causal relationships and rules, *Conceptual Models* describe an abstract representation of real-world processes including several components and their interactions, *Theoretic Frameworks* further provide a set of measurements and operators as a basis for evaluating models quantitatively, *Quantitative Laws* describe a set of causal relationships and measurements that can be shared within a *Theoretic System*.

This doctoral thesis approaches VA with a more holistic perspective on the entire knowledge generation process contributing mostly on the left hand side of the framework shown in Figure 1.2. It provides abstract representations of real world knowledge generation processes illustrating different human and machine processes and their interactions as *Conceptual Process Models*. These abstract representations are grounded in existing related theoretical frameworks but are also derived form real world example VA scenarios with the motivation to better support and integrate these processes with novel VA methods. However, on a meta-perspective, such process models are based on the identification and description of human and and machine concepts during knowledge generation together with relations as essential parts of *Ontologies*. Please note that ontological thinking differs

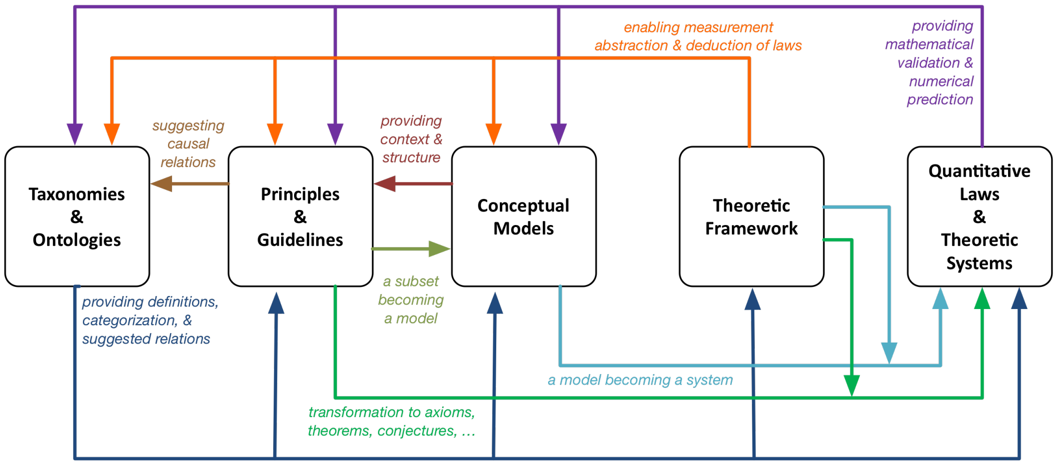



Figure 1.2: Major aspects of theory evolution adapted from Chen et al. [57].

from conceptual process modeling in terms of providing a higher-level, extensible, and more general framework for many possible examples (e.g., like a "dictionary" that can be used to write a sentence or a "map" that can be used to identify pathways) in contrast to process modeling that describes concrete scenarios with the aim to support the investigation and development of concrete applications. Hence, the major parts of this dissertation focus on conceptual process modeling while Section 6.2 will introduce a more general ontological framework for these conceptual process models. Figure 1.2 illustrates that these major aspects of a theoretical foundation influence each other, however, describing theoretic frameworks (which can lead to quantitative laws within a theoretic system in the future) are not within the scope of this dissertation. Instead, this doctoral thesis demonstrates how the presented conceptual process models can be used and instantiated to identify major drawbacks of current VA systems and major research opportunities. It further provides principles and guidelines to overcome and mitigate these drawbacks. A second methodological part of this dissertation derives novel methods from these conceptual process models and provides prototypical implementations. A detailed derivation and illustration of the major research areas and problems are provided in the next section.

## 1.2 Research Problems & Areas

The central question of this doctoral thesis is: **"How to achieve a tight integration between automated analysis and visual interaction to better support human knowledge generation in VA?"** Hence, the work focuses on the following major problem areas:

**RP1 Contributing to a Theoretical Foundation for Knowledge Generation in VA:** Current conceptual work, models, and frameworks provide guidance in designing and evaluating VA systems. However, they are lacking essential parts of the knowledge generation process, such as technical components, human-computer interaction and human sense-making/reasoning phases. A clarification of the human role within this process, a characterization of interactions, tasks, and kinds of provided user feedback is missing, especially when focusing on the underlying computational parts (data mining and machine learning techniques) that are crucial for VA. A unified and comprehensive perspective onto the entire knowledge generation process is the foundation for the identification and solution of major problems that hinder humans in generating knowledge from data.

**RP2 Identification of Analysis Problems and Research Gaps:** In a typical VA setup, domain experts (with often limited technical competence in data analysis) are using complex analysis methods in combination with visualizations. However, important aspects and functionalities to facilitate usage and support novice analysts are missing in many VA systems. Hence, a data scientist (or tool developer) has to guide the analyst in a time consuming analysis process. Analysts are often overwhelmed by an excessive number of parameters, unsure how to handle the user interface, unable to remember what has been done, or to validate the derived results. A major subject of this dissertation is to identify and put such problems into a broader context within the knowledge generation process and to provide guidance to VA researchers and system designers. A major research area and methodology will be to apply the developed conceptual models to existing VA systems (evaluative use) but also to identify under-explored research areas (generative use).

**RP3 Overcoming Problems During Knowledge Generation:** While some of the analysis problems during knowledge generation can certainly be mitigated by raising their awareness, or by teaching principles and guidelines, some others can be addressed technically. This dissertation will focus on investigating methods to support knowledge generation with the aim to make VA more effective, efficient, and transparent by addressing three major problem areas:

**RP3-1 Semi-Automation and Recommendation:** Some interactions and analysis tasks are relatively time consuming but simple. In these scenarios, it is beneficial to pre-configure the desired visualizations semi-automatically to speed up and facilitate the analysis process. Analysts are often unable to start or continue their analysis when they do not know how to handle the system, interpret the results, or how to interact. In these situations, they require guidance by the system.

**RP3-2 Analytic Behavior and Trust Building:** There are many, under-explored, human aspects that influence the analysis process: Cognitive biases, over- or under-trusting in (un-) certain computed results, and the humans reasoning process. We need to know more about analytic behavior related to trust building. Many VA systems fall short in supporting and integrating higher-level analytic processes (evidence-gathering and verification, hypothesis generation and refinement).

**RP3-3 Visual Interaction with Machine Learning:** Analysts without a technical background in data mining or machine learning are often not able to evaluate results and find it difficult to provide useful interactive feedback to semi-automated analysis methods. Interactive visualization can assist the analyst to understand and interpret the results. Similarly, a direct visual way to interact can make these methods usable and accessible to the analysts.

## 1.3 Research Methodology & Thesis Contributions

The research methodology and contributions of this doctoral thesis are two-fold (conceptual and methodological):

**C1 Model-Driven Approach to VA:** The core contribution of this thesis is a model-driven approach to VA that provides novel perspectives onto currently under-explored but important aspects within the knowledge generation process. The resulting artifacts (ontologies, guidelines, conceptual process models) build on existing previous works and are inspired by existing example systems. They are further instantiated by applying them to a rich set of examples to test their utility and to align them with existing conceptual models (evaluative use). This process results in both extending the process models to add missing features, and summarizing (simplification) common aspects across several example systems and existing theoretic work. This process also enables us to identify open areas of research for improving the state of the art and to generate novel methods that support knowledge generation (generative use). This approach is inspired by *Grounded Theory* [54], in which data (here papers and examples) is systematically analyzed to identify and refine categories, which are then used to incrementally build up a theoretical model. This approach has been used in visualization research [131, 167, 258] and related areas such as HCI before [125], and its importance for building up the much needed theoretical foundation in visualization research has been recognized [57, 210, 258]. The conceptual core contribution of this work is the **Knowledge Generation Model for Visual Analytics** that provides the basic context and characterization of human and machine concepts within the knowledge generation process. Based on this conceptual process model, the thesis also provides more specialized perspectives: A sharper focus on the definition of "valid knowledge" and its relation to the entire knowledge generation process (e.g., computational uncertainties or human trust building activities) leads to a novel conceptual process model defining the role of **Uncertainty, Awareness, and Trust** in VA. Another focus on interpreting and steering complex computational methods delivers a general human-centered **Visual Interactive Machine Learning Model** as well as a more specific and actionable conceptual process model for **Visual Interactive Dimensionality Reduction**. These process models shed light onto different kinds of user feedback and enumerate interactions along the visual interactive ML pipeline. C1 addresses the need for a theoretical foundation (**RP1**) and its evaluative and generative use (**RP2**) with a conceptual contribution to VA.

**C2 Novel Methods to Analyze & Support Knowledge Generation:** The thesis contributes four VA systems that address specific identified issues to analyze and support knowledge generation. These tools strive for a **"tight integration of visual interaction and automated analysis"**, supporting lower and higher-level analytic activities. The respective theoretical artifacts (C1) are leveraged to identify relevant research opportunities and to guide the development of novel VA methods. These methods are developed in close collaboration with domain experts having real data and analysis problems in a *User-Centered Design Process* [35, 225]. Therefore, the thesis adopts the *Design Study Methodology* approach [259] to build and evaluate research prototypes. I took the role of the VA system developer providing the VA methods to domain experts, but I became a Liaison [272] in most of these projects. The early prototypes have been used in *Pair Analytics* settings (where a subject matter expert collaborates with a VA expert [12]) to analyze the data, teach the analyst, observe the analysis process, and to collect qualitative feedback. We also conducted quantitative user experiments with more mature versions of our systems. The research areas to overcome current problems during knowledge generation (**RP3**) are addressed by novel VA methods (C2) that are informed by the conceptual contributions (C1). The first methodological contribution is a novel method for **Dynamic and Adaptive Visual Abstraction of Soccer Movement** data (geo-spatial). The analyst is enabled to dynamically navigate within a space of interlinked parameterizations of abstraction techniques and the visualization smoothly animates between the different abstraction levels. This allows the analysts to explore and understand the underlying computations and to assess the obtained results (**RP3-3**). A recommender system that learns configurations from explicit user feedback and facilitates the exploration process is added (**RP3-1**). Another methodological contribution is a generalized **Note Taking Interface and Analytic Provenance Component** that can be plugged to any existing VA system. The system implements a novel methodology to analyze higher and lower-level analytic behavior and trust building during the analysis process (**RP3-2**). It supports knowledge management, such as evidence gathering and hypothesis refinement, integrated with analytic provenance capabilities within the actual VA system and the note taking interface. It is integrated and evaluated with the soccer movement abstraction system to analyze human exploration and verification processes. Another novel visual interactive method for exploratory time series clustering with **Self-Organizing-Maps, Automatic Guidance and Analytic Provenance (SOMFlow)** is applied to the domain of linguistic intonation research. The system has been developed in close collaboration with domain experts and was iteratively improved and generalized to be used in other domains (e.g., for stock market analysis). The system tightly integrates visualization and interactive ML (**RP3-3**) but also includes integrated

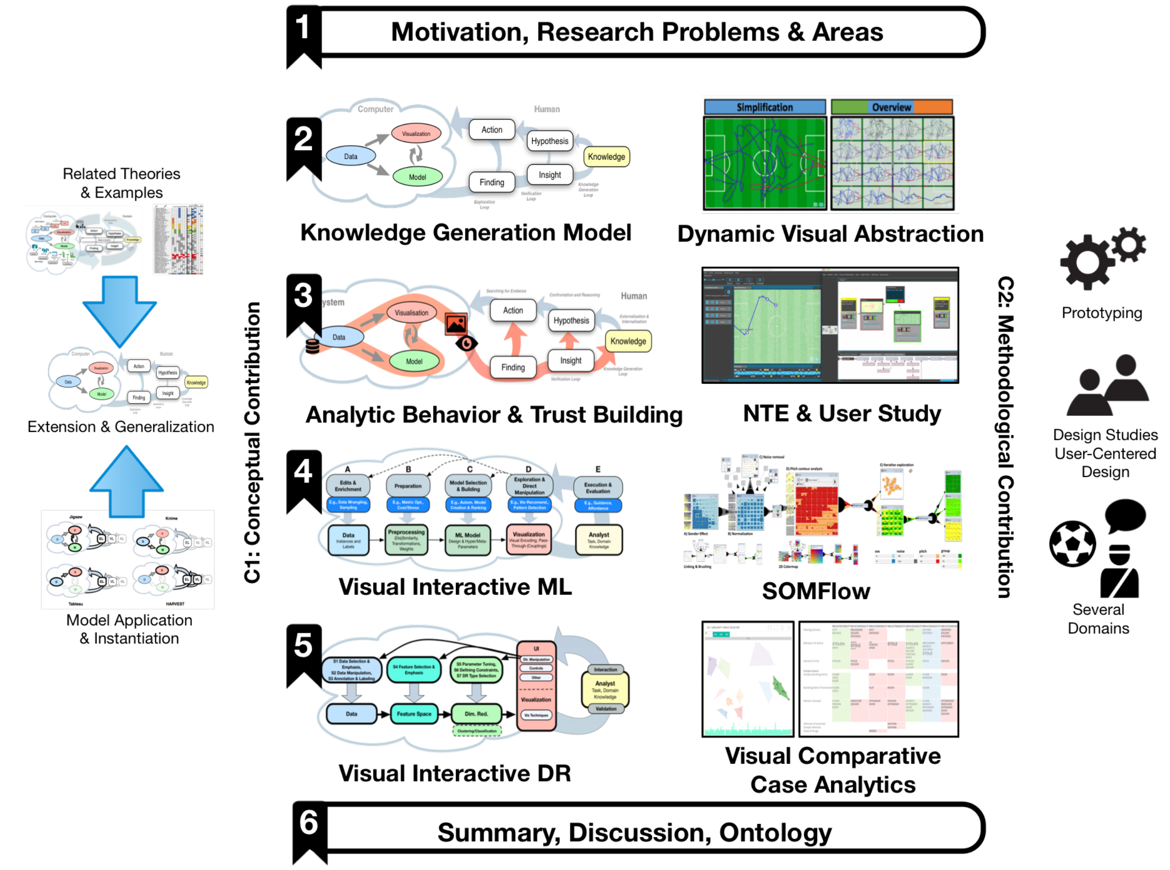



Figure 1.3: Thesis overview & research methodology. The thesis is organized with a conceptual (left) and methodological (right) stream wtihin the core chapters (2–5). These core chapters are bracketed by an introduction and summary.

note taking and analytic provenance capabilities to support knowledge generation (**RP3-2**). The system leverages quality and interestingness measures to guide the analyst (**RP3-1**) during knowledge generation. Finally, a system that applies different **Visual Interactive Dimensionality Reduction** techniques to the domain of crime intelligence analysis is described. It allows crime investigators to develop and evaluate crime clusters interactively while enabling them to test cluster robustness across algorithms and parameterizations (**RP3-3**). The system supports crime investigators, such as police officers, in conducting comparative case analysis (i.e., identifying similar crimes) based on high-dimensional features of the data. It also implements the capturing and visualization of user interactions (**RP3-2**).

## 1.4 Thesis Structure

The structure of this dissertation is illustrated in Figure 1.3. It is organized into six chapters, where the first chapter describes the research context and structure of this dissertation, followed by four main chapters and a final summary with conclusions. Each of the main chapters (Chapter 2–5) contains two sub chapters: the first part presents a conceptual contribution (conceptual process model) leading to a methodological contribution which is described in the second part. Figure 1.3 further relates the used research methodologies to these two parallel streams of conceptual (left) and methodological contributions (right).

**Chapter 1** motivates the topic and provides the research context, such as the problem areas, the thesis methodology and contributions. It also provides an overview of the thesis structure, clarifies the used terminology and contains a contribution clarification for reused published parts of this dissertation.

**Chapter 2** establishes the Knowledge Generation Model for Visual Analytics that emerged from a deeper analysis of related theories that focus on human-computer interaction in VA. The process model is applied to existing data analysis systems form different domains and it highlights major research problems that are subject to the following chapters. The second part of this chapter describes a novel dynamic visual abstraction technique for soccer movement data. It tightly integrates automatic simplification and abstraction techniques with interactive visualization enabling the exploration and understanding of movement patterns. We further add a recommender system to automatically configure the degree of abstraction to support the exploration process.

**Chapter 3** considers human higher-level processes within the knowledge generation model in more detail in relation to system components and analysis problems that are caused by computational uncertainty and complexity. An *uncertainty – awareness – trust* classification is introduced and guidelines for mitigating miss-calibrations and to support human knowledge generation processes are proposed. The second part of this chapter presents a note taking and analytic provenance component, which enables knowledge management and interaction capturing. We integrate this analytic provenance component with the soccer analysis system to capture and investigate human interaction and trust building processes in a quantitative user experiment.

**Chapter 4** focuses on human-centered machine learning by interactive visualization and characterizes specific analysis scenarios with domain experts in the visual interactive machine learning loop. The proposed conceptual process model is also applied to existing example systems and it is further used to describe future research opportunities. The second part describes the SOMFlow system as a concrete example that tightly integrates machine learning with visual interaction. The system further guides the analyst to perform iterative data partitioning tasks and it allows the analyst to reflect the analysis within an analytic provenance graph.

**Chapter 5** specializes the previous chapter for a concrete ML problem of dimensionality reduction with a structured literature analysis. The chapter addresses the core question of "exactly how analysts do interact with current dimensionality reduction techniques" and reveal seven common scenarios for interactive dimensionality reduction. The resulting process model is used to evaluate and compare existing examples and to generate future research areas. The second part of this chapter describes a novel visual comparative case analytics system that applies different dimensionality reduction techniques to criminal intelligence analysis. It enables crime analysts to develop alternative clusterings and to evaluate cluster robustness across different dimensionality techniques and parameterizations.

**Chapter 6** provides a summarization of the described conceptual and methodological contributions with a discussion of implications and open research opportunities in a broader research context. It also embeds the conceptual process models and example systems into a higher-level ontological framework.

## 1.5 Terminology

The term "model" is used in different contexts. It is generally defined as a "*standard or example for imitation or comparison*" or '*'a representation, generally in miniature, to show the construction or appearance of something.*"[1] However in a scientific context the term "model" is described as

*"[...] a representation of an idea, an object or even a process or a system that is used to describe and explain phenomena that cannot be experienced directly. Models are central to what scientists do, both in their research as well as when communicating their explanations."*[2]

The development, refinement and investigation of such scientific models is further describes as

*"[...] the generation of a physical, conceptual, or mathematical representation of a real phenomenon that is difficult to observe directly. Scientific models are used to explain and predict the behaviour of real objects or systems and are used in a variety of scientific disciplines, ranging from physics and chemistry to ecology and the Earth sciences. Although modeling is a central component of modern science, scientific models at best are approximations of the objects and systems that they represent—they are not exact replicas. Thus, scientists constantly are working to improve and refine models."*[3]

However, the scientific model term can still be used in different context within the visual analytics literature: For example, a *conceptual process model* can describe an aspect of theory evolution [57] (see Figure 1.2), a *machine learning model* can be used to predict class labels [293], while a *cognition model* describes human thought processes [109]. Therefore, this dissertation defines and uses the following terminology:

**Conceptual Process Models** describe visual analytics processes during knowledge generation. They include human (e.g., performing an interaction, reasoning, or evaluation) and computational (e.g., data transformation, visual mappings, or machine learning) aspects. Such a conceptual process model in visual analytics embeds machine learning and human thinking models.

**Machine Learning Models** describe technical artifacts that are generated during machine learning processes. They can be applied to unseen data to predict numeric values or class labels or can be used to analyze and reveal data structures (therefore, they can also be described as *data models*). A machine learning model is therefore part of a conceptual process model in visual analytics as a model for complex computations.

**Human Thinking Models** describe complex human analytic activities, such as perception, cognition, reasoning, sensemaking, biases, and mental models. Such human thinking models are therefore part of a conceptual models of visual analytics processes.

[1] `http://www.dictionary.com/browse/model`, accessed 12.07.17

[2] `https://www.sciencelearn.org.nz/resources/575-scientific-modelling`, accessed 12.07.17

[3] `https://www.britannica.com/science/scientific-modeling`, accessed 12.07.17

## 1.6 Own Thesis-Relevant Publications

Parts of this thesis have been published in:

### Journal Articles

- [248] **Sacha, D.**, Stoffel, A., Stoffel, F., Kwon, B., Ellis, G., and Keim, D. Knowledge generation model for visual analytics. *IEEE Trans. on Visualization and Computer Graphics 20*, 12 (2014), 1604–1613.
  I took primary responsibility for this publication. The paper contributions emerged from many group discussions and continuous feedback of all authors. A. Stoffel supported me in writing Section 2 about the different concepts of the knowledge generation model. F. Stoffel supported me to write parts on applying the knowledge generation model to existing VA systems (Section 4). B. C. Kwon helped me to write the introduction (Section 1) and discussion (Section 5) while G. Ellis helped me to improve the entire text. D. Keim commented on paper drafts. All parts of the paper were revised several times by me. Thus, in this thesis I reuse the paper text without citation marks in Chapter 2.1.

- [246] **Sacha, D.**, Senaratne, H., Kwon, B. C., Ellis, G., and Keim, D. The role of uncertainty, awareness, and trust in visual analytics. *IEEE Trans. on Visualization and Computer Graphics 22*, 1 (2016), 240–249.
  This paper is a follow up our workshop paper on uncertainty propagation and trust building in visual analytics [247]. I was leading this research project and responsible for most of the sections. H. Senaratne took responsibility for the parts that focus on uncertainty propagation within the knowledge generation model (Sections 2.1, 3.1., 4.1). G. Ellis wrote the section on awareness, including the awareness classification. All other parts of the paper were revised several times by me. B. C. Kwon and G. Ellis provided feedback, guided the project, and helped me to improve the text. D. Keim commented on paper drafts. In this thesis, I reuse the sections that have been written under my responsibility without citation marks in Chapter 3.1 and I shortened and rewrote Sections 3.1.2, 3.1.3, 3.1.4 to keep the context.

- [244] **Sacha, D.**, Sedlmair, M., Zhang, L., Lee, J. A., Peltonen, J., Weiskopf, D., North, S. C., and Keim, D. A. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing* (2017).
  This article provides significant extension to a previous conference paper [245] on human-centered machine learning. We revised major parts of the previous paper and added more information about related work, research methodology, example systems, human aspects, and research opportunities. I was leading the project, writing major parts of the paper by myself and incorporating valuable feedback of all co-authors. All authors were involved in discussing the most interesting extensions of the proposed conceptual process model. J. Lee and J. Peltonen helped me to adapt the paper for the ML audience, while S. C. North helped me to improve the text. The major parts of the paper were written by myself and revised several times by me. Thus, I reuse the paper without citation marks in Chapter 4.1.

- [249] **Sacha, D.**, Zhang, L., Sedlmair, M., Lee, J. A., Peltonen, J., Weiskopf, D., North, S. C., and Keim, D. A. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 241–250.
  This publication was the outcome of an interdisciplinary collaboration of visualization and ML researchers with focus on visual interactive dimensionality reduction. The main contributions emerged form continuous discussions of all authors. I was leading the project in organizing the meetings and discussions, providing summarizations of results and in structuring the work. I did

the implementations for the literature analysis and prepared the paper coding workflow. For the paper coding and validation I was supported by L. Zhang, J. Lee, M. Sedlmair and J. Peltonen. They also supported me to shape the paper and in discussing details. D. Weiskopf, S. C. North, and D. A. Keim provided feedback to intermediate results and guided the project. S. C. North further helped me to improve the text. The major parts of the paper were written by myself and revised several times by me. Thus, I reuse paper text without citation marks in Chapter 5.1.

- [238] **Sacha, D.**, Al-Masoudi, F., Stein, M., Schreck, T., Keim, D. A., Andrienko, G., and Janetzko, H. Dynamic Visual Abstraction of Soccer Movement. *Computer Graphics Forum* (2017) **Honorable Mention Award**
  This research has been initiated in the context of our soccer analysis team within our research group at the University of Konstanz with previous related publications (e.g., [240]) and the basic soccer analytics system [135] where H. Janetzko identified the need for abstracting soccer trajectories within arbitrary, user-selectable time spans. This paper is an outcome of a close collaboration between F. Al-Masoudi and myself (I supervised his Master project and thesis). F. Al-Masoudi was responsible for the major implementation efforts and I was responsible for leading the project, the major ideas, and writing the paper. D. A. Keim and H. Janetzko guided the project. M. Stein provided feedback during the project. T. Schreck and G. Andrienko were involved in a later stage to focus the project on the most interesting aspects and as devils advocates. H. Janetzko supported me in writing Section 3. M. Stein and T. Schreck were helping me with the related work and evaluation sections. All authors commented on paper drafts and helped to improve the text. I wrote the major parts of the text and revised all the sections several times. Thus, I reuse the text without citation marks in Chapter 2.2.

- [243] **Sacha, D.**, Kraus, M., Bernard, J., Behrisch, M., Schreck, T., Asano, Y., and Keim, D. A. SOMFlow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. *IEEE Trans. on Visualization and Computer Graphics* (2017)
  This paper is a result of a close collaboration between M. Kraus and myself (I supervised his Master project and thesis). I lead the project and developed the major ideas during our previous collaborations on linguistic intonation research with Y. Asano [14, 239] where D. Keim, M Behrisch, and T. Schreck have been involved to provide continuous feedback. With their experience in Self-Organizing Maps, J. Bernard and T. Schreck contributed significantly in discussing and fine-tuning details regarding the algorithm, calculating quality and interestingness measures, or regarding guidance during the analytic process. M. Behrisch and D. Keim guided the project and commented on paper drafts. I wrote the major parts of the text and revised all the sections several times. Thus, I reuse the text without citation marks in Chapter 4.2.

**Book Chapters**

- [307] Wagner, M., **Sacha, D.**, Rind, A., Fischer, F., Luh, R., Schrittwieser, S., Keim, D. A., and Aigner, W. Visual analytics: Foundations and experiences in malware analysis. In *Empirical Research for Software Security: Foundations and Experience*, L. ben Othmane, M. Gilje Jaatun, and E. Weippl, Eds. Taylor & Francis Group, LLC, 2017.
  M. Wagner and A. Rind took the major responsibilities of the project and wrote most of the text. I contributed with a general introduction of the knowledge generation model and a comparative evaluation of the investigated malware analysis systems. Therefore, I reuse this text without citation marks in Section 2.1.4.

**Conference Articles**

- [245] **Sacha, D.**, Sedlmair, M., Zhang, L., Lee, J. A., Weiskopf, D., North, S. C., and Keim, D. A. Human-Centered Machine Learning Through Interactive Visualization: Review and Open

Challenges. In *Proceedings of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN* (2016).
This research initiative between visualization and machine learning researchers was founded at the Dagstuhl Seminar 2015 with an initially published workshop report [152]. I took over the lead of this research collaboration from M. Sedlmair. For this conference paper, we re-discussed, revised, and tailored all essential parts to the ML audience and we focused the resulting conceptual process model on the different types of interactice feedback along the machine learning pipeline. The major contributions emerged from continuous group discussion involving all authors and external feedback. M. Sedlmair, L. Zhang, and J. A. Lee supported me in writing the text and in discussing the details. D. Weiskopf, S. C. North, and D. A. Keim provided continuous feedback and commented on paper drafts. S. C. North further helped me to improve the text. I revised all sections of the paper several times. Thus, I reuse the text without citation marks in Chapter 4.1.

- [240] **Sacha, D.**, Boesecke, I., Fuchs, J., and Keim, D. A. Analytic Behavior and Trust Building in Visual Analytics. *In EuroVis 2016 - Short Papers* (2016) E. Bertini, N. Elmqvist, and T. Wischgoll, Eds., The Eurographics Association.
The paper is an outcome of a close collaboration between I. Boesecke and myself to analyze analytic behavior and trust building in VA systems (I supervised her Bachelor project and thesis). I. Boesecke implemented the note taking interface and conducted the user experiment under my supervision. I initiated and lead the project. D. A. Keim observed and guided the project while J. Fuchs helped in designing and reporting the results of the user experiment. I wrote the major parts of the entire paper and revised all the sections several times. Thus, I reuse the text without citation marks in Chapter 3.2.

- [135] Janetzko, H., **Sacha, D.**, Stein, M., Schreck, T., Keim, D. A., and Deussen, O. Feature-driven visual analytics of soccer data. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2014), pp. 13–22.
This paper was the first publication within our soccer analysis research group. I was deeply involved in the implementation of the feature-driven clustering and visualization approach. For the paper, I was setting up the basic structure and mainly responsible for the content and use cases of the single and multi-player analysis, except the classification-based machine learning and the back-four formation parts. H. Janetzko took over the paper lead and rewrote most of the sections. Therefore, I don't directly reuse the paper content and phrases. However, the paper significantly contributed to my analysis experience in this domain (and VA in general) and the system framework was the basis for subsequent paper projects that are part of this thesis.

- [239] **Sacha, D.**, Asano, Y., Rohrdantz, C., Hamborg, F., Keim, D. A., Braun, B., and Butt, M. Self organizing maps for the visual analysis of pitch contours. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA* (2015), pp. 181–189.
This paper is an intermediate result of an ongoing collaboration with the linguistic department at the University of Konstanz. The collaborators provided the data and continuous expert feedback. We implemented a visual interactive system for analyzing time series (recorded spoken utterances of linguistic experiments) with Self-Organizing Maps that was a foundation for a subsequent contribution to visual analytics research [243]. I do not directly reuse the text of this paper, however, it was an important intermediate result that lead to a novel method that tightly integrates machine learning with interactive visualizations.

- [14] Asano, Y., Gubian, M., and **Sacha, D.** Cutting down on manual pitch contour annotation using data modeling. In Proceedings of the 8th International Conference on Speech Prosody (2016).

Y. Asano was mainly responsible for the paper and wrote major parts. I implemented the visual analysis system and provided the respective technical information. I do not reuse text in this dissertation, however, this study motivates and evaluates our semi-automatic approach to annotate captured pitch contours (utterances) using our visual interactive Self-Organizing Map system.

**Workshop Articles & Participation**

- [241] **Sacha, D.**, Jentner, W., Zhang, L., Stoffel, F., and Ellis, G. Visual Comparative Case Analytics. In EuroVis Workshop on Visual Analytics (EuroVA) (2017), M. Sedlmair and C. Tominski, Eds., The Eurographics Association.
  The paper is a result of our joint work within the EU project VALCRI and an intermediate version of the prototype is also described in a technical white paper [242]. F. Stoffel, W. Jentner and myself implemented the respective components (data processing - F. Stoffel, visual interactive dimensionality reduction - myself, crime table & weight observer component - W. Jentner). I took the main responsibility for this paper project and wrote most of the sections on my own. F. Stoffel, W. Jentner, L. Zhang, and G. Ellis commented on paper drafts. I revised the entire text several times and therefore reuse the text without citation marks in Chapter 5.2.
- [280] Stoffel, F., **Sacha, D.**, Ellis, G., and Keim, D. A. VAPD - A Visionary System for Uncertainty Aware Decision Making in Crime Analysis. In *Symposium on Visualization for Decision Making Under Uncertainty at IEEE VIS* (2015).
  In this workshop paper we sketched our ideas on implementing our prosed guidelines on supporting uncertainty aware decision making [246] for crime intelligence analysis. I was responsible for specific paper parts concerning human aspects and the envisioned user interface of the system. I do not reuse text of this publication, however, it is an interesting example on how our theoretic guidelines [246] can be applied to a real world setting and we realized some of these ideas (e.g., applying trust ratings to findings and interaction capturing) within our note taking and analytic provenance component [240].
- [247] **Sacha, D.**, Senaratne, H., Kwon, B. C., and Keim, D. A. Uncertainty Propagation and Trust Building in Visual Analytics. *Provenance for Sensemaking Workshop at IEEE VIS* (poster paper) (2014).
  We sketched and discussed our early ideas on uncertainty and trust building within the knowledge generation model with this workshop poster paper. I do not reuse the text of the paper, however, by participating at the workshop we could discuss our ideas with other researchers in the field and collect valuable feedback. This was an important milestone for our subsequent full paper publication [246].

**Technical Reports**

- [242] **Sacha, D.**, Jentner, W., Zhang, L., Stoffel, F., Ellis, G., and Keim, D. A. Applying Visual Interactive Dimensionality Reduction to Criminal Intelligence Analysis. Tech. rep., Universität Konstanz, Middlesex University, 2017. VALCRI White Paper Series.
  I was responsible for writing the technical progress report about our implementations of interactive dimensionality reduction techniques in the EU project VALCRI. F. Stoffel, W. Jentner and myself implemented the respective components (data processing - F. Stoffel, dimensionality reduction - myself, crime table - W. Jentner). W. Jentner was responsible for the table related sections and L. Zhang supported me in writing the part about dimensionality reduction foundations. F. Stoffel, G. Ellis, and D. Keim commented on paper drafts. I do not reuse the text of this report but it represents a preliminary version of our subsequent research paper [241].

# 2

# Building and Applying a Knowledge Generation Model for Visual Analytics

*"We are drowning in information but starved for knowledge."* – John Naisbitt

## Contents



Knowledge is the ultimate goal of data analysis processes. Data is processed by the computer and presented to the analyst by interactive visual user interfaces. The first part of this chapter introduces the knowledge generation model that is derived from related theories and applied to existing data analysis systems. This part is published within our knowledge generation paper [248]. In [307] we have further used the model to compare and evaluate state of the art malware analysis systems. This part has been added to the model application Section (2.1.4). The knowledge generation model allows us to derive open research challenges that aim to support human knowledge generation loops and we discuss future research. The second part of this chapter describes a dynamic and interactive approach for visual abstraction of soccer movement that is inspired by the knowledge generation model and aims to support exploration processes for different abstraction techniques by interactive navigation through the space of possible abstractions and smooth transitions between intermediate results. We further add a recommender system to automatically pre-configure the desired level of abstraction (LoA). This second part is based on our design study paper [238], and we added an instantiation of the knowledge generation model to the conclusion.

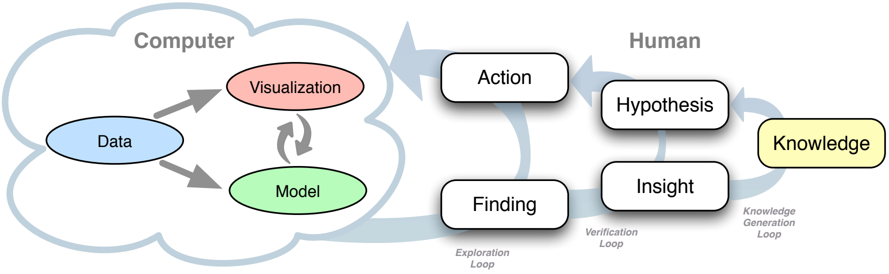



Figure 2.1: Knowledge generation model for visual analytics: The model consists of computer and human parts. The left hand side illustrates a visual analytics system, whereas the right hand side illustrates the knowledge generation process of the human. The latter is a reasoning process composed of exploration, verification, and knowledge generation loops. Visual analytics pursues a tight integration of human and machine by enabling the user to interact with the system. These interactions are illustrated in Figure 2.2.

## 2.1 Knowledge Generation Model for Visual Analytics

Visual analytics enables us to analyze huge information spaces in order to support complex decision making and data exploration. Humans play a central role in generating knowledge from the snippets of evidence emerging from visual data analysis. Although prior research provides frameworks that generalize this process, their scope is often narrowly focused so they do not encompass different perspectives at different levels. This section proposes a knowledge generation model for visual analytics that ties together these diverse frameworks, yet retains previously developed models (e.g., KDD process) to describe individual segments of the overall visual analytic processes. To test its utility, a real world visual analytics system is compared against the model, demonstrating that the knowledge generation process model provides a useful guideline when developing and evaluating such systems. The model is used to effectively compare different data analysis systems. Furthermore, the model provides a common language and description of visual analytic processes, which can be used for communication between researchers. In the end, our model reflects areas of research that future researchers can embark on.

### 2.1.1 Introduction

Visual analytics research made great strides in the past decade with numerous studies demonstrating successes in helping domain experts explore large and complex data sets. The power of visual analytics comes from effective delegation of perceptive skills, cognitive reasoning and domain knowledge on the human side and computing and data storage capability on the machine side, and their effective coupling via visual representations. Thus far, many application papers have tested and verified different ways to instigate this human and machine collaboration to reveal hidden nuggets of information.

Recent work emphasizes that visual analytics theories must go beyond “human in the loop” to “human is the loop” thinking in order to recognize and integrate human work processes with analytics (Endert et al. [87]). To achieve this goal, system, human, cognition and reasoning based theories have to be considered. Many theoretical works have also examined the role of visual analytics tools in data analysis, decision making and problem solving. Visual analytics processes span from human’s high-level analytic works using their domain knowledge to low-level activities such as interacting with tools. Many prior works investigated different levels of activities with regards to human’s cognition models (e.g., Green et al. [109], Kwon et al. [164]); interaction models/taxonomies (e.g., Brehmer and Munzner [38], Norman [207]); process models (e.g., Card et al. [48], Fayyad et al. [91]); sensemaking models (e.g., Pirolli and Card [221]). We initially sought interaction taxonomies that describe the aforementioned models. However, we lack an overarching pipeline that connects all the dots. Previously developed models (e.g., Keim et al. [150, 151]) are system-driven and are not describing in detail the human reasoning part in the visual analytics process. In particular, we have little idea how analytical components support knowledge generation processes and how analysts' intents drive the insight collection action forward. It would be valuable for future research to have an integrated framework of all processes and models relevant for knowledge generation with visual analytics.

This work aims to take the first step to establish the knowledge generation model for visual analytics. First, we build our initial process model by significantly extending the previous models [150, 151] (Section 2.1.2). Comparing with previously developing models and theories, we show how our model fits various kinds of models (Section 2.1.3). Using our model, we find areas that existing visual analytics tools can improve (Section 2.1.4). We discuss our model showing open issues, implications and future work (Section 2.1.5). Our contributions can be described as follows. The conceptual process model presented in this chapter provides a high-level description of the human and computer processes within visual analytics systems which facilitates an understanding of the functionality and interaction between the components. This can be used in the design of new visualization applications or in the evaluation of existing ones in terms of how their sub components support human's reasoning, decision making, and knowledge generation processes.

### 2.1.2 Knowledge Generation Model

Visual analytics uses data to draw conclusions about the world, a process, or an application field. It is a structured reasoning process that allows analysts to find evidence in data and gain insights into the problem domain. Our model of the knowledge generation process is based on the visual analytics process of Keim et al. [150, 151] and describes how knowledge is generated with this process. Pohl et al. [222] also discussed relevant theories that should be considered, besides a computer/system based view of the analytical process. They show that visual analytics system components, analytical procedures and human perceptual and cognitive activities and especially the interplay between these elements, lead to a complex process. The theories, namely sensemaking, Gestalt theories (describing problem solving), distributed cognition (interplay between humans and artifacts), graph comprehension theories (derive meaning from graphs/process visual information) and skill-rule-knowledge models (three-fold hierarchy of processing levels) are highlighted as important aspects of the visual analytics process. In the following sections, we define and relate common elements of the aforementioned models/fields/concepts in order to arrive at a better understanding of visual analytics in terms of computing and human processes.

Our visual analytics model (see Figure 2.1) is split into two parts. The computer system with data, visualization and analytic models, and the human component modeling the cognitive processes associated with an analytical session. The cloud in the model indicates that there is no clear separation between the computer and human part, as both parts are required for data analysis. Computers miss the creativity of human analysis that allows them to create surprising, often subtle or hidden connections between data and the problem domain. Humans are not able to deal efficiently and effectively with a large amount of data. In visual analytics, the connection between the human and computer uses the human‘s interaction abilities and perception.

Knowledge generation in visual analytics comprises of abductive, deductive, and inductive reasoning processes. For a detailed definition and discussion of these reasoning processes see Peirce [215] and Magnani [188]. Abductive reasoning processes formulate hypotheses from observations that are unexpected or cannot be explained with existing knowledge. Assuming that these hypotheses are true, expectations of effects, patterns, or relations in the analyzed data are deduced. Through interactions with visual analytics systems, analysts try to find evidence and detect patterns in data to verify or falsify the hypotheses, which is an inductive reasoning process. We decided to model human cognitive processes with loops because analysis does not follow deterministic rules but is rather chaotic or spontaneous nature. Analysts are often working on different hypotheses, tasks, or findings and can consequently be working on several loops in parallel.

### Computer

**Data** The starting point of all visual analytics systems is data. Data describes facts [91] in structured, semi-, or unstructured manner. It must be representative and related to the analytical problem, otherwise, the analytical process is unlikely to reveal meaningful relationships in the problem domain. In the visual analytics process, data creation, gathering, and selection processes often determine the quality of the data. During analysis, additional data can be created by automatic methods (e.g., clustering or classification) or by manual annotations. Hence, provenance information about data containing details about creation, gathering, selection, and preprocessing is important to estimate the trustworthiness of analysis results (see also [234]). The term metadata describes second order data or "data about data". The usage of this term is ambiguous depending on the domain and interests of users. In addition to provenance data, metadata describing the structure of data is usually handled by visual analytics system to access and display data appropriately. This sort of metadata is usually not subject to analysis as it describes mainly data formats, which is a necessity of any kind of data. The term metadata is also often used for data describing or summarizing other data, e.g., keywords or topics for documents or images. In the scope of data analysis, this descriptive metadata is treated similar to data, and we see no benefit or reason to treat this data differently. Therefore the term data also includes this sort of metadata.

**Model** Computational models can be as simple as descriptive statistics describing a property of a subset of the data or as complex as a data mining or machine learning algorithm. The KDD process leads to models from data (see Figure 2.3). These models serve different purposes in the visual analytics process. Simple analysis tasks might be solved by calculating a single number that confirms or rejects a preconceived notion/expectation. For instance, a statistical test can lead to a conclusion whether or not to trust a hypothesis. Complex patterns or abstractions found by data mining methods can be used in visualizations by showing, for instance, clustering or classification results visually. In addition, the automatically created model can be analyzed or visualized to derive insights. For example, logistic regression models learn weights of features, which can be used to identify most important features or feature combinations.

**Visualization** A different path from data to knowledge is the information visualization pipeline using data visualizations (see Figure 2.3). Visualizations use data or models generated from the data and enable analysts to detect relationships in the data. In visual analytics, visualizations are often based on automatic models, for instance, clustering models are used to group data visually. Also, a model itself can be visualized, for instance, a box plot shows the data distribution of a dimension. Visualization methods for a model may vary depending on the state of the visualization. For example, in semantic zooming, a visualization might use different properties of a model depending on the zoom level. Visualization is often used as the primary interface between analysts and visual analytics systems whereas understanding the model often requires more cognitive efforts.

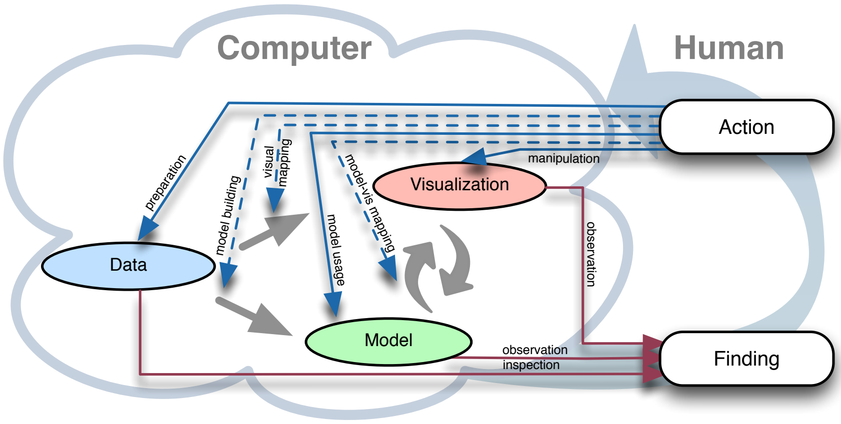



Figure 2.2: Detailed part of the process model including action and cognition paths. Actions can either lead directly to visual analytic components (blue arrows) or to their mappings (blue, dashed arrows). Humans can observe reactions of the system (red arrows) in order to generate findings.

### Exploration Loop

The exploration loop describes how analysts interact with a visual analytics system in order to generate new visualizations or models and analyze the data. Analysts explore data supported with the visual analytics system by interactions and observing the feedback. Actions taken in the exploration loops are dependent on findings or a concrete analytical goal. In case a concrete analysis goal is missing, the exploration loop becomes a search for findings, which may lead to new analytical goals. Even when the exploration loop is controlled by an analysis goal, the resulting findings are not necessarily related to it but can lead to insights solving different tasks or opening new analytical directions.

**Action** Different meanings of actions are present in the InfoVis community as they may be defined with different granularities or with differing relations. Pike et al. [220] illustrate that actions may concern user goals and tasks on the one hand and interactive visualizations on the other hand. According to recent interaction taxonomies (e.g., Brehmer and Munzner [38]), simple interactions, *How*, are related to higher-level concepts, *Why*. In our definition, actions refer to individual tasks that generate tangible, unique responses from the visual analytics system. For instance, a task can be to visualize a particular data property or calculate a model of a relationship in the data. Actions derived from hypotheses are usually complex actions, for instance, use a specific visualization method that has the potential to show interesting data. Actions derived from findings are normally simple actions, such as changing the visual mapping of a visualization or selecting a different parameter for model building. In visual analytics, analysts freely choose between visualization and modeling or a combination of both for their actions. Actions naturally provoke interactions of analysts with visual analytics systems.

We name actions dealing with data gathering or data selection as *preparation* action because these actions are used to prepare data for the visual analytics process (Figure 2.2). Actions taken to create models are summarized as *model building* actions which in turn are related to the KDD process and its configuration. Application of a model is termed as *model usage*, which refers to actions such as calculating a statistic or cluster data. *Visual mapping* actions are used to create data visualizations, and *model-vis mapping* actions are those which map models into visualizations. *Manipulation* of a visualization changes the viewport or highlights interesting data in the visualization without changing the mapping of data to the visualization. All these actions are observable as interactions between analysts and a visual analytics system.

**Finding** A finding is an interesting observation made by an analyst using the visual analytics system. The finding leads to further interaction with the system or to new insights. For example, findings from data inspection can be missing values or other data properties affecting the further analysis and require special data processing. In the case of visualizations or models, a finding can be a pattern, a conspicuous model result, or an unusual behavior of the system. Bertini et al. state that “a pattern is made of recurring events or objects that repeat in a predictable manner. The most basic patterns are based on repetition and periodicity.” [33, p. 13] Patterns in data can be detected with automatic analytical methods or humans may detect patterns using their visual perception and cognition skills. Findings are in principle not limited to data, visualizations, or models but comprise of anything interesting to an analyst, e.g., an unexpected high number or a word or statement in some text.

In our definition, a finding is independent of the problem domain, however, to make an analytical use of a finding, the analyst has to interpret them in the context of the problem domain. Although findings are usually associated with detecting a visual pattern, the lack of a pattern, when expected by an analyst, is also considered a finding. As shown in Figure 2.1, findings do not necessarily lead to new insights (see Section 2.1.2) but may trigger basic actions, such as manipulating the viewport of a visualization to show a region in more detail.

Analysts come across findings by observing the feedback from the system or examining visualizations and models, which in turn can lead to further actions. The exploration loop can be characterized as a searching activity by using the system to reveal useful findings to solve an analysis task. Analysts might frequently change their exploring strategies and switch between models and visualizations to collect different findings. The strategies and analysis directions in the exploration loop are guided by an exploration goal defined in the verification loop. The actions and findings in the exploration loop are closely related to the visual analytics systems. Analysts gain a new insight when they are able to understand the findings and are able to interpret them in the context of the problem domain.

### Verification Loop

The verification loop guides the exploration loop to confirm hypotheses or form new ones. To verify a concrete hypothesis, a confirmatory analysis is conducted, and the exploration loop is steered to reveal findings that verify or falsify the hypothesis. Analysts gain insights when they can interpret findings from the exploration loop in the context of the problem domain. Insights may lead to new hypotheses that require further investigation. Analysts gain additional knowledge when they assess one or more trustworthy insights.

Findings during exploration may contribute to the verification of a concrete hypothesis, but insights emerging from exploration may not be related to the examined hypothesis. It is often the case that new insights solve different analysis questions or open up new ones.

**Hypothesis** Hypotheses play a central role in the visual analytics process. A hypothesis formulates an assumption about the problem domain that is subject to analysis. Analysts try to find evidence that supports or contradicts hypotheses in order to gain knowledge from data. In this respect, the visual analytics process is guided by hypotheses. Concrete hypotheses can be tested with statistical tests or data visualizations when the data contains the necessary information. Unfortunately, hypotheses are often vague, such as the assumption that there are unknown factors that influence the problem domain. In such cases an exploratory analysis strategy allows analysts to come up with more concrete hypotheses that are used for further analysis.

**Insight** In the InfoVis community, insight has a variety of definitions. Saraiya et al. define insight as “an individual observation about the data by the participant, a unit of discovery” [250, p. 2]. North [208] takes another view and lists some important characteristics of an insight such as being complex, deep, quantitative unexpected and relevant, and going beyond an individual observation of the data. Yi et al. [319] go one step further and also consider the processes that involve insights. They focus on how people gain insight in information visualization and identify four types of insight-gaining processes, however, they argue that there is no common definition of insight. Chang et al. [53] suggest two different definitions of insight: the cognitive science insight as a moment of enlightenment, an "Ah Ha" moment, which can occur spontaneously, and an advance in knowledge or a piece of information. Our definition is closer to the latter, where the analyst interprets a finding, often with previous domain knowledge, to generate a unit of information. Hence, an insight can be quite small, such as realizing that there is a relation between several properties of the data, to something more important and potentially significant. So, insights are different from findings in the sense that insights have an interpretation in the problem domain, what we not required for findings. For instance, a finding might support a hypothesis, which may convince the analyst and lead to the insight that the hypothesis is reliable. An analyst gains insights by collecting enough evidence to create a new hypothesis about the application or, in the case of very strong evidence, even new trusted knowledge. We consider an insight not directly as knowledge because weak evidence might lead to an insight that needs further verification and becomes a hypothesis. For instance, finding a cluster in a visualization during an exploratory analysis might lead to the insight that there is a cluster with different properties, but this insight should at first be considered as a new assumption or hypothesis that has to be validated.

#### Knowledge Generation Loop

Analysts form hypotheses from their knowledge about the problem domain and gain new knowledge by formulating and verifying hypotheses during the visual analytics processes. When analysts trust the collected insights, they gain new knowledge in the problem domain that may also influence the formulation of new hypotheses in the following analysis process.

**Knowledge** Data analysis usually starts with data and one or more analysis questions. In addition, analysts bring in their knowledge about the data, the problem domain, or visual analytics tools and methodology. This prior knowledge determines the analysis strategy and procedure. During the visual analytics process, analysts try to find evidence for existing assumptions or learn new knowledge about the problem domain. In general, knowledge learned in visual analytics can be defined as "justified belief" [33]. The reasoning processes in visual analytics enable analysts to gain knowledge about the problem domain from evidence found in data. The evidence has different qualities, which directly affect the trustworthiness of the concluded knowledge. The evidence collection route also impacts the trustworthiness. For example, an outcome of a statistical test of a hypothesis may be perceived more trustworthy than a pattern found in a visualization. Depending on the collected evidence, an analyst has to decide whether enough evidence was collected to trust an insight and accept it as new knowledge or whether it is in need of further examination, e.g., analysis with different data or discussions with domain experts. Assessing the trustworthiness of new knowledge requires a critical review of the overall analysis process starting from data gathering. As well as new knowledge about the problem domain, analysts gain knowledge about data, e.g., patterns in the data or its quality. They also gain experience with the visual analytics systems and methodology.

### 2.1.3 Relation to other Models

This section covers the most important related models that have influenced the presented knowledge generation model and also aims to offer a big picture of the whole knowledge generation process in visual analytics. Figure 2.3 illustrates related models and our knowledge generation model for visual analytics. We have used the same color codes of the original model [150, 151] to highlight related areas in our model. Feedback loops of other models (e.g., InfoVis pipeline) are replaced by our

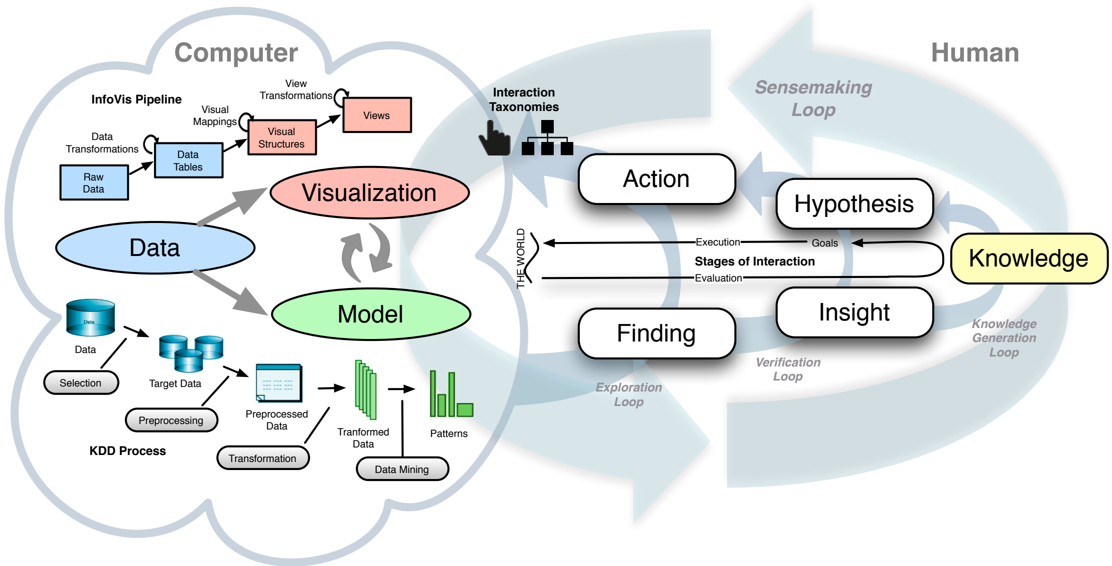



Figure 2.3: Relating the process model for knowledge generation in visual analytics to other models and theories. Similarity is illustrated by color and position. A detailed illustration of interactions between the human and computer is shown in Figure 2.2.

extensions on the human side. Each action results in a feedback loop via finding that may lead to new exploration loops or crosses over to higher-level loops. Relevant theories are grouped into three areas according to their focus on interaction, human aspects or systems.

### Systems

Card et al. propose the reference model for information visualization [48] that describes visualizations as data connected to visual mappings perceived by humans. The InfoVis-Pipeline contains the main components of *Raw Data*, *Data Tables*, *Visual Structures* and *Views* and transformations/mappings between these components which can be manipulated through *Human Interactions*.

Fayyad et al. [91] describe the Knowledge Discovery Process in Databases (KDD) as follows. KDD is the process of making sense out of data using data mining techniques at the core. The process includes the mapping of low-level data into more compact, abstract or useful forms using data mining models with the goal to discover or extract patterns that can be turned into knowledge. The KDD process consists of nine steps and represents an interactive and iterative process. Examples for interactions are selecting and filtering for the relevant data, choice of appropriate data mining methods or parameter refinement. The goal of these actions is to bring in the domain expertise of the human in order to find meaningful patterns. At each step, the user might make and add decisions that cannot be handled automatically. The KDD process steps are included in Figure 2.3.

As a basis for our knowledge generation model for visual analytics we take and extend the process model by Keim et al. [150, 151]. The main components of the model are *Data, Models, Visualization*, and *Knowledge* each representing a stage of the process. Also, a *Feedback loop* is present going from knowledge back to data. At and in between of each stage there are transitions. The visual analytics process consists of the two parts, *Visual Data Exploration* going from data via visualization to knowledge and *Automated Data Analysis* going from data via models to knowledge, as well as of their connection. Actions on the data may be performed in order to select, filter or preprocess the data. The data is then mapped to a visualization (visual mapping) and to models (data mining).

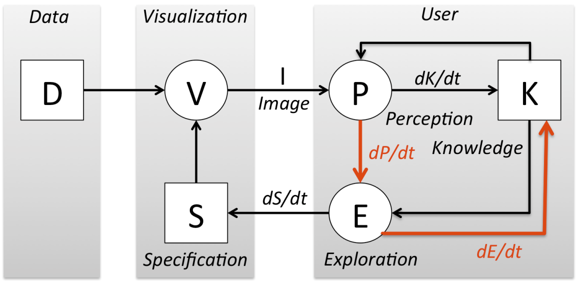



Figure 2.4: Van Wijks model including Green et al.'s changes [107, 109] and labels.

Model parameters and findings are visualized, and the user may interact with the visualization or refine parameters of the model in order to steer the analysis process. Knowledge can be derived from visualizations, automatic analysis and preceding interactions between visualizations, models and the human.

Figure 2.3 relates visual analytics components to the InfoVis and KDD system pipelines. The InfoVis pipeline corresponds to data, visualization, and the visual mappings, whereas the KDD process model is represented by data, model, and their mapping. Furthermore, the visual analytics process model includes human-computer interaction. Actions can be performed at each stage, namely data, visualization, model and their mappings.

The economic model of visualization by van Wijk describes contexts in which visualizations operate [298] (Figure 2.4). In brief, data is transferred into a visualization that can be perceived by a human through an image. Based on perception, the human generates knowledge over time which drives interactive exploration through changes to the visualization specification. Green et al. [109] add to the van Wijk model, arguing that perception has an important role in interactive exploration and that the act of exploration and associated reasoning often leads to knowledge acquisition. These relate to the exploration and knowledge generation loops of our model.

### Interaction

Norman describes a model for actions containing *Seven Stages of Action* [207]. At the beginning of each action, each human needs a goal to be achieved. Afterwards, the human has to perform an action to manipulate something. The last step is to check if the goal was achieved. These two subprocesses are called *Execution* and *Evaluation*. Based on previous actions the action cycle is traversed several times. The model also explains two major problems that occur when interacting with computer systems. The *Gulf of Execution* indicates if humans do not know how to perform an action while the *Gulf of Evaluation* indicates that humans are not able to evaluate the result of their actions. The *Goal* concept of the stages of interaction model matches to our *Hypothesis* as a starting point. The *Execution* path is leading from *Goal* via an action to the *World*. As a result, analysts evaluate observations of the *World* in several steps (see Figure 2.3).

Several interaction taxonomies exist which focus on different aspects, fields, and domains and attempt to structure different kinds of interaction at different levels of abstraction. Most of the well-known taxonomies focus on one or two fields, e.g., visualizations (Shneiderman [267]), reasoning (Gotz et al. [104]) or data processing (Bertini et al. [33]), rather than all possible interactions in visual analytics. Recent publications attempt to structure interaction using the dimensions of *Why* (the purpose of the task), *How* (methods used to achieve this) and *What* (necessary inputs/outputs) (Brehmer and Munzner [38], Schulz et al. [256]). There are also recent taxonomies that try to integrate existing taxonomies out of different fields into a sound interaction taxonomy for visual analytics (e.g., von Landesberger et al. [305]). Taking the taxonomy of Brehmer and Munzner [38] as an example, there are higher-level goals that an analyst might pursue (*why*) and lower-level operations that can be performed (*how*) on a specific target (*what*). High and low-level interactions are both covered by the action concept. However, they can be distinguished when considering the associated loops. High-level actions are inspired by the verification loop and are based on insights and hypotheses whereas low-level actions are influenced by the exploration loop that covers a sequence of simple actions and findings. Our process model implies that each action results in a feedback loop and is perceived and processed by the human. Our model description of action-types also shows the relevant interactions for visual analytics that are both visualization and model centric.

Also mentionable are two mantras characterizing the analysis process for information visualization and visual analytics because they shed light on human analysis strategies. Shneiderman proposes the *Visual Information Seeking Mantra* that summarizes the basic principles of many visual design guidelines [267]: *Overview first, zoom and filter, then details-on-demand.* Also Keim proposes a slightly different *Visual Analytics Mantra* [151]: *Analyse First - Show the Important - Zoom, Filter and Analyse Further - Details On Demand.* Both start with an overview/aggregation approach and end in a refinement of their hypothesis and analysis.

### Human Cognition, Sensemaking and Reasoning

Humans can observe visualization or model changes that can be used for the knowledge generation process. The data may also be inspected directly.

Pirolli and Card present the Sensemaking Process [221] as a description of intelligence analysis. Central terms are the *shoebox*, *schemas*, *hypotheses* and a *representation*. Sensemaking tasks are described as processes consisting of *information gathering*, the information representation as a *schema*, the generation of *insight* and finally the generation of some *knowledge product*. The first loop is called the *foraging loop* followed by the *sensemaking loop*. *Bottom-up* or *top-down* processes are possible. That indicates that the model does not have a fixed entry point which depends on the type of task. The model shows the data and process flow with many feedback loops. Our model splits the *Sensemaking Loop* into three sub-loops. In visual analytics, it is also possible that the system may learn from the analyst's actions allowing the system to support the user with visualization or action propositions. That is why the loop is leading through the system (see Figure 2.3). Hypotheses are generated as an entry point for an analysis process leading to repeated exploration cycles. The field of reasoning and decision making both depend on the construction of mental models or scenarios of relevant situations. According to Legrenzi et al. [172], the key components of decision making are *Information Seeking*, *Making Hypotheses*, *Making Inferences*, *Weighing Advantages and Disadvantages* and *Applying Criteria to make Decisions*.

The *Human Cognition Model* (HCM) proposed by Green et al. [109] can also be applied to visual analytics. A major problem in visual analytics is that human cognition is often assumed to be an over simplified black box. Information discovery and knowledge building are at the core of the HCM. Information is presented by the computer that humans can perceive and directly interact with in order to focus their attention. The process of discovery of patterns or relations is a primary stage of knowledge that can be created within the knowledge building process. The computer works to counter the human's limited working memory as well as some cognitive biases, such as confirmation bias. Central parts of the HCM are shown in Figure 2.5. The HCM also includes guidelines for discovery and knowledge building. The cognition of relevant patterns and schema that can be derived from insights (verification loop) and finally to some knowledge product (knowledge generation loop) plays a central role in the human part of our model. All elements of the HCM can also be found in the knowledge generation model. The key element *Discovery* describes the overall process, whereas the other sub-concepts all can be placed into our process model and related to our concepts. For example, the generation and analysis of hypotheses is a central part of our model, because evidence (proof/disproof) is collected as the basis for drawing conclusions.

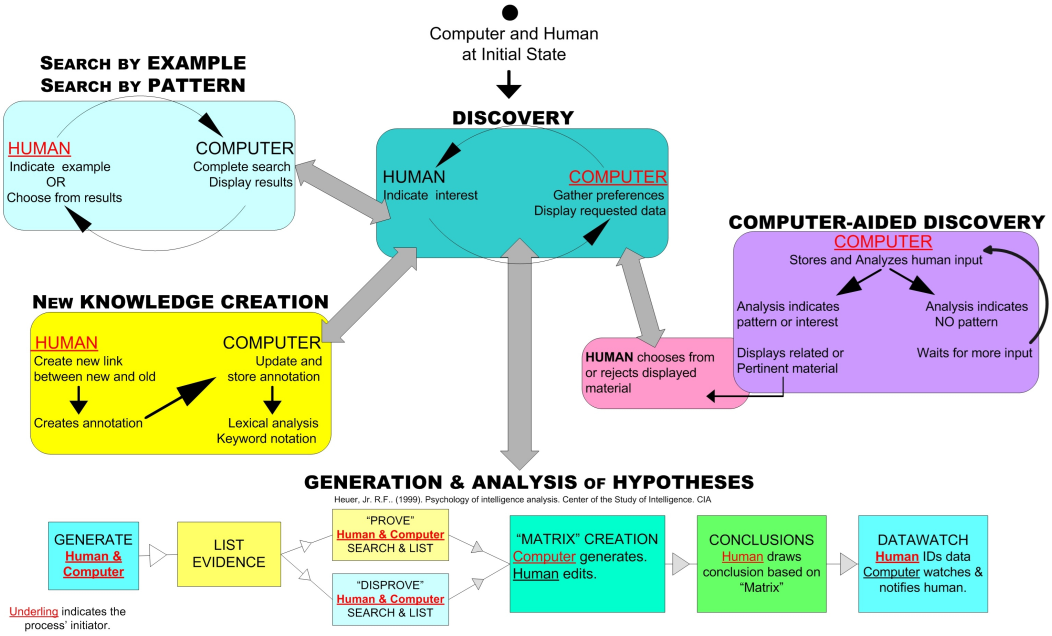



Figure 2.5: Human Cognition Model by Green et al. [107, 109].

### 2.1.4 Model Application

In this section, we illustrate that our model can be applied to systems on several levels. Firstly, the interaction possibilities can be examined according to our definition of actions in Section 2.1.2 and shown in Figure 2.2. Secondly, we can check if and how a system supports each of the individual loops. Thirdly, we can use our model for a comparative assessment. In the following, we will demonstrate a detailed model application with Jigsaw [101, 277] and a comparative high-level assessment of systems from different application domains. Finally, we apply the model to existing state of the art systems in the domain of malware analysis (cyber security) to reveal areas for improvement.

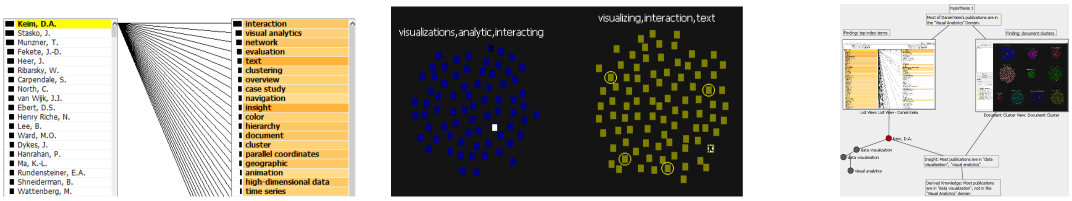
#### Actions in Jigsaw

In the following section, we investigate Jigsaw in terms of supported actions. It is important to note that in our model each of these actions leads to its own feedback loop. Therefore, we pay special attention to system components that are capable of accepting human inputs beyond fully automated implementations. We also highlight some areas of the system that might accept more user input for amplifying user interactions.

*Data preparation* is not fully supported. While loading the data, the user has no possibility of adjusting any of the data preprocessing or transformation steps. It is also not possible to change those once the data has been parsed and loaded. Users may find missing data, but such manipulation must be done outside Jigsaw.

*Model building* is done automatically, but some views provide possibilities of adjusting the underlying models. For example, the Document Cluster View allows users to select documents as a cluster seed, which can be used when the document clusters are computed. After that, the user-generated cluster seeds become part of the model used by the application.

Regarding the *visual mapping*, Jigsaw offers some basic functionalities. In some views, users can




(a) (b) (c)

Figure 2.6: Illustration of validation steps using Jigsaw. In 2.6a left, the person *Daniel Keim* (left list) is connected to the concept of *text*, displayed on the right. To find evidence for this fact, the Document Cluster View of the publications is opened (2.6b). After inspection of the cluster labels, a document from the *visualizing, interaction, text* cluster is examined in detail. This document is evidence that the fact presented in the list view in 2.6a is true. An example for the Tablet View can be seen in 2.6c.

select the background of the visualization or choose which attribute is mapped to the color of a visualization entity. Taking the color mapping as an example, there are a number of different possibilities to define such mappings, like a non-linear color scale, or the selection of the mapped colors. In this application, the authors left out any of those manipulation possibilities, and force the user to accept their pre-defined color mapping schemes.

*Model usage* is supported implicitly because most of the visualizations require special models. In some views, like the Document Grid, there are different document orderings and similarity measures to choose from. This is an example of a per-visualization model usage, which can be adjusted by the user.

The *model-vis* mapping is available in almost all views, but only at a basic level. This includes various ordering, sorting and filtering options. In general, though, this mapping is done as the developers designed it and does not allow the user to change much.

*Visualization manipulation* capabilities are mostly used to highlight document instances. There are also possibilities of manipulating and changing the visual appearance of a view. For example, in the Circular Graph View, single terms can be selected, and the connections of this term to all others are displayed.

In this examination, we show that Jigsaw supports all actions we proposed in our process model. It allows the visualization of different models based on the same data set, the visual mapping can be adjusted as designed by the authors, and the model-vis mapping can be modified to fit different analytical questions. In order to achieve a stronger coupling between the visualization and the underlying model, the interactions could be extended to modify the underlying model or algorithm parameters, e.g., when the user moves a document to another cluster in the Document Cluster View.

### Knowledge Generation in Jigsaw

Having examined what actions are supported by Jigsaw, we now focus on an evaluation of the three loops, which are part of the reasoning process as defined in Section 2.1.2.

Undoubtedly, describing the human reasoning processes using a visual analytic system is complicated. This involves a variety of aspects from perception, cognition, and reasoning. The detailed description and evaluation of these processes exceed the scope of this section and requires a study with expert users. We therefore limit this examination to a description of how Jigsaw supports human reasoning.

The *exploration loop* is broadly supported by Jigsaw, by providing a number of specialized visualizations for different analytical questions. For example, users can explore given data sets, to get an impression of the contained topics, by opening the Document Cluster view which gives a labeled view of document clusters. It is also possible to modify the number of generated clusters while exploring a once-generated cluster view. The history of consecutive runs of the clustering algorithm with different parameters is stored. This allows for assessment of parameter changes with respect to topic changes. All views include a bookmark feature, which can be used to switch between different saved states of the visualization. As a result, users can capture and annotate findings that might be of further interest or of high importance for further analysis. Bookmarks can be used to store the result of different runs of the exploration loop, which in turn is done by utilizing the actions as described in the previous Section 2.1.4.

The *verification loop*, tightly integrated with the exploration loop, guides users to develop findings into insights. These findings from the exploration loop can be used to verify or falsify a concrete hypothesis, which is the beginning of knowledge generation. The second investigative scenario given by Görk et. al. in [101] contains some questions which can only be answered by using the verification loop. In one of the examples, Jigsaw is used to examine the sentiment of car reviews. This is done to verify the car rating, which is given as a numeric score. The sentiment, displayed in the Document Grid, is expected to agree with the score in a way, that highly rated cars have more positive reviews and those with bad ratings have more negative reviews. In this example, the positive correlation of two measures is used to verify a hypothesis, which had been inferred based on product reviews contained in the analyzed data set. Natively, Jigsaw does not support the concepts of findings nor hypotheses. Although, the Tablet View displays bookmarked states of visualizations that can be organized, annotated and connected manually. This can be used to structure findings, derive insights, and connect insights to hypotheses which can be added to the view and refined by the user during the analysis process. This must be done manually because Jigsaw does not provide any automated support of this process. Figure 2.6c shows an example for hypotheses validation using the Tablet View.

Support for the *knowledge generation loop* is challenging, because knowledge generation is a process done entirely by the user, and involves concepts like trust or reasoning. In addition to these human factors, user's domain knowledge plays an important role, which is hard to incorporate, because it is difficult to externalize. Depending on the nature of the problem, the required kind and necessary level of support varies. An example would be the automatic search and suitable presentation of further evidence for a given hypothesis, which has been already verified but is not yet trusted enough to qualify as knowledge. If the Tablet View has been used during the verification loop to externalize the analysis process, the view can be helpful in the knowledge generation loop too.

We have shown that Jigsaw supports the three feedback loops which are part of the inductive reasoning process, and the amount of possible automated support varies. Starting with the exploration loop, automated support is based mainly on predefined models, and is therefore limited by the analytic possibilities of the system. Going a step further to the verification loop, it gets harder to provide adequate automated support. The variety of different hypothesis types is an important reason for this. It is even more difficult to extend a system to support the knowledge generation loop, due to the increasing influence of human and other external factors, which cannot easily be learned and represented by computers. This makes it difficult to incorporate them into an automated process supporting knowledge generation. Further implications of loop automation are discussed in Section 2.1.5.

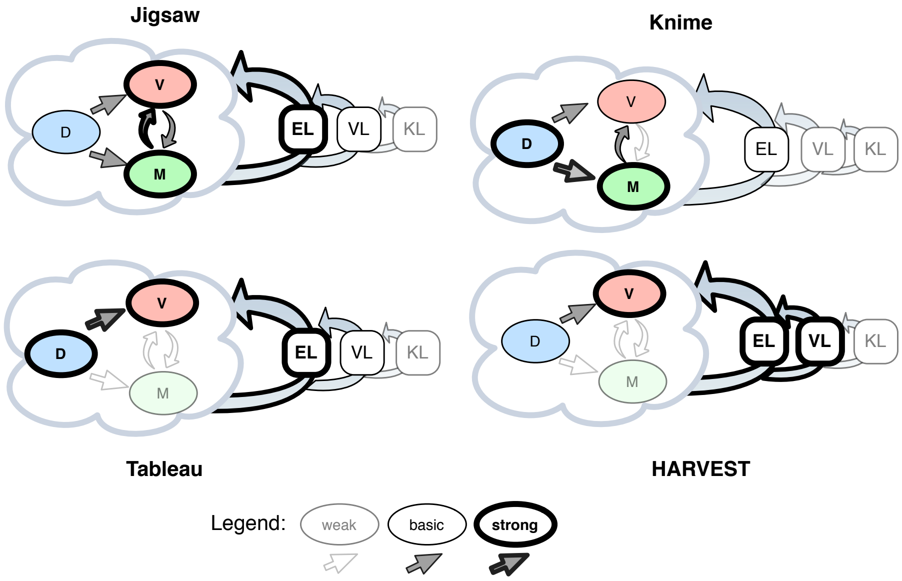



Figure 2.7: Comparative model application to different kinds of systems. Jigsaw represents visual analytics, Knime data mining, Tableau information visualization, and Harvest applications from the provenance domain. The strength of support for functionalities/components of our process model is indicated by the weight of the lines.

### Comparative System Assessment

In this section, we provide a high-level assessment of different systems from data mining, information visualization, visual analytics and provenance domain. The different natures of these systems illustrate the general applicability of our conceptual process model on applications that deal with data, provide visualizations, and are designed to generate knowledge. The comparison is shown in Figure 2.7.

At first, we assess Knime [32] (Version 2.9.1), which supports the interactive creation and execution of *data mining* pipelines. Data preparation and inspection, model building, model observation, and the model-vis mapping support is excellent, which is a good foundation for the exploration loops. The available data visualizations are on a basic level and allow brushing interactions only. Besides that, Knime provides no explicit support for the verification and knowledge generation loops, because there exist no tools to organize findings, derive insights, or connect hypothesis with insights. As a substitute, separate paths for the organization of findings and hypotheses can be added to an already existing pipeline. This is a possible way of recording the actions leading to a pipeline outcome, which can contribute to the knowledge generation process.

Next, we assess Tableau Desktop (Version 8.2 PE), which offers a number of visualizations that can be interactively adjusted by the user. Tableau is a representative of applications from the *information visualization* domain. Therefore, it has strong support for data-vis mapping, which is also the case for data preparation and data inspection. When it comes to model building, Tableau provides basic functionality. For specific data sets, it is possible to add a trend line or compute a forecast, where the model can be adjusted in designated bounds. The various manipulation and visualization options provide strong support for the exploration loops. Verification loops and knowledge generation loops support is also available with the story module, which allows free creation of reports and references to visualizations. In addition, all three loops are supported by the annotation tool, which allows the addition of persistent annotations to data points or specific locations and areas in the visualization.

Jigsaw is an example for a *visual analytics* application, which we already examined in detail in this Section. It supports all the actions we included in our model. In addition, all three loops are supported, but there is some potential for improvements. For example, a good support of the verification loops can be easily achieved by a tighter integration of the Tablet View in the system. Also, providing histories of actions leading to a specific state of a visualization (bookmark) is a step towards better support for the verification and knowledge generation loops.

At last, we examine Harvest [103, 269], a system which supports *provenance*. While using Harvest, all interaction is recorded and can be used in order to understand the way findings have been detected. Using this data, the system is able to support analysts during the exploration and verification loops, for example by an automatic ranking of manually created notes based on the user's behavior. This comes close to our definition of the knowledge generation process. The system also supports the analysis process by providing visualization recommendations or ordering of notes connected insights, or findings. Similar to Jigsaw's Tablet View, the note-taking-interface is capable of organizing, grouping, and ordering items, which supports the knowledge generation loops.

The assessment of the tools from different application domains shows that the model can be applied to applications, which work with data and visualization in general. The result also clearly separates the different application domains.

### Applying the Process Model to Malware Analysis Systems

In the following, we further compare three representative state of the art systems in the domain of malware analysis based on the components of the knowledge generation model. Zhuo and Yacin [324] developed an icon-based visualization system to analyze a malware sample executed within a sandbox environment. A circular timeline is augmented with a set of streams ("cilia") that can be investigated. SEEM as described by Gove et al. [105] is another system that allows feature-based malware comparison for large attribute sets of malware samples. It provides a visual overview with set-based similarity histograms (Venn-diagram) and provides details on demand with a relationship matrix. KAMAS described by Wagner et al. [306] supports rule-based analysis of malware samples. Malware samples are processed and analyzed using a knowledge base to extract and categorize rules of interest. Each of the systems is described in detail in our book chapter on visual analytics foundations for malware analysis [307].

**Data Preprocessing** All of the tools incorporate a complex data extraction process. MalwareVis extracts documents from pcap files, SEEM extracts malware samples, and KAMAS builds on malware sequences extracted by using the Sequitur algorithm. While the preprocessing is a complex system component and often tailored to specific analysis tasks, we observe that none of the tools enables the analyst to adapt the preprocessing interactively.

**Models** SEEM and KAMAS make use of models to extract and visualize patterns. SEEM calculates the Jaccard distance and enables the analyst to filter in the similarity histograms. However, these interactions do not influence the similarity calculation at hand. KAMAS incorporates a knowledge base that contains categorized rules for pattern extraction. KAMAS enables the user to enrich and maintain the knowledge base in a tree-based visualization. In general, we observe that models are rather used in a static fashion. Interactive model building and usage is rarely present in the current state of the art/presented tools.

**Visualizations** From the view of the used visualization techniques, SEEM and KAMAS are both using standard 2D/3D visualization techniques combined with geometrically transformed displays. In

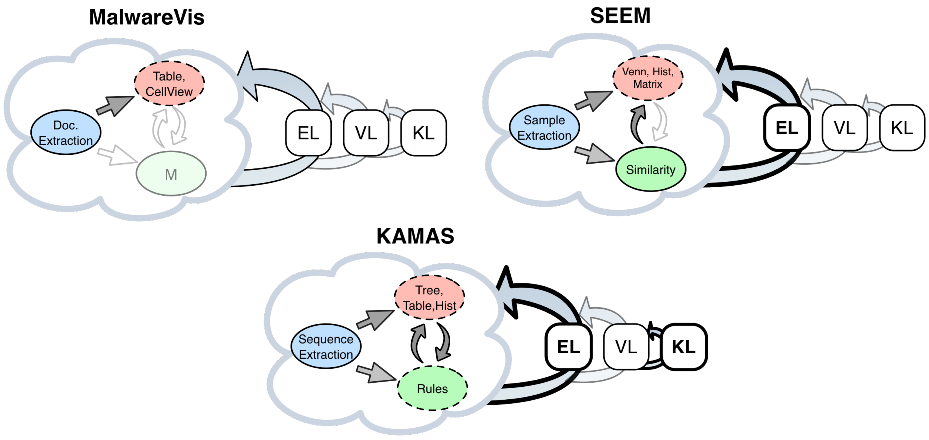



Figure 2.8: A comparison of the three tools based on the knowledge generation model. Interactive components are illustrated with dashed borders. The strength of the loops (color and border) denote how well the loops are supported by the tools.

contrast, MalwareVis uses iconic displays combined with geometrically transformed displays for the data representation.

**Supporting exploration** All the tools enable exploration interactions, and the visualizations are tuned to reveal the desired patterns. The visual interfaces enable data filtering and selection, (visual) configuration, and encode interactions. Beyond, SEEM and KAMAS make use of models to point the user to potentially interesting patterns.

**Supporting verification** The interactions in the tools are mainly limited to details-on-demand interactions to lookup, interpret, and verify spotted patterns. Furthermore, SEEM reveals certainty information within the matrix visualization to support the analyst in his verification process. In addition, the analyst can add data annotations (tags) to enrich the investigated data items with expert knowledge. KAMAS, on the other hand, enables the analyst to enrich the knowledge base that is used for rule-based sequence detection and analysis by adding useful/relevant rules to a structured categorization (rule tree). Overall, however, we observe that verification activities are rarely supported. It is not possible to bookmark and extract useful visualizations to foster evidence collection and to review the analysis.

**Supporting knowledge extraction** KAMAS makes use of a knowledge base to distinguish known from novel potentially interesting patterns. The analyst may investigate the automatically extracted patterns in more detail and assign the patterns to a family/category in the knowledge base. It is further possible to enable/disable different parts of the knowledge base to interactively seek for known patterns with specific characteristics stemming from previous investigations of multiple analysts (if they all share the same knowledge base).

In summary, we observe that data extractions are very specific and static. User interactions are mainly restricted to navigation, exploration and filtering/selection and do rarely feed back to used models/algorithms. Higher-level human analytic activities (collection and verification of evidence) are limited to simple data annotations (none of the tools provides analytic provenance information/capabilities). Only KAMAS makes use of a knowledge base to present the analyst with a categorization of malware sequences and to highlight novel patterns. This comparative evaluation illustrates that the knowledge generation model can be used to reveal major areas of improvement for future investigations.

### 2.1.5 Summary and Discussion

This study identifies new perspectives on visual analytic processes beyond weaving existing frameworks into one. Our model highlights that human and machine are a loop in the knowledge generation process using visual analytics. While existing models focus on one of these actors, our model integrates human thinking whilst describing visual analytics components. Most of the visual analytics systems do not fully cover or properly treat all aspects that our model requires as it includes the holistic process of knowledge generation which involves visual analytics systems and human users in the loop. Our model also specifies how human loops are intertwined with the subprocesses of visual analytics tools. We look at systems in a skeleton view using Keim's previous model and show how the interplay between each subcomponent can be influenced by human decision making and reasoning processes. This is important for researchers as it enables discussions of specific functions and their impacts on reasoning processes more explicitly because our model can describe the whole path from data to knowledge and vice versa. Besides connecting to system components, the model also defines human concepts and introduces three self-contained loops/stages of reasoning/thinking. We can also use it to assess a visual analytics system in terms of its functions toward human analytic outputs. For instance, we can detect areas of visual analytics systems that tend to cause biases within the knowledge generation process by aligning experts' analytic processes and outcomes against our model. Then, designers could improve corresponding visual analytics components to enhance early detection of such analytic failures: wrong hypotheses, conflicts between findings and insights, and dead ends of exploration cycles. We also find results that resonate with sensemaking, cognition and reasoning models (e.g., Pirolli and Card [221]). Our model even specifies where our current visual analytics systems fall short of, which is to support higher-level loops, namely the exploration, verification and knowledge generation loops. Supporting higher-level loops is more complex, however, this would be a useful addition to many of the current systems [234].

#### Collaboration and Communication

Our model describes the knowledge generation processes for visual analytics. There are specific types of visual analytics, which our model does not explicitly mention. Visual analytics processes could be collaborative, so multiple stakeholders asynchronously or synchronously analyze data together and gain insights through verbal communication between them. Our model currently assumes an individual's analytic process, so it is missing collaborative components. We can simplify this collaboration process with different but possibly shared knowledge generation loops of all humans (i.e., white nodes in Figure 2.1). We can explain that a number of users perform actions and interpret findings together to improve the quality of their knowledge. Also distributed cognition theories have to be considered when examining representations and interactions among humans and artifacts (Liu et al. [178]). Nobarny et al. [206] already developed a system and performed studies focusing on distributed cognition in order to support collaborative visual analytics.

Especially the externalization and communication of information plays a crucial role that needs more detailed investigations with regards to the presented concepts of findings, insights or knowledge. The importance of externalization in visual analytic processes can be observed by heavy reliance on note-taking throughout analysis processes [189]. In real world, groups of users often take a variety of approaches to synthesize information with their own organizational language [235]. Communicating knowledge in visual analytics is actually the communication of evidence found in data supporting a belief. This evidence is shared with findings or insights, e.g., a commented visualization revealing an interesting relationship in data. The communication counterpart can follow the evidence and, depending on the level of trust, gain own insights or knowledge out of it. Collaborative analysis scenarios allow communication partners to verify this evidence with the system. We should also note that collaboration between multiple analysts opens up chapters about maintaining exploration awareness [268]. In case of presentations or static documents, the missing visual analytics system is replaced by the presenter or reporter. During presentations questions and answers are able to replace the knowledge generation loops. Static documents have to convey all information from hypotheses, findings, and insights in a way that readers can follow the conclusions. Interactive reports or documents (e.g., infographics) go beyond the aforementioned scenario as they can be seen as a report combined with a limited visualization that is tailored to show the relevant findings of insights.

### Visual Analytics of Streaming Data

In addition to collaborative visual analytics, we can apply our process model to visual analytics of data streams (e.g., monitoring twitter data). In this situation, our data node is dynamically changing, so our process model reformulates its visual representations and analytic models according to substantial changes made by the data streams. Further investigation is necessary since a number of requirements on data management, knowledge discovery and visualization need to be researched due to the dynamic nature of streams. Mansmann et al. [191] illustrate this and highlight that the analyst role changes since exploration is extended to include real-time monitoring tasks where situational awareness and complex decision making come into play.

### Automatic Support of Knowledge Generation

We will now consider how novice users can be given guidance using the system and supporting analysis can aid knowledge generation. Systems can make automatic suggestions based on their current state, e.g., choosing a suitable color mapping or selecting parameters based on data.

**Exploration Loop** The exploration loop is the basis of all knowledge generation in visual analytics. An important trigger to observe findings is how the system handles interactions. Analysts can learn from the causality between interactions and reaction, hence, supporting the exploration loop requires a system to respond with an immediate observable reaction to any interaction. Many algorithms in data analysis require complex computations and are not able to calculate a complete result immediately. In these cases, analysts should at least receive feedback that the algorithm is still running. Systems should provide the ability to switch between the states before and after calculation so analysts can learn from interactions. If the final result can be estimated, for instance, by intermediate results of an incremental algorithm, the estimation could be shown to the analysts with the additional ability to abort the calculation.

Findings are related to unexpected results or patterns in models or visualizations. Automatically detecting unexpected results is complicated, because it requires a definition of what analysts are expecting, which is usually not known to systems. On the other side, pattern mining algorithms are extracting patterns directly from data, and in visualization, patterns could be detected with automatic methods as well. Even automatic methods to judge the usefulness of visualization exists, e.g., Bertini et al. [34] give an overview of quality metrics approaches.

Actions are dependent on findings and the goal of the analysis. Visual analytic systems could provide suggestions for further actions in the analysis process, as in behavior-driven visualization recommendation [102]. Based on findings, these suggestions could offload some burden from analysts of having to choose the right action from all different possibilities. Novice users would benefit from such suggestions because the system would present proper actions allowing users to learn the abilities of the system. For expert users, the interaction with the system would be more efficient compared to navigating a large set of options. It is important to find an adequate level of suggestions based on user experience. A solution to this problem could be a learning technique, such as active learning (e.g., Settles [263]), that adapts the suggestions to users and minimizes the interaction costs.

**Verification Loop** The verification loop is the central part of the knowledge generation loops. Analysts combine findings from data with their domain knowledge and gain new insights into the problem domain. The knowledge of analysts play an important role in the verification loop, therefore automatic support for the verification loop is limited to helping analysts record their analysis results.

Systems are not directly generating insights, but analysts gain new insights from data when they are able to interpret findings. In this respect, systems can support this process by providing useful summaries of findings by allowing analysts to organize findings, hypotheses, or insights. Often insights are not dependent on a single finding but are hidden in complex relationships in the data and often difficult to find without prior knowledge. Systems addressing this problem are, for instance, Shrinivasan et al. [269] and Wright et al. [316].

To formulate hypotheses about the problem domain, findings are not enough as analysis requires insights into the domain. However, systems can automatically formulate hypotheses about the analyzed data from findings. These hypotheses can help analysts get a faster overview of unknown data sets but their use for complex analysis tasks is limited because only hypotheses about relations in data can be generated automatically.

**Knowledge Generation Loop** Automating the steps from insights to knowledge or from knowledge to new hypotheses is according to our definition not possible. Analysts gain new knowledge when the evidence collected with a visual analytics system is convincing. The best way to support knowledge generation with visual analytics are systems with the ability to look at data from different perspectives. This gives analysts the possibility to collect versatile evidence and increases the level of trust in findings or insights.

### Future Investigations on Visual Analytics Systems

Our model specifies some areas that our current research can further investigate. We find some interaction types missing in many systems and interaction models, especially in the model construction or the coupling of models with visualizations. Many visual analytics tools tend to maintain preloaded models or to provide a very limited capability to manipulate models by adjusting some parameters. We observe that data become more and more dynamic and unstructured and human interaction on the model part is therefore crucial to analyze such data. Secondly, we see that many improvements could be made to further support human actions. Systems can actively learn from user behavior and adapt its models and visualizations, too. Visual analytics systems could proactively seek next candidate actions based upon user-generated logs. Our model points out that we need more explicit support to transfer findings, insights, and knowledge. As Endert et al. [87] states, visual analytics should recognize and integrate human working processes into the system. Different ways of how systems could offload some burdens from human users exist. The key is the interaction with visual analytics, although we should be aware of interaction costs (Lam [166], van Wijk [298]). Frequent interactions triggered by the system could demand too much effort, which may discourage user's exploration.

After analysis, the results have to be documented or communicated to others. An interesting ability of visual analytics systems could be a semi-automatic approach for generating documentations. Algorithms could detect interesting new findings and put them together in a convincing form, removing dead ends and duplicate findings. Such intelligent journals would make the creation of reports or presentations much easier. Alternatively, written reports could be enriched with findings supporting statements in the document.

The analysis process often follows one direction, and many findings are not explored in depth. Visual analytics systems could provide functionalities to backtrack former analysis results and suggest interesting but not investigated findings for further analysis. This would require algorithms to judge the interestingness of findings in the context of the conducted analysis results. This functionality together with a short summary of previous results could be helpful for continuing an interrupted analysis session.

#### Real World Scenario

Our conceptual process model reflects an ideal scenario where an analyst uses one single visual analytics system that is capable of handling all requirements, but real world scenarios are different. On the one hand, users are often not aware of the system's capabilities and lack the required expertise to understand complex analysis methods (Gulf of Execution). On the other hand, the system's capabilities are not sufficient to solve an analysis task. As an outcome, analysts may stop their analysis in order to consult domain experts or continue their analysis with another system. In addition, our process model implies that each action ends in an observable reaction of the system. Real world systems often lack this capability, which is a known problem in interaction science (Gulf of Evaluation).

The analysis of real world problems requires both expertise about the analysis and the domain. Domain experts often lack experience in understanding computer systems, visualization techniques, and analysis methods, whereas visual analytics experts lack sufficient domain knowledge. Thus, analysis requires collaboration between them. Without domain knowledge, a visual analytics analyst is able to generate findings and insights concerning the data, not the domain. The domain expert is responsible for the formulation of problem hypotheses, the detection and interpretation of patterns. Domain experts need to be familiar with visual analytics methods and systems. On the other hand, analysts have to learn about the problem domain.

One thing to keep in mind is that our process model simplifies many different processes, and it contains inherent fuzziness, especially on the human side. That is the reason why our model consists of various loops that represent several levels of thinking. Obviously, no visual analytics tools can differentiate exactly between reasoning processes. Furthermore, the processes can disseminate many influences into other processes, which may not clearly appear in our model. Often many conflicting hypotheses are investigated in parallel and derived findings, insights or knowledge may affect each other.

#### Teaching Visual Analytics

Our process model can be useful to provide a general overview for novice students, designers, and researchers in visual analytics. Teaching visual analytics often require teachers to provide fundamental concepts commonly appearing throughout applications, which inevitably involves domain knowledge. Our model can be used to explain the knowledge generation process without the need for such expert language. In addition, this process model touches upon various components of visual analytics, such as interactions and automatic algorithms, in the perspectives of visual analytics applications. This guideline could point researchers to relevant literature in case they want to find out about specific methods. Our model also highlights the importance of the interplay and collaboration between human and machines. This rather obvious but easily forgotten notion could be highlighted, as illustrated in Figure 2.1.

### 2.1.6 Conclusion

We introduce a process model for knowledge generation in visual analytics that integrates system and human aspects. The model defines and relates relevant concepts and provides a knowledge generation process from knowledge to data and vice versa. It embeds concepts into a three loop framework and illustrates possible human-machine pairings that are fundamental in visual analytics. We illustrated that our process model integrates with existing models and theories that focus on specific parts within the overall context. We demonstrated the process model's application using Jigsaw as an example system as well as undertaking a comparative assessment of other applications (general well-known systems and a comparison within the domain of malware analysis). Finally, we discussed implications, named open issues, and pointed to future directions. The right-hand side (human part) of our process model is not restricted to visual analytics and can also be relevant for other disciplines as it combines computer and human-based theories. The process model aims to give a basis for more detailed compositions of theories. Whilst it is not our intention to cover every single detail of such processes, we do provide an overview that commonly appears in visual analytics processes. We also acknowledge that human's knowledge processes cannot be linear or clearly subdivided into components of our process model (e.g., insight). However, we provide inherently limited but meaningful distinction between human's knowledge gaining processes.

The described knowledge generation model depicts the theoretical foundation for the remainder of this dissertation. The following section describes a novel dynamic and interactive visual abstraction technique to analyze soccer movement that is inspired and derived from the knowledge generation model. A central idea is to develop the solution in close collaboration with domain experts within a user-centered design process to address many of the issues raised in this section (e.g., interactive model steering, automatic support, bookmarking). We further explicitly support the exploration process of the resulting abstraction space with a recommender system that learns from explicit user feedback.

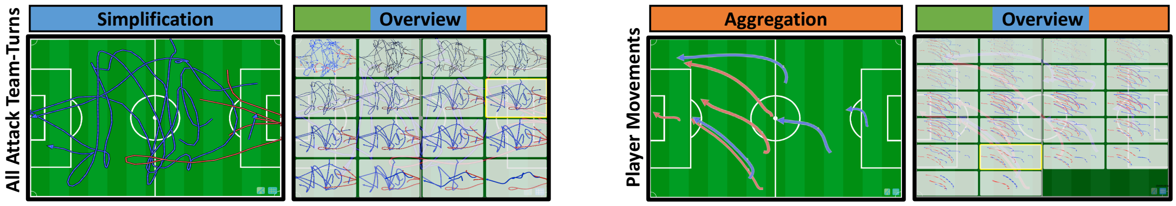



Figure 2.9: Abstracted movements are shown together with overviews on varying levels of abstraction. On the left, all attack turns of both teams with a shot on goal event of an entire soccer match are shown revealing the different attacking styles of the teams. On the right, several player trajectories are shown for a few seconds of movement revealing similarly moving groups on the soccer pitch.

## 2.2 Dynamic Visual Abstraction of Soccer Movement

Trajectory-based visualization of coordinated movement data within a bounded area, such as player and ball movement within a soccer pitch, can easily result in visual crossings, overplotting, and clutter. Trajectory abstraction can help to cope with these issues, but it is a challenging problem to select the right level of abstraction (LoA) for a given data set and analysis task. We present a novel dynamic approach that combines trajectory simplification and clustering techniques with the goal to support interpretation and understanding of movement patterns. Our technique provides smooth transitions between different abstraction types that can be computed dynamically and on-the-fly. This enables the analyst to effectively navigate and explore the space of possible abstractions in large trajectory data sets. Additionally, we provide a proof of concept for supporting the analyst in determining the LoA semi-automatically with a recommender system. Our approach is illustrated and evaluated by case studies, quantitative measures, and expert feedback. We further demonstrate that it allows analysts to solve a variety of analysis tasks in the domain of soccer.

### 2.2.1 Introduction

The recent advance of modern motion tracking technologies makes movement analysis practically applicable in many different domains. An example is professional competitive team sports (such as soccer) where movement is bound by a specific area, restricted to specific rules, and coordinated by tactical and reactive behavior. Visual Analytics (VA) techniques for movement analysis can support users from media, sports analysis, coaching, and management to extract valuable insights from this data. However, raw tracking data is hard to process and interpret due to severe overplotting and clutter when visualized. This is true even for a few minutes of movement to be visualized (see Figure 2.10 for an illustrative example). A generic solution to handle the trajectory overplotting problem is to simplify, abstract, or aggregate the trajectories, a concept that is known as *Generalization* in cartography [187]. Many trajectory generalization techniques have been proposed and applied to a variety of data types and application domains [70]. However, the techniques are often specific and hard to generalize. For example, individual techniques may be useful for a specific amount of tracked data points (e.g., hundreds vs. thousands of tracked trajectories), or designed for a specific analysis task and application domain, or restricted by computational complexity. Further, analysis goals are often ill-defined requiring adaptive exploration of abstractions.

To address this problem, we propose a novel VA approach that combines several state of the art generalization methods, and allows an analyst to interact and explore the space of trajectory abstractions, to find the most appropriate for the analysis task and data set at hand. Specifically, we implement trajectory simplification and aggregation techniques as layers of a generalization hierarchy. As a running use case, we apply and evaluate our approach to soccer trajectories. As the basis for

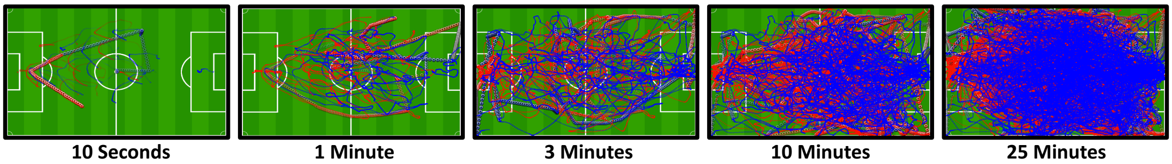



Figure 2.10: Raw trajectory plots of 22 player and the ball movements for increasing time intervals. The visualizations suffer from overplotting. Our work tackles the problem of applying an adequate generalization technique that dynamically adapts to the data that needs to be visualized.

steering the generalization techniques, we introduce a *level of abstraction* (LoA) parameter that is mapped to specific abstraction layers and configurations. This enables us to provide the analyst with smooth transitions between the used techniques and to support analysts in building a mental model of the underlying computations. Our application to soccer movement was inspired by tactical board drawings (sketches) of soccer experts, where abstraction is naturally done. We implement interactive visualizations supporting a variety of analysis task and further support the analyst in finding an adequate configuration of the technique by adding a recommender system that learns and predicts the desired LoA based on captured measures from explicit user feedback. The applicability and usefulness of our technique is illustrated and evaluated for the domain of soccer movement analysis with two qualitative and one quantitative user studies. Especially the interactive and dynamic computations in real-time helped the analysts to understand and assess the generated visualizations.

### 2.2.2 Related Work

We review general related work in visual analysis of trajectory data and applications including team sport. Afterwards, we detail methods for trajectory reduction and aggregation, followed by a discussion of existing works in user guidance and recommending for visual exploration. Finally, we emphasize the novel aspects of our approach as compared to existing works.

#### Visual Analysis of Trajectory Data and Applications

As a basis, object movement can be described by trajectories, i.e., positions in a movement space as a function of time. Due to advances in sensing technologies, trajectories can be recorded at high accuracy and spatio-temporal resolution, e.g., for traffic vehicles, animals, or the members of a sports team during a match. The recent textbook on visual analysis of movement [7] proposes a comprehensive approach covering multiple analysis perspectives: focusing on moving objects, locations, and time intervals and proposing corresponding methods suitable to various application domains.

Analysis of trajectory data is important in many application contexts. For example, the study of *movement ecology* in biology concerns understanding movement patterns of organisms in nature. An encompassing overview of movement ecology analysis is given in [70]. The authors distinguish tasks in analyzing spatio-temporal patterns, classification and identification of behavior, and the relation between movement and the environmental surrounding. Also, trajectory-based analysis of team sport data has recently gained interest, with a number of works addressing the soccer case. For example, Perin et al. [217] introduce visual designs to present and summarize game situations in soccer matches. Soccer pass analysis and methods to cluster player trajectories are addressed by Gudmundsson and Wolle [112]. In own previous work, we describe feature-based techniques to visualize, segment, and classify soccer data [135] . We also describe work for the visual exploration of soccer player interactions and free space situations in an interactive system [278] and, furthermore, introduce an approach for sketch-based search and comparison of soccer movements [265]. There are manifold other applications for visual trajectory analysis, including analysis of moving persons in buildings [132] or vehicles in a street network [2].

### Visual Aggregation, Reduction and Simplification of Trajectory Data

Next, we detail several existing approaches to reduce the amount of rendered information for trajectory data. In *visual abstraction*, the goal is to summarize large trajectory data using appropriate visual representations. Willems et al. [311] compare the effect of density-based aggregation, animated dots, and the space-time-cube technique regarding movement pattern understanding. *Data reduction* techniques can be applied to reduce data size prior to the analysis and visualization. Data clustering is a well-known data reduction tool that forms groups of similar data items. Based on appropriate trajectory similarity functions or feature-based representations [216], clustering algorithms, e.g., k-means, hierarchical or density-based clustering can be applied [116]. For example, Andrienko et al. [9] use density-based clustering to analyze a large set of traffic trajectories via a smaller number of clusters. Schreck et al. [255] employ the Self-Organizing Map algorithm for trajectory clustering to generate overview visualizations and support pattern comparison. Besides clustering, also filtering approaches can reduce trajectory data. We note that clustering is usually applied to complete trajectories, while filtering finds sub-trajectories. Von Landesberger et al. [304] apply a moving average analysis to certain time-dependent trajectory features, with the goal to filter for potentially interesting sub-trajectories. Trajectory filtering based on predefined trajectory features or user-sketched trajectory outlines is further proposed by Hurter et al. [127]. Additionally, Janetzko et al. [135] propose trajectory feature visualizations for interactive filtering and identification of relevant sub-trajectories.

While clustering and filtering reduce the number or size of trajectories, *simplification* techniques [187] are available to reduce the level of detail of trajectories. Andrienko and Andrienko [10] propose a tessellation of the space by Voronoi polygons that reflect the density of key points of trajectories, and represent trajectories as flows between the polygons. Several works consider trajectory simplification to analyze trajectories of groups of objects or subjects. The work of Laube et al. [168] represents trajectories as sequences of turns, supporting the analysis of coordination of moving objects. Andrienko et al. [11] propose to re-map trajectory coordinates to a so-called group-space, thus allowing to find different roles of group members. Furthermore, a similar transformation is used for characterizing football situations in time intervals and clustering time intervals by so defined feature vectors [8].

### User Guidance and Recommending

Our approach includes learning of trajectory abstraction levels from user feedback, hence we relate to recommending and relevance feedback. The main goal of recommender systems [136] is to support user search by suggesting previously unseen yet potentially relevant information. This can be seen as a problem of classification of relevance. Recommender systems often rely on information from a given user basis, such as user profile (e.g., classes of users), explicit or implicit user feedback information (e.g., provided ratings or reviews), and log data. In Information Retrieval, relevance feedback techniques [18] distinguish relevant and irrelevant information items based on explicit user feedback. Recent works have applied approaches from recommending and relevance feedback to support visual-interactive exploration. Relevance feedback can be applied to predict relevance of previously unseen scatter plot views, effectively narrowing the search space [23]. Healey and Dennis [118] train a classifier from user-selected views in a larger geospatial data set, supporting navigation to unexplored data areas. Finally, in own previous work we train classifiers to find potentially interesting scenes in soccer matches based on trajectory features [135] and explicit user feedback.

### Summary and Novel Aspects of our Work

Related work on the visual analysis of trajectory data provides useful building blocks for an interactive exploration system, but not yet allows for real-time interactive data reduction, simplification, and aggregation. Our work addresses this challenge by combining such existing trajectory abstraction techniques with the aim to enable real-time interactive search of appropriate data reduction and simplification levels for visual trajectory exploration. With such a formulation we bridge the gap between the prior work and our novel developments. We also support this exploration by applying a simple recommender module that can suggest potentially useful levels of aggregation and abstraction, based on an existing set of training observations. Our approach hence can improve the efficiency of trajectory analysis and dynamically adapt to user preference and analysis context. By application, our work contributes to data analysis for soccer data, however, is not limited to this.

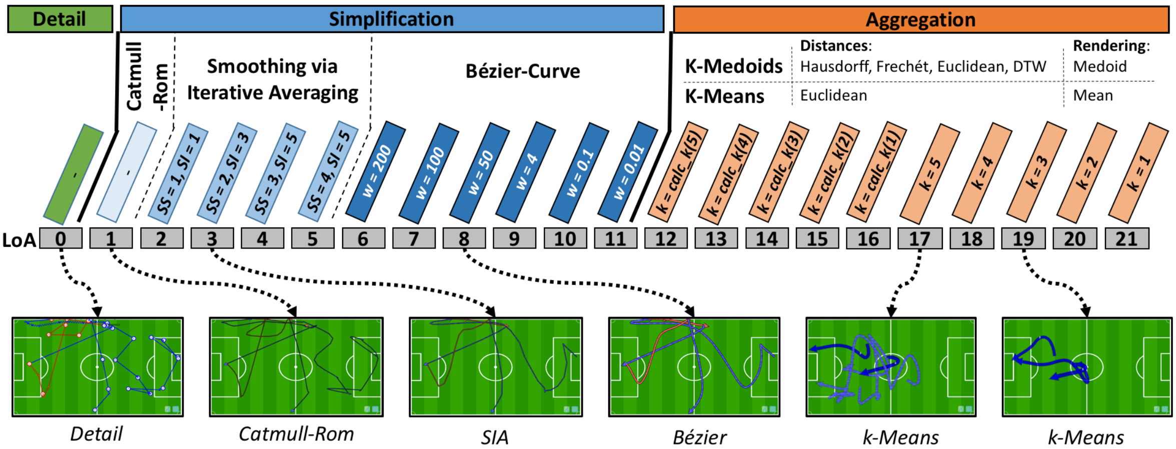



Figure 2.11: The image illustrates our basic idea and implementation of chaining several different trajectory abstraction techniques. The different layers and used techniques are shown on top. In the center of the figure, we can see how the techniques are parameterized and mapped to a global level of abstraction parameter. The bottom shows the visual outputs for some of the steps within this abstraction space.

## 2.2.3 Dynamic Visual Abstraction of Soccer Movement

Our approach combines three fundamental principles of data or movement generalization, which arose from our discussions and have been identified in related works:

- *Focusing on Relevant Information* **Detail**
- *Trajectory Simplification* **Simplification**
- *Summarizing Similar Movements* **Aggregation**

Following our "focus - simplify - aggregate" mantra, we perform visual abstraction and describe in the remainder of this chapter how we realized and implemented them in the domain of soccer movement.

### Design Study Methodology

We conducted a user-centered and iterative design study methodology process [259] to build our visual abstraction system. We used real data from professional soccer matches and work in close collaboration with a domain expert (E1) that already worked with us in previous studies. He works for the German soccer club FC Bayern München as a certified coach in the youth sector. Furthermore, he has been an active soccer player for 24 years. During our previous projects, we came across the problem of abstracting the movement depending on the selected amount of data.

To identify the challenges and requirements of a meaningful visual representation as described in the next section, we invited the expert for a first round of interviews followed by a research group discussion about framing and defining the project. We used the resulting abstraction requirements as an input for our initial design phase choosing proper (from data perspective) simplification and aggregation algorithms. With this bundle of methods and a first prototypical implementation, we were able to discuss with E1 in a more concrete manner. On the one hand, E1 was able to provide feedback about the implemented abstraction layers, used techniques, and parameterizations. On the other hand, we were able to elaborate on a variety of concrete analysis tasks (see Section 2.2.4). We iteratively implemented and revised the order of simplification and aggregation methods until the sequence of abstraction steps felt natural to the domain expert. In a subsequent and longer implementation phase, we built the major parts of our final system.

Hence, we conducted a qualitative user study with the aim to validate our design. For this study, we were able to recruit a second (unbiased) soccer expert (E2). He was invited from the group of active amateur players and is playing soccer actively for 16 years and has been part of several teams and leagues (e.g., the third-highest league of Switzerland). We decided for this second expert to have a "broad" representation from people of different levels of professionality and education that could make use of such a system. Additionally, we decided to fine-tune the detailed parameterizations with another qualitative user study with E1 (see Section 2.2.5). Based on the feedback and our observations of both studies, we were able to apply final adaptions and to define the "default" parameterizations.

We also observed that a recommender system could be a useful addition to support exploration. After the implementation of a recommender system, we conducted a quantitative user study with both experts to measure its accuracy and were able to identify next steps (see Section 2.2.6). Finally, we reflected our design process (see Section 2.2.7) during the writing of the paper [238].

### Data and Domain Requirements

We derive our work motivated by soccer analysis. The typical data set of interest contains 22 captured trajectories (one trajectory for each player on the soccer pitch) and manually annotated event streams. Events are captured for passes, shots, throw-ins, fouls, cards etc., each event containing a timestamp, position (x,y), actor-id, and an event-type. For some matches, the ball trajectory is available but it is possible to derive and interpolate the ball movements from respective events (e.g., passes and shots). The trajectories are captured with a resolution of $100ms$ resulting in approximately $1,242,000$ data points (10 points per second $\times$ $60s$ $\times$ 90 (minutes per match) $\times$ 23 (trajectories)) per match. Our system is able to handle soccer specific changes to the dataset as for example player substitutions or team changeovers after the half-time break. We implemented an optional pre-analysis routine that rotates all movements of the second half by 180 degrees to enable an half-time independent analysis.

Our visualization methods support the understanding of complex and dense movement behavior by visual abstraction. While short time windows in principle allow very detailed analysis views, we also need means of abstraction for single and multiple time spans. Of course, abstraction techniques suitable for movement patterns are highly dependent on the length of the selected time span. Dealing with intervals ranging from seconds over minutes to hours requires adaptive and dynamic techniques. However, the design space of visual abstraction is large and needs to be narrowed down. Consequently, we involved a domain expert and discussed the needs for abstraction methods. We derived the following four requirements:

**R1** The abstracted movement needs to express that we do not visualize the raw movement data but rather an approximation.

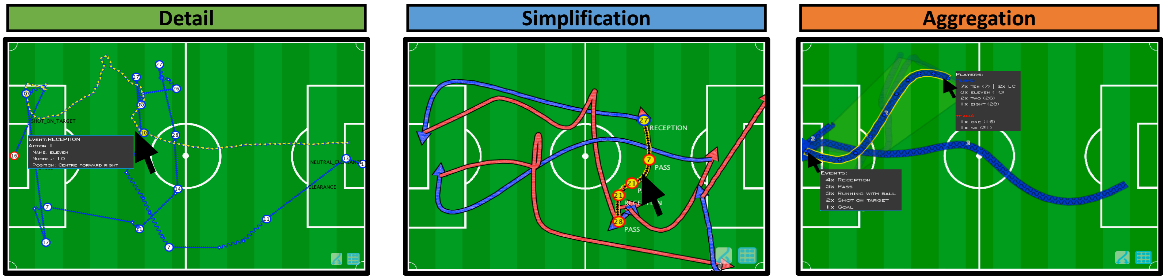



Figure 2.12: Hovering interactions are provided in each layer: In the detail layer, hovering specific events will reveal tool-tips and player trajectories. In the simplification layer, the trajectory is highlighted and the events are shown. In the aggregation layer, the cluster is highlighted and different tool-tips reveal the involved players and event counts. Further, the sample points of the cluster representation are shown and the cluster member trajectories are visualized in the background including their convex hull.

**R2** All data points, including outliers as they represent important tactical features, need to be reflected in the abstraction.

**R3** The overall abstraction algorithm needs to be parametrized and allow for varying levels of abstraction.

**R4** Computing the abstracted visualization must be possible in interactive time.

Our initial idea was to visualize the ball movement enriched with important events and players. We therefore focus on these developments in the following paragraphs and discuss in detail our chosen abstraction methods depicted in Figure 2.11. Note, that we selected the methods based on the application needs and expert feedback. Nevertheless, the proposed techniques are exchangeable depending on the respective analysis and visualization requirements.

### Detail Layer - Focusing on Important Information

We define a "team-turn" as the sequence of actions (or events) of a specific team in ball possession. Obviously, the ball trajectory is the most important information in this case, as it determines all the other movements. Consequently, we visualize the movement of the ball and numbered glyphs for the involved players (Detail in Figures 2.11 and 2.12). Movement types (e.g., passes, dribbling, ball reception) are displayed by common used strokes in the soccer domain (straight lines for passes and curly lines for dribbling). By focusing on the ball movement, we are able to reduce the amount of data visualized. Details on demand (e.g., movement of the involved players) are enabled and triggered by mouse hovering. The movement of involved players is visualized as small triangles pointing in direction of the movement. It is also possible to show (and abstract) specific player trajectories of interest permanently.

### Simplification Layer - Trajectory Simplification

We employ several simplification techniques ranging from low to high simplification illustrated by Simplification in Figures 2.11 and 2.12. We specifically selected methods that use data points as control points for the simplification enabling outlier-aware abstraction as stated in our second requirement **R2**.

We use Catmull-Rom-Splines as an abstraction method resulting in trajectories very close to the observed ball movement. As an intermediate abstraction step we apply Smoothing via Iterative Averaging (SIA) [192]. SIA offers two parameters *SI* (Smoothing Iterations) and *SS* (Smoothing

Sensitivity) that allow to control the line smoothing. Increasing these parameters will increase the degree of simplification. The strongest simplification is performed by Bézier-curves. Rational Bézier-curves allow us to add adjustable weights ($w$) to control points determining the force of attraction for the shape simplification. Applied to our movements, we can decrease $w$ to increase the degree of abstraction. In our soccer case, we linearly select ten control points. The first and last points are the most important information for analysts: The begin tells where and how the team captured the ball, while the end represents the outcome of a team-turn. Consequently, to retain these important features of the trajectory, we do not smooth the begin and end of the trajectory and keep their weightings fixed ($w = 200$) while the other control points will be decreased. Note, that other application domains might require different control point weightings.

We combine all the mentioned simplification techniques and parameterizations to a chain of increasing simplification levels exemplified in Figure 2.11. The specific ordering and parameterizations of the techniques have been developed and evaluated in close collaboration with domain experts. It is worth mentioning that we incorporated the SIA technique in a later stage based on the feedback that the transition between Catmull-Rom-Splines and Bézier-curves was not smooth enough. In our development, we first focused on simplifying the ball movements of team-turns. Although ball and players have different physics of movement with very different movement properties, it was possible to apply our implementations also to the player movements in a later stage.

### Aggregation Layer - Summarizing Similar Movements

We aggregate trajectory segments according to similarity employing clustering (Aggregation in Figures 2.11 and 2.12). We chose algorithms that produce $k$ clusters. This parameter perfectly fits to our intended LoA as stronger abstraction can be obtained by decreasing $k$. For our application domain, two different well-established algorithms are proposed: k-Means and k-Medoids with standard Euclidean distance measures. For k-Medoids we additionally implemented further state of the art distance measures in the area of trajectory clustering (Frechét, Hausdorff, and dynamic time warping - DTW). These clustering methods are easily explainable and per se define cluster prototypes. K-Means creates an artificial cluster prototype while k-Medoids determines a cluster member to this end. From the trajectories we sample the same amount of points resulting in same-length feature vectors. For our application, ten sampling points are sufficiently describing usual observed team-turn segments. A cluster is rendered by a simplified cluster centroid based on the ten sampling points. Both methods are easily controllable by the number of desired clusters ($k$) and efficient enough to support real-time interactions required by **R4**. The granularity of decreasing $k$ is dynamically obtained: according to our soccer experts the abstraction levels from $k = 1$ to $k = 5$ have to be always available. Of course, if there are for example only two trajectories to be aggregated, the maximum supported $k$ will be 2. Additionally, we provide five additional clustering levels $calc\_k(X)$ depending on the number of trajectory segments as defined in Equation 2.1.

$$calc\_k(X) = \frac{\max(|S|, kMax)}{5} \cdot X \tag{2.1}$$

In Equation 2.1, $X$ is the abstraction level ranging from 1 to 5, $S$ is the set of trajectory segments to be clustered, and $kMax$ is the maximal number of clusters. In several expert discussions, we found a value of $kMax = 30$ suitable to the analysts needs, being a good compromise between details and overplotting. The decision whether to apply k-Means or k-Medoids (as well as the different distance functions) is dependent on the respective analysis task as shown later in the evaluation (Section 2.2.5). Please note that the trajectory segments are obtained based on the ball possession for team-turns. Further segments (for each moving object) can be obtained from manual time interval selections or in combination with feature-based segmentation techniques (provided by Janetzko et. al [135]).

### Method Chain - Combining Generalization Techniques

We combine and chain the methods according to their ability to handle growing amounts of data and to their degree of abstraction as postulated in the third requirement **R3**. We combine the methods described above into a linear abstraction process as shown in Figure 2.11 controllable by mouse wheel. The crucial point of our sequence of different simplification and aggregation techniques is the proper setting of parameters. We iterated several times over the proper order and parameters with subject matter experts in soccer analysis. Consequently, we are able to visualize smooth model transitions and let the analyst interactively explore the *Abstraction Space*. It is further possible to generate an Overview of all LoA parameterizations (e.g., shown in Figure 2.9) that lets the analyst compare and choose an adequate visualization. At this point we want to refer the reader to the video[1] illustrating our approach.

We visualize the different abstraction layers in different ways conveying that we do not show the raw movement as required by **R1**. In the Detail layer, the raw team-turn data is visualized using the ball trajectory with specific events and movement types (pass, dribbling) where we use the color of the team in ball possession. Raw player trajectories are visualized as little triangles pointing into the movement direction. In the Simplification layer, the trajectory thickness is mapped to the LoA and we use lighter colors and textures for simplified player trajectories. In the Aggregation layer, the trajectory/cluster thickness is mapped to the number of cluster members, which is also shown by a label. We again apply different textures to distinguish players from team-turns (players with lighter color and texture, team-turns with darker color and texture). We treat the team-turns as well as the players of the particular teams as separate aggregations (e.g., it is not intended to aggregate player trajectories with team-turns or team-turns of opposing teams).

## 2.2.4 Analyzing Soccer Movement

This section describes our visual abstraction implementation integrated into an existing soccer analysis system described by Janetzko et al. [135] and showcases how it supports soccer experts in their analysis. The default user interface comprises a soccer pitch with the trajectory (layer)-rendering, an option panel to define configurations, and a timeline view that shows selected time intervals. For our visual abstraction work, we added the abstraction layers with its configurations in the option panels and a snapshot view that allows the analyst to define bookmarks. Additionally, we enable the analyst to create specific filters using the Move-Filter (see Section 2.2.4). Finally, it is possible to annotate and add visualizations to a note-taking interface developed by Sacha et al. [240], which is also able to capture user interactions (this will be described in detail with a user study in Section 3.2).

### Interactions

We provide the analyst with several interactions and configurations to enable interactive exploration of soccer movements.

***Object & Time Selections:*** The analyst is able to select multiple time intervals in the timeline. The option panel lets the analyst select the moving objects to be visualized (turns of team A or B, and players). We also developed a filter component enabling the analyst to define conditions to select specific team-turns of interest. It is, for example, possible to add players that have to be involved in a team-turn or specific events that have to occur (e.g., crosses). Team-turn clusters can be selected (or removed) or removed directly on the soccer-pitch if the Aggregation layer is shown.

***Abstraction Layer Interactions:*** It is possible to navigate within the abstraction space (adapting the LoA-parameter) by simple mouse scrolling. It is further possible to bring up an *Abstraction* Overview -

[1] https://www.youtube.com/watch?v=AXEJU5nW_A4, accessed 28.04.17

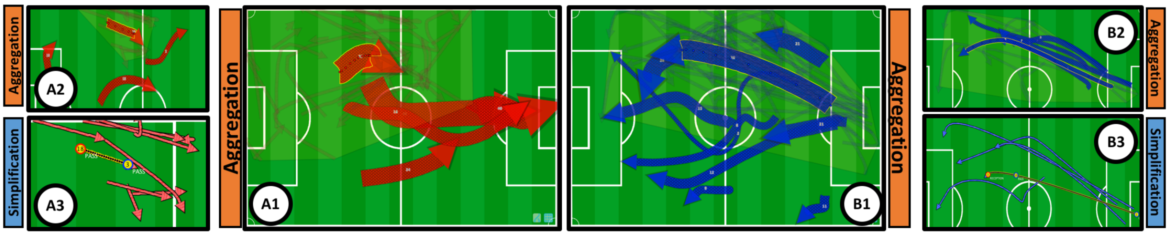



Figure 2.13: Exploratory Team-Turn Analysis: All turns of the teams are shown in A1 and B1 revealing different tactical attacking styles. The system lets the analyst explore and select clusters of interest interactively drilling down into particular team-turns of interest. In A1-A2-A3, the analyst investigated "unsuccessful" turns. In B1-B2-B3 the analyst investigated turns initiated by a long ball of the goal keeper.

overlay that computes and visualizes all steps of the global LoA-parameter. This enables the analyst to compare and select different levels of abstractions (see Figure 2.9). Detail on Demand interactions are provided in each layer: In the Detail layer, the analyst is able to hover on events and players to reveal a tool-tip and the player trajectory (Figure 2.12-left). In the Simplification layer, hovering a trajectory will reveal the important events (stations) of a team-turn (Figure 2.12-center). In the Aggregation layer, hovering a cluster will reveal all the cluster members and the convex cluster hull in the background (Figure 2.12-right). This supports the analyst in evaluating the cluster quality and to fine-tune different clustering configurations. Additional tool-tips show the frequency of events and involved players. The analyst can choose the clustering algorithm (k-Means or k-Medoids) and the used distance metrics for k-Medoids (Fréchet, Hausdorff, Euclidean, DTW) in the option panel.

***Meta Interactions:*** Analytic provenance interactions let the analyst save and add visualizations to the snapshots bar and to the note-taking interface. The analyst can re-arrange all the views and open other analysis capabilities of the soccer analysis system.

### Soccer Analysis Tasks

The presented system supports several different kinds of analysis. On the one hand, it is possible to focus on different moving objects (team-turns, players). On the other hand, the analysis can focus on continuous (temporal sequence is preserved and important) or discrete movements (spatial similarity is more important). We observed for each abstraction layer advantages and drawbacks. The Detail layer is good for fine-grained and exact analysis as no distortion is applied but limited to the amount of data. The Simplification layer works well when the analyst wants to follow a continuous movement of the ball or players and simplifies complex "branched" movements. However, the events and movement points get distorted and do not refer to the exact positions anymore. Finally, the Aggregation layer works well in identifying similar movements and to generate overviews of discrete movements, but obviously, a lot of details are lost. Putting all these methods together enables us to overcome a lot of these drawbacks and to satisfy many analysis tasks. In this study, we identified four different kinds of investigations that can be performed with our implemented system.

***Exploratory Team-Turn Analysis:*** The system visualizes all the team-turns of an entire match allowing the analyst to iteratively select and explore clusters of interest in order to drill down into specific turns. Figure 2.13 shows all the turns of a soccer match for the red team A (A1) and the blue team B (B1). We can clearly spot differences. While the red team has more turns through the middle, the blue team tries to attack via the wings (a typical V-shape is visible in Figure 2.13-B1). The analyst is now interested in the upper red cluster, because it ends at the center/midfield line of the soccer

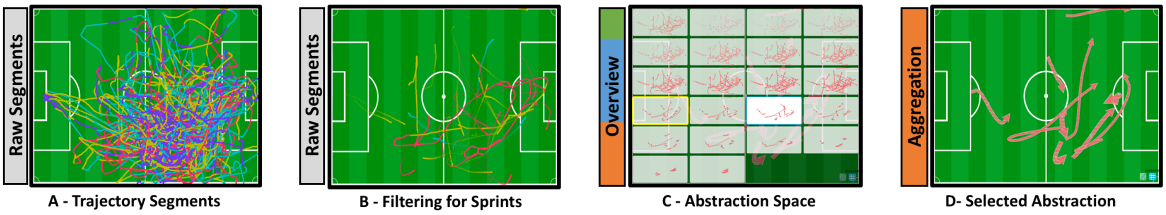



Figure 2.14: Analyzing Player Segments: Player trajectory segments are created based on movement speed, acceleration, and straightness (A). All the sprint segments are filtered (B). The resulting sprint segments of this player can be abstracted using our approach (C). The aggregated sprint view allows us to discover the main sprint areas and a few outliers (D).

pitch representing “unsuccessful” turns where the red team loses the ball too early. By selecting this cluster the analyst is provided with another abstraction of the selected turns (Figure 2.13-A2) and in a subsequent selection of the upper red cluster with a more detailed representation (Figure 2.13-A3). Hovering the trajectories revealed that player 18 of the red team was involved in most of these turns loosing the ball too early and player 3 of the blue team was able to capture the ball. A similar approach of interactively selecting clusters allowed us to identify and investigate all right wing attacks that are initiated by a long ball of the blue goal keeper (Figure 2.13-B1, B2, and B3).

***Analyzing Specific Team-Turns:*** The move filter lets analysts filter the dataset for specific team-turns. Figure 2.9-left shows all attacks with a “shot on target” event. We can identify two different tactics. The blue team shows long turns with a zic-zac pattern (a tactic with short passes and long ball possessions), whereas the red team shows relatively short turns with the ball capturing near the opponents goal and early finish (also known as “pressing”).

***Analyzing Player Segments:*** Another analysis task is to analyze the movement segments of a particular player. Player trajectory segments can be obtained, e.g., in combination with existing feature-based approaches implemented by Janetzko et al. [135]. We used their implementation to segment the movement of a midfield player based on speed, straightness, and acceleration (Figure 2.14-A) and filtered the segments for high speed and acceleration (Figure 2.14-B) in order to analyze sprints. We can apply our techniques to these trajectories (Figure 2.14-C) and investigate an aggregated representation of the players sprints (Figure 2.14-D) which are mainly in the opposing teams half of the soccer pitch on the right wing with a few outliers. It further reveals that this player always attempts to reach the left full-back when sprinting. This player might be advised to occasionally take the ball outside the full-back as predictable play is easy to defend against. Note, that the segmentation and filtering have to be applied according to the analysis task at hand. We could also apply our abstractions to all the trajectory segments or filter for different movements (e.g., stop moments with a high negative acceleration). However, such segmentations are essential preprocessings for our abstractions to support meaningful analysis tasks.

***Analyzing Collective Movements:*** Finally, we can apply the same technique to identify and investigate collective movements of several players. To do so, we selected a short time span but all player trajectories and apply our trajectory abstraction technique (Figure 2.9-right). The visualization allows analysts to identify different groups of players. Obviously, the two goal keepers are shown as separate clusters as their movement is different from the other players.

### 2.2.5 Evaluations

We conducted two qualitative user studies to validate our design and collect feedback about our implementations.

*Apparatus:* The studies were conducted in a lab setting using a 24 inch screen to show video sequences or the soccer analysis system. For the first study, the participants were provided with a sheet of paper for each task, showing an empty soccer pitch or a printout with trajectories that should be abstracted. The participants were provided with a pen to draw on the sheets. For the second study, the only input device was a common computer mouse. The participants were seated approximately 50 cm away from the screen. The experimenter was present during the study for answering questions, introducing the study procedure, and collecting feedback.

#### Visual Design

We conducted a qualitative user study with both domain experts to validate our abstraction layers and if the visual results meet their expectations. Expert E1 was involved in the design process of our visual abstraction technique, expert E2 was just involved during the evaluation phase. The general idea of this experiment was to let the experts draw the abstractions on a sheet of paper and to compare the results with our computed visualizations afterwards.

*Tasks and Procedure:* The participants had to produce drawings that are comparable to each of our abstraction layers. In the first task (T1), both participants were shown six specific team-turns from a video recording. We showed them a video sequence and paused the video when the participants requested it. For each sequence the experts had to draw the team-turn in an empty soccer pitch on a piece of paper. The experts were asked to focus on the sequential events to draw an abstract representation of the team-turn. We told them that they would not need to provide an exact drawing. The rational behind this task was to let the experts produce team turn drawings that can be used to evaluate our **Detail** and **Simplification** layer. In the second task (T2), both experts were provided with several printouts of a soccer pitch containing many trajectories representing all turns of a single team. The participants had to draw the general flow of movement on the sheet of paper on top of the trajectories. The experts got no further restriction or hints on performing the task. The aim of this task was to let them produce trajectory summarizations that can be used to evaluate our **Aggregation** layer.

***Results:*** The generated drawings are provided in Appendix A.1 and three representative results for each abstraction layer are shown in Figure 2.15. In T1, expert E2 (who was not involved in the design process) produced drawings that are very similar to our **Detail** layer basically connecting annotated circles of important events. Note, that this layer has been designed together with expert E1 who produced simplified representations of the team-turns that come very close to our **Simplification** layer. In T2, both experts produced results similar to the **Aggregation** layer by creating thick arrows above the main movements that can be seen on the soccer pitch. Both experts used *thickness* of the arrows to communicate aggregation of the underlying trajectories. This study helped us to demonstrate our abstraction and visualization concepts that are in line with the mental models of the soccer experts.

#### Comparing Clustering Results

We implemented two different clustering methods (k-Means and k-Medoids) and specifically for k-Medoids, we also implemented four different distance functions. We conducted another deeper qualitative study with one of the experts (E1) to evaluate the distance functions and to collect feedback about the clustering techniques.

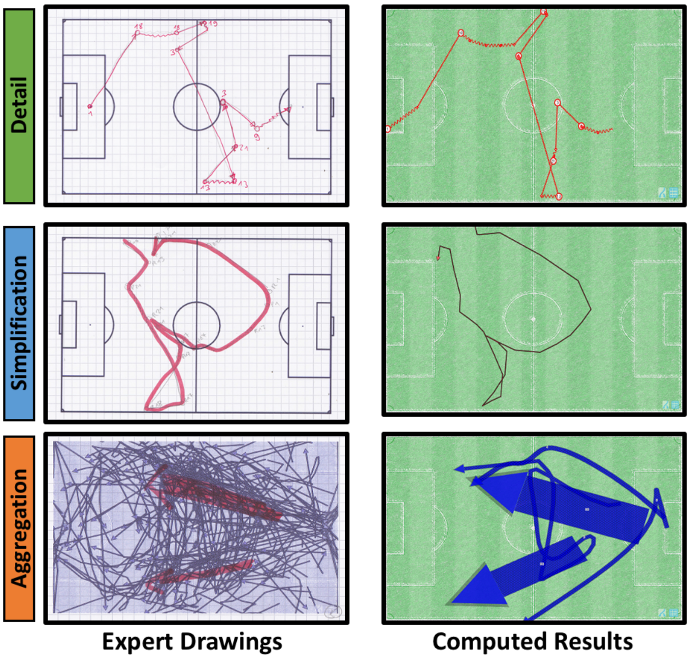



Figure 2.15: The participants had to perform the visual abstraction tasks manually on a sheet of paper. A few examples of the participants drawings are shown on the left and the computed visualizations are shown on the right (one example for each layer).

*Tasks and Procedure:* We chose eight different time and team selections of a soccer match M1: Whole match, first/second half, and team-turns with crosses (all for team A and B). We first showed the Simplification layer representation to illustrate all the trajectories that have to be clustered. Subsequently, the participant had to compare and provide feedback about the different distance functions (T1) and the preferred clustering technique (T2) within the Aggregation layer. The first task (T1) was to compare distance functions with the k-Medoids method. Expert E1 had to configure the clustering and compare the results of all the distance functions. He was further asked to provide specific feedback and decide for a preferred distance function. In the second task (T2), the expert was asked to compare the k-Medoids with the k-Means clusterings. The aim of these tasks was to collect feedback about the used methods (metric, clustering) for different match situations.

***Results:*** In T1, the expert preferred for six situations DTW and for two situations the Euclidean distance. The Hausdorff distance did not provide good results because it was not possible to spot the desired tactical patterns in the visualizations ($V$-, or *cradle*-shapes). The DTW turned out to be the expert's favorite as it was possible to reveal the desired patterns for most of the situations. However, the expert also emphasized that he likes the ability to compare and explore the different distance functions. Therefore, we provide the DTW distance metric as a the default configuration but retain the ability to change it. In T2, the expert reported advantages and disadvantages for both clustering methods depending on the analysis task and selected situation. The k-Means method was useful to estimate an aggregated direction of the movements, whereas k-Medoids was considered more useful to detect and analyze soccer-specific patterns. For example, Figure 2.16 visualizes the cluster representation (foreground) and the aggregated trajectories (background) for k-Means (a) and k-Medoids (b). The general movement direction is shown with k-Means while the *cradle*-shape pattern is preserved by k-Medoids. As a result we provide the analyst the ability to change the clustering method on demand.

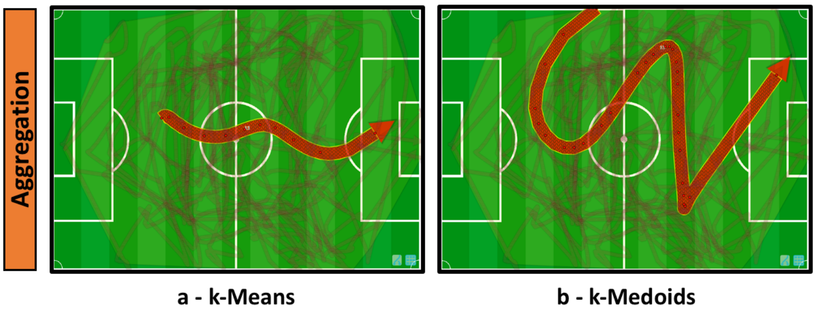



Figure 2.16: Two clustering representations for the same trajectories. The k-Means method was considered useful to analyze an aggregated movement direction, whereas k-Medoids was considered more useful to identify soccer-specific patterns.

### 2.2.6 Recommending the Level of Abstraction

We observed that setting the LoA parameter (by scrolling the mouse wheel) is useful for understanding the abstractions, however, it also turned out to be time consuming in exploratory analysis tasks. Additionally, further parameters (e.g., clustering, distance function) have to be chosen by the analyst. To support such parameterization tasks we started to build a recommender system that provides the analyst with automatic pre-configurations. We further observed that such a goal is user dependent (they may prefer more/less abstraction).

***Measures and Recommender Training:*** A domain expert provides explicit feedback about specific useful visualizations by pressing a “learn from current view” button. This feedback will create a record for the current visualization in an (individualized) recommender database. We defined several measures to capture data and visual characteristics in combination with configuration information. Firstly, we measure the sum of selected time interval lengths (data). Secondly, we measure the amount of trajectory crossings, the crossed surface area, and the sum of trajectory surfaces (visual). Finally, we capture the used abstraction layer with its local and global abstraction parameter (configuration). The system offers a 50 step training-process covering randomized and specific situations (randomized time intervals as well as match specific team turns). During the actual analysis process, further explicit feedback will individualize and adapt the recommender system. Instead of performing the personalized initial training, it is also possible to choose a “default” (pre-trained) recommendation database, which was created during our evaluation sessions.

***Providing Recommendations:*** We analyzed a captured training dataset and found a significant positive Pearson correlation between the LoA and the selected duration ($r(48) = .68$, $p = .00$). This simply means the more time and data is selected, the more abstraction is needed. As a result of this statistical test we select the time intervals as a primary feature to predict the global LoA (which is composed of the abstraction layer and the local abstraction parameter). We decided for a simple and efficient *lazy learning* classifier [15] as they are able to handle and solve multiple and changing problems. Furthermore, additional training data shall be considered during runtime. The *k-Nearest-Neighbor* (kNN) algorithm satisfies these requirements and has been implemented to determine the abstraction layer and its local abstraction parameter. Note that we cannot simply predict the global LoA because its range is dynamic (dependent on the number of trajectories, see Section 2.2.3). The process is realized as follows: First, all the learned records in the training database are sorted according to the time selection length (Euclidean distance) to identify the $k$ closest neighbors. The number of considered neighbors is determined by Equation 2.2, where $T$ is the total amount of training data, 5 an upper and 1 a lower bound.

$$f(T) = max\left\{min\left\{\frac{|T|}{10}, 5\right\}, 1\right\} \tag{2.2}$$

The kNN-records are used to determine the most frequent abstraction layer and uses the remaining records to calculate the mean value of the local abstraction parameters. The results are then used to pre-configure the LoA.

***Accuracy:*** We conducted a quantitative user study with both soccer experts to evaluate the usefulness and precision of our recommender system. The basic idea of this experiment was to compare the recommended LoA with a user defined LoA and to measure the difference as an error $e$. The participants were presented with a specific time selection and had to set the desired LoA parameter by scrolling the mouse-wheel and to save their decision using the "next" button. In the first "personalized" evaluation (T1), our goal was to measure the accuracy of a previously trained classifier for two different matches. The participants had to train the recommender using the 50 test cases for a specific match M1. In a subsequent iteration, the participants had to perform another 50 steps for a different match M2. For every step, we calculate the difference as error $e$ and added a penalty of 10 ($e = e + 10$) when two different layers were used as they support different analysis goals (e.g., Aggregation instead of Simplification layer). In the second "default" evaluation (T2), we tested the accuracy of our "default" classifier that is not trained by the participant. Therefore, we created a "default" database trained by several users for matches M1 and M2 with a size of 400 records. We then evaluated match M3 with 50 test steps for each participant and calculated $e$ similar as in the previous case.

***Results:*** The results for both participants and study tasks are shown in Figure 2.17. For the "personalized" cases (T1), we found an average error of $e = 1.43$ for expert E1 and $e = 1.45$ for expert E2. Hence, we are able to state that even our simple kNN approach is able to provide a "good" pre-configuration and is in average not worse than 1.5 steps (within the 21-step abstraction space). However, in Figure 2.17 we can also spot three outliers with very high values for the cases when the the wrong abstraction layer was recommended. For the "default" cases (T2), we found an average error of $e = 1.92$ for expert E1 and $e = 2.06$ for expert E2. In this case, the recommender system performed not as good as in the personalized case. However, the error is in average still as close as 2 steps to the user preference for both participants. Similar to the previous case, we also spot three outliers greater 10 where the wrong abstraction layer was chosen (out of 100). This study showed that good recommendations can be provided even without a personalized training before usage. However, the personalized still outperforms the default case.

***Next Steps:*** We are well aware that more sophisticated recommender systems and approaches do exist and that we just provide a simple and early proof of concept to illustrate its usefulness in our setting. Our results reveal a multitude of interesting research areas. Firstly, we will leverage more measures that can be obtained from a visualization. Examples include data-related metrics, such as the number of moving objects, but also visual measures. Note, that we can pre-compute visual measures for different parameterizations (e.g., the used clustering technique) in order to optimize the visual configuration beyond the LoA parameter. Secondly, it will be very interesting to train separate recommender systems for different analysis tasks. We envision that we could even leverage such task-databases to detect user intent (determining the "closest" task) and adapt the visualizations accordingly. Finally, our approach learns from explicit user feedback. Another interesting research direction is to obtain the training data from implicit user feedback (e.g., once a visualization is bookmarked or annotated).

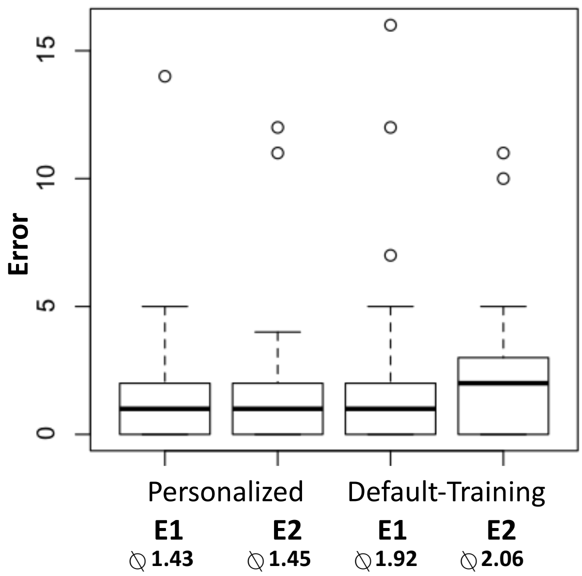



Figure 2.17: The error rates between the manually chosen and recommended LoA parameters for both participants and two evaluation cases (personalized vs. “default” training).

### 2.2.7 Discussion

We developed the system together with domain experts that iteratively helped us to adapt our implementations and refine our ideas. Our design process and evaluations focused on the visual abstraction of team-turns and a domain requirement was to strictly separate between the opposing teams and players. It will be interesting to evaluate and fine-tune our system with focus on player movements and apply our computations across teams.

By reflecting our design process we further noticed that it was “easier” for users to map the abstraction parameters to a LoA for the **Simplification** layer techniques than for the **Aggregation** (the analyst has to set more parameters). That is also why we had to conduct evaluations for the clustering. However, we also did not implement and evaluate all the visual design alternatives. We mapped, for example, the LoA intuitively to trajectory thickness (based on **R1**), however, thick arrows introduce additional overplotting. It would be interesting to parameterize thickness and investigate how much is needed and appropriate. Similarly, one could focus on the used textures and colors. Another configuration for the cluster rendering could be to show the means and medoids concurrently (if $k$ is small).

Furthermore, we designed the system with two domain experts and derived some methodological choices that worked well in our setting. However, further requirements might become relevant for a broader set of users and more specific analysis tasks. For example, the number of sampled control points and their weighting (for both the **Simplification** and **Aggregation** layer) could be parameterized and further investigated. We note our approach supports an adaptive level of simplification per given situation, and that all trajectories within that situation are simplified using the same parameters. In future work, we may also consider adaptive simplification on a *per-trajectory* level. An idea to do so is to classify the type of trajectory, e.g., pertaining to different player moves or roles, and then to adapt the number of sampled control points and weightings to this class.

Another way to improve the abstraction techniques is to take semantics or additional features into account. For example, in the **Simplification** layer, we can select and weight control points based on the events that are contained in the trajectory. Similarly, for the Aggregation layer, we can include further meta-data (e.g., events) and features (e.g., speed) into the similarity calculation.

Finally, we want to emphasize that we selected methods that naturally matched our domain requirements. In other application domains it could be interesting to investigate other abstraction techniques (e.g., grid-based or edge-bundling). We tried to complement our visualizations with heatmaps or concav/convex-hulls but we had to abandon these investigations to keep our work focused. It will be interesting future work to generalize our approach and to enrich it with further abstraction techniques.

### 2.2.8 Conclusion

This chapter presents a novel approach to define and combine visual abstraction techniques interactively to overcome over-plotting and clutter. We specifically focused on the domain of soccer movement data and designed a system that supports soccer analysts in solving a variety of analysis tasks. The interactive navigation and smooth model transitions allow the analyst to track and understand the underlying abstraction computations. A recommender system has been added as a proof of concept to automatically pre-configure the LoA based on the amount of data to be shown. In the future, it will be interesting to improve the obtained recommendations for a variety of tasks. Furthermore, we will apply this approach to other movement data types (e.g., animal movements) and adapt (or replace) the abstraction techniques with the ultimate goal to come up with a reusable and extensible abstraction framework.

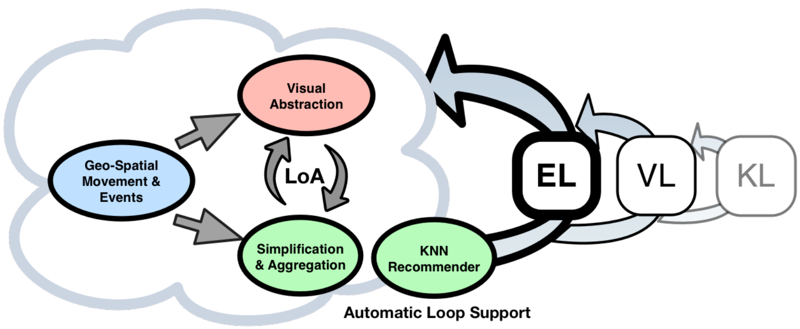



Figure 2.18: Knowledge generation model applied to our visual abstraction approach.

Applied to the knowledge generation model, we can illustrate how our visual abstraction approach supports knowledge generation. Figure 2.18 shows the systems main components (movement data that is mapped to the abstraction models and visualization) in relation to the human knowledge generation loops. We can highlight the most interesting aspects using this model: On the one hand, we achieve a tight coupling between the abstraction techniques and visualizations by introducing the LoA that can be dynamically controlled by the user. On the other hand, we support the exploration process semi-automatically with a recommender system that learns from explicit user feedback. The following chapter will dive deeper into humans‘ higher-level trust building and verification processes. In addition, it will introduce a note-taking component that can be plugged to existing VA systems to support such higher-level loops.

# 3

# Analytic Behavior and Trust Building During Knowledge Generation

*"To know what you know and what you do not know, that is true knowledge."* – Confucius

## Contents



Information needs to be credible, reliable and verified in order to become trusted or accepted knowledge that results from data analysis processes. However, we know little about humans' analytic activities during knowledge construction in typical visual analytics scenarios. The first part of this chapter will leverage the knowledge generation model for visual analytics to reveal such higher-level analytic activities in relation to technical aspects, such as errors or uncertainties stemming from automatic data processing methods. This first part is based on our paper on the role of uncertainty, awareness, and trust in visual analytics [246] where we focus on humans' analytic behavior which is influenced by technical aspects. Therefore, I shortened and revised all original parts that describe uncertainty aspects in detail to keep this section focused. The second part of this chapter describes a note-taking interface that can be plugged to existing visual analytics systems. The component supports verification activities, such as evidence collection, annotation, or knowledge management but we also propose an evaluation methodology that can be used to investigate human-computer interactions in relation to human trust building processed. This work has been published in a short paper [240] where most of the content is reused in this chapter. This chapter illustrates that the knowledge generation model can be refined by revisiting its concepts with a deeper focus on uncertainty and trust building. In a second step, the chapter reveals that such models (and more detailed perspectives) can be used to develop novel systems and approaches to investigate and support analytic behavior in visual analytics.

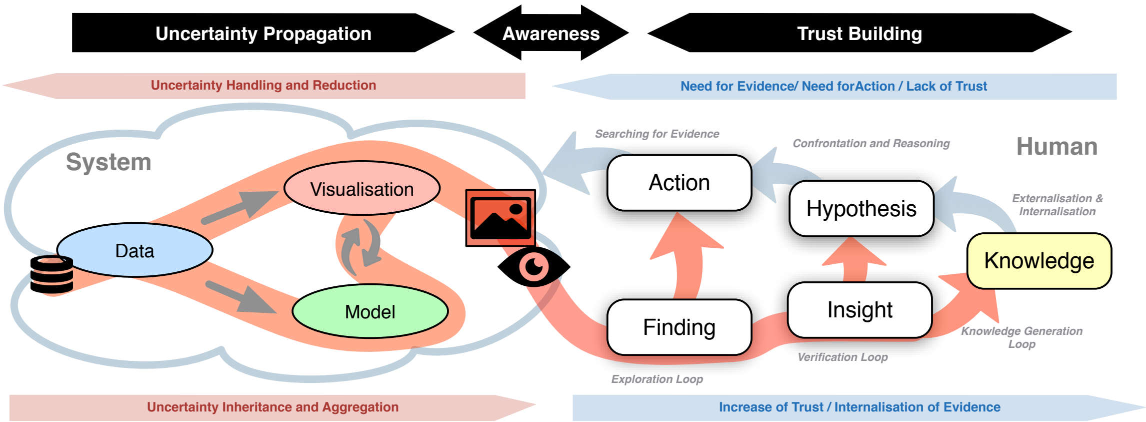



Figure 3.1: Knowledge generation model for visual analytics including uncertainty propagation and human trust building. Uncertainty originates at the data source and propagates through the system components which introduce additional uncertainties. Uncertainty awareness influences human trust building on different knowledge generation levels.

## 3.1 The Role of Uncertainty, Awareness, and Trust in Visual Analytics

Visual analytics supports humans in generating knowledge from large and often complex data sets. Evidence is collected, collated and cross-linked with our existing knowledge. In the process, a myriad of analytical and visualization techniques are employed to generate a visual representation of the data. These often introduce their own uncertainties, in addition to the ones inherent in the data, and these propagated and compounded uncertainties can result in impaired decision making. The user's confidence or trust in the results depends on the extent of user's awareness of the underlying uncertainties generated on the system side. This section unpacks the uncertainties that propagate through visual analytics systems, illustrates how human's perceptual and cognitive biases influence the user's awareness of such uncertainties, and how this affects the user's trust building. The knowledge generation model for visual analytics is used to provide a terminology and framework to discuss the consequences of these aspects in knowledge construction and through examples, machine uncertainty is compared to human trust measures with provenance. Furthermore, guidelines for the design of uncertainty-aware systems are presented that can aid the user in better decision making.

### 3.1.1 Introduction

In the visual analytics process, users arrive at new knowledge after performing numerous sensemaking activities. The goal of visual analytics is to foster effective collaboration between human and machine that improves the knowledge generation process. To succeed in this process, end users need to be able to trust their knowledge generated by means of visual analytics. Analysts can often be unaware of uncertainties in their data sources, pre-processing, analysis processes or visualizations that are hidden by a "black box" approach of visual analytics systems.

In criminal investigation analysis, where analysts use a visual analytics application to analyze a collection of reports and to identify crime suspects, the system may hint at otherwise hidden connections between pieces of evidence using a trained machine learning algorithm. To progress, the analyst needs to trust this outcome. However, if the analyst is not aware of the inherent uncertainties, they may waste their time following wrong leads and may, in the worst case, incriminate innocent people. Likewise, overestimating uncertainties can have a negative impact on decision making. It is therefore crucial for users to be provided with an accurate estimation of uncertainties from visual analytics systems so that they can trust acquired knowledge.

The literature describes some parts of uncertainty propagation and trust building in visual analytics processes, however, the interplay of trust and knowledge within the knowledge generation process in visual analytics has not yet been established. Prior studies have investigated sources of uncertainties in subsets of the visualization process (e.g., [65]). Other studies have looked at human analysts' behaviors while building trust in the knowledge generation process, with respect to perception [309], cognitive biases [109], and analytic roadblocks [164]. What is missing is a unified framework that bridges the concepts of uncertainties on the machine side and the trust building process on the human side. Recently, the IEEE VIS2014 Workshop on Provenance for Sensemaking called for research in defining uncertainty, trust, and data quality. MacEachren also highlighted human's decision making and reasoning processes under uncertainty as future research direction [183]. Building such a framework can provide a common language of the concepts that are largely uncharted in the visualization domain.

Our goal is to investigate uncertainty propagation, trust building, and the interplay between uncertainty and trust during the knowledge generation process within visual analytics. Building on the related work in *Uncertainty Propagation* and *Human Trust Building under Uncertainty*, this chapter describes a novel model of uncertainty and trust using the knowledge generation model [248] as a framework and brings in human cognition and perception issues through the concept of *Awareness*. We choose the most recent and complete knowledge generation model for visual analytics because it "integrates human thinking whilst describing visual analytics components" [248]. Furthermore, this conceptual process model was the foundation for our initial investigations and discussions about defining and relating its concepts to uncertainty and trust. To extend the usefulness of the process model, we provide guidelines on how to improve decision making, avoiding misperceptions and pitfalls generated in the visual analytic processes. Finally, we explore future directions and opportunities for handling uncertainties and trust.

### 3.1.2 Related Work

We group related work into *Uncertainty Propagation* and *Human Trust Building under Uncertainty*. The former covers work on capturing and deriving uncertainty measures within visual analytics pipelines. The latter covers the knowledge generation and the trust building processes in visual analytics.

#### Uncertainty Propagation

This subsection introduces different *Uncertainty Types*, *Uncertainty Propagation*, *Uncertainty Visualization* and *Data Provenance* theories.

**Uncertainty Types:** We can roughly distinguish uncertainties that are inherently contained in datasets and uncertainties that arise from computations and models. *Source* uncertainties are, e.g., addressed by Lush et al. [181] supporting users to assess properties of geospatial datasets (e.g., producer information). *Model* uncertainty is caused by computations, such as data transformations and model parameterizations. Many works have considered different main sources of uncertainties within data models that describe real-world phenomena (e.g., Chatfield [55]), or provide methods to handle model variability and uncertainty (e.g., Cullen and Frey [67]).

**Uncertainty Propagation:** Uncertainties are included in the data, which is further processed, passed to machine learning models or data mining algorithms, and finally used to produce a visual representation. Haber and McNabb [114] describe different stages of uncertainty propagation within a traditional visualization reference model. Correa et al. [65] introduce another uncertainty propagation framework and example applications to propagate and visualize uncertainties caused by different components of a visual analytics system. Further workflow or pipeline models of Pang et al. [213] or Zuk and Carpendale [325] consider several stages of propagated uncertainties within the visualization and reasoning process.

**Uncertainty Visualization:** A plethora of works have considered the problem of visualizing such propagated uncertainties (e.g., [120, 186, 281]). Howard and MacEachren [182, 126] provide design guidelines to relate data and uncertainties in a meaningful way in order to produce visualizations that are appropriate to the data and task at hand. Additionally, uncertainty visualization techniques have been evaluated in different tasks and scenarios (e.g., [262]). However, less research has been put into the evaluation of visualization effects that impact user perception (e.g., visual clutter) and human problem-solving (e.g., cognitive load).

**Data Provenance**: Data provenance describes the approach to capture data derivations from its origin until the final data product, information that can be used to support the humans' verification process. The survey of Simmhan et al. [271] describes a taxonomy that covers *Usage*, *Subject*, *Representation*, *Storage* and *Dissemination* aspects of provenance.

### Human Trust Building under Uncertainty

We distinguish the relevant human focused theories in *Knowledge Generation*, *Trust Building*, and *Analytic Provenance*.

**Knowledge Generation:** Tory and Möller [288] give an introduction to human factors and highlight that visualizations serve as cognitive support and address human computer cooperation. They suggest that analysts perceive visualizations and match them to their mental model of the problem. Other human factors include a users' knowledge, expertise, and tasks but also factors on perception and cognition. Zuk and Carpendale [325] extend the typology on uncertainties by Thomson et al. [287] for reasoning. Both of the typologies include a category about *Subjectivity* that represents the "amount of interpretation or judgment that is included" [287] or the "amount of private knowledge or heuristics utilized" [325]. Green et al. [109] propose the *Human Cognition Model* that covers various human aspects of knowledge creation with visual analytics. They point out that hypothesis generation is very much influenced by the human tendency to accept confirmatory evidence more than dis-confirmatory and that the computer can help to mitigate this cognitive bias. Winters et al. [312] show in their study that humans can have different roles, expertise, and knowledge that can be applied during the analysis process. This influences how individuals approach visualizations and also how they reason about their problems. Gahegan [98] summarizes different kinds of reasoning and relates them to human activities, visualization tools or computational tools. MacEachren et al. [185] mention that analysts and decision makers behave differently with and without the usage of uncertainty visualizations, whether they are aware of the uncertainties or not. They differentiate between information uncertainty and an "analysts' or decision makers' uncertainty" and also suggest that to capture, represent and understand these uncertainties are future research challenges.

**Trust Building:** Muir [202] discusses trust relations between humans and machines and builds on Barber's trust dimensions, which are *Persistence*, *Technical Competence*, and *Fiduciary Responsibility* [19]. Furthermore, Muir gives the following recommendations for improving trust calibration: "(1) improving the user's ability to perceive a decision aid's trustworthiness, (2) modifying the user's criterion for trustworthiness, (3) enhancing the user's ability to allocate functions in the system, (4) identifying and selectively recalibrating the user on the dimension(s) of trust which is (are) poorly calibrated" [202]. Dzindolet et al. [79] investigate how trust develops during the usage of a system. Initially, all participants considered the decision aid as trustworthy and reliable. Observing errors caused the participants to distrust the systems unless an explanation was provided. Understanding the errors helped the users to increase their trust in the decision aid, even under uncertainty. Castelfranchi [50] relates trust to the process of knowledge management and sharing and provides a theory that considers the process to be a decisional act of *passing* and *accepting* knowledge. Trust is related to these activities as a mental attitude, but also a decision (e.g., intention to delegate trust) and a behavior (e.g., the relation between trustor and trustee). Uggirala et al. [291] studied humans using systems that include

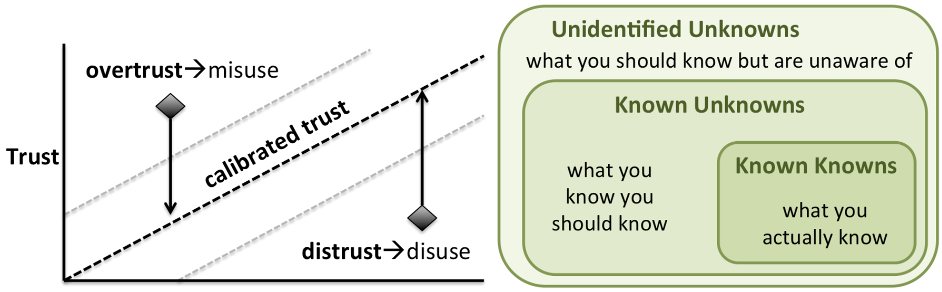



Figure 3.2: *Left:* Trust calibration adapted from [69], *Right:* Awareness classification adapted from [274].

uncertainties by having the users rate their trust at each level through questionnaires. Their study showed that trust relates to competence and an inverse relation to uncertainty, meaning that an increase in uncertainty decreases trust in the systems. Visser et al. [69] provide a design taxonomy for trust cue calibration that includes all kinds of information that may influence human judgment. Figure 3.2 (left) illustrates trust calibration and the included problems. A miscalibration between the humans' trust and the systems' trustworthiness leads to over- or distrust that is directly connected to disuse and misuse of automation. Skeels et al. [274] deliver a comprehensive perspective on uncertainties for information visualization and also briefly discuss the role of awareness (see Figure 3.2 right). They identify *Unidentified Unknowns* as the worst kind of missing information because in that case, humans are building more trust than they should do.

**Analytic Provenance:** Recent research focuses on tracking interaction in order to investigate human analytic processes. Dou et al. [73] present an approach to capture human reasoning processes that distinguishes between internal (interactions within the system) and external capturing (outside the system). Ragan and Goodall [227] go on to mention that provenance tools support the human in memorizing and communicating analytic processes. Nguyen et al. [205] survey analytic provenance and show that typical provenance consists of three stages: *Capturing*, *Visualizing* and *Utilizing* with capturing on different levels (events, low-level actions, high-level sub-tasks, top-level tasks). They also describe the benefits of analytic provenance that go beyond re-calling the analysis process to support "evidence in constructing the reasoning process, and facilitating collaboration [...]"[205]. Examples of leveraging analytic provenance will be given in the guidelines for handling uncertainties in Section 3.1.4.

In summary, our review discovers two distinctive groups of literature. One group deals with uncertainty propagation and visualization on machine aspects; the other investigates human-machine trust relations. We observe important gaps in research within visual analytics frameworks. These are uncertainties in the visualization itself, uncertainties in the coupling between model and visualization, and uncertainties in the model building. Only a few studies relate uncertainties to these human trust building processes. Furthermore, there is no clear terminology that differentiates between uncertainties from machine and human. In the following, we address these issues and provide a process model that integrates uncertainty propagation and human trust building.

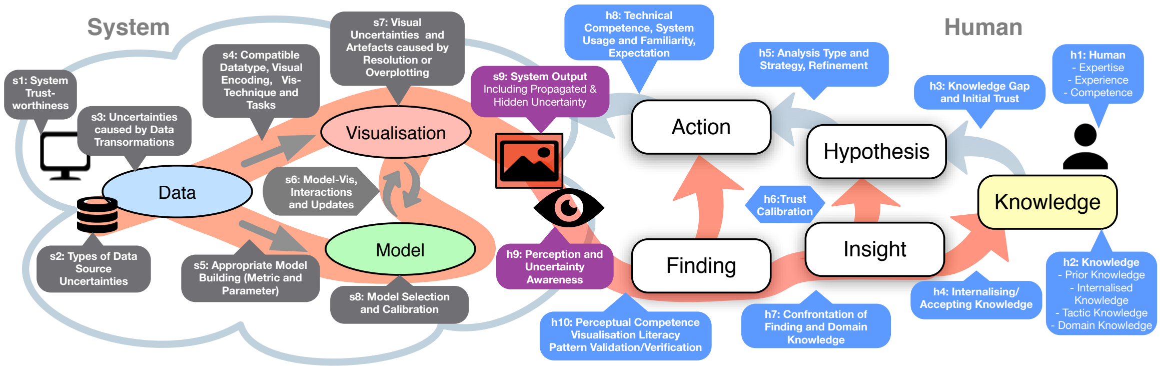



Figure 3.3: Each system component may change the data and consequently introduce additional uncertainty. Human trust building within knowledge generation processes is affected by many human factors. The relation between uncertainty and trust is included as the awareness of uncertainties.

### 3.1.3 Uncertainty Propagation and Trust Building within the Knowledge Generation Process

Within the knowledge generation model for visual analytics, uncertainties are propagated in the different components, causing various issues of trust on the human side. In the following, we give an exposé on how uncertainty is propagated on the machine side and trust is calibrated on the human side. Finally, we discuss the concept of awareness that glues uncertainty and trust together.

#### Uncertainty Propagation on the Machine Side

Several concepts are considered as uncertainty, such as *errors*, *imprecision*, *accuracy*, *lineage*, *subjectivity*, *non-specificity*, or *noise* (Griethe and Schumann [110] provide a more detailed description of all terms), however, in this section, we consider any of these concepts under the umbrella of uncertainty. Source uncertainties (Figure 3.3-s2) are inherent in the data source itself and depend on data collection. We can distinguish authoritative data (stemming from a trusted source) and non-authoritative data (such as social media with a lack of professional gatekeepers and control standards) [96]. Propagated uncertainties are due to data transformation and visualization stages, where uncertainties travel from the data to the visualization components through the system side of the knowledge generation model in order to produce a visual representation that can be used for reasoning (Figures 3.1 and 3.3). Most of the uncertainties have been identified, e.g., in Correa et al. [65] but we further include model building (s5), model-vis coupling (s6), and visualization (s7) uncertainties. In the following, we briefly describe each of these stages with respect to uncertainty.

*Data Processing* (Figure 3.3-s3): Data is transformed in order to prepare it for further processing within the visual analytics pipeline (e.g., data cleansing). Common techniques are sampling, segmentation, normalization, or interpolation. Some of these uncertainties can be measured, e.g., by using statistics [122].

*Model Building* (Figure 3.3-s5): Analysts can apply different configurations or parameterizations to machine learning models or data mining techniques. Uncertainty arises from the complexity of such configurations and the sensitivity of the parameter space. Hence, model calibration introduces additional uncertainty by estimating unknown configurations, parameter values, or distance metrics. Chatfield [55] describes model building uncertainties as caused by model misspecification.

*Model Usage* (Figure 3.3-s8): Chatfield [55] emphasizes that a lack of knowledge about analyzed real-world phenomena may cause inadequacies of the applied model. Furthermore, analysts may choose between alternative models resulting in further uncertainties of using and applying inappropriate models for data and task at hand. Brodlie et al. [39] state that such uncertainties have not yet been investigated in detail within visualization research.

*Visual Mapping* (Figure 3.3-s4): Errors or approximations can also cause uncertainties within the mapping process between data/model results and visual variables that are the basis for generating visual representations of the data. Additionally, the mapping process can introduce further uncertainties if the visualization techniques or visual variables are not suitable for the underlying data types and analysis tasks.

*Visualization* (Figure 3.3-s7): The rendered visualization image can include uncertainties due to clutter, resolution, or contrast effects. Such uncertainties may cause perceptual biases and hinder analysts in spotting patterns and gaining insights causing uncertainties within the human reasoning process (as discussed by Zuk and Carpendale [325] and MacEachren and Ganter [184]).

*Model-Vis Coupling* (Figure 3.3-s6): Uncertainties can also be caused by the coupling between visualizations and models. For example, Endert et al. [87] make use of "semantic interactions" to translate direct manipulation interactions of visual elements to model configurations (e.g., moving objects will adapt the similarity model based on user defined distances). Such technical translations (or mappings) have to fit the user intent. In many cases, visualizations are updated once the final model result is computed, however, it is also possible to visualize partial, incremental, or progressive outputs of the computations (e.g., [95]).

The propagation of uncertainties affects the final system output (Figure 3.3-s9) which will be observed (Figure 3.3-h9) by the analyst to generate knowledge from data. Please note that the selection of uncertainties to account for is highly dependent on the application domain, data, and analysis task at hand.

### Trust Building within Human Analytic Processes

We define *trust* on the human side as a counter part to the machine's *uncertainties* (similar to MacEachran et al.'s [185] distinction between human and machine uncertainties). In the following, we walk through the human concepts of the knowledge generation model [248] and describe them with respect to trust building and highlight influences that are caused by uncertainties. Human trust building can be described as a process of passing and accepting knowledge [50], in our case between human and machine. On the other hand, there are many individual factors that indirectly influence trust building, such as the technical competence [202] and visualization literacy that is dependent on the user's expertise and previous experience with a system.

*Trust Calibration* (Figure 3.3-h6): In each knowledge generation step, users have to calibrate their trust towards their counterpart, the system (or automation as in [69], Figure 3.3-s1), and they also need to calibrate their trust between their own previous knowledge (Figure 3.3-h1), hypotheses (Figure 3.3-h2) and the information that is presented by the system (Figure 3.3-s9, h10, h7). Trust calibration is influenced by all the dimensions mentioned by Muir [202]: The "expectation of the persistence of natural physical laws" allows for the creation and usage of mental models (or rule bases). Further, Muir distinguishes three types of technical competence (expert knowledge, technical facility, and everyday routine performance) that are essential for trust building. Finally, fiduciary responsibility "is invoked as a basis for trust when the trustor's own technical competence is exceeded by the referent's [(in our case the visual analytics system)], or is not known to him" [202]. In visual analytics, this is often the case when complex processing or data mining algorithms are applied but hidden behind the final visual output.

**Knowledge Generation Loop** The knowledge generation loop steers the whole analysis process and consists of knowledge externalization (there is a knowledge gap) and internalization (when sufficient trust has been developed). We start our description with the knowledge generation loop because the analysts' initial trust and hypotheses are based on the prior knowledge and are the foundation for all trust building activities [248].

*Knowledge*: In general, knowledge can be split into many subparts such as domain-, tactic-, data-, system- or experience-based knowledge (Figure 3.3-h2) and has consequently a very individual nature [288, 312]. However, we can distinguish prior knowledge from the knowledge that is gained during analysis and has to be internalized, synthesized and related to the prior knowledge. Within this process, trust develops and pieces of evidence that match or contradict the mental model of the problem are collected and increase or decrease human trust levels. Therefore, trust building depends heavily on the trustworthiness of the machine counterparts (system or data). At the beginning, the prior knowledge is assumed to be valid or verified until the analysis reveals evidence that strengthens or weakens it. Through evidence collection supplemental trust emerges and finally transfers gained information to internalized knowledge (Figure 3.3-h4). Within this process, humans utilize their "private knowledge" in order to judge or interpret pieces of evidence [287, 325]. It is also possible to gain knowledge with analytics, even though the knowledge is based on high uncertainty (if the uncertainty is known and understood as described in [79]). At this stage, we also consider the type of user as an important factor (Figure 3.3-h1). A domain expert will behave differently from a machine learning-expert or a novice user. The relation and former experiences with data analysis systems also play a crucial role in trust building. The claim by Muir that "the trust in a machine, once betrayed, may be hard to recover" [202] is backed up by Manzey's study that revealed a similar relationship between error and subjective trust [193]. Furthermore, knowledge includes many sub-components that influence trust building (e.g., the technical competence or subjective attitudes [202]).

**Verification Loop** This loop describes higher-level trust building and covers confrontation (of information) and human reasoning. If the trust in the combination of all insights related to the hypothesis exceeds a certain amount and integrates with prior knowledge, we leave the verification loop and arrive at new accepted knowledge (by induction).

*Hypothesis*: Hypotheses are derived from prior knowledge (because there is a gap) and are the foundation for each verification and exploration cycle (abduction). Initially, the trust in a hypothesis is derived from prior knowledge and develops during the analysis process by revealing pieces of evidence (insights) that support or contradict them (Figure 3.3-h3). With that respect, humans calibrate their trust and refine their hypothesis in order to come up with an explanation [98]. Also, the type and the initial trust in this hypothesis more or less defines the following analysis type (Figure 3.3-h5) as the verification loop steers the exploration loop [248]. A very vague and open hypothesis that is weakly trusted will originate analysis with an exploratory fashion that solidifies the analysis step by step, whereas a very defined and a highly trusted hypothesis generates a confirmatory analysis. In reality, there are often multiple, conflicting or dependent hypotheses that can be resolved with the detection of a single expected or unexpected insight.

*Insight*: Insights are directly related to hypotheses and can be seen as the pieces of evidence for or against them. The trust in insights relates to the number of similar findings that were produced by the system that support the insight and also on their match to the domain knowledge or the user's mental model of the problem (i.e. what is expected, Figure 3.3-h7). If there is a mismatch between the user's mental model and the gained insight, one of them has to be adjusted. In other words, the user has to decide whether he trusts himself or the information that was obtained using the machine. As described by Green and Maciejewski [108], human higher-level ("System 2") analytical reasoning is able to modify the mental model. Another aspect is that there might be more than one possible interpretation (*insight candidates*) for a finding that can be tested. The analyst develops trust in the alternative interpretations and may (or may not) be able to verify them. However, insights are more likely to occur if they contribute to a plausible narrative (because confirmatory evidence is more likely accepted by humans than dis-confirmatory [109]) and therefore calibrate their trust towards evidences that they are comfortable with.

**Exploration Loop** The exploration loop covers lower-level trust building and evidence search processes through analysis (deductive). Humans stay in the exploration loop until they develop enough trust in their findings and gain insights by applying their domain knowledge.

*Action*: Actions reflect many aspects in trust building (Figure 3.3-h8) as they are the direct interface between human and machine. In general, actions can help the human to develop more trust in the system itself if the user feels in control of the system. On the other hand, hardly operable systems and unexpected behavior (or errors) may result in a general decrease of trust in the system [79]. This trust depends on the human‘s technical competence and familiarity with the system and its sub-components (as described by Muir [202]). Actions further help the human to understand and decrease uncertainties. The ability to explore different kinds of uncertainties enables the user to develop an understanding of these uncertainties, where they arise and how they impact the whole system pipeline (or the sensitivity of data items [65]). Another approach is to actively reduce uncertainties by changing the pipeline or the data (e.g., by choosing more suitable processing methods, mappings or models that introduce fewer uncertainties). Data changes can be done manually by correction or enrichment. In this case, experts are enabled to bring in their expert knowledge in order to change data. However, at this stage users adjusting the data according to their needs are at risk of introducing new human created uncertainties.

*Finding*: As immediate results of an action, users observe a system reaction (Figure 3.3-h9, h10). This reaction (either expected or unexpected) contributes to human trust building [79]. If users develop enough trust in their observations they make them to findings. Or in other words, they stay in the exploration loop until they are able to trust what they see. On a perceptual level, users have to consider being misled by their interpretation of visual elements. For example, a user might spot a visual artifact that is not there in reality (e.g. Muller-Lyon illusion). Furthermore, a finding may include many known and visualized, but also hidden, uncertainties that have been propagated through the system. With this respect, human trust building differentiates when uncertainty information of a finding is communicated and considered [98, 291]. Findings are directly related to insights, which are themselves directly related to gained knowledge. Consequently, uncertainty propagates from its root, the data source, through the system and human reasoning until knowledge. This is illustrated by the red flow lines in the knowledge generation model in Figures 3.1 and 3.3.

### Awareness

We have considered the relationship between trust and uncertainties within the system. Thus far, we assumed that the user is aware (Figure 3.3-h9) of the uncertainties. We now consider the possible effect on trust when the user is unaware of uncertainties and how this might manifest itself in subsequent errors (similar to [274]). In addition, we illustrate the situation that the user mistakenly believes there are no uncertainties when in fact there are, and vice versa. A proposed classification of the different states of awareness and uncertainties is shown in Table 3.1.

We can see that trust it is highest when the user is either aware of no uncertainties or mistakenly believes there are no uncertainties. The latter is a case of over-trust leading to a high chance of errors.

Table 3.1: Awareness classification

| | | system | |
|---|---|---|---|
| | | no uncertainties | uncertainties |
| human | aware | trust = high<br>chance of human error = none | trust = med-low[1]<br>chance of human error = low |
| | mistaken[2] | trust = med-low[1]<br>chance of human error = med-low | (*over*) trust = high<br>chance of human error = high |
| | unaware | (*under*) trust = medium<br>chance of human error = none | trust = medium<br>chance of human error = high |

[1] depends on degree of understanding

[2] 'mistaken awareness' is when the user wrongly believes the opposite, e.g. no uncertainties when in fact there are uncertainties in the system

The lowest trust is when the user is either aware of uncertainties or mistakenly believes there are uncertainties. These are given a value of medium to low as it depends on the user's understanding of the uncertainties, the higher the understanding (or mistaken understanding) the higher the trust. The situation where the user is unaware that there are no uncertainties is a case of under-trust. Making the user aware would increase their confidence and hence trust. In terms of the chance of errors occurring, this is highest when the user is either unaware of uncertainties or wrongly believes that there are no uncertainties, and lowest when the user is aware or unaware that there are no uncertainties.

Whether or not a user becomes aware of uncertainties and indeed takes note of information presented to them, can be influenced by cognitive biases. These, so called, cognitive biases, first introduced by Kahneman and Tversky in the 1970's [143], are deviations in judgment from what rational decision models would predict that occur in particular situations. Importantly, they are involuntary, affect most people to some degree and generally have a negative impact on decision making. For instance, most people have a poor understanding of statistics and instead apply simplifying heuristics to cope with the uncertainty, which leads to irrational decisions. Arnott [13] lists such statistical biases which highlight the inability to comprehend the practical implications of randomness, base rate, sample size, correlations, regression to the mean and probability in many guises. Visual analytic systems allow the user to explore datasets but this relies on the user wanting to seek further information. Unfortunately, confirmation bias is the tendency to ignore information that does not agree with the user preconception or hypothesis [144]. In a recent study, Phillips et al. [219] demonstrate that users of information systems, tend to reduce the perceived usefulness of information that does not reinforce their current premise, which in turn reduces their likelihood to explore the data. Over-confidence and perceived expertise have a similar effect. Completeness bias, where the user perceives that the data is logical and correct, without uncertainties, may also reduce information seeking. We need to be aware of other perceptual and behavioral traits when utilizing visualization and automated systems. For instance, our visual perceptual system is subject to errors due to effects such as contrast, color, clutter and pre-attentive processing. In addition, automation bias can lead the user to over-trust and rely on wrong information that is produced by an automation, overriding their own ability to judge the situation ("looking-but-not-seeing effect"[193]).

As suggested at the start of this section, awareness of uncertainties can reduce errors and increase the user's trust in the data. However, cognitive biases may impede the user's awareness and additionally may lead to poor decisions, especially when the user is in a state of uncertainty. Principally due to the involuntary nature of cognitive biases, reducing their negative effects has proved difficult, even when the user is informed of the possible impact of particular cognitive biases. In the next section, we will enumerate some methods to reduce the impact of cognitive biases, perception effects, and the automation bias.

### 3.1.4 Guidelines, Examples, and Challenges for Handling Uncertainties

We formulate guidelines for handling uncertainties and illustrate them with examples from literature. G1 and G2 are the foundations for uncertainty communication by tracking, quantifying and combining uncertainties. G3, G4, and G6 aim to improve the perception of a systems' trustworthiness through the communication of uncertainty information. G5, G7, and G8 take human issues into account in order to enhance, identify and recalibrate poorly calibrated trust dimensions. Our guidelines have been influenced by Muir‘s recommendations for improving trust calibration [202] (see Section 3.1.2). Additionally, we put forward some extensions and challenges that suggest future research directions.

#### Uncertainties in a System

**G1: Quantify Uncertainties in Each Component:** Our first recommendation is to measure and quantify uncertainties in each pipeline component (shown on the left-hand side of the knowledge generation model in Figure 3.3). *Data Source* uncertainty can be measured based on the capturing process (sensor, completeness, etc.) or further qualitative measures (documented meta-data by the data producer). *Data Processing* uncertainty stemming from data transformations can be handled using probabilistic approaches (e.g., [171]), statistics (e.g., standard deviation, variance, and range), or distance based functions (e,g, comparing inputs and outputs [51]). *Model Building* uncertainties can be reduced by expert knowledge and previous experiences from analyzing similar data [55] (e.g., knowing to include/exclude specific variables). An alternative of avoiding model building uncertainty is to use non-parametric methods (if available) or to apply parameter space or sensitivity analysis. *Model Usage* uncertainty can be, e.g., addressed by using a Bayesian averaging approach with replicated studies [55]. Several works have demonstrated the use of this approach to deal with model uncertainty (e.g., [94, 153]). *Visual Mapping* uncertainties can be mitigated by evaluating the chosen visual variables and metaphors with respect to existing guidelines and systematic categorizations (e.g., [267]). Furthermore, uncertainty itself has to be considered as additional visual variables in the visualization (e.g., [182, 44, 261]). *Visualization* uncertainty may cause errors such as spotting (artificial) artifacts that are not there or over-seeing artifacts (e.g., caused by lower visualization resolution as compared to the data resolution [39]). Consequently, we recommend considering perceptual or contrast effects within the visualizations [197]. We are not aware of existing methods to quantify *Model-Vis Coupling* uncertainties. However, an approach could be to bridge and compare model and visualization measures (e.g., visual 2D compared to high-dimensional [284]). Furthermore, model changes (caused by human interaction or data streaming) cause an iterative re-computation of results. In such scenarios, we recommend to characterize and visualize the uncertainty of incremental results on the fly (e.g., [95]).

**G2: Propagate and Aggregate Uncertainties:** We recommend to provide the analyst with an aggregated and propagated uncertainty measure. Existing approaches, such as the work of Correa et al. [65], quantify and aggregate propagated transformation uncertainties via error modeling and sensitivity analysis. A similar approach to aggregate uncertainties along the visualization pipeline is described by Klir and Wierman [156]. Van de Wel et al. [297] build a weighted uncertainty aggregation measure using entropy measures based on weighted criteria. An alternative approach would be to let the user weigh each kind of uncertainty based on their relevance to the analysis task at hand.

**G3: Visualize Uncertainty Information:** Uncertainty visualization is an effective medium for analyzing source and propagated uncertainties. Griethe and Schumann [110] present an uncertainty visualization pipeline differentiating between four different kinds of data flows (data transformation process, uncertainty acquisition, dependencies between visualization and uncertainty, parameterization of the pipeline). Furthermore, different design principles for visualizing uncertainties have to be

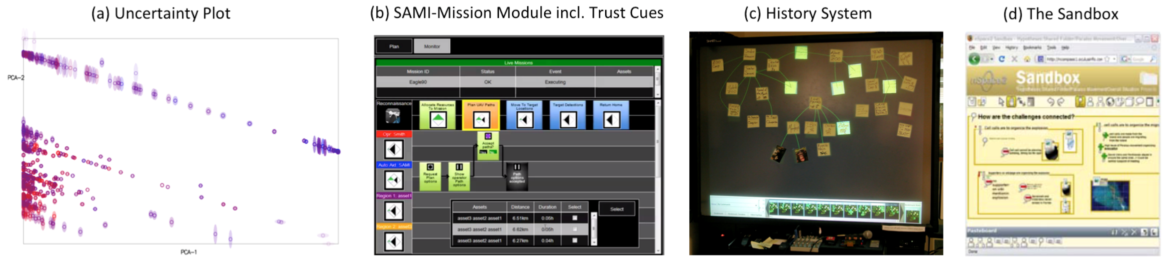



Figure 3.4: Examples for uncertainty aware trust building from different domains: (a) An uncertainty projection to explore how data items are affected by uncertainties [65], (b) decision support including trust cue design from [69], (c) deriving important user notes based on user tracking [155], (d) integrating evidences for computer assisted knowledge management [58].

considered (e.g., the work of Pang [212] focuses on multi-dimensional uncertainties, Griethe and Schumann [110] consider a user's experience and visualization principles, Cai and Lin [45] suggest systems to report self-confidence via cognitive cues). These uncertainty visualizations aim to support users in adjusting their trust appropriately.

**G4: Enable Interactive Uncertainty Exploration:** We recommend enabling the analyst to explore visualizations for the different uncertainties stemming from all components of the system in order to enrich the analysts‘ understanding of the data and how it is affected by different uncertainties. Illusion type cognitive biases, such as clustering and correlation, can be mitigated by providing the analyst the ability to use a variety of visualizations and perspectives onto the data. This can be achieved by enabling the analyst to interactively control the importance of different uncertainties and to explore how they affect the final output. The example by Correa at al. [65] describes such an approach that allows the analyst to explore and understand uncertainties (Figure 3.4-a).

### Supporting Human Factors

**G5: Make the Systems Functions Accessible:** Accessible, intuitive and easy to use interaction techniques will increase the technical competence of the analyst and consequently enhance human trust building [202]. In this respect, different user groups have to be considered. Expert and power-users of an analysis tool will need different interaction possibilities and guidance than novice users. For example, with visual analytics tools, we often observe users having problems with model steering interactions such as parameter setting [248]. In this case, switching between expert or learning mode is the first step in that direction. Endert et al. give a nice example of "semantic interaction" [84] where direct manipulation interactions are directly translated to model steering interactions. Furthermore, Chuang et al. [61] provide guidelines for designing trustworthy and understandable visual analytics systems. Their recommendations can verify modeling decisions and provide model interactions during analysis.

**G6: Support the Analyst in Uncertainty Aware Sensemaking:** Human sensemaking can be supported by offering note-taking or knowledge management components connected to the systems, where humans can externalize and organize their thoughts in order to bridge data and knowledge management [248] (see Figure 3.4-d, e.g., the Sandbox for Analysis [58, 316]). Our recommendation is to transfer and visualize uncertainty information to the findings that are imported from the analysis part in order to support humans in calibrating their trust in the findings' trustworthiness [69]. We can imagine that a system will automatically take care of relations between (conflicting) hypotheses, findings, insights and take the role of an unbiased counterpart to the human by including uncertainty information at any stage [109]. A system could, for example, calculate aggregated uncertainty scores for pieces of evidence that have been grouped by the user. In addition, evidence connected to hypotheses and insights may be explicitly marked by the user as *trusted* or *unknown* (or intermediate value). This would be a form of trust annotation that can be matched to uncertainty measures. Furthermore, humans can integrate external evidence from other systems or their own knowledge that might complete the big picture of the analysis. Utilizing all the connected information enables a system to offer uncertainty and trust cues, e.g., as glyphs connected to the items (such as traffic lights, radar charts or avatars as described in [69]). Figure 3.4-b illustrates an example view of automation blocks that are enriched with specially designed glyphs that serve as trust cues. We will describe such a note-taking component in the second part of this chapter (Chapter 3.2).

**G7: Analyze Human Behavior in order to Derive Hints on Problems and Biases:** Tracking human behavior can be beneficial in deriving hints on the users of a system. We therefore recommend leveraging analytic provenance approaches suggested by Nguyen et al. [205]. Low-level interaction tracking can be used to predict users performance [41] or infer the user frustration [117]. These methods could be enhanced for predicting a users trust level. Closer measures related to uncertainties and trust building can be captured by the rate of overall decision switching. Goddard et al. [100] measured automation bias by noting correct to incorrect prescription switching. Furthermore, Klemmer et al. were able to detect important notes or items based on user tracking [155] (see Figure 3.4-c). These methods could be leveraged by a system to automatically suggest alternative visualizations or items that have not been utilized. The latter may be useful in mitigating some selection based cognitive biases such as confirmation bias [161]. Another approach to derive human trust measures is to analyze user generated contents. A system could automatically seek for signal words such as "unsure, uncertain, maybe, perhaps ...". Zhou et al. describe 27 language features grouped as: quantity, informality, expressivity, affect, uncertainty, nonimmediacy, diversity, specificity, and complexity [323]. Also, Tenbrink [285] investigated how to derive cognitive analytical processes based on language data. Physical or other human sensors such as eye-tracking can also be used. Kurzhals et al. give an overview on the potential for eye tracking in visual analytics [163]. Furthermore, user analysis may be used during system development and evaluation. Scholtz describes five evaluation areas: Situation awareness, collaboration, interaction, creativity, and utility [253]. We imagine protocol analysis [90] as a useful method to interpret "think aloud" evaluations. User interviews using trust questionnaires could also be conducted [291] in order to investigate the relationship between uncertainty and trust for system evaluations. In addition, Bass et al. propose a method to analyze and predict a humans understanding of automation [20]. Chapter 3.2 will describe an approach that captures user interactions within different components of visual analytics systems.

**G8: Enable Analysts to Track and Review their Analysis:** This guideline points to post-analysis activities as a method to detect and mitigate biases. During analysis, users often focus on searching potential evidence without considering alternatives, errors or uncertainties. In addition, users in their "work flow" should not be interrupted [130]. Therefore, we recommend that the analyst is able (or even encouraged) to look and think about his analysis afterward, without interruption during the analysis. In our opinion, this is a better way than warning users during their analysis (e.g., by popup dialogs) as recent studies show that too often warnings may lead to the opposite [6]. Support to mitigate statistical biases (see Section 3.1.3) should be provided, such as presenting the user with base rate information (e.g., typical distribution), estimating realistic probabilities or indicating that a particular 'behavior' is expected rather than a special case, as with regression to the mean. Structured analytic techniques such as a devil's advocate may also be ways that help the user to detect problems and lessen the impact of confirmation bias in particular. Furthermore, analysis process visualization enables involving other users and story telling. Provenance systems such as CzSaw [142], but also systems including story telling [160], are a good starting point in that direction. The note-taking and interaction capturing component that will be described in Chapter 3.2 supports the analyst in the reflection and verification of the analytical process.

### 3.1.5 Discussion, Limitation, and Conclusion

In this section, we discuss our findings, provide limitations and open questions of our study, and conclude with some takeaway messages.

#### Discussion

Our process model shows that human trust building under uncertainty is an extremely complicated, individual process and is influenced by many factors directly as well as indirectly. Furthermore, users informed with uncertainty information can avoid falling into traps concerning mistaken uncertainties and unaware uncertainties. Readers also need to note that the process model has to be tailored to concrete, individual cases where the scope of uncertainties, users, and their tasks are known. The core value of our process model is to provide a balanced view on the role of uncertainty propagation and trust building in visual analytics by considering human and machine aspects together.

The guidelines will be useful to estimate the dynamics of uncertainties in developing visual analytics applications. The terms and structure we outlined in Section 3.1.3 provide an overview of uncertainty propagation, both from the source data and from algorithmic manipulation. With this structure, practitioners and researchers can attempt to quantify uncertainties through the process of data transformations. Depending on its use, this quantification will help users determine effective visualization techniques by thinking of the trade-offs between gaining insights and showing uncertainties.

The process model we provide in Section 3.1.3 can be used to educate users of visual analytics applications about uncertainties and their impact, so that they might reduce errors (e.g., cognitive biases) and build trust whilst analyzing data. We recommend system developers to provide a simple tutorial of their visual analytics applications using some usage scenarios. In addition to assisting users, the material itself provides a groundwork for the education of uncertainties and trust building for designers, practitioners, and researchers. We believe that the impact of uncertainties will decrease as users gain awareness.

There are many implications of this process model. As explained, it is necessary for us to capture human's perceived uncertainties and trust levels at a given moment of analysis. One way is to intervene in the analysis process by asking the users to input this directly, which will ensure accurate estimates of their current status. However, to avoid interruptions it would be useful to compute this automatically. This may be possible through tracing and interpreting usage logs to estimate the level of trust. Data provenance may be an effective method to track uncertainty propagation that enables us to increase uncertainty awareness. On the other hand, if analytic provenance methods are used to infer human measures this may give hints on trust building processes. Combining measures/methods from both sides has the potential to identify relations between uncertainty propagation and human trust building.

Our contribution is to categorize types of uncertainties, awareness of them, and human trust building process. However, there are many external factors that can influence individual processes. For example, our model assumes a single analyst perspective, which simplifies the knowledge generation process. In the real world, many interdependent knowledge generation loops run in parallel and often conflict each other, which can result in uncertain outcomes. Furthermore, taking into account collaboration between human analysts would extend the process model to explain the dynamics of real world scenarios with a team of analysts.

#### Limitations and Open Questions

The scope of this study provides a conceptual process model of unpacking uncertainty propagation within visual analytics processes as well as discovering the human trust building process. Here we provide limitations of our approach as well as open questions that future researchers can investigate.

Uncertainties are difficult to be quantified and categorized into a single process. In visual analytics systems, uncertainties can be propagated and implied through the pipelines, as we discussed. Thus, the combination of uncertainties from multiple sources could be larger than the sum. Our process model does not provide a quantified model of such intertwined process of uncertainty propagation just yet. As outlined in G1 and G2, some efforts have been made to quantify and aggregate different subsets of uncertainty propagation within visual analytics process. Researchers may need to integrate such efforts using our overarching process model and predict such uncertainty propagation in a specific context.

Another open question is whether the transparency of uncertainty propagation is always good and how much of it is beneficial to users. Our process model builds upon an assumption that making the uncertainty propagation transparent will let users be aware of variation in their outcomes. However, providing too much information could confuse, overwhelm, and mislead users, thereby making unwanted human errors. Furthermore, it is also a trade-off between efficiency and accuracy. For instance, applications for human safety, where uncertainty can result in catastrophic results, may need to consider as much transparency as possible. On the other hand, some business analytics may require fast and reasonable analysis results. Thus, it will be interesting to investigate what are proper amounts and methods to communicate uncertainty information to different groups of visual analytics users.

In line with previous points, it is also an open question whether the awareness of uncertainties leads to increasing or decreasing trust. This question may be from the human's trust building process. To build trust in visual analytics outcomes, users may need to build trust in the visual analytics system first. In this process, the awareness of uncertainties may lead to increasing the awareness of visual analytics process but not to increasing trust in the outcomes. Thus, future research may investigate the sophisticated process of human's trust building steps under uncertainty.

In this regard, we may think of the awareness provenance to verify human's understanding. We introduced the concept of awareness to bridge between machine's uncertainties and human's trust. The awareness again is highly subjective to individuals like the trust level, so it will be difficult to quantify the information. Nonetheless, the awareness indeed affects the entire process, so we call for research into capturing it.

These points above do not capture all limitations and open question from our study but will be an interesting start for future work.

### 3.1.6 Conclusion

In conclusion, we have illustrated how uncertainties arise, propagate and impact human knowledge generation processes by relating the concepts of trust, calibration, and awareness. Further, we have given hints on misconfigurations of uncertainty awareness that may cause human errors in data analysis. We provide guidelines that describe various ways to handle uncertainties and to include human factors in order to enhance human trust calibration. Finally, we put forward open research areas that will contribute to more reliable knowledge generation in visual analytics in the future. The following section will describe a note-taking and capturing system that is inspired by the presented uncertainty and trust building process model, the proposed guidelines, and the open research questions.

## 3.2 Investigating Analytic Behavior and Trust Building

Visual Analytics (VA) is a collaborative process between human and computer, where analysts are performing numerous interactions and reasoning activities. This section presents our current progress in developing a note-taking environment (NTE) that can be plugged to any VA system. The NTE supports the analysis process on the one hand and captures user interactions on the other hand. Our aim is to integrate human lower- (exploration) with higher- (verification) level analytic processes and to investigate those together related to further human factors, such as trust building. We conducted a user study to collect and investigate analytic provenance data. Our early results reveal that analysis strategies and trust building are very individual. However, we were able to identify significant correlations between trust levels and interactions of particular participants.

### 3.2.1 Introduction

VA allows analysts to generate knowledge from data through visual interactive interfaces. During this process, analysts have to perform numerous lower-level interactions (with the VA system) and higher-level reasoning activities in order to arrive at the desired and verified pieces of information [248]. In order to validate the different pieces of gathered information, these facts have to be reviewed, related, organized, and combined with the analysts prior knowledge and assumptions (also described as "connecting the dots" [269]). However, we know little about all the human factors affecting this entire process. Human analytic activities may be very unstructured and unorganized due to interruptions or unexpected events, such as spotting unforeseen patterns in the data. To cope with these issues, many systems allow analysts to bookmark, annotate, and organize interesting visualizations (e.g., [316]). In addition, approaches for capturing the human analytic process (by means of interaction logging) have emerged, enabling researchers and analysts to review their analysis processes and to retrieve (or "jump back" to) specific states/visualizations [317]. However, relating human lower and higher-level activities remains an open research challenge. Recent research identifies very individual human factors, such as trust building and the awareness of uncertainties [246] (see Section 3.1). In fact, these factors are hard to measure and to investigate.

This section presents our work in progress for combining and analyzing these diverse aspects together. Our approach is based on a note-taking environment (NTE) that can be connected to VA systems, supporting analysts in organizing their findings and externalizing their thoughts. The NTE offers interfaces for external interaction capturing (e.g., for VA tools) with the aim to combine provenance information of different levels. In addition, the NTE enables analysts to apply trust ratings or textual input as a third source of captured information. We conducted a user study to capture all the different kinds of data in order to create an initial dataset to be investigated. Our results reveal different analysis behavior among the individual participants between exploration and verification activities. Furthermore, we found a significant positive correlation between local and global trust ratings for a subgroup of participants. However, we did not discover any positive correlation between trust and the analysis efforts (on an exploration, verification or total measure). In contrast, two exceptional cases show a significant negative correlation.

### 3.2.2 Related Works

Many theoretical works on human thinking, reasoning, and sensemaking during data analysis exist. Pirolli and Card describe this process as foraging and sensemaking loops [221] iteratively traversing several analysis stages. Sacha et al. [248] propose that the knowledge generation process is assembled by three loops (*exploration*, *verification*, and *knowledge generation*). More recently, Sacha et al. [246] describe the role of uncertainties, their awareness, and human trust building within this process (see Section 3.1). They propose guidelines for supporting human factors, such as 1) supporting analysts in uncertainty aware sensemaking, 2) enabling analysts to review the analysis process, or 3) to analyze human behavior in order to derive hints on problems (see Section 3.1.4–G6–8). Interaction categorizations (e.g., [104], [38]) offer a useful foundation for capturing human analytic processes. Nguyen et al. [205] survey several emerging approaches for analytic provenance and highlight that there are different stages of capturing, visualizing and utilizing provenance information.

We found several related systems in our literature review. Examples are Jigsaw's tablet view [177], the Sandbox for analysis [316], HARVEST [269], Aruvi [270], or VisTrails [251]. Typical components are the capturing of visualization states, history visualization, and the management of bookmarks, which is usually represented as a graph composed of entities, images, annotations and connections (from now on called a "knowledge graph"). Despite the fact that these tools share the same goal of supporting the analysis, we are able to identify some differences. Some focus on note-taking capabilities (e.g., Sandbox [316]), whereas others focus on capturing system states (e.g., VisTrails [251]), or leverage the captured interactions for ranking or organizing bookmarks automatically (e.g., HARVEST [269]).

Aruvi [269] and Jigsaw [147] have been used to study analytic behavior. Additionally, other user studies investigated further human factors. For example, the works by Harrison et al. [117] investigates user frustration and interaction. Dzindolet et al. [79] analyzed trust development between humans and automated decision aids. Observing errors caused a decrease in trust towards the system unless an explanation was provided. However, understanding the uncertainties caused in turn increasing trust towards the decision aid, even under uncertainty. Uggirala et al. [291] tried to find a way to measure trust in complex and dynamic systems and showed that an increase in uncertainty caused a decrease in trust towards the used system. Bass et al. [20] investigated human judgment and proposed a method to measure and predict a humans understanding of automation. They offer a trust questionnaire for distinguishing high- from low-trusting participants. These studies are interesting illustrative examples for investigating further human factors that haven't been focused in the VA community so far.

In summary, we found very inspiring works on supporting, capturing, and analyzing analytic processes. However, we are not aware of a system that tracks interactions beyond system borders (either exploration or verification), enables knowledge management (by means of note-taking) enriched with further capabilities to gather human inputs (such as trust ratings), with the goal to analyze these aspects together.

### 3.2.3 Note-Taking and Capturing Approach

The foundation of our approach is the capability to build a knowledge graph composed of gathered information and human assumptions. The note-taking capability (verification) is smoothly integrated with the actual analysis within a VA system (exploration). Figure 3.5 shows a knowledge graph that has been built by a soccer analyst. At the very first, the analyst defined a hypothesis widget in the NTE interface (Figure 3.5-a). In our example, he had watched the game and assumed that "the red team mostly attacked via the right wing". In order to prove his assumption, he switched to the soccer analysis tool and created a heat map for all ball movements of red team attacks. Subsequently, he imported the bookmarked finding including annotations to the NTE (Figure 3.5-b). The analyst marked the finding as a verifying piece of evidence (assigning a "falsifying" or "neutral" tag is also possible). However, by looking at the peaks within the heat map, he found out that the peaks may be caused by standard situations (e.g., free-kicks) where the ball is not moving. Consequently, this lowered the trust in this finding. Therefore, the soccer analyst adjusted the trust rating slider (value range 1-7). As a second step, the analyst visualized all right wing attacks, imported again the bookmarked visualization and applied again a note, a trust rating, and marked this finding as verifying (Figure 3.5-c). In the following, the analyst started to seek for counter evidence. He explored the left wing attacks and added them to the knowledge graph (Figure 3.5-d). Subsequently, the analyst produced a finding that clearly illustrates attacks for both wings (Figure 3.5-e). Finally, the analyst had collected enough pieces of evidence for rejecting his hypothesis. The evidence bar placed at the hypothesis supports

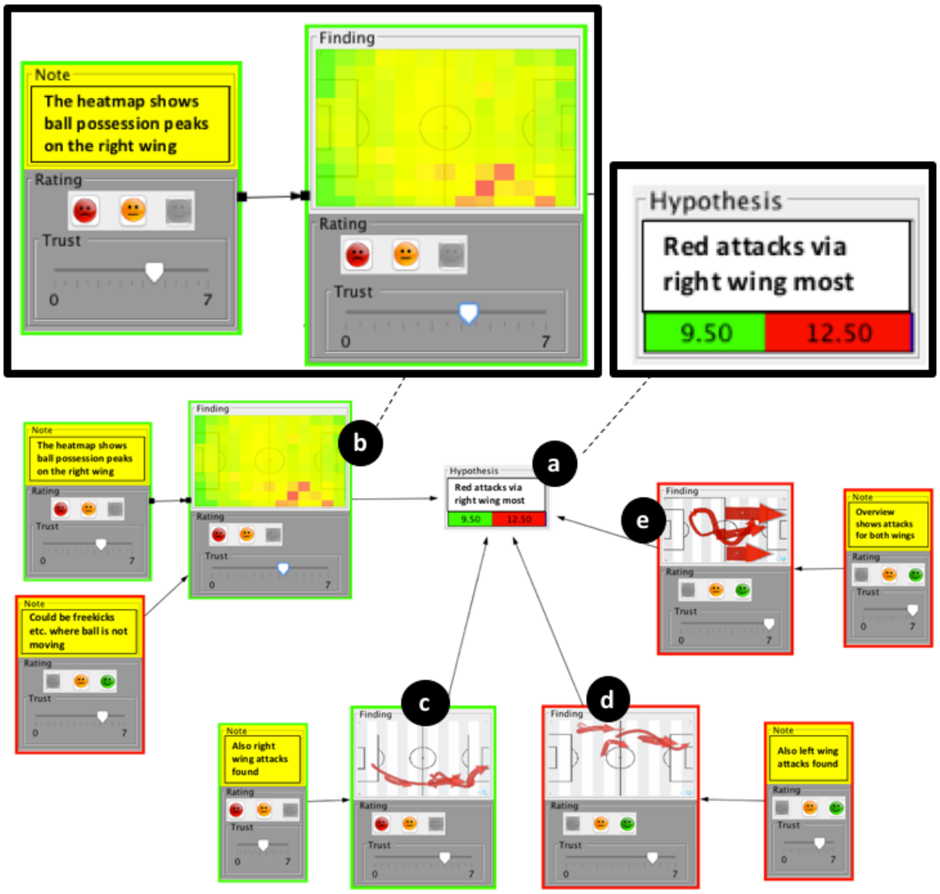



Figure 3.5: Knowledge graph that has been built during the analysis process. Verifying as well as falsifying findings for confirming or rejecting the hypothesis have been collected. Additionally, the analyst added notes and trust ratings to the elements. The evidence bar at the hypothesis widget indicates that the hypothesis is rejected.

this conclusion (Figure 3.5-a) by aggregating the trust inputs with verifying or falsifying information. During this process, the analyst switched between the two provided tools, revisiting and refining bookmarked visualizations.

This simple example illustrates that our approach smoothly supports and integrates the knowledge generation process. The NTE design is based on the knowledge generation model for VA [248] (Chapter 2.1) and offers different widget types according to the concepts: *Hypotheses*, *Actions*, *Findings*, and *Insights* which are part of the NTEs data model. *Actions* are captured automatically (in the NTE and the VA system) and additional *Notes* are created by the analyst. Note that *Findings* are the bookmarks that are imported from an external VA tool (also importing external images as *Findings* is possible). Beyond, the NTE incorporates functionality for externalizing and supporting the analysts' trust building process by means of the guidelines (on human factors - G6, G7, and G8) provided in Section 3.1.4 and [246]. First, it enables the analyst to apply trust ratings and annotations to the graph elements. Second, it is capable to capture and visualize further measures that belong to a finding (e.g., uncertainty measures) and provides visual cues by aggregating them (evidence bar-Figure 3.5-a). Further, it is possible to map the measures (e.g., trust or uncertainty) to the elements of the knowledge graph (encoded with transparency). Third, it enables the analyst to review her/his analysis processes and to "jump back" to the specific system state once a finding in the NTE is clicked. Additionally, interaction sequence visualizations are provided (e.g., one line in Figure 3.6). Furthermore, it is possible to save and load the knowledge graph and captured interactions.

The NTE captures human behavior on different levels and spaces. On the one hand, all the exploration interactions in the VA system are captured (according to [38]) and low-level operations (e.g., mouse clicks or moves) are counted. On the other hand, the NTE captures all the verification interactions in the knowledge graph and allows analysts to provide further inputs (such as the trust ratings). As a third source of information, we consider individual aspects that can be captured by a questionnaire before and after the analysis process. This information provides hints on the different user characteristics (e.g., their experiences and attitude towards automated analysis systems [20]). In sum, our capturing approach enables us to investigate different factors of the knowledge generation process.

Our process is realized as a note-taking component (written in JAVA) providing an application programming interface (API) for integrating external VA systems. The API offers two different interfaces. One interface allows sending bookmarks (visualization images) including additional information (annotations, measures, or a callback function) to the NTE. The other interface provides an interaction logger, that offers different logging levels and types. Implementing the desired interaction logging and callback functions have to be done by the VA system developer. However, in this way the NTE is completely independent of the VA system. More technical details about the NTE (implementation, API, and user interface description) can be obtained from the supplemental materials of [240].

### 3.2.4 Experiment

We conducted a user study to prove our concept and to create an initial captured data set for investigation. We wanted to observe how the NTE is used and to gather feedback about possible areas of improvement. Our main goal was to measure the amount of actions per phase (exploration and verification) and to capture trust ratings on a global level (per task) and on a local level (per finding). Therefore, we formulated the following research questions (RQ):

**RQ1:** "Is it possible to identify different user groups based on their analysis strategies?" (e.g., performed analysis efforts between exploration and verification phase).

**RQ2:** "Is there a positive correlation between trust in findings and the overall trust in the system?" (e.g., a low trusted finding causes a general decrease of trust towards the system as a whole).

**RQ3:** "Is there a positive correlation between interaction activity and trust in findings?" (e.g., the more analysis effort is spent, the more trust should be built by the analyst).

**Design and Procedure:** *Participants:* We recruited 9 participants from the local student population (4 female, 5 male, age 23-29 (median: 23)). All participants reported normal or corrected to normal vision, had mixed experiences with soccer and only little background in using data analysis systems.

*Apparatus:* The study was conducted in a lab setting using two 24 inch screens. The only input device was a common computer mouse and a keyboard for textual input. The participants were seated approximately 50 cm away from the screen. The experimenter was present during the study for answering questions and introducing the study procedure. For recording the user input we plugged our NTE component to a soccer analysis tool [135, 238] (see Chapter 2.2) and implemented the API for importing the bookmarks and interaction logging.

*Task and Procedure:* We defined six analysis tasks that had to be solved using the soccer analysis tool. Each task comprised a given hypothesis (similar to the example at the beginning of Section 3.2.3) and soccer data with which the hypothesis had to be proven or rejected. After introducing the two systems and our intention of trust, participants had to work on all six tasks independently. The tasks

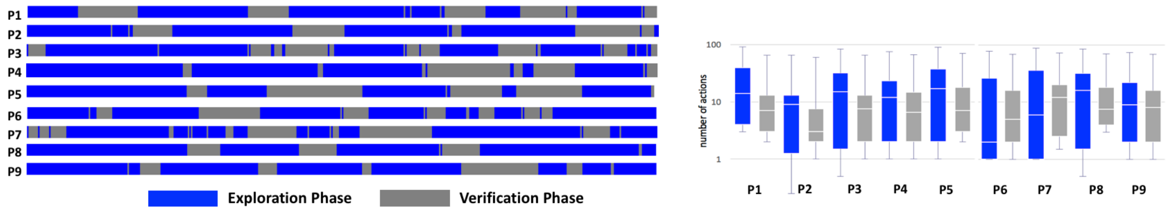



Figure 3.6: Captured analysis phases of the participants solving the same tasks. Left: Phases are visualized for task 1, Right: Boxplots for the number of actions per phase (logarithmic scale) are shown for the whole user study (all tasks).

were ordered according to their difficulty. Between two tasks participants were told to take a short break and assign a trust rating to their answer, as well as to the overall system. In order to enforce/influence a trust variation, we told the participants after the fourth task that the soccer analysis tool might not work as accurate as expected due to interpolation operations. Exploration and verification interactions have been captured and counted for the particular phases. Additionally, participants had to answer a questionnaire (designed according to Bass et al. [20]) at the end of the study.

**Results:** For the analysis of RQ1, we visualized the amounts of captured interactions per phase (exploration and verification). For the investigation of RQ2 and RQ3, we analyzed the captured trust values to calculate the Pearson correlation index and report only on significant results.

*RQ1-Exploration and Verification:* Figure 3.6 shows the beginning of the analysis process of the participants (task 1). The visualization reveals that there are different analysis strategies. For example, P8 and P4 have very long exploration phases interrupted by brief verification phases. They started to collect and refine several findings before they switched to longer verification phases. In contrast, P7 and P3 directly put more efforts in verifying their findings. These user characteristics are also reflected in the respective derived phase measures (for all tasks), as shown on the right-hand side of Figure 3.6. In general, exploration phases tend to be longer than verification phases. In contrast to the other participants, P7 and P6 invest fewer efforts in exploration and consequently start verifying earlier. This characteristic may be identified automatically by measuring the ratio between exploration and verification efforts. Since we identified different analysis strategies we are able to validate **RQ1**. An interesting observation is that the majority of participants only searched for findings verifying the given hypothesis (instead of collecting findings to reject them). Even when we raised their awareness on this issue, they did not start seeking for counter evidence.

*RQ2-Global and Local Trust:* In order to investigate our second RQ, we determined the finding with the lowest trust rating per task (local) and compared it to the general trust in the analysis system after solving the task (global). We calculated Pearson correlations for each individual participant. The results showed a significant positive correlation for three participants (P1: $r(4) = .82$, $p = .05$ - P6: $r(4) = .87$, $p = .02$ - P8: $r(4) = .84$, $p = .03$). On the one hand, this means that if the trustworthiness of a finding was declared as low, also the trustworthiness of the general system was determined by this particular trust rating (independent from the presence of highly trusted findings). On the other hand, global trust increased when the trust of the (lowest) finding increased. Further, another group of three participants showed smaller positive correlations but included a slightly higher $p$ value (P4: $r(4) = .79$, $p = .06$ - P5: $r(4) = .63$, $p = .18$ - P9: $r(4) = .78$, $p = .07$). The last group of participants had very low or even negative correlations and a very high $p$ value, indicating that there was no relation between local and global trust (P2: $r(4)= .09$, $p = .86$ - P3: $r(4) = .31$, $p = .56$ - P7: $r(4) = -.08$, $p = .88$). In summary, these results showed that trust building is very individual, but we were able to identify

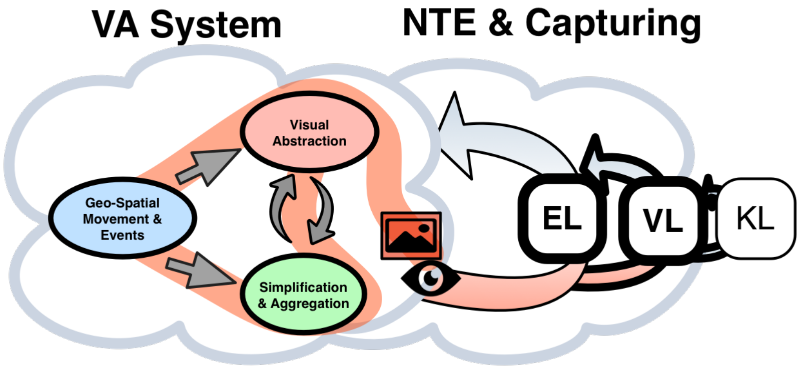



Figure 3.7: Note-taking and analytic provenance component "widens" the capabilities of existing visual analytics systems to support higher level human analytic activities and interaction capturing.

different user groups based on our correlation analysis. Therefore, we are able to verify **RQ2** only partially for a particular sub group (P1, P6, P8).

Interestingly, our hint on potential faults/uncertainties after task 4 did not decrease the system's trustworthiness for each participant. For some participants, the trust value already decreased before or even increased after task 4. A possible explanation could be the diverse background of the participants and previous experiences with the system before task 4 (e.g., user frustration/success).

*RQ3-Trust and Interaction:* We analyzed for each participant if a positive correlation between the assigned trust ratings and the analysis effort exists (for all the findings that have been collected and rated). Therefore, we measured the amount of exploration and verification interactions as well as their sum. These measures have been calculated for each finding and correlations have been calculated for each participant. Among all users, we did not find a general hint on significant positive correlations to the trust value ($p > .07$). Consequently, we are not able to prove **RQ3**. Interestingly, we observe two exceptions. P1 shows a significant negative correlation to the total analysis efforts ($r(15) = -.57$, $p = .02$) and to his explorations ($r(15) = -.52$, $p = .03$). Furthermore, for P3 we found a significant negative correlation for his verification efforts ($r(16) = -.60$, $p = .01$). In these cases, the participants assigned higher trust ratings to findings with fewer analysis efforts (however, on different levels). This could mean that these participants trusted their findings and therefore analyzed less. These results indicate the opposite of our initial assumption (that trusted items are investigated more intensively), however, for a small set of our participants.

*Qualitative Feedback:* After analyzing the answers to the questionnaire the participants stated that they felt on a medium to high level annoyed by adjusting the trust value (range: 4-7, median: 5), although, it was easy to learn how to do it. In addition to the 9 participants, we invited a soccer expert to work with our prototype. He also stated that adjusting the trust value was annoying (7) but useful. He reported that while assigning a trust value he reflected about his own work. Furthermore, he thought that the NTE is a useful extension, as it provides an overview of the current analysis.

### 3.2.5 Conclusion and Future Work

In this chapter, we presented our general approach and early results for integrating, capturing, and analyzing exploration, verification, and trust building activities. Figure 3.7 illustrates how our note-taking and analytic provenance component "widens" capabilities of current visual analytics systems to support and investigate higher-level human analytic activities. We are able to mention interesting findings: 1) Different analysis strategies/behavior can be analyzed and identified (based on exploration and verification capturing), 2) a positive correlation between local and global trust exists (for a particular user group), and 3) there is no significant positive correlation between trust and the amount of interaction (in two exceptional cases there is a *negative* correlation). However, so far we just reported on little first steps of our work in progress and the gathered results are hard to generalize and need to be verified by more focused and extensive investigations (e.g., more participants). In addition, our derived interaction measures could be defined in more detail for specific activities (such as navigation, configuration, recording or annotation) to further distinguish user characteristics with respect to trust building activities. This would also enable us to analyze specific interaction sequences and patterns. Additionally, we found out that especially assigning trust ratings delivers a very subjective measure that has to be analyzed individually (or the trust has to be normalized among participants). Furthermore, we assume that human trust building is highly influenced by the analysis case and the impact of the decisions that have to be made during the analysis process. Therefore, we want to conduct similar studies in other (more critical) domains, such as crime analysis, or crisis management. Our vision is to investigate methods that enable us to identify different user characteristics automatically with the final goal to support the analysts adaptively according to their needs. This chapter has put a sharper focus of human analytical processes to the knowledge generation process. It further demonstrated with the NTE that humans‘ exploration and verification processes can be captured, investigated, and supported. The following chapter will focus on a tighter integration of interactive visualization and machine learning techniques.

# 4

# Visual Interactive Machine Learning

*"If you don't know how to ask the right question, you discover nothing."* – W. Edwards Deming

**Contents**



Machine Learning (ML) offers many methods to analyze data, such as supervised methods to classify data records or unsupervised methods to reveal otherwise hidden data structures (e.g., clustering). However, real world analysis problems often require the ability to explore data and ML results where the default preprocessing, data selections, configurations, and parameterizations have to be adapted to specific domain tasks and problems at hand. This often leads to complex processing pipelines and costly design processes. Interactive visualization can bridge this gap by effectively involving the analyst in the machine learning process. Tightly intertwined solutions are needed to advance in VA. The first part of this chapter approaches this problem with a conceptual process model for human-centered machine learning (Section 4.1) revisiting typical steps of interactive ML pipelines by reviewing existing theoretical frameworks and existing example systems. This process model (Section 4.1) is based on its initial publication in ESANN 2016 [245] and an extended version in Neurocomputing [244]. This extended version includes an extended review of prior work, a more detailed process model, examples of providing automated support for each stage, identification and description of scenarios where analysis and feedback take place, and additional discussion throughout. The second part of this chapter (Section 4.2) describes a concrete VA system for visual interactive cluster analysis that is inspired by this conceptual process model. This section is based on our

VAST paper [243] and describes a system that integrates an iterative ML approach with interactive visualizations and provides user guidance to support the analyst during the analysis process.

## 4.1 Human-Centered Machine Learning By Interactive Visualization

Visual analytics (VA) systems help data analysts solve complex problems interactively, by integrating automated data analysis and mining, such as machine learning (ML) based methods, with interactive visualizations. We propose a conceptual process model that relates human interactions with ML components in the VA process, and that puts the central relationship between automated algorithms and interactive visualizations into sharp focus. The process model is illustrated with several examples, and we further elaborate on the interactive ML process by identifying key scenarios where ML methods are combined with human feedback through interactive visualization. We derive five open research challenges at the intersection of ML and visualization research, whose solution should lead to more effective data analysis.

### 4.1.1 Introduction

Real-world data analysis usually relies heavily on both: automatic processing and human expertise. Data size and complexity often preclude simply looking at all the data, and make machine learning (ML) and other algorithmic approaches attractive, and even inevitable. However, the power of ML cannot be fully exploited without human guidance. It remains a challenge to translate real-world phenomena and analysis tasks, which are often under-specified, into ML problems. It is difficult to choose and apply appropriate methods in diverse application domains and tasks. More importantly, it is crucial to be able to incorporate the knowledge, insight, and feedback of human experts into the analytic process, so that models can be tuned and hypotheses refined.

In a typical setting, domain experts use ML and visualization methods provided by common software tools (*e.g.*, SPSS, R, Tableau) "out of the box". Realistically, the domain experts' proficiency in ML may be limited, and the underlying computations may not be transparent and comprehensive enough to provide the feedback needed to guide model refinement. Visualizations are often used to display the ML model results without offering interactions that trigger recalculations. This results in a very standardized configuration of the ML and visualization pipelines based on default parameters that domain experts may not know how to adapt. The situation may be improved by having domain experts collaborate with data scientists, improving the effectiveness of analysis, but also leading to a much more costly iterative design process. ML researchers usually know how to tune models directly in ML platforms (*e.g.*, Matlab, R, Python) and provide results to domain experts. However, domain experts generally find it necessary to learn how models behave and how to evaluate results to provide useful feedback.

By integrating ML algorithms with interactive visualization, visual analytics (VA) aims at providing visual interfaces for analysts to interact directly with data and models [150]. Tam et al. [283] illustrated in case studies that human-centric ML can produce better results than purely machine-centric methods. In such cases, an analyst is enabled to steer the computation and interact with the model and data through an interactive visual interface. Despite many efforts to date, though, solutions from ML and VA are still not interwoven closely enough to satisfy the needs of many real-world applications [88, 248]. For example, in existing toolkits (such as WEKA, Elki, or javaML), tight integration between interactive visualization and ML process is missing. Most of these tools present modeling results as static visualizations and interactions are often limited to command line interfaces or user interface controls that are not intuitive and accessible to end-users. Toward a better integration of ML and VA, in recent years conceptual frameworks that characterize the interplay between them have been proposed [150, 88, 248, 4]. It appears most frameworks were designed from the perspective of interactive visualization, focusing on the role of the "human in the loop". A closer connection between visualization and common ML paradigms (such as unsupervised and (semi-) supervised learning; classification, regression, clustering, etc.) including specifics of these methods (*e.g.*, SVM vs. random forests in classification) and their implementations are needed. We put a sharper focus on scenarios in which complementary ML and VA methods are combined, and propose a conceptual process model for a tighter relationship between ML and VA. To do so, we identify aspects of automated ML techniques that are amenable to interactive control and illustrate these with examples. We further describe human factors within this process that should be considered carefully in the design of interactive visual ML systems and enumerate analysis scenarios. The proposed conceptual process model opens perspectives on new ways of combining automated and interactive methods, which will lead to better integrated, and, ultimately, more effective data analysis systems.

Researchers in both ML and visualization have realized for some time that closer collaboration could help to solve this problem. An interdisciplinary team of experts from the ML and visualization communities was formed at a Dagstuhl Seminar on "Bridging Information Visualization with Machine Learning" [152]. The process model proposed in this study is the outcome of several iterations of discussions, feedback, and framework refinements made by this team.

The initial version of this process model [152] was based on a survey of several earlier frameworks and systems combining ML and interactive visualization. Subsequently, the process model was refined by applying it to a larger set of example applications (identified in the visualization, ML, and HCI literature) and by incorporating external feedback from experts, such as conference submission reviews [244, 245]. This led us to a refinement process, carried out over 1.5 years, including extensions and simplifications, validation, and evaluation by analyzing existing VA systems and ML techniques.

The rest of this section is structured as follows. Section 4.1.2 discusses related work on the interplay of machine learning and human feedback. Section 4.1.3 introduces our conceptual process model and the key stages in its interactive pipeline, and they are illustrated with examples in Section 4.1.4. Section 4.1.5 examines the human interaction loop in more detail, describing the stages of action and analysis scenarios where interaction occurs. Section 4.1.6 identifies five challenges and associated opportunities in creating systems that fully use the process model. Section 4.1.7 gathers conclusions and final discussions.

### 4.1.2 Related Work

The literature describes related models that capture the interplay between ML system components and human feedback loops. We will discuss several different perspectives on this topic, divided into VA models, interaction taxonomies, interactive ML, and human-centered design. This section concludes with a high-level summary for interested readers without ML expertise.

**Visual Analytics Models** Pipeline-based models such as the *Reference Model for Information Visualization* [48] or the *Knowledge Discovery Process in Databases* (KDD) [91] usually contain feedback loops that cover all the subcomponents with the potential for user interaction. In the standard VA model [150], the analysis process is characterized by interactions between data, visualizations, models of data, and users, for knowledge discovery (see Figure 4.1). ML interaction in this process is aimed at model building and parameter refinement. Sacha et al. extended this model [248] to encompass the process of human knowledge generation (see chapter 2.1). This extended model clarifies the role of humans in knowledge generation, and highlights the importance of supporting a tighter integration of human and machine. Several other models focus on a clear depiction of the human data analysis process, including Pirolli and Card's sensemaking process [221], and Pike et al.'s science of interaction [220]. Endert et al. characterized the interaction process between a human analyst and automated analysis techniques as the "human *is* the loop" [87] and proposed a model for coupling cognition and computation [88]. More recently, Chen and Golan [56] provided an abstract

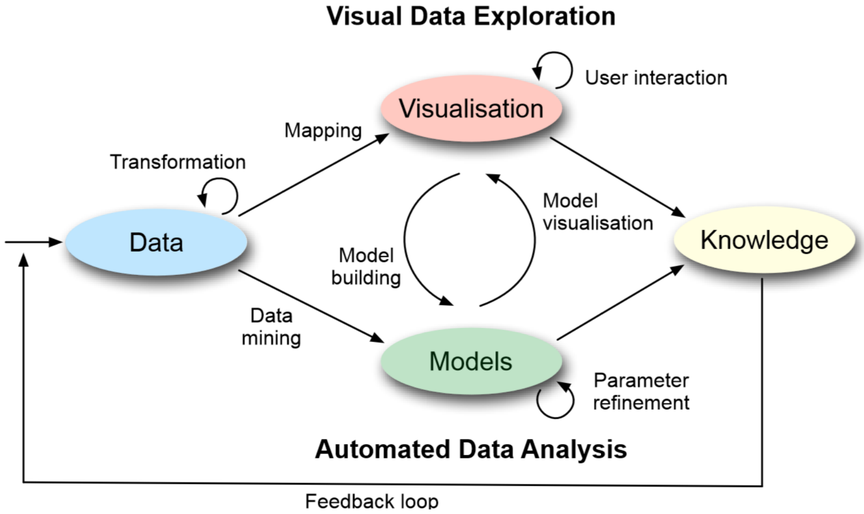



Figure 4.1: Visual analytics process by Keim et al. [150]. ML interactions are related to model building and parameter refinement.

model to describe six classes of human-machine workflows in combination with an information-theoretic measure of cost-benefit. Their model allows one to analyze workflows composed of machine computations and human interactions supported by different "levels" of visualizations. All these models reflect a high-level understanding of system and human concepts.

**Interaction & Task Taxonomies** Another set of models related to our endeavor seek to characterize and organize the tasks and interactions in a visual data analysis process. For example, Brehmer and Munzner [38] propose a comprehensive visualization task taxonomy. However, model interactions only arise in tasks they refer to as "aggregate" or "derive" tasks. Landesberger et al. [305] define a taxonomy that includes interaction and data processing. Their taxonomy provides two types of data processing interactions: data changes, such as editing or selecting data, and processing changes, such as scheme or parameter changes. They incorporate Bertini and Lalanne's [33] distinction of human intervention levels, that distinguishes, for example, between scheme tuning (*e.g.*, parameter refinement) and scheme changing (*e.g.*, changing the model) interactions. Mühlbacher et al. [201] investigate and categorize several types of user involvement for black box algorithms with different characteristics. The characterization of interactions in our process model is orthogonal to these taxonomies and extends them with a dedicated view on interaction with ML components.

**Interactive Machine Learning** While the above models were strongly framed from the viewpoint of visualization and VA, there is also growing interest in the ML community to incorporate human interaction more fully in the analysis and learning processes. A typical scenario would be that a human observes or explores the current state of a learning system, and explicitly or implicitly guides an ongoing training process. This is often referred to as *human-in-the-loop machine learning*, or more broadly, *interactive machine learning*. The classical example is recommender systems that infer users' personal preferences from previous choices. User can provide continuous feedback, such as by recording additional choices, or by explicitly scoring (liking/disliking) individual items [137, 194, 232]. In similar scenarios, the user provides class labels, for which a machine learning classifier is (continuously) trained. This concept is strongly linked to the topic of *active learning* [62, 264], which aims at efficient choices of samples during training epochs to achieve fast convergence of a learning algorithm. An increased demand for involving user feedback is underscored by recent work investigating more

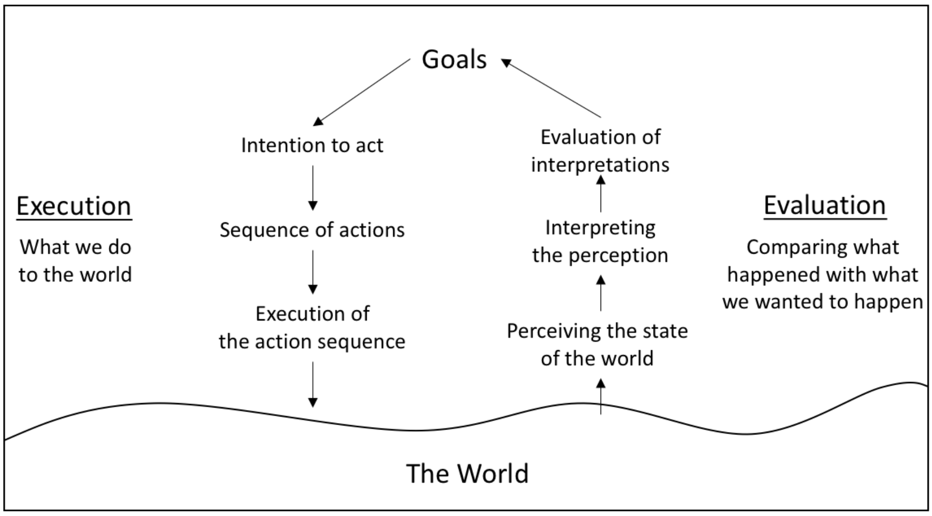



Figure 4.2: Stages of interaction [207].

elaborate user models or more intricate forms of interaction. For example, Amershi et al. [4] propose a set of high-level paradigms by which user involvement in ML may be characterized, similar to models that have been discussed in the VA community [150]. They also stress the importance of accounting for user behavior, and the potential benefits of collaborative research between ML experts and the human-computer interaction community. Similarly, Groce and colleagues investigate sample selection strategies to test classifiers effectively via systematic feedback requests to end users [111].

**Human-Centered Design** Another perspective on the analysis process is provided by the human-computer interaction (HCI) domain. We can adopt commonly known concepts and terms in our interactive ML setting, considering the interplay between human and machine. A famous example is Norman's *Stages of Action* cycle [207], shown in Figure 4.2. At the center of the cycle are the *Goals* that a human analyst wants to achieve. Norman distinguishes between two major stages of an interaction: *Execution* and *Evaluation*. In the *Execution* phase, the human (1) forms an intention to act and (2) specifies a sequence of actions that is (3) finally executed to the world. Subsequently, in the *Evaluation* phase, the state of the world has to be (1) observed, (2) interpreted, and (3) finally compared and evaluated with respect to the initial goals. Norman further describes the distances or "gulfs" between the human goals and the world that need to be bridged when humans interact with (digital) interfaces. The *Gulf of Execution* describes the problem when the human does not know how to perform an action, whereas the *Gulf of Evaluation* indicates that humans are not able to evaluate the result of an action. Human-centered user interface design attempts to bridge these gaps. User interface features need to be visible and offer *Affordances* to the end user. These (perceived) affordances are relationships between a person and a physical/digital object and suggest how the object might be used [207]. *Visual Cues* (*e.g.*, visual elements, icons, or animations that attract attention) may guide the end user during the analysis process. On the one hand, a system should communicate the progress of ongoing computations or the quality of results to the analyst. On the other hand, visual cues may guide the analyst to "handles" or objects that can be manipulated within the interface. In this respect, the concept of *direct manipulation* [266] has been demonstrated to enable intuitive operations to end users. Interactive visualizations of ML model structures and data items allow direct interaction that is more convenient and more easily interpreted than text commands.

**Machine Learning Overview** A wide range of ML algorithms and methods have been proposed and employed in practice. One way to distinguish these methods is based on their learning paradigm, which is either *supervised* (examples of system inputs and desired outputs are both provided), *unsupervised* (no desired outputs are specified), or *semi-supervised* (not all outputs are available, typically only a few). While supervised methods aim to learn the input-output relationship from the provided examples, unsupervised methods attempt to extract hidden structures from them. These learning paradigms can be instantiated into specific categories of ML tasks. *Regression* aims to best predict any form of continuous outputs as a function of the inputs. *Classification* aims to predict class labels or memberships associated with the inputs. On the unsupervised side, *clustering* aims to identify groups or hierarchies present in data. Similarly, *dimensionality reduction* and *manifold learning* both aim to identify linear or nonlinear relationships between the observed variables and to represent the subspace where most of the data variation happens with fewer latent variables. These are a few examples of emblematic ML tasks, among many others, like association rule learning, missing value imputation, time series prediction and outlier detection, novelty detection. In practice, several of these abstract tasks are combined in a data flow to solve real-world problems and analysis. For example, one might apply dimensionality reduction (to mitigate the computational impact of working with high dimensional data) before applying regression or time series prediction. Such combinations of methods, though useful in practice, lead to composite models with heterogeneous parameterization, which are difficult to train, and time-consuming to validate.

In summary, both VA and ML communities have noticed the gaps between automatic ML and human interactions in data analytics systems, which limit their effectiveness in solving real world application problems. Various models to conceptualize the potential integration of ML and interactive visualizations have been proposed. These models, however, still have either a strong human/visualization focus or a strong algorithmic focus. We propose a new conceptual process model that covers both aspects with the objective of providing a more systematic view of how interactive visualization and ML algorithms can be integrated in practice.

### 4.1.3 Human-Centered Machine Learning Process

As shown in Figure 4.3, our conceptual process model unifies, embeds, and extends existing theories on interactive ML and VA by integrating and generalizing observations from emergent case studies and examples. The process model combines typical ML and VA pipeline components (*A–D*) with an analysts' iterative evaluation and refinement process (*E*). An analyst can interact with the individual stages in this pipeline through a visual interface (*D*), which acts as a mediator or "lens" between the human and the ML components (*dashed arrows*). Changes are then sent back to the visual interface and presented to the analyst (*solid arrows*). The dark blue boxes in the figure denote examples of automated methods that support the analyst in performing specific interactions. Our process model illustrates that a multidisciplinary perspective combining ML and VA is needed to provide usable and accessible access to end-users (domain experts). For example, data operations, visualization techniques, and human-computer interaction (blocks *A*, *D*, and *E* in Figure 4.3) are addressed in the visualization community, whereas ML algorithms, setups, and optimization (blocks *B* and *C*) are core to ML research. Next, we detail the interactions involved in each stage of the analysis process and discuss possible automatic support to facilitate these interactions.

**Edits & Enrichment (A)** In ML, data is often seen as fixed or immutable, but many VA tools support data cleaning, wrangling, editing, and enrichment, which is essential in many applications [145]. For example, in a typical active learning scenario in ML, a domain expert may want to incrementally add labels to data while training a classifier in order to inject domain knowledge and improve the

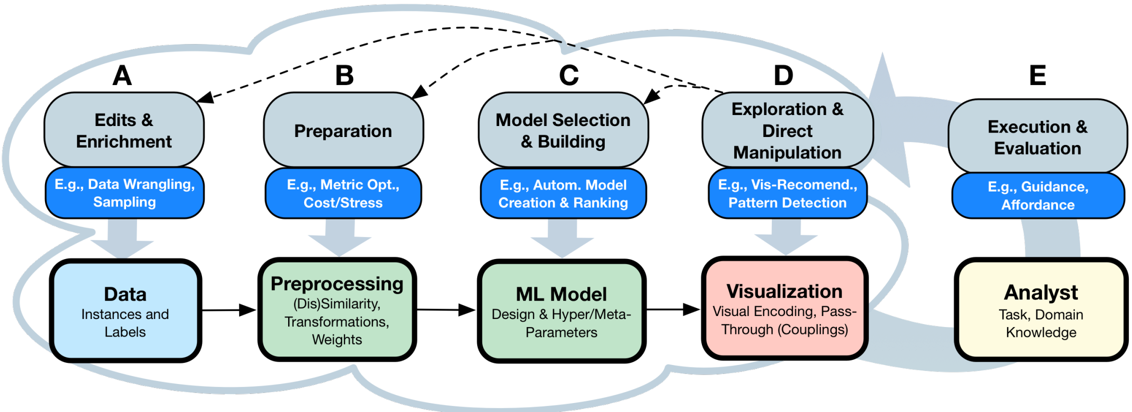



Figure 4.3: Proposed conceptual process model: A reference interactive VA/ML pipeline is shown on the left (*A–D*), complemented by several interaction options (light blue boxes) and exemplary automated methods to support interaction (dark blue boxes). Interactions derive changes to be observed, interpreted, validated, and refined by the analyst (*E*). Visual interfaces (*D*) are the "lens" between ML models and the analyst. Dashed arrows indicate where direct interactions with visualizations must be translated to ML pipeline adaptations. The colors of the pipeline components refer to the ones defined in the VA process model [150] shown in Figure 4.1.

quality of the classifier. Another example of Edits & Enrichment interaction is the testing of "what-if scenarios" on the data. The analyst might want to change or remove some data points and see the effect to test certain assumptions about the data. Data editing is often followed by a "warm restart" of the ML pipeline, iteratively propagating results to the analyst. From an ML perspective, data editing combined with user feedback can be seen as a form of cross-validation/bootstrap. In these techniques, the ML model is re-trained with "modified" data, either data held out in cross-validation and leave-one-out, or changed into some other observed instance that is then reintroduced in the bootstrap sample. However, traditional cross-validation/bootstrap are performed with strict rules about how data are held out or modified, to pursue statistical goals about generalization performance, whereas the editing and feedback discussed here are performed by the user to carry out a task, not constrained to a specific mathematical formalization of the task. Hence data editing might be seen as a kind of "meta" cross-validation, requiring proper quality assessment for the user's task.

*Automatic Support:* Various statistical and ML techniques exist for preconditioning and processing data. These techniques can be implemented in interactive systems for data wrangling, such as missing value detection and replacement, sampling, and data transformation. When datasets are large, the analyst can apply sampling techniques (*e.g.*, vector quantization or hierarchical clustering) to derive a representative subset for interactive analysis. Similar methods can also be used for efficient labeling, for example, by adding an annotated class label to all items in a particular cluster.

**Preparation (B)** Many ML pipelines or VA workflows incorporate preprocessing steps that are selected and adjusted by a-priori domain experts. Being outside of the scope of the central ML model, the design options and parameters in these steps often have a different status. For instance, they may not be subject to cross-validation in some cases. While edit-and-enrich interactions focus on persistent changes to data (possibly individual items), preparation interaction applies a uniform, transient transformation of features to a larger set of observations. Typical preparation activities include transformation of data such as standardization, scaling, Fourier or wavelet transforms, and weightings. Weightings include filtering (0-weights) of data items, as well as feature selection [113]. In this respect, we often observe a gap in the "judgment of (dis)similarity" between a human and the "default" metrics used in ML methods. Analysts usually focus on specific features or subsets within their data. This requires feature weightings or defining more complex (dis)similarity functions.

*Automatic Support:* ML offers several measures and methods to optimize *Preparation*. For example, feature weighting can be supported in the form of relevance [141], metric [252, 43], or kernel learning [228]. Other setups make use of cost functions or stress for optimizing parameterizations of preprocessing steps. Furthermore, other methods such as correlation (e.g., Pearson correlation) or factorial analysis support analysts in understanding feature dependencies within data.

**Model Selection & Building (C)** An essential idea of VA is to enable analysts to interact directly with ML models so they can integrate domain knowledge into the analysis process. In *Model Selection*, analysts choose among various ML algorithm families or a set of pre-built model results. Another possibility is to build ML model ensembles interactively. *Model Building* interactions focus on changing a given ML model through the adjustment of model parameters. While internal model parameters are usually optimized automatically, others, such as design or form, and meta- or hyper-parameters, need to be tuned by the analyst according to their assumptions. Model building interactions can lead to ML model changes that affect its *form*, *constraints*, *quality*, and *accuracy*. *Form* parameters define basic structures (such as the number of nodes in a neural network), whereas *constraints* reflect more detailed assumptions (*e.g.*, defining fixed anchor points in a dimensionality reduction algorithm). In some applications, it is also desirable to tune the *quality* and *accuracy* of the ML result (*e.g.*, by interacting with the confusion matrix of a classifier).

*Automatic Support: Model Selection* can be supported automatically or semi-automatically, for example, with Akaike or Bayes information criteria, cross validation, bootstrap [80], etc. These techniques assess the quality of a model based on its complexity (roughly, the number of free parameters) and the generalization error, which allows different ML models to be compared and ranked. When there are too many models to exhaustively compare all using model selection methods, higher-level *Model Building* can also be supported by automatic methods, especially for metaparameter optimization (i.e., parameters that cannot be tuned by optimization within the model family and where the analyst is not able to provide "useful" feedback). For example, heuristic approaches such as genetic algorithms can be applied to select features and a regularization parameter for support vector machine classifiers based on the quality (cross-validated performance or theoretical performance bounds) of the classification result [97].

**Exploration & Direct Manipulation (D)** The various characteristics, parameters, and results of all pipeline stages can be presented to the analyst as visualizations in a user interface. On the one hand, data can be visualized using a plethora of known visualization techniques. On the other hand, visualizations of ML components can be presented as well. VA aims to combine the two variants by incorporating ML results or patterns (e.g., identified groups, classes, or outliers) into data representations. We found visual representations for different parts of the ML pipeline, such as data and model spaces (Figure 4.4-a), pre-built model variants including their characteristics (Figure 4.4-b) and quality (Figure 4.4-a/b/d), but also the ML structures (Figure 4.4-d). Interactive visualizations that allow for *Direct Manipulation* of visual objects make ML interactions amenable to analysts. Usually, simple *Exploration* interactions, such as changing a graphical encoding, or navigating within views, do not feed back to ML components but help the analyst to understand and interpret the visualization. However, the preceding discussion also mentioned several situations where interactions in visual interfaces are "passed through" to ML changes that trigger a recalculation of the ML pipeline, as indicated by dashed arrows in Figure 4.3. This concept has become known as "semantic interaction" that maps intuitive observation level interactions in a visualization to appropriate ML changes [85].

*Automatic Support:* Automated methods can be used to detect and highlight specific visual patterns, such as class separation, correlation, outliers, or sequences. These methods imitate human perception

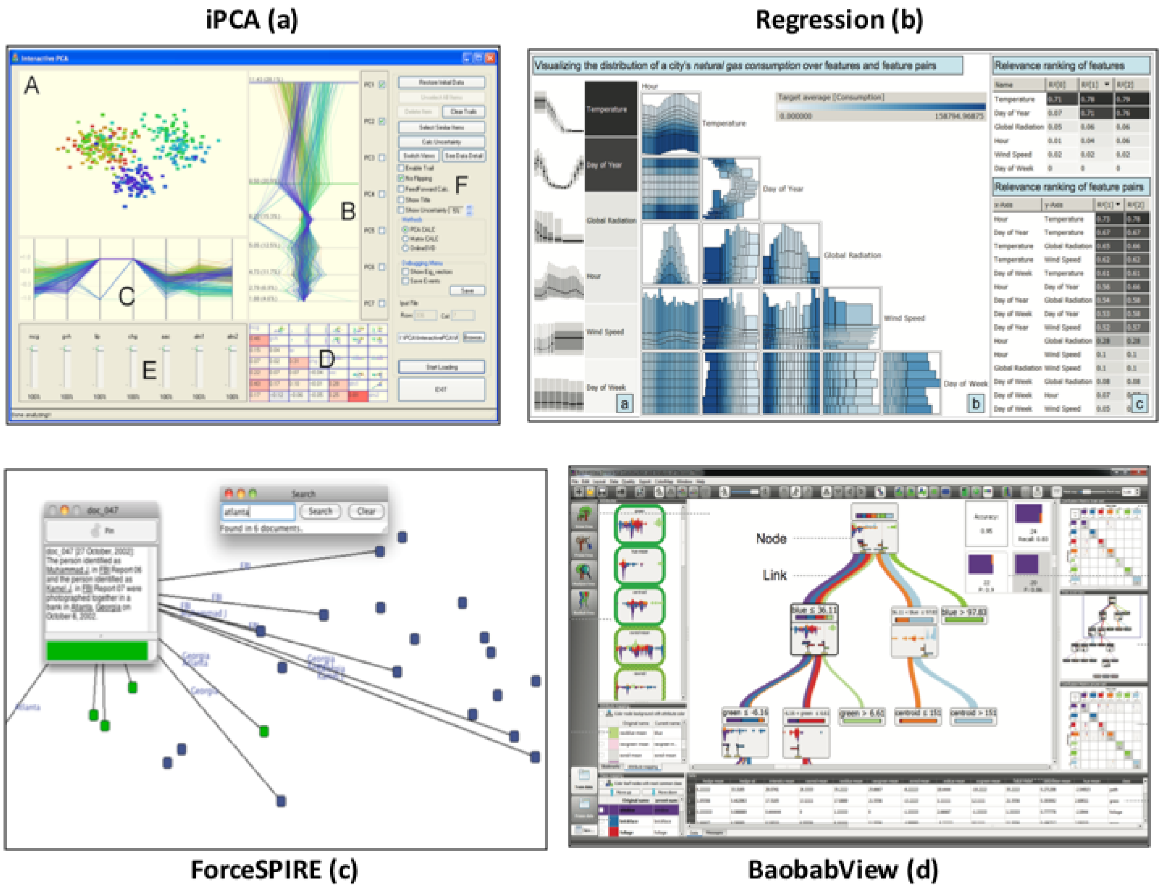



Figure 4.4: A selection of examples that effectively involve analysts into the ML process by interactive visualization. Courtesy of Jeong [138], Mülbacher [200], Endert [85], van den Elzen [293].

with the goal of better helping human analysts find interesting, visible patterns in the data. Aupetit and Sedlmair [16], for instance, provide a rich set of over 2000 measures that automatically detect visual class separation patterns. Similarly, different visualization techniques (e.g., scatterplots, parallel coordinates, or matrices) may be employed to provide different perspectives of data and ML results. In summary, recommending "interesting' visualizations has excellent potential for improving the effectiveness of data analysis.

**Execution & Evaluation (E)** This step involves the entire interactive and iterative analysis process, including all the interaction components mentioned above *A*-*D*. Interactive visualizations (*D*) not only serve as an aid or "lens" that facilitates the process of interpretation and evaluation of ML results but also make the execution of ML interactions amenable to analysts. In an ideal VA system, analysts actively engage in an iterative process of observing, interpreting, and evaluating the system's outputs, followed by subsequent execution of interactions to refine the analysis. This duality of interaction design goals has been characterized by Norman's pioneering work on *Stages of Action* [207] (see Section 4.1.2 for more details). However, the system should actively enable and support this duality by providing usable and interpretable visual interfaces considering human-centered design, such as affordances [207, 66], direct manipulation [266], and interpretable representations. We will provide a more detailed perspective on this human loop in Section 4.1.5 describing an analyst's thinking, sense-making, and reasoning process influenced by various human factors.

*Automatic Support:* Boy et al. [36] investigated visual cues as perceived affordances in a "suggested interactivity" study, with the goal of guiding analysts. Furthermore, the analysis process itself can be recorded (as a sequence of interactions) and visualized to enable browsing through various analysis states, adding analytic provenance capabilities [317]. In this context, the quality of an interaction result can be measured and tracked within such a sequence to automatically distinguish beneficial from detrimental changes. For example, Kapoor et al. measured the accuracy of a classifier before and after interaction [148]. However, such measures are rarely available today, especially in more exploratory or speculative types of analysis.

### 4.1.4 Example Systems

Our process model was inspired by studying current data analysis systems that engage analysts through interactive visualization. In this section, we discuss ForceSPIRE [85] as an example of how interactive visualization can be integrated into each stage of an automatic analysis process. We will also briefly review some other relevant examples, and map their interactions to our process model to show how it covers many different types of interactive visualization, as well as its potential for identifying interactions missing in the analysis process.

#### ForceSPIRE

ForceSPIRE [85] is an interactive text visualization and analysis system. It takes a collection of text documents as input and shows them in a force-directed layout, driven by document similarity (measured by comparing common terms). The analyst can explore documents in this spatial representation, and take advantage of domain knowledge about them by means of interactions such as document movement, highlighting, annotation, and search. Below is a mapping of these interactions to our process model:

**Edits & Enrichment (A)** The data (text documents) can be enriched with annotations, for example, by adding topic-terms that do not explicitly occur in the text.

**Preparation (B)** Document similarities are derived from common terms, which are transferred into a weighted feature vector for each document. Term weights are adapted based on user interactions, such as highlighting and searching for specific terms. In addition, term weights may be updated if the analyst rearranges document positions in the layout. Documents that are moved closer are considered more similar, and the term weights are adapted accordingly.

**Model Selection & Building (C)** A force-directed graph model is derived to create a two dimensional spatialization of the documents. Document nodes are treated as physical objects, where the number of entities or terms per document defines its mass, so larger documents move more slowly. Document similarities are represented by graph edges or ideal springs connecting related document nodes. Spring forces are calculated from common terms and an importance value or weight per term. The analyst may add constraints to the document layout by pinning specific nodes to fixed positions.

**Exploration & Direct Manipulation (D)** Documents are visualized as nodes that can be opened or closed on demand. This allows the analyst to explore documents, inspect details on demand, and provide feedback as needed. The visualization offers direct manipulation interactions that can be translated to ML-pipeline adaptions. These "semantic interactions" adapt the underlying term-weight model or add constraints into the document layout. In this way, the analyst can enrich the spatialization with semantic meaning to support the human reasoning process.

**Execution & Evaluation (E)** Each performed interaction results in an observable behavior within the visualization. For example, new clusters may emerge as a result of pinning a document in the layout, as some documents may move further away, and others may move closer to the pinned document. In such cases, the analyst needs to interpret and validate the results and provide further feedback.

### Other Examples

We found many other examples of VA systems described in the literature that support aspects of the proposed analysis process. In this section, we briefly review a few examples that illustrate a wide variety of realizations. Table 4.1 summarizes these examples in terms of the process components (A–E).

**Inter-Active Learning [123]** is a concept proposed by Höferlin et al. aimed at extending active learning. In line with standard active learning approaches, the proposed system allows the analyst to iteratively add class labels (A) to train a classifier. In contrast with traditional active learning, the analyst can also pose queries to identify points to be labeled. Additionally, the analyst is provided with multiple views that visualize the classifier quality (D) and let the analyst tune parameters of the classifier (C).

**DataWrangler [146]** provides a good example of supporting Edits & Enrichment (A). The system automatically validates selected data and suggests data transformations to the analyst to implement data editing operations.

**iPCA [138]** (Figure 4.4-a) is an interesting example that offers different perspectives and interactions within data and model space. The analyst can edit or remove data items (A) in linked views (e.g., scatter plot, parallel coordinates, or matrix views) (D) and observe the results (E). It is also possible to define dimension loadings using sliders (B).

**Dis-Function [40]** learns distance functions from user feedback. When the analyst rearranges data points in a two dimensional embedding (D), a calculation of a new distance function (or feature weights) is triggered (B). By immediately revealing the resulting changes (updated scatter plot and bar chart), the approach gives analysts a convenient way of exploring alternative configurations of preprocessing steps (E).

**The partition-based framework** by Mühlbacher and Piringer [200] (Figure 4.4-b) provides the analyst with different pre-built regression model variants that can be selected and refined (C). The system visualizes these regression models in a matrix view in combination with further features and quality information (D). The analyst can apply feature-transformations and tune preparation parameters that are used for feature partitioning (B). All interactions trigger recalculations of the regression model variants, and quality metrics enable the analyst to evaluate these changes (E).

**BoababView [293]** (Figure 4.4-d) visualizes the structure of a decision tree in combination with data, feature, and quality information (D). The analyst can perform tree operations (e.g., split nodes) and adapt specific parameters, such as split points values (C). The analyst can inspect and follow changes of the data flow within the tree, and evaluate the precision of the classifier at any stage (E).

**EnsembleMatrix [148, 282]** lets analysts build classifier ensembles by discovering several combination strategies (C). The system offers several confusion matrix visualizations for each classifier with a combined main matrix, as well as a linear classifier combination view (D). An extension of the tool further measures the accuracy of an ensemble classifier before and after user interaction to support analysts in their evaluation process (E).

Table 4.1: Examples grouped by process model components (A–E).

| Part | Examples |
|---|---|
| **(A)** | Adding class labels [123], adjusting & removing data [138], adding textual annotations [85], suggesting transformations [146] |
| **(B)** | Re-arrange data points [40], dimension loadings [138], term weightings [85], preparation parameters [200] |
| **(C)** | Parameter tuning [123], making model selections [200], building ensemble classifiers [282], defining constraints in a force-directed layout [85], tree operations [293] |
| **(D)** | Multiple linked views [138, 123], 2D-spatialization with direct interactions [85, 40], pre-built ML variants [200], tree-visualizations [293], (confusion) matrix [138, 293, 282], browsing visualizations [315] |
| **(E)** | Responsive visualization updates [138, 200, 293, 85, 40], ML workflow design [129], measuring interaction quality [148] |

**Voyager [315]** provides the analyst with visual recommendations for faceted browsing through a series of automatically generated visualizations (D) that match the underlying data's characteristics with user preferences.

**DimStiller [129]** allows analysts to design and validate several steps of a dimensionality reduction workflow (A–D). This approach makes it possible to compare alternative workflows and validate each step of the ML pipeline to identify which phases can be improved (E).

### 4.1.5 Human-Centered Machine Learning Loop

In this section, we focus more closely on the analyst's feedback loop, described as an iterative cycle of *Executions* and *Evaluations* (shown in Figure 4.3-E and in more detail in Figure 4.5). This loop "connects" an analysis system and an analyst whose ability to provide feedback depends on individual factors, as well as the visual interfaces of the system. We will outline individual human factors of the analyst, and elaborate on potential stages of action in more detail (Figure 4.5-left). Subsequently, we enumerate several analysis scenarios to illustrate types of feedback that can be provided by an analyst (Figure 4.5-right).

#### Stages of Action

We adopt Norman's *Stages of Action* [207] to distinguish two phases within the human-in-the-loop model, shown in Figure 4.5. This model describes the interplay and collaboration between the system and the analyst. While the previous section described details of the analysis system, we now shift our focus to the analyst and describe the phases of *Execution* and *Evaluation* in more detail.

**The Analyst** The analyst forms goals based on individual prior knowledge, and assumptions about the data/visual interface. The ability of analysts to provide feedback depends on factors such as technical competence (*e.g.*, expertise in data analysis, ML, mathematics, statistics, or visualization) and application domain knowledge (e.g., biology, business, digital humanities, or sports). Analysts may have highly diverse backgrounds and therefore varying level of skills and related capabilities. Consequently, they may provide different kinds of feedback and "take on different roles". Typically, data scientists, such as ML experts with strong mathematical skills, can train specialized ML models and techniques but may miss significant anomalies in data generation or collection. Conversely, domain experts may be very aware of these details, but overlook important properties of models they might

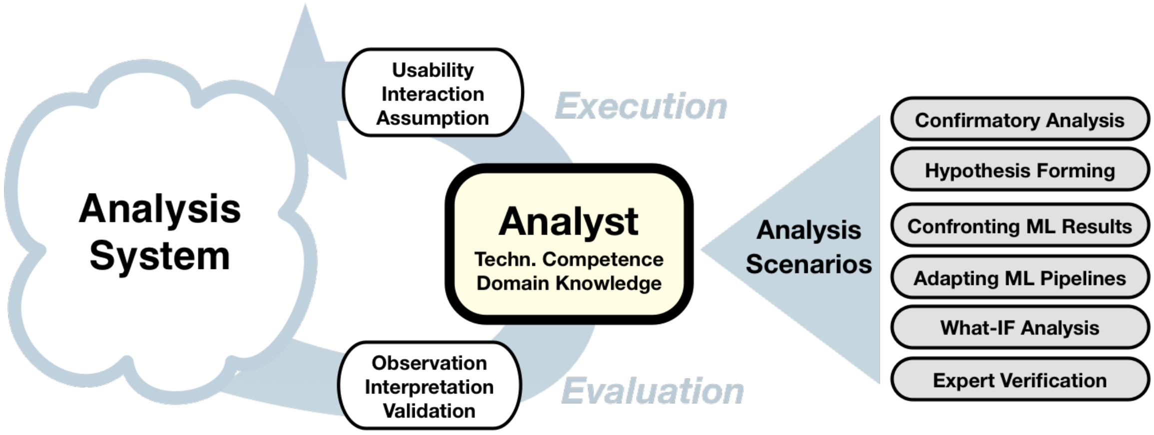



Figure 4.5: Human-centered ML process loop shown in more detail. The loop reveals important characteristics of analytic activity; the right hand side shows different analysis scenarios.

approach as off-the-shelf black boxes. To overcome this, in interdisciplinary research, ML experts, visualization experts, and data owners usually work in collaborative teams. In this case, visualization provides a common platform for communication. These differences and gaps between different types of users can be addressed by a *Liaison*, a person sharing language and knowledge from the application and the visualization domain, with the goal of mediating communication issues [272].

**Execution** Applying Norman's *Stages of Execution* to our interactive ML setting (1) the intentions to act may be based on assumptions about the ML model or the data at hand, (2) the sequence of actions describes the different ML pipeline adaptions, and (3) the actual execution is realized through the visual interface (or visualization) and heavily depends on its usability. As pointed out in the previous paragraph, analysts usually need user interfaces tailored to their individual capabilities. Visual metaphors and actions (e.g., moving points, or providing labels) need to be accessible and familiar. Command line interfaces and specific parameters are often operable for data scientists. However, application domain experts often expect simple and intuitive user interfaces to provide feedback. Note that the visual interactions should faithfully reflect and translate the analysts‘ assumptions to ML pipeline adaptions. If the analyst is not able to perform the desired action (e.g., because of poor usability of the user interface) there is a gap between human and machine, also known as “gulf of execution” [207].

**Evaluation** Before the analyst is enabled to provide (further) feedback, he/she has to *Evaluate* the current state of the system. In our described VA/ML pipeline, changes made by the analyst or feedback given by the analyst (should) cause (1) observable reactions in the analysis system. These observations—in our context usually represented as visual patterns (e.g., groups, sequences, outliers)—have to be (2) interpreted by the analyst who can leverage his/her domain knowledge. Finally, the analyst has to (3) validate and verify the derived insights according to previous goals and assumptions. Visual interfaces should, therefore, allow the analyst to compare different states of the analysis system, by switching between visualization results before and after the computations. Animations and transitions between states or progressive/intermediate ML results may enhance the interpretability of complex ML models. Design studies have to be conducted in order to “bring the entire ML pipeline closer” to the domain experts mental models, language, and metaphors [259]. Interpretability is essential for evaluating the obtained results and also for providing further feedback in subsequent loops/iterations [300]. Note that misinterpretations may cause poor feedback and therefore impair the ML pipeline configuration. This gap between machine and the human is known as “gulf of evaluation” [207]. Especially in ML when the analyst is presented with a final result, it is often a challenge to find out “why” the result is not good enough. Several methods may be combined into complex pipelines, making it hard to assess the quality of the individual blocks.

### Analysis Scenarios

This section enumerates six analysis scenarios illustrating a variety of strategies and feedback that can be incorporated in a visual interactive ML setup. Notice that some scenarios overlap, and can be combined or switched during an analysis session. Specifically, the first two scenarios (*confirmatory analysis* and *hypothesis forming*) can be seen as higher-level analysis goals, in contrast with the latter four scenarios.

**Confirmatory Analysis** An ML model is built on assumptions about the domain and data at hand. In an interactive ML session, analysts may make use of different ML types to model and confirm hypotheses. In this activity, they often correct and refine model parameters to more closely match their assumptions; they also may need to generate and collect evidence to either verify or falsify hypotheses [248]. Such evidence may be provided by statistical tests, or by inspecting visual patterns generated by a more complex ML algorithm. For example, a grouping of similar observations can be computed by clustering or classification. However, analysts also have expectations about groupings and may need to check whether their assumptions are consistent with the ML results. In many cases, techniques do not fit “out of the box” and need to be refined by an expert.

**Hypothesis Forming** Another analysis goal is to generate, form and refine hypotheses. In this case, the analysis is more exploratory, and ML models can be invoked and visualized to get broad overviews. Several unsupervised ML methods are effective for revealing certain structures that are otherwise hidden in data (e.g., feature selection, dimensionality reduction, clustering, outlier or novelty detection). Visualizations support the analyst in spotting patterns that can be investigated in more detail. Such patterns can be, for example, manifolds, outliers, sequences, clusters, or trends. Note that spotting patterns may be the result of pure serendipity during analysis. However, once a pattern has been spotted, an analyst generally needs to discover “why” the pattern exists, and consequently forms more concrete hypotheses and may switch to confirmatory analysis.

**Confronting ML Results or Structures** In an iterative ML process, the analyst provides feedback about results to the ML pipeline. Domain experts, who are able to exploit domain knowledge, can effectively adapt ML results if they spot errors within visualizations that do not match their assumptions or prior knowledge. For example, they can re-organize automatically generated groups [5] or adjust class labels [23]. Furthermore, parameters or weights can be tuned to adapt ML structures to focus the analysis on specific features, or to determine the granularity of the ML algorithms (e.g., the number of clusters desired). Reorganizing points (“declaring distances”) can correct distances between specific observations when they are known to the analyst (e.g., [40]).

**Adapting ML Pipelines** Depending on the data and analysis task at hand, the ML pipeline may not be “comprehensive” enough and may require reconfiguration to accommodate additional ML computations or features of the dataset. ML models can be thought of as atomic components that can be combined and then require some meta-assessment, with the difficulty that validation faces combinatorial growth and can become intractable. An ML algorithm could, for example, require additional pre-processing, specific feature selections and transformations. If ML models become “too complex” some parts of the ML pipeline may need to be simplified or even removed (for example, in case of model overfitting).

**"What-If"-Analysis** Interactions enable the analyst to experiment with an ML pipeline and observe how it reacts to changes or feedback. This may contribute to better understanding of how the model behaves, even without ML expertise. An example is to investigate how the final ML results are affected by adapting dimension loadings in a problem of dimensionality reduction (such as in iPCA [138]). In such a case, the analyst can identify which data items are affected and related to specific features. The same can be done with manipulating data observations. The analyst may examine what happens if a particular observation is present or absent in the data.

**Expert Verification** ML models aim to detect structure and patterns in data, such as trends. However, in a real world use case, patterns have to be cross-checked based on "external" knowledge. One possibility is to apply the ML model to other external data sources to test whether the pattern recurs as in a kind of manual validation or test procedure. These data sources may be obtained from another database or repository. However, if such data is not available, the domain expert has to judge whether observed structures or patterns are plausible and useful. In this case, several domain experts may collaboratively discuss the outcome, or design further experiments.

### 4.1.6 Challenges & Opportunities

On the path toward systems that fully implement the proposed process model, we encounter several important research challenges that must be overcome. We identified five relevant challenges at the intersection of ML and visualization research. We will describe how joint research in these areas opens up novel opportunities to advance practical data analysis.

**Designing Interaction for ML Adaption** A variety of ML algorithms, offering a broad set of design options and parameters, have been described in the literature. Yet, we find no generic way to interface ML with visualization. Current visual analytics systems are often restricted to working with a small set of ML techniques and parameters. Furthermore, within current interfaces, switching between ML models is likely to disrupt a human's sense of context in the analysis process. To address, this, new approaches are needed that support analysts in making sense of such model changes. In addition, existing examples such as ForceSpire and iPCA nicely illustrate how understandable direct interactions can be combined with model changes in a straightforward manner. Direct manipulation has proven effective and easy-to-learn for accessing computational tools [266]. It has, however, not been extensively explored in the context of ML so far. Often, ML models are designed for unique, static configurations, whereas in VA iterative refinement is needed. Mapping user inputs to more complex algorithmic actions along the entire ML pipeline remains an open problem. One key question is how to translate "simple" interactions within the visual interface to data manipulation, preprocessing, or ML model-adaption operations and combinations thereof. *—Opportunities:* Central to our conceptual process model is the idea that the underlying ML design options and meta-parameters, which usually cannot be optimized automatically, can often instead be steered by convenient, iterative user interactions. Accessible interactions and smooth transitions between different ML models will help analysts to develop intuition or form mental models [179] about the underlying data, as well as about the function or behavior of complex ML methods. Consider the case of switching between different ML models: at what point does the system realize—from user feedback—that the current ML model might not be the most appropriate one anymore? It could then suggest an alternative model and smoothly transition to it. Instead of linear projection with PCA, it might, for instance, suggest a more complex nonlinear dimensionality reduction method like multidimensional scaling or $t$-SNE. Continuous model spaces [301, 170] contribute some preliminary ideas towards such solutions, which are dependent on the ML models' meta- or hyper-parameters and their interpretability. Further, more general ways to apply and adapt ML through expert feedback (*e.g.*, labeling or rating) would allow us to take advantage of a larger, more powerful set of ML methods. The previous examples demonstrate that there is vast space for future research, given the great variety of available ML techniques and their associated parameter spaces. A joint effort from both the ML and VA communities is needed to face this research challenge.

**Guidance** Another major challenge is to adequately support application domain experts in steering the ML pipeline. Analysts can be overwhelmed by the wide range of ML models and parameters, along with the challenges of working with large data sets. Moreover, analysis problems are often incompletely defined, and change over time, resulting in a complex analysis process with much trial-and-error. Consequently, analysts may change, adapt, or switch tasks frequently. While analysts may bring crucial domain-dependent information to problem-solving, they often lack advanced programming skills and statistical expertise, and therefore require assistance and guidance (*e.g.*, by providing recommendations about operations on data and alternative models.) *—Opportunities:* It is important to understand better the tasks, practices, and stumbling blocks of domain experts (which likely differ from those of visualization or ML experts). Adopting a design study methodology is a viable approach towards gaining a better understanding of such user characteristics [259] and providing appropriate guidance. Furthermore, enhanced measures and tools could point analysts to interesting data, parameterizations, and ML models through automatic recommendations. While many measures exist, both depicting data and perceptual characteristics (e.g., [34]), currently it is not well understood how they can be effectively exploited in interactive analytical processes. Consider a relevance feedback approach where the learning system retrieves a set of interesting visualizations based on iterative user feedback. In each iteration, the analyst marks the presented results either as positive (interesting) or negative (uninteresting) [23]. How could the system detect if a pattern was spotted and the analysis task changes from overview to detail? Therefore, we envision the usage of analytic provenance which "captures the interactive exploration process and the accompanied human reasoning process during sensemaking" [317]. This information could guide the analysis process to meet the analysts' needs, which might be derived from their behavior. In the VA community, research has been carried out on capturing, visualizing, and reusing analytic provenance. However, more work is needed on modeling such information to shape or refine the overall analysis as well as specific ML methods. Doing this is an interesting research problem that will require expertise from the ML community.

**Measuring Quality & Consistency** In the rich human-in-the-loop analysis process we envision, it is crucial to ensure both *ML model quality* and *visualization quality*. Yet, these two types of quality assurance do not always align well. For example, in a visual representation of a data embedding, after dimensionality reduction, there might be a trade-off between preservation of the original data structure, and readability of patterns, due to intrinsically high dimensionality. While measures have been developed that describe each of these aspects of quality in data analysis (*e.g.*, [301], [170], [34], and [273]), the challenge is to help analysts to find the right balance between them, so that meaningful analysis is enhanced. Beyond measuring ML and visualization quality, our process model suggests a third type of quality assessment, which is the *level of consistency* between the ML model and the analyst's expectations. The goal of data analysis is to extract reliable and relevant knowledge from data. Assuming that there exists some "ground truth" to back up such knowledge, it is the goal of ML and visualization to reveal it with high fidelity. At the same time, the user will have a priori knowledge and expectations, which in the ideal case should closely match what the analysis reveals. While an ML model will surely seek to accurately describe the data, essential pieces of information or context known by analysts may be unavailable to an algorithm. In this case, the set of ML assumptions may be incomplete, a challenge often encountered in exploratory data analysis. *—Opportunities:* To externalize this missing information, it is important to check consistency between what the ML model presents and what the analyst expects. If inconsistent, the analyst should either suspect a problem with the ML model and provide feedback about missing information or accept that the expected patterns were not found in the data. If consistent, analysts usually conclude there is a confirmation of their expectation. Note, although consistency between human and machine is desirable, it does not guarantee correct reflection of the underlying ground truth in the data per se. Currently, consistency checks are often done manually. Automatic methods that systematically check consistency, highlight inconsistencies and recommend any needed remediation could help. A joint effort from both ML and VA communities is needed to enhance these measures, especially by combining and bridging them.

**Handling Uncertainty** There are several stages in our process model where uncertainty might be dealt with explicitly. Uncertainty may arise from several sources of unreliability or vagueness, such as data described by probability density descriptions, missing data, or even systematic errors in modeling. Visualizations themselves can also introduce uncertainty, for example, due to resolution or contrast effects [325]. In our process model, uncertainty implicitly propagates through the pipeline and eventually may affect the analyst's trust-building process [246] (see Chapter 3). Properly describing, quantifying, and formally propagating uncertainty in all pipeline stages will be a major challenge in developing robust, effective tools. *—Opportunities:* Alternative visualizations can be generated and investigated to raise the analyst's awareness of the sources of uncertainty within the ML pipeline. Furthermore, analysts can be supported in (interactively) exploring, understanding and reducing uncertainty [65]. Better integration of uncertainty measures from data, preprocessing, ML models, and visualization can be expected to provide a holistic perspective and understanding of uncertainty. There is much previous work on visualization techniques to display data uncertainty of spatial data, such as volume or flow visualization [213, 39]. We find less work on uncertainty visualization of abstract data, such as high-dimensional data visualization [274]. As abstract data is typical in ML applications, there is a need for improved uncertainty visualization along the analytical pipeline outlined in our process model. A joint effort by ML and visualization researchers is needed to handle uncertainty within the entire pipeline.

**ML-Vis Interoperability** A final challenge arises because most existing ML algorithms, toolkits, and libraries were not designed to support interactive visualization. Scalability problems in computation may cause long delays that impede interaction; parameters may not be adaptable or visible; and relevant information (e.g., quality measures or internal ML structures) may be inaccessible. The ability to communicate these types of algorithmic information and to take advantage of them to construct better user interfaces is often described as "opening the black box" [201]. Especially in the case of direct human interactions, it is often difficult to speed up the ML computations enough to provide the desired responsive behavior of the visualization. Another challenge is to train ML techniques from interactive inputs, which typically are few in number. *—Opportunities:* The visualization community could benefit from ML algorithms and libraries that meet specific requirements, such as exposing intermediate or progressive results, and providing meaningful and interpretable parameters or handles to integrate them with interactive visualizations. Additional information, such as model structures, preprocessings, and quality information can be visualized. Recently, novel VA systems have been described that provide approximated or progressive computations with interactive and steerable visualizations (e.g., progressive t-SNE [218] or progressive PCA [290]). These examples suggest considerable potential for an expanded, generalized integration of ML with visualization. In summary, both communities could gain much from library and framework designs informed by the requirements of both interactive visualization and ML.

### 4.1.7 Conclusions

We propose a conceptual process model that characterizes important forms of interaction that are possible with ML components in a VA process. In general, such interaction offers considerable potential for improved support of ML interpretability, understandability, evaluation and refinement. We found that a multidisciplinary perspective can bridge the gaps between automated ML methods and human reasoning. The proposed process model offers a balanced perspective on the design and configuration of the analytic pipeline, incorporating important aspects of both automated techniques and human interaction. Of course, current VA tools and ML components pose many interesting challenges for future work at the intersection of visualization and ML. To address these challenges, closer collaboration between ML and visualization researchers will be vital.

The conceptual process model proposed in this chapter was developed over a period of 1.5 years. During this process, it was iteratively validated and refined, comparing it with previous models and real-world systems as illustrated in Section 4.1.4. By doing this, we were able to verify how well our process model fits such models and systems. This process lead to both extending the process model to add missing features, and summarizing and generalizing common aspects of it across multiple example systems and existing models. The main benefit was to ground our theoretical process model in real-world systems and applications. A full evaluation of any theoretical work, assessing how useful it will be to others, is beyond the scope of such a single chapter. An effective theoretical process model should be expected to stand the test of time, and to repeatedly demonstrate its validity [224]. We hope that empirical evaluations of future systems and solutions implementing the proposed process model will add evidence to prove its worth [71]. By guiding researchers and practitioners in the design of novel data analysis solutions, in the long run, its “real” value can be established. The next section will demonstrate how such a conceptual process model can be used to design a novel VA approaches that tightly integrates ML (e.g., clustering using Self Organizing Maps) with interactive visualizations, guidance, and analytic provenance.

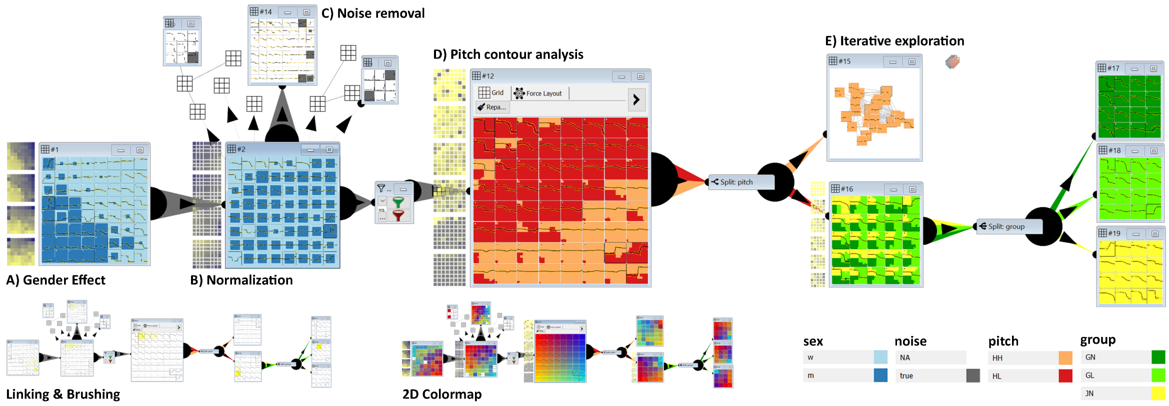



Figure 4.6: Overview of a SOMFlow clustering graph that was created during our expert study to analyze speech intonation: First, a gender effect is identified (A) and removed using a domain-specific semitone normalization (B). The analyst created more detailed SOMs for artificial cells and added manual annotations (C) to filter noise caused by measurement errors. The resulting SOM reveals a relation to the pitch meta-attribute (D) and further data partitions allow the analyst to compare pitch contours of different speaker groups.

## 4.2 SOMFlow: Guided Exploratory Cluster Analysis with SOMs and Analytic Provenance

Clustering is a core building block for data analysis, aiming to extract otherwise hidden structures and relations from raw datasets, such as particular groups that can be effectively related, compared, and interpreted. A plethora of visual-interactive cluster analysis techniques has been proposed to date, however, arriving at useful clusterings often requires several rounds of user interactions to fine-tune the data preprocessing and algorithms. We present a multi-stage Visual Analytics (VA) approach for iterative cluster refinement together with an implementation (SOMFlow) that uses Self-Organizing Maps (SOM) to analyze time series data. It supports exploration by offering the analyst a visual interface to analyze intermediate results, adapt the underlying computations, iteratively partition the data, and to reflect previous analytical activities. The history of previous decisions is explicitly visualized within a flow graph, allowing to compare earlier cluster refinements and to explore relations. We further leverage quality and interestingness measures to guide the analyst in the discovery of useful patterns, relations, and data partitions. We conducted two pair analytics experiments together with a subject matter expert in speech intonation research to demonstrate that the approach is effective for interactive data analysis, supporting enhanced understanding of clustering results as well as the interactive process itself.

### 4.2.1 Introduction

Clustering can be used to analyze large unknown collections of time series data, such as stock market prices, temperature changes, movement features, or spoken utterances, to form subsets of similar data items and to reveal otherwise hidden patterns (e.g., cluster properties and relations). However, analysis problems are often ill-defined or imprecise (neither knowing where or what to seek), interesting patterns (e.g., relations to further metadata) are hidden within particular subsets, and it remains a problem to identify relations among a series of obtained clustering results. Furthermore, the underlying computations need to be adapted to reveal the desired structures for the analysis task at hand.

Hence, this large problem space specifies a need for interactive data exploration in different "directions". Visual Analytics (VA) aims to provide the analyst with a visual interface to explore automatically obtained results to form and refine hypothesis and to interact with the underlying computations if necessary [150, 248]. Tightly intertwined solutions (computations, visualizations, interactions) are needed to cope with nowadays real-world analysis problems [245, 249, 244] and users which are typically experts in their domain, but novices when it comes to VA, require specific guidance during exploration [52].

To cope with these challenges, we propose an interactive partition-based clustering approach that allows the analyst to drill down into subsets of interest (top down, divide & conquer) based on different division strategies. This approach emerged from our ongoing collaborations (started 3 years ago) with linguistic researchers from the domain of prosodic research (i.e., speech intonation) analyzing time series data of recorded speaker utterances [239, 14]. Our initial VA system used the Self-Organizing Map (SOM) algorithm to create data overviews, and it iteratively enabled the analyst to select data subsets of interest. However, we observed that our users were sometimes overwhelmed by the number of obtained SOMs requiring a visual overview of the analysis process as well as user guidance to support costly and time-consuming analysis tasks (e.g., relation seeking or data annotation).

Inspired by existing hybrid visualization systems (e.g., [121, 293]) we developed the idea to embed interactive SOM visualizations into a graph structure as an analysis overview representing the clustering and interaction flow that further supports higher-level human analytic activities (e.g., organizing and memorizing what has been done, comparison-, or verification tasks). Our resulting SOMFlow system supports four abstract exploration tasks with a rich set of visualization and interaction techniques to (1) analyze and assess the quality of the obtained clusterings, to (2) adapt the computations, and to (3) create further data partitions while (4) keeping the overview. We further leverage quality and interestingness measures to guide the analyst. Hence, SOMFlow instantiates our conceptual human-centered machine learning process model (see Chapter 4.1, [244] and Figure 4.10). We contribute a general clustering approach and an implemented SOMFlow system (focusing on the SOM algorithm and time series as primary data) that tightly integrates interactive visualization, machine learning (ML), and quality measures, embedded into an analytical reasoning space representing the analysis process.

Next, we provide background information and discuss related work before we describe our approach in detail along four abstract exploration tasks (Section 4.2.4). In Section 4.2.5, we explain different ways to guide the user during the analysis and describe our SOMFlow system in Section 4.2.6. We report on two pair analytics experiments that evaluate our SOMFlow system in a real-world setting (Section 4.2.7). Finally, we discuss remaining issues and enumerate promising future research directions (Section 4.2.8), before we come up with a conclusion (Section 4.2.9).

### 4.2.2 Background on Self-Organizing Maps

We chose the SOM algorithm (also known as Kohonen Maps [159]) as a fundamental approach to automatically generate data overviews. The algorithm introduced by Kohonen has been widely applied to clustering problems and data exploration [157]. It is based on a neural network that can be represented as a grid of cells (neurons, or tiles). Each cell contains an artificial vector (e.g., time series) with the dimensionality of the input data. During the training phase, the vectors are subsequently adapted towards the information provided with the input data. In each step, the input vector is assigned to the best matching unit (most similar cell), and this cell, as well as a subset of spatial neighbors of the grid, are modified for better matching [157]. The result is a grid that represents the data based on their prototype vectors. In the final topology, more similar cells are closer and less similar cells will be farther away. As a result, the input data is distributed across the SOM in a similarity-preserving way. In summary, this algorithm provides data reduction (vector quantization with means vectors), dimensionality reduction (two dimensional embedding), and data clustering/classification (assigning data items to cells). Note that in cluster analysis, the data items within a cell do not necessarily form a single cluster as a set of similar cells can be considered as a cluster as well.

Depending on the use case at hand, SOM visualizations either directly show the information of mean cell vectors, or use concrete data items for cell representations (e.g., [302, 303, 255, 30]). The SOM algorithm also allows the visualization of the structure of the grid (e.g., neighborhood information) and quality measures. For our collaboration with linguistic domain researchers (e.g., in our previous work [239]), we visualize the artificially created pitch contour of recorded speaker utterances (sound of the pitch over time) as a thick black line in each cell (see, e.g., Figure 4.8). Data items of every cell (in our case real pitch contour vectors) can be shown on demand. In addition, the analyst can inspect relations between clusters and available metadata which can be used to color the cells. A linguistic task is then, e.g., to analyze how often a certain pattern (pitch contour) appears and if it is related to specific speaker properties (e.g., nationality) to identify differences. Many visualization techniques for SOMs exist and have been applied in different domains. Our work relies on such visualization techniques (that mainly focus on single SOMs) embedded into a workflow supporting iterative data partitioning and analysis steps.

### 4.2.3 Related Work

Our work is related to *Visual Interactive Cluster Analysis* in general where we put specific focus on using the SOM algorithm to analyze time series data. The second part describes existing *Hybrid and Provenance Visualization Systems*. Then we focus on *Quality-based Guidance for Visual Exploration* before we emphasize on the novelty of our work.

#### Visual Interactive Cluster Analysis

Visual interactive clustering solutions exist for a variety of data types. E.g., the work of Andrienko et al. [9] proposes methods to group movement trajectories, Ruppert et al. [237] describe visual interactive workflows to cluster textual documents, Cao et al. [47] focus on the interactive analysis of multidimensional clusters, and the approach by Nam et al. [203] focuses on high-dimensional data. Some of them also let the user select a specific subset of interest where another subsequent computation of the clustering/classification is applied (e.g., the work by Choo et al. [60]). Furthermore, specific visualization approaches focus on the visual analysis of time series data [1] with different analysis goals (e.g., segmentation, clustering, classification, motif-detection) [229]. With respect to our clustering scenario, we focus on the grouping of time series based on their similarities. A further task is then to seek relations between the obtained clusters and metadata attributes (if available) or to apply data annotations (labels) manually.

Many different clustering algorithms exist (e.g., k-means, hierarchical clustering, etc. [116]) and have been applied in VA, often in combination with different metrics and a dimensionality reduction step [249] to obtain a two dimensional embedding of the clusters. Therefore, many VA approaches make use of the SOM algorithm [159] that naturally comprises both steps. Vesanto [302, 303] early described several techniques to apply and visualize the obtained results. Existing SOM implementations and toolboxes that offer gird visualization exist (e.g., Java SOM toolbox [1], or som pak [158]) and further interactive VA systems have been developed. E.g., Schreck et al. [255] describe a trajectory clustering system that offers the analyst visual representations to provide interactive feedback to the algorithm. Further works by Bernard et al. focused, e.g., on time series research data [25] and motion patterns [30]. SOM visualizations have also been used to speed up expensive data labeling tasks. E.g.,

[1] http://www.ifs.tuwien.ac.at/dm/somtoolbox/index.html, accessed 23.03.17

the work by Moehrmann et al. [198] allows users to apply image labels using SOM visualizations. Finally, the predecessor of the presented work [239] proposed an iterative refinement approach of SOM cell selections and computation adaptions to arrive at subset visualizations of interesting speech intonation patterns. However, a lot of results (SOM instances) were produced making it hard for the analyst to compare and reflect the analysis. Furthermore, it only offered a few visualization techniques and did not support the analyst with automatic recommendations based on quality measures.

### Hybrid and Provenance Visualizations

Another area of related works describes hybrid visualization approaches that embed smaller visualization types into another visualization technique encoding a particular structure (e.g., a tree, graph, or network). A famous example is the Node-Trix system by Henry et al. [121] that embeds matrix representations as aggregated nodes within a social network (node-link diagram). A similar approach is adopted in the OntoTrix [17] system. Other techniques embed several different visualizations into a structure representing the data or analytic flow. Gratzl et al. [106] describe the domino system that enables users to apply data subset selections and manipulations using several dependent visualizations. More recently, Stitz et al. [279] propose a data workflow-based visualization system for biomedical research. The work by van den Elzen and van Wijk [294] provides a visual exploration method based on small multiples and large singles. The same authors proposed another technique called BaobabView [293] that explicitly uses the structure of a ML algorithm (decision tree) augmented with smaller visualizations that describe, e.g., data distributions and flows.

Further related systems can be found under the heading of “data or analytic provenance” that enable the analyst to track the history of data transformation or interaction steps. Early works of Young and Shneiderman [320] provide a graphical interface for data filter flows to define and analyze boolean queries. Similarly, Elmqvist et al. [81] describe the DataMeadow system that lets the analyst visually construct queries using graphical set representations within a canvas. Another famous example is the VisTrails system by Callahan et al. [46] visualizing scientific workflow evolutions. The GraphTrail system by Dunne et al. [77] is another example that tracks users’ interactions and embeds respective visualizations into an “exploration workspace”. Other VA systems, such as Jigsaw [177], offer the analyst specific visual components (e.g., the tablet view) to organize, manage, and annotate bookmarked visualizations.

### Quality-based Guidance for Visual Exploration

Another branch of related research concentrates on guiding the user during the analysis and different approaches to build, e.g., mixed-initiative [63] or relevance feedback [23] systems have emerged. However, the basis for such systems are task models, data-, quality- or interestingness measures.

Bertini et al. [34] survey existing approaches that make use of quality measures in high-dimensional data analysis and propose a quality-based analysis framework that includes measures (e.g., cluster, correlation, or outliers) in data and image space. Sips et al. [273] make use of class consistency measures to determine the quality of cluster mappings from n-D into low-D (centroid-based and entropies of spatial distributions). Tatu et al. [284] describe further quality measures for different high dimensional visualizations (e.g., scatterplots or parallel coordinates), and the work of Aupetit and Sedlmair compares visual separation measures [16]. General cluster validity measures such as compactness and separation [115], or silhouette coefficient [236] further exist.

We are also aware of SOM-based quality measures to calculate, e.g., quantization errors within cells, or topological errors [159, 223]. It is further possible to visually assess the quality of a SOM result [29] using SOM-grid/network visualizations, such as the u-matrix [292], or s-map (smoothed data histograms) [211]. Further work by Bernard et al. [26, 27] describes approaches to measure the

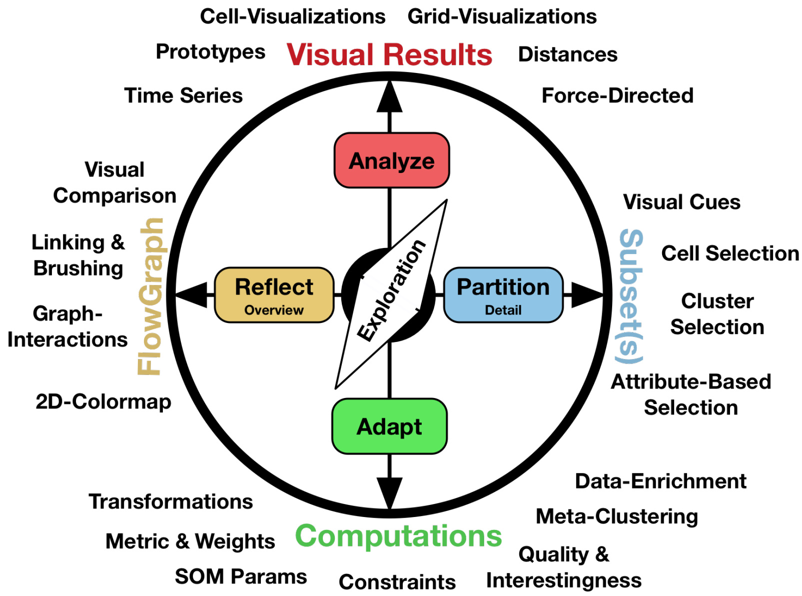



Figure 4.7: Our approach supports four abstract exploration tasks: The analyst can **analyze** visual results of the current SOM and **adapt** the computations if necessary. New SOMs are generated with several data **partitioning** approaches. All intermediate cluster results are embedded into a flow graph that enables the analyst to **reflect** the analytic process. For each exploration task, we offer a set of visualization and interactions techniques that are shown outside the circle.

strength of relations between data content and metadata, such as using Simpson's diversity or Shannon entropy measures.

#### Novelty and Contributions of our Work

Our first subsection on *Visual Interactive Cluster Analysis* describes works in the VA domain that leverage the SOM algorithm to obtain clusterings but it also reveals that most of the work focused on one single SOM (or clustering) result. The subsequent section on *Hybrid and Provenance* reveals that many such visualization techniques exist, however, they are rarely integrated with complex ML methods, such as iterative cluster exploration processes. The *Quality-based Guidance for Visual Exploration* section offers a variety of approaches and measures that can be used to guide the exploration process, however, concrete real-world applications are rarely described. Hence, our contribution is a hybrid approach to support explicit visualization of the clustering interaction process and we further leverage quality and interesting measures to guide the analyst during the analytic process demonstrated with a real-world setting.

### 4.2.4 Partition-based Cluster Exploration

A fundamental idea of our approach is the interactive and iterative construction of analysis workflows for the user-centered partitioning of large complex datasets. The SOM algorithm serves as a powerful visual-interactive data partitioning tool that is widely applied and delivers robust results. The analysis workflows are visually represented within a graph serving as a means to reflect analytical provenance and support workflow navigation. In every analysis step (node of the graph), users are enabled to analyze partitioning results, adapt algorithmic models and parameters, proceed with downstream partitioning routines, or step back to compare previous results. As such, our hybrid interactive graph implements the overview and details paradigm, facilitated with VA support in every step. In addition,

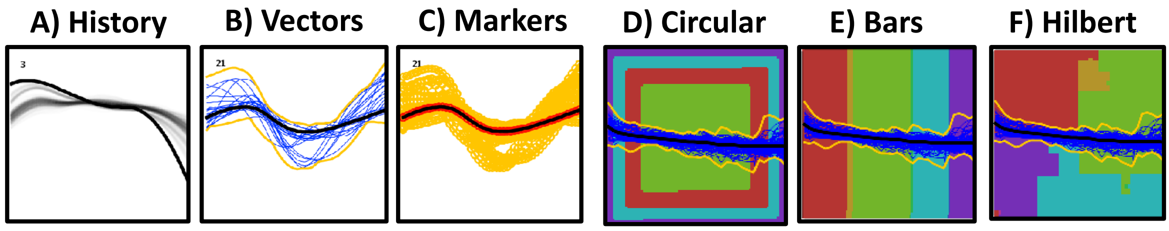



Figure 4.8: Cell visualizations: A)–C) depict time series and prototype visualizations. A) illustrates the training history of the prototype, B) renders the prototype vector (black) with the actual time series vectors (blue) and a yellow min/max bandwidth, C) renders the single data points for the prototype (red) and data items (yellow). E)–F) show different pixel-filling techniques to color the cell according to metadata.

the graph at a glance provides provenance information and can be used for the navigation from coarse to fine-grained analysis. The analyst is presented with a visualization of the entire dataset at the beginning of the analysis. Then, the human task is to decide, how the data can be partitioned, or how the computations can be adapted. These decisions result in new SOMs (or new computations) and iteratively enable the human to navigate into subsets of interest.

Hence, our approach comprises four abstract exploration tasks enabling analysts to **partition** data, **analyze** data partition results, **adapt** data partition models, and **reflect** the analysis process. We structure them along two orthogonal axes (Figure 4.7): The vertical axis corresponds to the "current" state of the analysis process that can be visually **analyzed**, as well as **adapted** to improve model and parameter settings further. The horizontal axis corresponds to the analysis granularity from coarse (left) to fine grained and detail-rich (right), where new subsets can be created by **partitioning** the data. Exploring the clustering flow graph enables the analyst to **reflect** what has been done in the past. Three of these abstract tasks (analyze, adapt, and partition) directly correspond to building blocks of conceptual VA models (e.g., [150, 244, 245] or Chapter 4.1) while reflect enables higher-level verification activities [248, 244] by comparing the graph elements. Each exploration type is supported by a variety of visualization and interaction techniques shown outside the circle in Figure 4.7. In the following, we describe these techniques in more detail.

### Analyze Visual Results (Single SOM)

The SOM algorithm is used to enable users to analyze and partition large unknown data collections. Our decision for using the SOM is based on its special characteristics conflating data clustering, vector quantization, dimension reduction, and the ability for cluster visualization [159, 255, 29]. Hence, we make use of existing techniques to support the analysis of the SOM grid visually.

**Cells and Time Series:** According to the nature of the SOM being a neural network-based clustering algorithm, the output of the algorithm is a (2D) grid, containing a matrix of cells. Each cell represents a portion of the high-dimensional data space, comparable to the cell of a Voronoi diagram. The data items mapped to a respective cell are represented by a representant or, like in our case, a means vector that is visualized as a bold black line chart (Figure 4.8). These vectors are created during an animated training phase allowing users to observe and follow (see Section 4.2.7) the algorithm and a training history for each prototype vector can be shown as gray background (Figure 4.8–A). The actual data items that are assigned to the cells are visualized as thinner blue vectors. We also visualize a yellow bandwidth for each cell by drawing the min/max values (Figure 4.8–B) representing the cell uncertainty. Finally, it is possible to show all the data points of each time series step for the prototype (red) and the vectors (yellow, Figure 4.8–C).

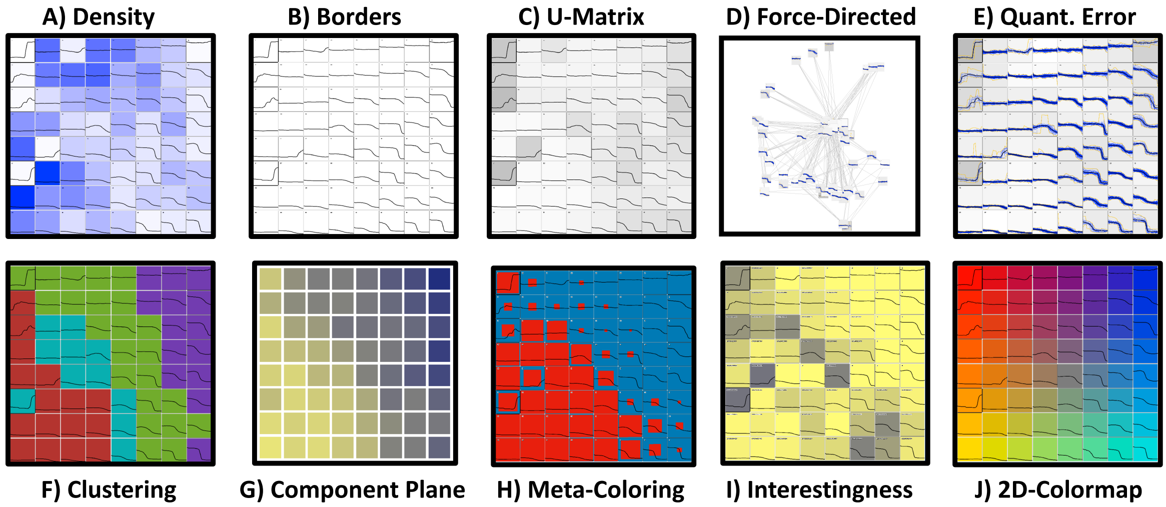



Figure 4.9: Grid visualizations: A) Density map, B) distance borders, C) u-matrix, D) cell topology as a force-directed graph, E) quantization error, F) meta-clustering, G) feature component plane, H) meta-coloring, I) cell interestingness, and J) global 2D colormap overlay.

**SOM Topology:** The grid-based output of the SOM provides the natural structure (topology) of the cells to be visualized (e.g., Figure 4.9). A series of visualization techniques exist to analyze topological properties in detail. The amount of data items in a cell can be encoded as a density map (Figure 4.9–A) using a linear color coding (i.e., the more blueish, the more data items are contained). Distances between the cells are visualized as black borders within the grid where darker lines indicate larger distances (Figure 4.9–B). The u-matrix can be computed by obtaining the Euclidean distances of all neighbors of a single cell (i.e., it determines how similar or separated a cell is compared to its neighbors). It can be visualized by a gray scale cell coloring where light colors depict most similar neighborships (potential clusters), and dark colors represent more widely separated cells (potential cluster borders or noise, Figure 4.9–C). In addition, we allow the user to switch to a force-directed graph layout that is using the distances of cell neighbors as spring forces (Figure 4.9–D). These techniques support the analyst in understanding the topology of the SOM and in identifying clusters within the SOM grid. It is also possible to calculate the quantization error (*qe*) within a cell by comparing all member vectors to its representant/means vector (see also Section 4.2.5). Visualizing the *qe* as another gray scale overlay (Figure 4.9–E) allows the analyst to understand the quality of the quantization (aggregation) within each cell (e.g., dark gray indicates heterogeneous cell content while lighter cells contain data items very similar to the cell vector). We can also apply a meta-clustering (e.g., k-means) to the SOM grid (based on the distances between the cell representants) to automatically obtain and visualize cluster regions on the grid (Figure 4.9–F).

**Component Planes:** This technique reveals relations between individual dimensions (components) of the dataset and the SOM result [255]. Given a feature vector representing the aggregated temporal domain, component planes support the comparison of different temporal phases (user-defined set of components/parts) of a time series by coloring each cell according to the average value of the time series (see Figure 4.15). This allows the analyst to spot specific SOM-regions that have in average high (blue) or low (yellow) values (Figure 4.9–G).

**Visualizing Metadata:** Our approach supports the combined analysis of (time series) clusters and additional metadata attributes, such as sensor devices of an experiment, subject measured with

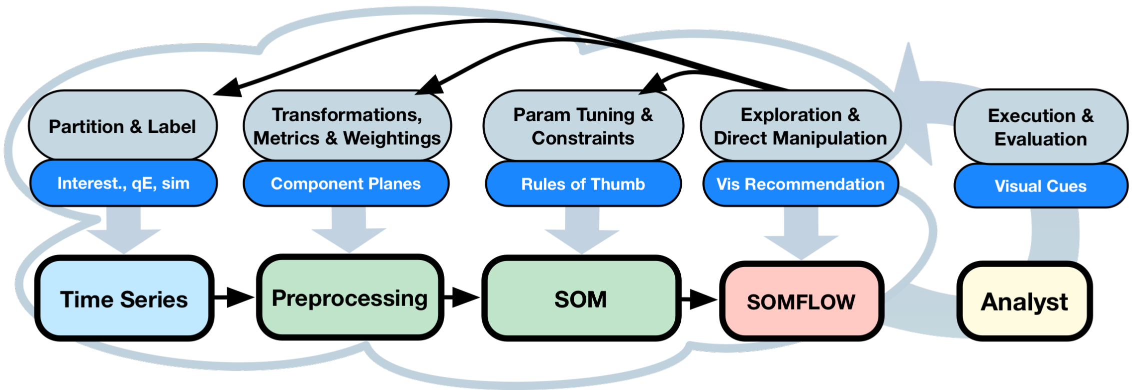



Figure 4.10: SOMFlow process model instantiating the human-centered ML model (Figure 4.3) described in Chapter 4.1. The analyst can interact with each component of the pipeline. Actions (gray boxes on top) cause a re-computation of the pipeline (bottom boxes), and the results are embedded into to SOMFlow graph. The analyst can then evaluate the changes and provide further feedback (loop of executions and evaluations). The dark blue boxes in each stage enumerate methods to automatically support the analyst in each of the stages.

a time series recording, distinction between male and female, or the day of the week. If the data contains such additional information, its category/value frequency can be visualized. We implemented different meta-coloring techniques, such as circular, bar, or Hilbert pixel filling (Figure 4.8–D,E,F) to reveal relations between the SOM results and the metadata (e.g., see Figure 4.9–H). Those techniques produce different visual patterns and are more or less suitable for specific analysis tasks and application domains (see Section 4.2.7). The coloring allows the analyst to distinguish homogeneous (single value) and heterogeneous (mixed) cells. With that respect, it is possible to visualize how interesting each cell is using the Simpson's Index (see also Section 4.2.5) considering a specific metadata attribute. The obtained interestingness value can also be normalized and encoded on the entire SOM grid (see Figure 4.9–I) [27]. Yellow denotes interesting cells that include a relation to a specific attribute value while dark gray depicts uninteresting cells without relation to the metadata attribute.

### Adapt the Underlying Computations

Users can adapt the underlying computations in every analysis step if the obtained results do not sufficiently meet the analysis requirements. Examples include the wrong preprocessing (e.g., sampling or normalization) of the time series, the distance calculation, or parametrization of the SOM algorithm. Hence, the SOM algorithm may not be able to grasp the desired properties of the data. Our approach offers interactions for each block of the ML pipeline [244, 249] (Chapter 4.1) that correspond to the data, the feature space, and the SOM algorithm. Figure 4.10 (bottom) illustrates the underlying machine learning components (time series, preprocessing, SOM) that are mapped to interactive visualizations within the SOMFlow user interface. The analyst is able to inspect and evaluate the results visually. The user interface further allows the analyst to explore and adapt each of the pipeline components interactively. These interactions are shown on top of the pipeline in Figure 4.10 augmented with ways to support these actions (blue boxes).

**Data:** Analysts can assign user-defined labels to each cell. These labels can be used as additional metadata provided by the human (data enrichment) who can, e.g., mark cells (or data items) as "uninteresting" or "interesting". Note, that the computations will automatically consider these labels (e.g., interestingness measure) for providing user recommendations (see Section 4.2.5). Iterative data selections are explicitly realized with our data partitioning tasks (Section 4.2.4).

**Transformations:** The performance of the analysis depends on the data cleansing and preprocessing strategy. Our tool is able to apply normalization techniques (min/max, logarithmic, square root, etc.) to transform the data values. As the SOM algorithm requires time series vectors of equal lengths, it is also possible to adjust the time series by different strategies, such as simple approaches of adding mean-values (mean–padding) or 0s (zero–padding), or linear interpolation (pair–wise).

**Metric and Weightings:** The Euclidean distance measure builds the reference metric. We use a weighted variant, allowing the user-based weighting of different temporal intervals of the time series feature vector. The metric can be switched to Manhattan or more expensive computations, such as dynamic time warping (FastDTW) or Earth Movers distance. We also offer an editor to weight different parts of the time series more or less important (Figure 4.15–C).

**Parameter Tuning and Constraints:** The SOM algorithm can be parameterized in different ways. Training parameters such as the number of iterations ($i$), learning radius ($r$), or learning rate drop ($d$) can be set in a control panel for each SOM. We animate the training phase by updating the visualization after 1000 steps. Another parameter denotes the form of the SOM (number of cells, rows, and columns). For a default configuration of these parameters, we use the "rules of thumb" [159, 303]. Accordingly, we apply a two-step training process: The first step is a rough training ($i = 200000$, $r = 0.3$, $d = 1$) while the second step is a finer training ($i = 1500000$, $r = 0.1$, $d = 1$). Within the training progress, users can fix individual cells of the SOM grid to enforce a specific topology [255]. Adapted configurations can be applied in a new SOM or replace the current SOM. This allows the analyst to compare the different configurations. The parameters can be freely adjusted based on the analysis task at hand. It is, e.g., possible to create SOMs with different sizes. This allows the user to analyze the same data with different aggregation levels (e.g., step A and B in Figure 4.14).

### Partitioning Data

Our approach enables the analyst to partition the data into subsets of interest to iteratively arrive at a more fine-grained analysis. We are able to group the partitioning task into three categories.

**Cell Selections:** The analyst can select cells that either represent an interesting subset of the data that can be investigated in more detail, or depict imprecise cells (high $qe$) that need further refinement in order to arrive at more fine-grained visualizations. Cell selections can be performed based on the primary data and the SOM result quality (Figure 4.8–A–C, Figure 4.9–A–E).

**Cluster Selections:** Similar cells (or cell clusters) that partition the data based on their neighborhoods can be selected to form new subsets. The analyst can either select specific clusters to create new partitions or just split the data based on cluster labels. Clusters can be identified manually supported by the SOM topology visualizations (Figure 4.9–A–E) or using automatic meta-clusterings (Figure 4.9–F).

**Metadata-Based Selections:** On the one hand, metadata can be used to create hypothesis-driven subsets of the data by applying filters or splits based on attribute values. On the other hand, the analyst can seek relations between clusters of the SOM result and metadata. A task for the analyst is to overlay the meta-colorings to reveal specific areas on the grid that are represented by a particular data attribute (see, e.g., Figure 4.9–H). These attributes (or the respective clusters) can then be used to split the data further. Another reason to partition the data is to select heterogeneous (mixed value) cells (e.g., at the decision border, or outliers cells) to explore more detailed differences within these subgroups. The

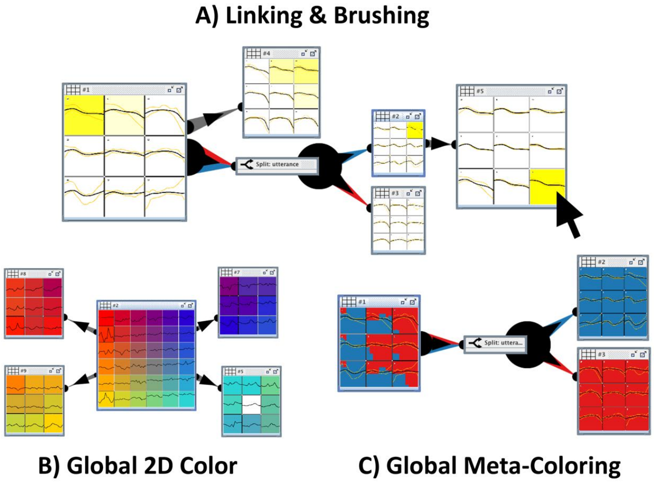



Figure 4.11: Graph exploration techniques: A) Linking & brushing across the entire graph on cell hovering, B) A 2D colormap can be created for a selected SOM and shown in the entire graph, C) Meta-coloring for the same attribute for all SOMs.

metadata-based partitioning is supported by the meta-coloring and interestingness overlays (Figure 4.9–H and I).

### Reflect the Analysis within the Flow Graph

The entire analysis is embedded into a flow graph connecting the SOMs based on their hierarchical relations. It serves as an analytic provenance [317] component that supports higher-level verification and evaluation (see Figure 4.10–right hand side) activities with a visual comparison of the obtained SOM results.

**Flow Graph Elements:** Each SOM is a node connected by shared data items (links). The graph is built by the human who is supported by visualizations and quality-based recommendations (see Section 4.2.5) to create data partitions. The connections show the number of data items flowing from the parent into the child SOMs. Different elements (direct flow, splitters, and filters) can be created. The arrow size is mapped to the number of data items and metadata can be used to illustrate the data flow.

**Interactive Exploration:** Our approach offers specific interactions to explore relations within the graph. Hovering a cell will highlight all the cells that contain shared data items. The strength of the highlighting is mapped to the number of common data items (Figure 4.11–A) by comparing the hovered cell to all other cells of all SOMs. Selecting a specific SOM will highlight the original cells within the parent SOM by adding an orange border around these cells (e.g., Figure 4.14–B). When users want to analyze distributions of cell contents of an entire SOM with all SOMs of the analysis graph, the 2D colormap technique can be used. 2D colormaps [28] dye the cells of a SOM with similarity-preserving colors, either depending on the input or the 2D output space. We transmit the color-coding to all SOMs in the analysis graph, allowing the lookup of similar cells in different SOMs, as well as the comparison of cluster structures across SOMs. An example for this color-linking strategy is depicted in Figure 4.11–B. Similarly, it is possible to select a global meta-coloring (Figure 4.11–C), and finally, we let the analyst switch between a local (per SOM) and global (per graph) min/max-normalization for the data-rendering.

**Meta Interactions:** In real-world analysis tasks, the flow graph can grow fast, and different analysis branches can be created. The graph elements can be re-arranged, resized, maximized, and minimized. It is further possible to navigate within the canvas (zoom & pan) and to annotate graph elements with textual descriptions. This all supports the analyst‘s verification activities, such as knowledge management, adding interpretations, remembering results, and drawing conclusions.

### 4.2.5 Providing Guidance

Our approach offers a rich set of visualization and interaction techniques to support our four abstract exploration tasks. To enhance the usefulness, we integrated a series of guidance techniques to overcome costly investigations of uninteresting data properties. Figure 4.10 illustrates these techniques as blue boxes placed at the respective interaction types.

#### Computing Groupings, Quality, and Interestingness

We automatically calculate groupings, quality, and interestingness measures based on data, SOM, and metadata properties.

**Automatic Clustering:** It is possible to apply a meta-clustering to the SOM-grid that is computed based on the cell prototypes. We implemented different algorithms (k-means, k-medoids, coweb, SOM) that can be chosen by the analyst who may then decide to split the data based on the obtained cluster labels.

**Similar Cells:** Once a user selects a cell (that is considered as interesting), we can compute if there are neighboring similar cells that can be suggested for extending the selection (*simCells*). We make use of the normalized cell distances (*dist*) to identify the relevant neighbors that have a smaller distance than a similarity threshold $simT = 0.3$.

**Cell Quality:** We can automatically point the analyst to imprecise cells with a high *qe* as candidates for further refinement. We compute the *qe* according to [159] by calculating the mean Euclidean distance of all cell members compared to the cell prototype vector. *qe* is normalized for each cell over the complete SOM, and we introduce a threshold $qeT = 0.1$ to distinguish good from imprecise cells. We also leverage the SOM topology and put neighboring cells with $qe > qeT$ into a common SOM resulting in new child SOMs that show imprecise areas in more detail.

**Interestingness:** Other measures support the identification of interesting relations between time series clusters and metadata properties. Similar to [27], we calculate an interestingness score for each metadata attribute and SOM cell (i.e., calculate a diversity score of contained attribute-values for each cell) using the Simpson’s Index (*simpIdx*). We can make use of this measure to identify interesting metadata attributes with potential relations to the SOM result.

#### Providing Recommendations with Visual Cues

This section describes how we leverage the described measures to provide the analyst with visual recommendations.

**Interestingness Ranking:** We append a thumbnails bar at the left hand side of each SOM visualizing ranked attribute interestingness overlays (Figure 4.12–A). Similar to the approach of Bernard et al. [27], the rank of a meta-attributes is determined by a three-step calculation: 1.) For each SOM cell and metadata attribute we calculate the interestingness (*simpIdx*), 2.) The average interestingness over

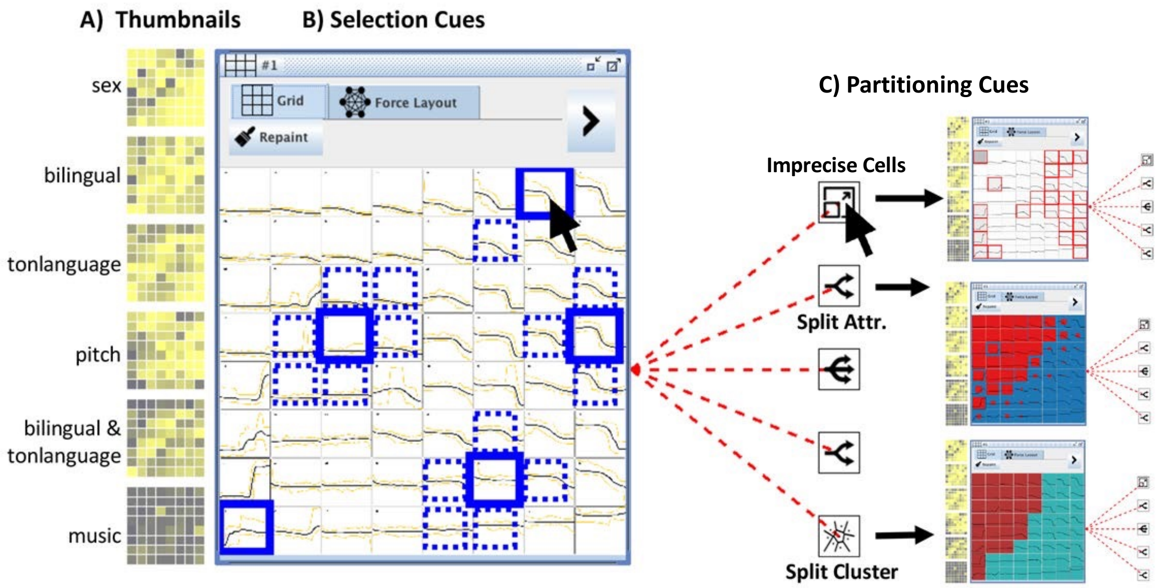



Figure 4.12: Visual cues recommend ways to partition the data: A) Ranked attribute interestingness, B) selection extensions, C) partitioning cues.

all cells and attributes is calculated, 3.) For each attribute, we count how many cells are more interesting than the average and we use the obtained value to rank the meta-attributes. If two meta-attributes share the same rank, a second-level ordering is done by considering the average interestingness value of each attribute.

**Extending Cell Selections:** Once a user selects a cell of interest, we obtain relevant neighbors ($simCells$) and visualize a dashed selection border as a visual cue to extend the current selection (Figure 4.12–B).

**Partitioning Cues:** We visualize the most significant recommendations for creating further data partitions on the right hand side of the SOM (Figure 4.12–C). Hovering the partitioning cues will reveal the respective visualization overlay on the SOM grid (Figure 4.12–right). In case of imprecise cells, we show the respective cells with red borders. In case of recommended attribute splits the meta-coloring is shown, and finally, in the cluster split case, we show the cluster coloring. Our recommendation system contains three different types of actions. The first type of recommendation is the SOM refinement by retraining imprecise cells as new SOMs. Therefore it suggests retraining cells with a high $qe$ ($> qeT$). Neighboring imprecise cells are aggregated and trained in a single SOM. This option is always the first recommendation, if available. The second type of action is the split option. It suggests partitioning the data into subsets with equal meta-attribute characteristics based on its interestingness value as high interestingness implies high homogeneity in the SOM cells. Therefore, it might be interesting to analyze each meta-attribute characteristic separately. A maximum number of five meta-attributes with high interestingness average values are suggested ($avgInterestingness > 0.6$) and ranked, similarly to the thumbnail previews. Last but not least, the third type of action recommends splitting the data by a meta-clustering. Clicking on any partitioning cue will trigger a data partitioning action. We are well aware that our thresholds appear a bit arbitrary and need to be adapted based on the data and analysis tasks at hand (see Section 4.2.8).

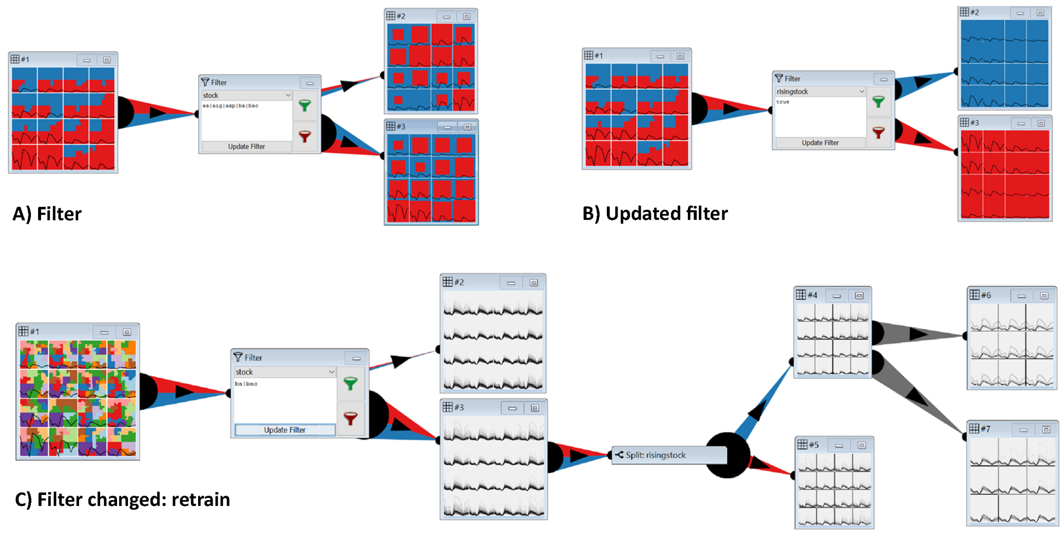



Figure 4.13: A) Filter divides data into two parts by some meta-attribute, B) filter is changed with different constraints, C) filter updates affect all following SOMs in the hierarchy.

### 4.2.6 The SOMFlow System & Use Cases

All the described methods are implemented within our SOMFlow system. In the following, we introduce the remaining details with exemplary use cases.

**Implementation:** The system is implemented in Java using the Java 2D Graphics API and the Swing library for rendering. The SOM algorithm is also implemented in Java what enabled us to tightly integrate the computations with the visualizations. We further make use of prefuse[2] to generate a force-directed layout and included javaML[3] for basic ML functionalities.

**Data Handling:** The system can handle any data in JSON format with the only requirement that each data object has to contain a numeric array that can be used as primary (time series) data. As the SOM algorithm requires data vectors of equal length, we offer several data processing operations (see Section 4.2.4). All remaining data attributes are parsed as categorical metadata.

**Attribute Manager:** The meta-coloring (for single SOMs or the entire graph) can be controlled using the Attribute Manager. This component automatically assigns a default color to each attribute value but also allows to assign custom colors using a color chooser. It is further possible to add new user-defined attributes (adding a name and class labels with colors) that can be used for data annotations.

**SOM Interactions:** It is possible to interact with the SOM cells: 1.) hovering will trigger the linking & brushing for the respective cell, 2.) left-click will select a cell of interest (and trigger the similar cell recommendations), and 3.) right-click will open a context menu for applying manual data labels. It is further possible to enable global or local rendering options within a control panel next to the canvas. Another controls bar on top of each SOM allows to switch between the grid and force-directed layout, while another controls bar can be revealed on the right hand side of each SOM (e.g., Figure 4.16–F). This bar offers controls to 1.) annotate the SOM or the graph (adding notes), 2.) create a new SOM for selected cells with a default configuration, 3.) create a new SOM for selected

[2] http://prefuse.org/, accessed 24.03.17
[3] http://java-ml.sourceforge.net/, accessed, 24.03.17

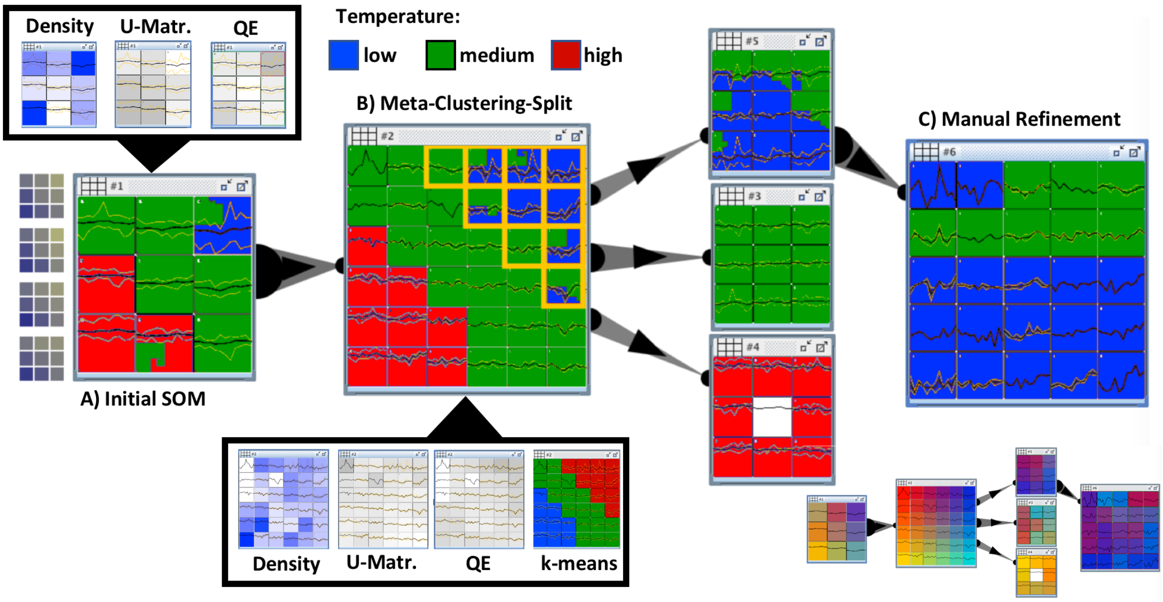



Figure 4.14: Yearly temperature changes of the southern hemisphere: A) Initial SOM, B) a bigger SOM is retrained with a meta-clustering, C) cluster labels are corrected manually.

cells with a custom configuration (a configuration panel to set the data processing, SOM parameters, and metric will be opened), 4.) define data filters, or 5.) splitters.

**Filters:** We offer filters for metadata-attributes. Figure 4.13 shows the application of filters with a stock market analysis example[4]. If a filter is applied on the meta-attribute "stock" (to filter, e.g., for specific stocks based on their abbreviations), the SOM is split into two parts: A SOM which contains only data items matching the regular expression in the filter (e.g., stock = $aa|aig|axp|ba|bac$) and a SOM which contains all remaining data items. Links connecting filter and SOMs depict the amount of data flowing in each SOM. Filters can be altered by changing the meta-attribute by which it filters or by changing the regular expression (Figure 4.13–B). All children are recursively updated and retrained as their data changed (Figure 4.13–C). This data-update and retraining process is restricted to children who have been created by split, filter or retrain options. SOMs that have been created based on a selection can not be updated due to the loss of information.

**Splitters:** We provide data splitters which divide the data based on attribute values, or meta-cluster labels. An example is shown in Figure 4.14, where yearly temperature times series (1 value per month) of the southern hemisphere[5] are analyzed (A) and retrained within a bigger SOM (B) to reveal more detailed variations. Then, a k-means meta-clustering was applied to split the yearly temperature progressions (B) and the data has been split to obtain three separate SOMs (high, medium, and low temperatures) that can be corrected manually (C). Note, that the manual annotations are back-propagated within the entire graph.

**Component Planes and Weighting.** The interestingness thumbnails can be replaced with the component planes on the left side of each SOM (Figure 4.15–A). The component planes editor (Figure 4.15–B) and a weight editor (Figure 4.15–C) can be used to apply configurations. The example in Figure 4.15–A shows that most prototypes in the bottom left corner of the SOM have relatively

[4] http://www.stockhistoricaldata.com/nasdaq, accessed 24.03.17
[5] https://data.giss.nasa.gov/gistemp/, accessed 24.03.17

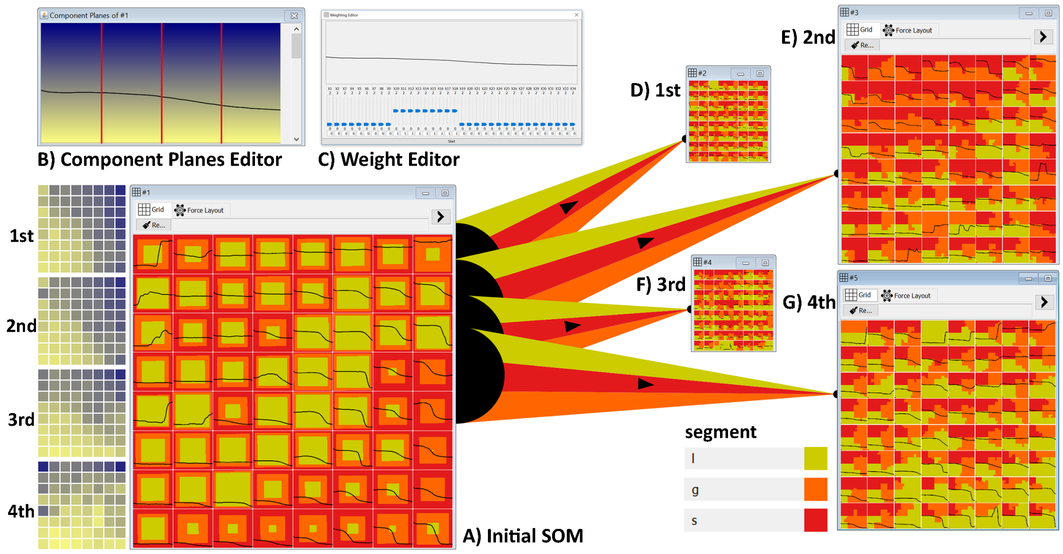



Figure 4.15: A) Initial SOM with component plane thumbnails on the left, B) component planes editor allows to generate an arbitrary number of feature components, C) custom weighting for future training can be set, D)–G) component planes 1-4: Each component plane SOM only uses the respective part of the time series for calculation. Visible differences between the second (E) and fourth (G) component with regard to the distribution of some meta-attribute.

low values and high values in the top right corner. We can also see that most prototypes are relatively stable in the first three components (1-3) and change in the end (4). The weight editor (Figure 4.15–C) can be used to apply custom weights to the SOM training and to generate new SOMs only based on the comparison of a certain part of the time series (Figure 4.15–D–G).

### 4.2.7 Evaluations

We conducted two pair analytics [12] experiments to analyze empirical linguistic datasets with a subject matter expert (SME) from the domain of speech prosodic research (intonation). The aim was to explore the datasets and to discuss the new system functionality, also compared to our previous version of the system (that focused on one single SOM) [239, 14]. All the used datasets contain a set of recorded utterances for different speakers. We use the utterance pitch-contours (i.e., a curve that tracks the perceived pitch of the sound over time) as primary data for our SOMFlow system and further information about the speaker, utterance, or the experiment as metadata.

*Apparatus:* One VA expert (VAE1, tool developer) was controlling the system guided by a linguist (SME) who had to interpret the visualizations and point VAE1 to interesting aspects. A second experimenter (VAE2) was observing the study and available for explanations and discussions. We recorded the study, saved important screenshots, and took notes. The system run on a desktop computer using a display with screen resolution of 3840x2160 pixels. The SME was familiar with the basic concepts of our system (SOM, exploratory data analysis) based on our previous collaborations. We also introduced the new SOMFlow functionalities at the beginning of the session.

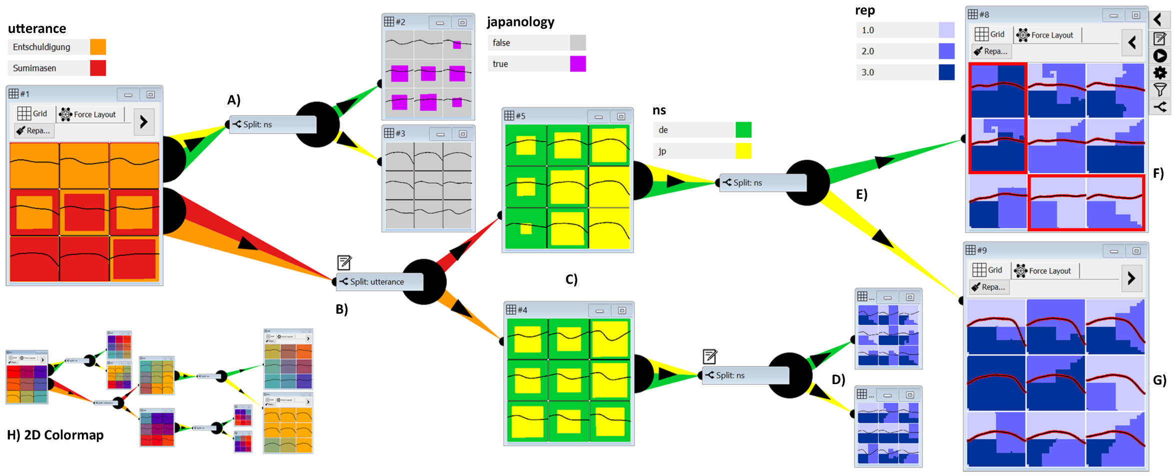



Figure 4.16: A part of the SOMFlow graph that has been produced during the first study. An unexpected finding was that some metadata is only present for one of the speaker groups and requires different labels (branch A). The system automatically pointed the analyst to speaker differences within the "sumimasen" utterances (C – SOM #5) revealing a steeper pitch fall for Japanese native speakers. Further splits and investigations (E) revealed stronger pitch variations (G) for German speakers (because they use pitch to express emotions, in contrast to Japanese native speakers). H) overlays a 2D colormap to compare all SOM cells within the flow graph.

### Study 1 – Confirmatory Analysis

The first experiment captured data to investigate to what extent a speaker's first language (German or Japanese) influenced the production of intonation when reinforcing an utterance in first and second language. To this end, German and Japanese participants produced the word "Entschudligung" and "sumimsen", both with the meaning "excuse me". The participants had to repeat each utterance three times to attract the waiter's attention within a crowded bar (with the assumption to produce more emotional utterances under increased frustration). Japanese speakers were also learners of German and vice versa. Our SME expected a significant difference between the two speaker groups within the "sumimasen" utterances, and we were especially interested in the usefulness of the recommendations that should guide the analyst to answers for this hypothesis.

*Dataset:* The dataset contains 185 recorded pitch contours (pitch value over time) with metadata about the speaker (e.g., nationality, age, etc.) and the utterance (word, repetition). The time series have been well pre-processed by the SME to make them comparable (smoothed using B-splines [68]).

*Tasks and Procedure:* We presented the SME with the initial SOM and explained all recommendations. The SME had to comment and assess the quality of each recommendation, and the task was to decide which actions are most interesting to pursue in order to derive findings and explanations from the visualizations. We were especially interested in whether the recommendations automatically point the analyst to the predicted differences within the "sumimasen" utterances.

A part of the resulting SOMFlow graph is shown in Figure 4.16. The first SOM is shown on the very left (#1, without the meta-coloring) and the first system recommendation was to train new SOMs for the cells with high *qe*, however, the SME favored to keep the current aggregation and to look for the other metadata recommendations before. These recommendations pointed the analyst to the attribute "japanology" (indicating if a speaker studied Japanese) where the SME discovered that this attribute is only tracked for one of the speaker groups. To visualize this effect, we decided to split the data based on the groups (Figure 4.16–A) and can reveal that only the German speakers contain "true" values. The SME further reported that the cells with a magenta-color filling look "more Japanese-like". By investigating the recommendations for the obtained subsets, we were able to identify further attributes that are only tracked for one of the subgroups or contained coding errors.

The next recommendation was to split the data based on the different utterances (see color overlay in SOM#1). By comparing the obtained SOMs (Figure 4.16–C) the SME was able to interpret that the "Entschuldigung" (#4) utterances have more variations than the "sumimasen" (#5) because in German pitch is primarily used to express emotions, while this is not the case in Japanese [299, 310]. To reveal speaker differences, the SME decided to use the nationality color overlay for both SOMs and to split the data further in order to investigate the utterance repetitions (1, 2, 3–blueish colors). As the SME did not find any patterns in the lower branch (Figure 4.16–D), we focused on the "sumimasen" utterances. The SOM clearly revealed that Japenese (native) speakers have a steeper pitch fall in the end (yellow cells in SOM#5). Splitting the SOM according to the speaker groups (Figure 4.16–E) also reveals that the contours produced by Germans have a stronger variation (Figure 4.16–F) than the ones of Japanese speakers (Figure 4.16–G). The SME concluded that the Germans ignored basic linguistic rules of the Japanese language [299]. Using the blueish color overlay for the repetition attribute also revealed that the Germans produced a raising pitch for the first and second repetition (politeness) and a falling pitch for the third repetition (impoliteness). The SME concluded that the German participants tried to adapt their habits to Japanese. In the end, we used the 2D-colormap to reflect the analysis.

***Results**:* Especially in the beginning of the analysis, we observed that the SME investigated each recommendation in detail. However, during an "analysis branch", the SME formed (novel) specific hypotheses that could be tested by manual color overlays, splits, or cell selections. After these hypotheses have been confirmed or rejected, the SME was able to come back to focus again on the previous recommendations (it was good to have the graph to remember that there was another recommendation). The SME also mentioned that the bar coloring (e.g., see Figure 4.8–B) is not useful for their domain because it could communicate a relation between the different parts of the pitch contours and the colors. The study took much longer than expected because most of the system recommendations were interesting and also helped the SME to identify errors in the data (e.g., missing or wrong labels for subsets where the system recommended interesting attributes). We further observed that the SME focused on the attribute-based selections and did not follow up on the SOM or cell properties. We can also confirm that the recommendations automatically pointed the SME to the predicted speaker differences in the "sumimasen" case and that the SME was able to derive further insights about the data. We also observed that the graph grew very fast and the SME liked the ability to reflect the analysis by having an overview with a 2D-colormap and linking & brushing functionality. The note taking functionality was also considered as useful, and the animated SOM training with the history overlay helped the SME to get an intuition about its function. The most interesting question of the SME during the study was if it is possible to rate the recommendations with respect to their usefulness depending on the analysis task at hand. In general, the SME was very satisfied with the results and arrived at a useful overall picture of the different groupings within the dataset.

### Study 2 – Exploratory Analysis

We used another bigger and unprocessed dataset to test which functionality of the tool is used and needed to arrive at interesting insights about the data.

*Dataset:* 7179 utterances of non-sense words (e.g., gubbu, punnu, nunnu, etc.) with High-Low-contour (HL) or High-High-contour (HH) have been imitated by 48 German learners of Japanese (=GL), 24 Japanese (=JN) and 24 German non-learners (=GN). The data contains metadata about the speakers and the pitch contours. They also contain manually annotated labels indicating if the pitch is HL or HH. The HH-condition was considered as a reference for all three participant groups. For the HL-condition, it was expected to find a difference between the groups as the contours in this condition were Japanese-specific. In contrast to the previous dataset, the pitch contours were not normalized nor smoothed. We only applied our linear interpolation processing to obtain vectors of equal length.

*Tasks and Procedure:* We started with the initial SOM and spotted a gender effect by browsing through the visual recommendation cues. Figure 4.6–A shows the metadata overlay for the gender information (sex) encoded with blueish colors. We can see that the upper right area is dominated by higher pitch values produced by female speakers and the lower left cells are dominated by male speakers that produce lower pitch values. This effect is also visible within the component planes to the left of SOM#1 where the main difference of these areas appears within the first two components (i.e., the first half of the pitch contour). The SME reported that it would be useful to apply a semitone normalization (a domain specific normalization to remove the pitch differences caused by general pitch height differences by female and male speakers). We applied this normalization and obtained a second SOM that clearly visualizes that the gender effect was removed (Figure 4.6–B), except some cells at the SOM borders that are specific to female speakers. By inspecting the SOM cells (prototypes, bandwidth, and *qe*) the SME spotted artificial contours that could be caused by measurement errors during the experiment. Therefore, we started a noise annotation graph for each selected cell of interest (Figure 4.6–C) and added a filter to create a subsequent noiseless SOM. The result (Figure 4.6–D) offers the metadata attribute "pitch" as the most interesting meta-overlay that differentiates red (HH) from orange (HL) contours, and the SME reported that the remaining mixed cells could be checked in more detail to validate these manually annotated labels. However, the SME asked us to split the data based on the pitch label (SOMs #15, #16) and we then focused on the HL data (#16) to further explore the data as it was expected to find differences between the groups in this condition. The SME was interested in the different speaker groups shown as color overlays in SOM#16. Due to the different numbers of data in the three participant groups, we trained separated SOMs for each group. The SME was now able to compare the contours across the SOMs using the linking & brushing functionality to identify speaker differences. In the end, we again zoomed out, activated the 2D-colormap and reflected the analysis steps.

***Results:*** We observed that the first part of the study focused on the data processing, SOM/cell quality, and noise removal while the second part of the study turned over to investigate interesting metadata attributes. During the study, the SME identified an uneven distribution within the speaker groups (GN, GL, JN) and the SME reported that it would be useful to see the number of items as further histograms or simple numbers within the attribute manager. Furthermore, the force-directed SOM layout was considered as "a nice feature", but it was not really used by the SME to create subsets. We also observed that the functionality of the system was overwhelming, but the visual cues and the VAEs were able to provide recommendations and explanations. Hence, we conclude that using the system and understanding the concepts/approaches requires training. Finally, the SME emphasized that our approach enables to "freely" explore the data to identify subsets of interest (that include significant effects). These findings could then be verified using conventional statistics.

The two studies demonstrated that our approach was useful to accomplish a variety of analysis tasks. Please note that we report on another data annotation study in the appendix (see Appendix A.2). However, we also received useful feedback to improve SOMFlow for the domain of prosodic research.

### 4.2.8 Discussion and Limitations

Our study and ongoing discussions revealed remaining open issues and interesting future work.

We recognized in our user studies that the SME did not fully exploit the functionality of the system (e.g., meta-clustering). Therefore, we aim to improve and fine-tune the recommendations. As a first action, we implemented sliders for our recommendation thresholds ($qeT$, interestingness rank, $simT$, $k$-clusters) to steer and test different configurations as an intermediate preparation step to leverage ML techniques to derive good recommendations from explicit user feedback ("guiding the guidance", learning the thresholds). Further improvements can be achieved by considering the current analysis state and previous decisions within a SOMFlow (e.g., by considering already selected/split attributes for interestingness calculations). We also envision to experiment with automatically starting computations of subsequent SOMs (or even complete SOMFlow branches) and to investigate how users react to such recommendations. It will also be interesting to revisit, incorporate, and compare other existing automatic approaches to create hierarchical SOMs (e.g., [230]) for our SOMFlow graph (in contrast to our human-in-the-loop approach). Furthermore, we aim to implement "semantic interactions" [82] that automatically adapt the underlying computations. E.g., we can automatically adapt the feature weighting based on manual user annotations (using relevance feedback [23]) or enable the analyst to navigate (semantic zoom) through different SOM-grid dimensions.

We focused on categorical metadata, and it would be useful to consider numerical attributes as well. On the one hand, we will implement several binning approaches to transfer numeric metadata into meaningful categories. On the other hand, we can offer further metadata color overlays and quality measures for numeric data (e.g., avg/min/max value color encoding). Similarly, the visual design of the SOMs could be further tailored and evaluated for specific data and domain requirements (e.g., removing the bar-coloring for prosodic data or adding other cell visualizations for other data types than time series).

We discussed focusing in more detail on the analytic provenance aspect of the resulting SOMFlow graph. We could map further data characteristics to the graph (besides link sizes/colors etc.). Similarly, we want to track user interactions for each graph element (e.g., hovers or clicks) that can be mapped to graph properties (e.g., node sizes). Finally, we want to experiment with different automatic layouts (e.g., temporal or SOM similarity based). This will enable us to conduct further studies with the aim to compare and evaluate visual results. Another related aspect is a collaborative analysis setting.

We noticed that the resulting graph can be used as a classifier (similar to the decision tree in [293]). It would be interesting to "keep the flow but to change (or enrich) the data" like in common ML scenarios (e.g., cross-validation, training vs. test set). We also noted that our approach "strictly" focuses on a particular dataset that is iteratively partitioned. In contrast, we can experiment with other "flow" paradigms, such as starting from multiple SOMs that merge during the analysis. Another idea would be to freely drag & drop cells to re-assign data. This would, e.g., enable the analyst to create "SOM bins" to organize the data. (e.g., put all good ones into one SOM). However, this would also require adapting the guidance (automatic recommendations) to these paradigms.

Scalability can be discussed in several ways. Firstly, computation time of the SOM algorithm depends on available recourses of the machine and increases with the size of the data (number of items and vector lengths), the SOM grid (and additional SOM parametrizations), as well as the used metric. Complexity further increases with parallel SOM computations (e.g., after splitting data) and with the quality measures (e.g., attribute interestingness). To avoid long response times, we visualize the iterative process of the SOM training [255], while threading allows continuing the analysis process in parallel to model (re) computations. Threading also allows parallel computation of multiple SOMs. Secondly, the visual and perceptual scalability of the SOM representation depends on screen size and resolution. In case of bigger SOMs (beyond the data sizes of our examples), the cell prototypes or the linking & brushing might not be visible anymore requiring further visualization alternatives to our tile based representations (e.g., aggregates, glyphs, lenses). Thirdly, SOMFlow graphs can become very complex beyond perceptual and cognitive capabilities of the human analyst. Therefore, we can further investigate graph simplification and layout techniques.

Finally, we want to emphasize that our approach is in principle not limited to SOM and could be implemented for other clustering and dimensionality reduction algorithms (or even combine several algorithm types). That would make the approach applicable to a broader range of domains and problems and additionally foster a tighter integration of automatic clustering techniques with interactive visualizations.

### 4.2.9 Conclusion

We proposed a visual interactive clustering approach with an implementation that allows the analyst to iteratively partition the data while keeping the overview. The described SOMFlow system provides a variety of visualization and interaction techniques to support four abstract exploration tasks (analyze, adapt, partition, reflect) and offers additional user guidance. We leverage quality and interestingness measures to provide the analyst with visual recommendation cues and demonstrated their usefulness in a real-world setting. Hence, we were able to derive useful findings of the data and additionally derived interesting future research areas from our observations. As a next step, we will focus on automatic recommendations and fine-tune usability issues with the ultimate goal to offer a powerful and freely available SOMFlow implementation. We demonstrated how SOMFlow instantiates our conceptual human-centered machine learning model (Chapter 4.1) in order to generate novel and more tightly intertwined visual interactive machine learning systems. In the following chapter, we will present a more specialized conceptual process model that focuses on dimensionality reduction (DR) with another instantiated system that tightly integrates a variety of DR techniques with interactive visualizations.

# 5

# Visual Interactive Dimensionality Reduction

*"If you torture the data long enough, it will confess."* – Ronald Coase, Economist

## Contents



High and multi-dimensional data constitute challenges in many current real-world data analysis scenarios. Nowadays big datasets comprise a high number of multivariate features, and it becomes a challenge to analyze such datasets due to the "curse of dimensionality" [72]. Automatic Machine Learning (ML) algorithms can be applied to analyze such high-dimensional dataset in order to identify data structures, similar data items, or clusters. However, it remains a research challenge to effectively involve the analyst in the process to arrive at a useful low-dimensional representation of the data. The first part of this chapter (Section 5.1) describes a structured literature analysis to identify seven common interaction scenarios in visual interactive Dimensionality Reduction (DR). This part is based on our TVCG publication [249] where we further specialize our general process model on visual interactive ML (Chapter 4, [244]) based on our findings for interactive DR. We apply and instantiate the resulting visual interactive DR process model in the second part of this chapter (Section 5.2) with a real application of visual interactive DR methods to crime intelligence analysis which is based on [241]. This application has been investigated within the EU funded project "Visual Analytics for Sense-making and Criminal Intelligence Analysis" (VALCRI)[1] and published as a technical report [242] and as a research paper [241] at the EuroVis workshop on visual analytics.

[1] `http://www.valcri.org/`, accessed 03.07.17

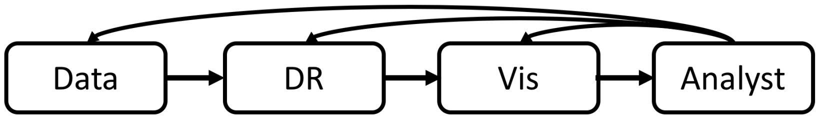



Figure 5.1: A basic DR pipeline maps data to a DR algorithm. The results are visualized and presented to the analyst. Interaction feeds back to the pipeline components.

## 5.1 A Structured Literature Analysis

Dimensionality Reduction (DR) is a core building block in visualizing high-dimensional data. For DR techniques to be useful in exploratory data analysis, they need to be adapted to human needs and domain-specific problems, ideally, interactively, and on-the-fly. Many visual analytics systems have already demonstrated the benefits of tightly integrating DR with interactive visualizations. Nevertheless, a general, structured understanding of this integration is missing. To address this, we systematically studied the visual analytics and visualization literature to investigate how analysts interact with automatic DR techniques. The results reveal seven common interaction scenarios that are amenable to interactive control such as specifying algorithmic constraints, selecting relevant features, or choosing among several DR algorithms. We investigate specific implementations of visual analysis systems integrating DR, and analyze ways that other machine learning methods have been combined with DR. Summarizing the results in a “human in the loop” process model provides a general lens for the evaluation of visual interactive DR systems. We apply the process model to study and classify several systems previously described in the literature and to derive future research opportunities.

### 5.1.1 Introduction

Dimensionality Reduction (DR) is one of the major data abstraction techniques in Visual Analytics (VA). In a typical setup, data is processed by a DR algorithm, and the output is visualized and presented to the analyst (Figure 5.1). DR aims at representing multidimensional data in low-dimensional spaces while preserving most of its relevant structure, such as outliers, clusters, or underlying manifolds [169]. DR is commonly applied to map data from many dimensions down to just 3 or 2, so that salient structures or patterns can be perceived while exploring data visually, for example, distances between data points in a scatterplot. It is also used as preprocessing for other algorithms, to improve performance by mitigating the curse of dimensionality [72].

Faced with a plethora of existing DR methods [296], it can be difficult for analysts to choose a good one, interpret the results, and apply DR to the best advantage in a broader VA process. A common approach to overcome this challenge is to involve analysts more closely, enabling them to investigate and adapt standard methods through interactive visualizations [174]. In such situations, tight integration of algorithmic techniques and visualizations is essential. Contributing tools that support this duality is one of the major goals of VA [150]. Indeed, many VA applications have been proposed that offer solutions for specific DR methods and analysis problems. In these examples, the goal is usually to support the analyst in steering the underlying algorithms through effective interactions in a visual interface (e.g., [37]), a concept that has become known as “semantic interaction” [83].

Despite these efforts, more general solutions that blend machine learning and VA still do not exist. Yet, it is these more general tools that are needed to deal successfully with real-world challenges [87, 248]. Aiming at a more general understanding of how to integrate algorithmic and visual components, a wide variety of theoretical VA models and frameworks have been proposed [59, 86, 150, 245, 248]. These models, however, often focus on high-level, abstract views, and fail to successfully characterize how a strong interplay between algorithms and visualizations would be realized and exploited.

To better understand the integration of DR and visual user interfaces, we formed an interdisciplinary group of VA and machine learning researchers. The motivating questions considered were **"Exactly how do analysts interact with the DR pipeline?"** and **"How can we incorporate our findings into the interactive DR process?"**. To answer these questions, we conducted a semi-automated review of 1850 papers from the visualization and VA literature. In the first step, 377 relevant papers were selected and subsequently reviewed to identify specific examples of how DR interactions are realized and to get a comprehensive, well-grounded understanding of the overall area. We summarize our main findings in the form of seven guiding scenarios that describe ways of combining DR with visualization (to an extent, inspired by previous work on guiding scenarios for visualization evaluation [167]) (Section 5.1.4). We also present some relevant statistics about DR and interaction techniques (Section 5.1.5). To relate our work to existing theoretical models in VA, we incorporate the findings of the literature study in a conceptual process for interactive DR [245]. We illustrate how such process models describe and support reasoning about dedicated systems, and enumerate five open research opportunities derived from our analysis (Section 5.1.6). Finally, we consider limitations of our work, and outline topics we plan to address in the future (Section 5.1.7 and 5.1.8).

### 5.1.2 Related Work

This study is related to previous work in several ways: it is concerned with general theoretical models of VA and their relationship to machine learning; it makes use of DR methods; it adopts basic ways of interacting with data visualizations; and it is related to the general idea of self-reflection in the visualization and VA community.

#### Theoretical Models

In the standard VA model [150], the discovery process is characterized by interaction between data, models of the data, visualizations, and the analyst. User interaction in this process is aimed at model building and parameter refinement. Sacha et al. [248] extended it to describe the human knowledge generation process (see Chapter 2.1). The extended process model clarifies the role of the analyst in knowledge generation, and highlights the importance of a tight human-machine integration by enabling interaction with the system. The previous process models apply to VA in a generic manner. In contrast, the study presented here focuses specifically on interacting with DR methods. Another framework describes the problem of DR as a two-stage process [59]: it first maps high-dimensional data to a lower-dimensional space, then allows another stage to reduce it to 2D for visualization. While this framework generalizes specific DR methods, it focuses on a specific application to clustered data and is limited to the two-stage process as described. The framework for observation-level interaction with statistical models [86] focuses on interaction by direct manipulation of visualization by different projection techniques. Therefore, it yields a generic approach toward interacting with the output of DR methods, which is one part of our human-in-the-loop process model; i.e., observation-level interaction directly fits in our proposed process of interaction with DR methods. Another general model is semantic interaction [83], taking acquired interaction data as a means to build user models and guide the VA system.

#### Surveys of DR and Interaction Techniques

DR maps data into fewer dimensions aiming to preserve structure like cluster gaps or local manifold continuity. In linear DR, output axes are linear combinations of original features, for example, directions of largest variation in principal component analysis (PCA), maximally statistically independent directions in independent component analysis (ICA) [128], directions of maximal between-class and minimal within-class variation in linear discriminant analysis (LDA), or directions of maximal correlation between feature subsets in canonical correlation analysis (CCA). Nonlinear DR finds either a mapping function or only output coordinates for the data set, interpreted through proximities or distances of output data; for example, mappings are sought to preserve pairwise data distances in multidimensional scaling (MDS), small distances in Sammon mapping, distances along a neighborhood graph in Isomap, or neighborhood relationships in neighbor embedding methods [296, 301]. Some methods seek mappings onto a regular grid of units as in self-organizing maps (SOMs) or generative topographic mapping (GTM). Details on PCA, MDS, Sammon mapping, Isomap, SOM, and GTM are available in books such as [169] and for LDA and CCA in [3].

Van der Maaten et al. [296] offer a comparative review of the state of the art in DR techniques, focusing on the performance of nonlinear techniques from the machine learning perspective. Similarly, Wismüller et al. [314] survey nonlinear DR, manifold and topological learning techniques. Bengio et al. [24] give an overview of representation learning in the context of deep learning. However, all the aforementioned works do not take into account VA or user interaction. A survey by Liu et al. [174] covers visualization of high-dimensional data, including DR as one of the main techniques. They include a short discussion of interaction and embed examples into the traditional visualization pipeline. However, they focus on general interaction techniques and not specifically how users interact with DR. Furthermore, they enumerate interactive model manipulation as a future research opportunity. Similarly, Buja et al. [42] review interaction techniques in the general setting of high-dimensional data visualization. Hoffman and Grinstein [124] and Bertini and Lalanne [33] discuss visualization methods for high-dimensional data mining, including projection and interaction methods. Keim [149] structures such visualization approaches according to the type of data to be visualized, the actual visualization technique, and the interaction and distortion method. However, none of these surveys performed a systematic exploration of the existing literature, nor did they focus on interaction techniques for DR.

#### Interaction Taxonomies

Our study addresses interaction in the context of DR. Therefore, related work includes general models of interaction for visualization. For example, Yi et al. [318] identify seven interaction method categories: select, explore, reconfigure, encode, abstract/elaborate, filter, and connect. Brehmer and Munzner [38] provide a comprehensive description of visualization tasks, leading to a multi-level typology of abstract tasks (which includes the ones by Yi et al.). However, model interactions only arise in tasks they call “aggregate” or “derive” tasks. Von Landesberger et al. [305] define an interaction taxonomy that is suitable for tracking and analyzing user actions in VA and provides two types of data processing interactions: data changes, such as editing or selecting data, and processing changes, such as scheme or parameter changes. In contrast, our work focuses less on a general description of user tasks, but rather on the process of interacting with DR methods.

#### Self-Reflection in the Visualization and VA Community

Because our study is based on a systematic review, coding, and analysis of previous work in the visualization and VA community, it is also related to previous work on self-reflection of empirical studies in information visualization [167], evaluation in visualization research in general [131], or affordance in human computation and human-computer interaction [66]. While we adopt the methodology of systematic analysis of previous work, our research has a very different focus.

### 5.1.3 Methods

To obtain a general understanding of visual interactive DR systems, we systematically reviewed the IEEE InfoVis, IEEE VAST, TVCG, and EuroVis literature. We first automatically identified a relevant subset of papers from these conferences and journals. Then, we carried out a qualitative

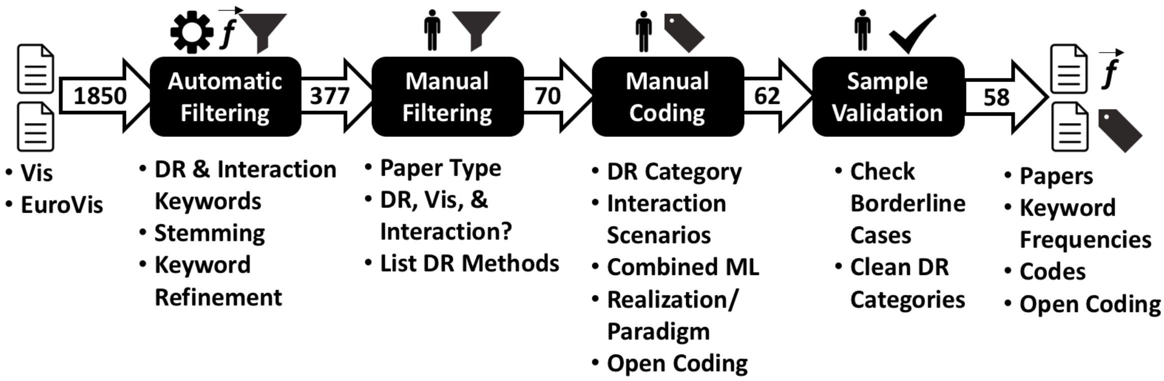



Figure 5.2: Our four stage analysis process: 1. Automated filtering, 2. Manual filtering, 3. Manual coding, 4. Manual sample validation.

in-depth analysis of the relevant papers, iteratively extracting and refining visual DR interaction characteristics. Our overall approach to this analysis was inspired by *Grounded Theory* [54], in which data is systematically analyzed until meaningful categories emerge (see Section 5.1.4). This methodological approach is based on identifying and refining categories from a representative set of qualitative data, here papers, which are then used to incrementally build up a theoretical model (Section 5.1.6). This approach has been used in visualization research [131, 167, 258] and related areas such as HCI before [125], and its importance for building up the much needed theoretical foundation in visualization has been recognized [210, 258]. We next describe our method, followed by more detailed sub-sections on our analysis procedure and findings.

### Methodological Choices

We began our endeavor with a curated list of landmark publications in interactive machine learning and visualization. Using these candidate papers, we first tried an open coding approach to identify "interesting" aspects at the intersection of VA and machine learning in general. This approach turned out to be very time consuming, and, ultimately, impractical. While it led to a high-level framework [245], our initial goals of thoroughly and systematically depicting how the VA and machine learning can be combined were largely unsatisfied. Hence we decided to analyze a much larger set of sample papers, resulting in three implications for our methodological choices. (1) We realized the need to focus on a specific machine learning problem (in our case, DR) to make the analysis more concrete, relevant, and actionable. (2) We needed automated methods to reduce the set of potentially interesting papers. (3) We opted for crisp, clear criteria for manual coding and filtering of papers. During this process, we refined the process, filtering criteria, and coding options several times. Our final workflow was then composed of four major steps, shown in Figure 5.2: *1.) Automated keyword-based paper filtering*, *2.) Manual paper filtering*, *3.) Manual paper coding*, and *4.) Manual sample validation*.

### Sample Set of Papers

Our overall goal was to identify which DR methods are used, and how interaction is implemented in the VA and visualization communities. We decided to take a representative sample of papers, constituted of all IEEE VIS papers (1221) and EuroVis papers (629) from 2005 to 2015, for a total of **1850 papers**. From EuroVis we included all full and short papers, as well as EuroVA publications. The IEEE VIS papers included all InfoVis, VAST, and TVCG papers. Our main focus was abstract, multi-dimensional data; consequently, we did not include IEEE SciVis/Vis papers in the analysis, which generally focus on 3D spatial data (e.g., flow and volume rendering).

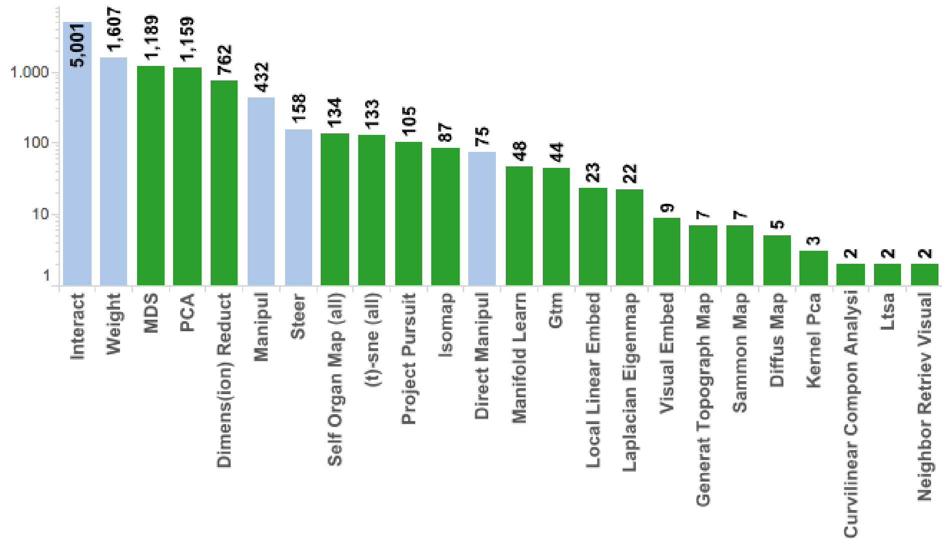



Figure 5.3: The top keyword occurrences in the automatically identified papers shown in a log-scale histogram. DR keywords are colored in green and interaction keywords are colored in light blue.

### Automated Keyword-Based Filtering

We implemented a basic NLP pipeline to analyze the initial set of papers. The pipeline parses the full text of each paper, applying a tokenizer and a snowball stemmer implemented from StanfordNLP components[2]. The same was done with keyword lists, one list for DR and another for interaction keywords. From this, a feature vector of all keyword occurrences was derived. Papers without any keyword occurrences were deemed irrelevant and filtered out, and the remaining papers were listed in a csv file with associated keyword counts. This file was the basis for the subsequent manual filtering and coding steps.

For the keyword definition, we examined previously published surveys and taxonomy papers in related fields and formed a set of primary papers in DR and interactive visualization. The keywords of these papers were extracted and processed using the NLP pipeline. A manual validation process then followed to refine the keyword lists. For example, ambiguous abbreviations (such as, *LLC*), or words that become ambiguous after stemming (such as *projection*, which stems to *project*, or *some*, which stems to *SOM*) were removed.

After the automated process, the initial set of 1850 papers was filtered to 382 relevant papers based on DR keywords, then reduced to 377 papers (108 EuroVis, 247 VIS) based on interaction keywords. Figure 5.3 illustrates a histogram of the keyword frequencies in a logarithmic scale. DR keywords are colored green and interaction keywords are shown in light blue. *Interact* is the outlier with the maximum occurrence in interaction keywords, while *MDS* and *PCA* are the most frequently occurring DR methods.

### Manual Expert Filtering

The remaining 377 papers were manually checked using the following criteria. First, we checked if the paper is a visualization application or technique paper, and if it handles "abstract data." (We intended to exclude theory and evaluation papers, as well as papers focused on unrelated or tangential topics such as volume rendering or physical flow data). Second, we checked if the paper addresses the combination of visualization, DR, and interaction, and if the interaction feeds back to the DR. For example, Joia et al. [140] present an interesting technique for sampling and feature selection. However, there is no

[2]http://nlp.stanford.edu/software/, accessed on 30.06.17

interaction that causes a recalculation of the DR. Given our focus on interactive DR, we excluded interactions that do not feed back to the analysis pipeline, such as exploration/navigation/DoD (Details on Demand) interactions. Finally, we listed employed DR techniques. Based on this, we obtained a candidate set of 70 relevant papers.

#### Manual Paper Coding

We next analyzed these 70 papers in detail, by open coding the "interesting" aspects of interaction described in each paper. For each paper, we extracted a brief description of the proposed interaction, including how interaction is performed and which parts of the DR pipeline are affected. In addition, we iteratively identified and refined a set of criteria. A more general process model on human-centered machine learning [245] (see Chapter 4.1) and the different components of the DR pipeline (data, preprocessing, DR) served as initial set of criteria to encode which parts are affected by the analysts feedback. However, we had to adapt, split, and refine these criteria several times. As a result we arrived at seven scenarios for DR interaction, encoding "how the DR pipeline is changed" (see Section 5.1.4), the interaction paradigm ("how the interaction is performed", see Section 5.1.5), the *DR Method(s) or Algorithm(s)*, and combined machine learning techniques such as *clustering* or *classification*. During our process, we had to discard several aspects that we initially were interested in. We started, for example, to encode "who" is expected to perform the actual interaction, and "why" the human input is needed. However, investigating these aspects turned out to be challenging as the necessary information was not provided in many cases. A more detailed description of the final criteria and options is provided in the following sections. 8 more papers were filtered out in this iteration.

#### Manual Sample Validation

In a final validation iteration, we aimed at more detailed analysis of borderline cases and ended up removing 4 more papers. Our final corpus included 58 relevant papers, with the encoded information and the corresponding feature vector of keyword occurrences. We "cleaned" the encoded information and grouped the identified DR methods into higher-level categories (see Section 5.1.5).

### 5.1.4 Seven Guiding Scenarios for DR Interaction

We next describe the interaction scenarios that emerged from our literature review. By examining the interactive machine learning pipeline proposed in [245] (see Chapter 4.1), we identified the main potential interactions in data analysis and classified them into seven guiding scenarios. This categorization is based on the outcome of several iterations of the paper filtering and open coding process and is one of the major findings of our study. It enables us to evaluate various methods for "how the DR pipeline is controlled through interaction". In the following, we briefly describe these seven DR interaction scenarios "along the DR pipeline" and illustrate them with examples:

**S1 Data Selection & Emphasis**: This group of interactions affects the data records (or observations) that will be supplied to the actual DR method. We found many examples in which a filter is applied to the data, and the DR pipeline is re-run on the remaining subset. In this scenario, we further identified several realizations. In some situations, analysts select subsets directly in a two-dimensional visual representation. In others, analysts specify conditions or filters through control panels. Furthermore, we identified various preprocessing configurations or parameters that can be adjusted by the analyst. An example is Jäckle et al.'s temporal MDS plot technique [133], where a parameter sets the size of a sliding window. The resulting slices are taken as input for subsequent DR by one-dimensional MDS. *S1 Data Selection & Emphasis* was identified 26 times.

**S2 Annotation & Labeling**: The second group of operations enriches data with annotations or labels on instances. In some systems, data may be enriched with additional information. For example, StarSPIRE [37] allows analysts to annotate documents with additional terms that will be included in the similarity calculation. Other systems enable the analyst to assign classification or cluster labels if the DR is combined with another form of machine learning (e.g. [119]). The cluster or classification labels, as well as data structures (such as a hierarchy obtained from hierarchical clustering), are then translated into constraints for the DR algorithm (e.g., cluster preservation).

In the classification case, the analyst provides class labels within the two-dimensional embedding for training a classifier. Labels are provided for data instances, or in some settings, for pairs of instances. The classification result influences subsequent DR (e.g., [99]). In the clustering case, the analyst defines cluster memberships, such as by grouping elements into clusters, by adding or removing elements, or by splitting or merging clusters. Resulting clusters are used by the next iteration of DR. For example, the Bubble Cluster approach [119] lets the analyst re-position points or draw cluster boundaries in a 2D projection of the data, and use the new cluster assignments to update the projection. *S2 Annotation & Labeling* was found in 15 papers.

**S3 Data Manipulation**: Some VA systems let the analyst explicitly manipulate data values by moving points in a spatialization, or by editing data in a table view. This interaction helps analysts to investigate “what if” scenarios. For example, the iPCA system [138] allows the analyst to re-position a point in the 2D projection, and see how other values change. Interestingly, Jeong et al. reported that adjusting data values could be counter-intuitive to some of the subjects in their study. However, they argued that these interactions are still useful for revealing relationships in the data that might otherwise not be recognized. *S3 Data Manipulation* was only rarely used (7 times).

**S4 Feature Selection & Emphasis**: We found many interaction examples that feed back to the initial data space by adapting the metric for calculating similarities or dissimilarities between data instances. Many DR applications adopt a “default” metric such as Euclidean distance. However, the default metric may not correspond well with the analyst’s “notion” of dissimilarity, and the metric needs to be adapted to the application. One way to do this is to associate adjustable weights with each data dimension. Distances can be calculated accordingly, giving more influence to relevant dimensions. For example, iPCA [138] provides the analyst with weighting sliders for each dimension. Another possibility is to infer the dimension loadings from direct manipulation interactions of visual elements. An example can be found in [199] where the analyst rearranges points serving as control points for a subsequent optimization of the projection matrix. Similarly, in Dis-Function [40], an analyst drags and selects points on a 2D scatterplot, and the system learns a compatible distance function. When the user is finished with manipulations, a button is pressed to learn the distance function and re-render the result. Other systems such as [204] provide analysts with drop-down menus to select a distance metric. Further options are to let the analyst determine interesting features in combination with subspace clustering (e.g., [176]) or quality metrics (e.g., [139]). *S4 Feature Selection & Emphasis* was the most frequently implemented interaction scenario (37).

**S5 DR Parameter Tuning**: Some DR algorithms contain specific parameters that can be tuned, such as LDA regularization in [60]. An approach proposed by Schreck et al. [254] allows the analyst to set the grid dimensions of a self-organizing map (number of neurons, DR structure). Some systems have parameters related to quality and accuracy, such as thresholds or level-of-detail parameters. Garg et al. [99] provide a similarity cutoff parameter that determines edges with low similarity to be removed from a graph layout. Others have parameters affecting visual appearance. For example, [74] allows adjusting node padding or forcing strength in a force-directed embedding. We also found examples where the analyst can define algorithmic variants (by setting parameters), that animate or show transitions between multiple DR results. In [199] a transition parameter (slider) is set to compare and track changes. Finally, parameter sets or configurations can be set indirectly, such as when the analyst is offered several visualization recommendations or previously defined parameter sets and may compare them to select the most appropriate one. However, we did not identify any mature, ready-to-use system incorporating this kind of parameter tuning. *S5 DR Parameter Tuning* was found in 20 papers.

**S6 Defining Constraints**: Interactions can be translated directly to DR algorithm constraints. We identified several examples in which an analyst directly arranges points in the visualization. These modified points are interpreted as anchor points in the subsequent DR iteration, in which their positions should remain fixed to help the analyst track other changes. For example, Endert et al. [86] introduced Guided MDS, where user-defined anchor points are used to fix positions and adjust similarities for maintaining consistency in visualization. A similar example can be found in [37], where nodes representing objects are marked as "fixed" and subsequently not rearranged by a force-directed algorithm. Constraints such as region or containment, as well as visual constraints have also been proposed. For example, the technique introduced by [78] allows analysts to group points and define regions that should not be split or overlap with others. In addition, constraints for the edges may be defined, such as pointing edge downward. Note, that in combination with another ML method, the ML output can be thought of as a constraint for DR, e.g., items that belong to the same cluster or classification should be placed close to each other, or a hierarchy obtained from hierarchical clustering should be preserved. In some systems, these constraints can also be interactively controlled (providing labels, setting parameters for the clustering, etc.). *S6 Defining Constraints* was described in 15 papers.

**S7 DR Type Selection**: Visual embeddings of high-dimensional data can be generated by various DR algorithms and vary in terms of layout and quality. For example, linear methods project data to new axes, such as directions of maximal variance in PCA, whereas methods such as MDS aim to preserve distances or neighborhoods of data records. While some systems, such as iPCA and StarSPIRE, focus on one DR technique, others implement multiple algorithms so the analyst can select and compare their results while analyzing data. A system by Rieck and Leitte [233] visualizes and ranks embeddings from several DR algorithms according to quality measures. Another system by Liu et al. [175] lets the analyst select DR algorithms and compare them based on visualization of distortion measures. We can even envision approaches for indirect *S7 DR Type Selection*. Although we did not find examples, it seems potentially useful to infer an appropriate DR Type from user inputs automatically, on the fly. We elaborate on this idea in Section 5.1.6. *S7 DR Type Selection* had the lowest occurrence (4) among the seven scenarios.

Note that some of the seven guiding scenarios overlap. For example, S1-S3 affect data items, and S5-S7 involve the choice of DR algorithm. However, we identified these particular scenarios as useful descriptions of the papers we studied. We found it useful to distinguish scenarios based on the way interaction affects the DR pipeline. For example, to distinguish *S2 Annotation & Labeling* from *S6 Defining Constraints* interactions, we note that both add information to data items (e.g., a class label vs. a "pinned" information), but *S2 Annotation & Labeling* focuses on information about input data items, while *S6 Defining Constraints* involves information about desired results or outputs of the DR. Note also that the role of the VA system is to translate these similar inputs to different interaction scenarios (see Section 5.1.5).

**Observations**: The final result of our coding process is shown in Table 5.1 and Figure 5.4. To provide an overview of the coded results, we created a 2D projection of the papers using Multiscale Jensen-Shannon Embedding [170], which aims to place papers with similar codes nearby in the

Table 5.1: Result of the proposed coding process. Blue, orange, yellow and green setups appear more frequently. Red cells denote papers implementing 4 different interaction scenarios. The three main column groups specify interaction scenarios, combinations with other machine learning methods, and interaction paradigms.

| Number | First Author | S1 | S2 | S3 | S4 | S5 | S6 | S7 | Clust | Class | Dir. Man. | Contr | CLI | Other | NA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Garg (2008) | | | | 1 | | | | | | | 1 | | | |
| 2 | Yuan (2009) | | | | 1 | | | | | | 1 | | | | |
| 3 | Bremm (2011) | | | | 1 | | | | | | | | | | 1 |
| 4 | Brown (2012) | | | | 1 | | | | | | 1 | 1 | | | |
| 5 | Boukhelifa (2013) | | | | 1 | | | | | | | 1 | | | |
| 6 | Yu (2013) | | | | 1 | | | | | | 1 | | | 1 | |
| 7 | Hu (2013) | | | | 1 | | | | 1 | 1 | 1 | | | | |
| 8 | Liu (2015) | | | | 1 | | | | 1 | | | 1 | | | 1 |
| 9 | Kim (2015) | | | | 1 | | | | | | 1 | 1 | | | |
| 10 | Wang (2006) | 1 | | | | | | | | | 1 | | | | |
| 11 | Brandes (2007) | 1 | | | | | | | 1 | | | 1 | | | |
| 12 | Van Ham (2009) | 1 | | | | | | | | | | 1 | | | |
| 13 | Jianu (2009) | 1 | | | | | | | 1 | | 1 | 1 | | | |
| 14 | Ward (2011) | 1 | | | | | | | | | 1 | 1 | | | |
| 15 | Yuan (2010) | 1 | | | 1 | | | | | | 1 | | | | |
| 16 | Correll (2011) | 1 | | | 1 | | | | | | | 1 | | | 1 |
| 17 | Turkay (2011) | 1 | | | 1 | | | | 1 | 1 | | | | | 1 |
| 18 | Yuan (2013) | 1 | | | 1 | | | | 1 | | | 1 | | | |
| 19 | Jäckle (2015) | 1 | | | 1 | | | | 1 | | | 1 | | | |
| 20 | Janoos (2007) | | 1 | | 1 | | | | | 1 | | 1 | | | |
| 21 | Nam (2007) | | 1 | | 1 | | | | 1 | 1 | 1 | 1 | | | |
| 22 | Peltonen (2013) | | 1 | | 1 | | | | | | 1 | 1 | | | |
| 23 | Gleicher (2013) | | 1 | | 1 | | | | 1 | 1 | | | | | 1 |
| 24 | Ingram (2010) | | | | 1 | 1 | | | | | | 1 | | | |
| 25 | Engel (2011) | | | | 1 | 1 | | | 1 | | 1 | | | | |
| 26 | Johansson (2009) | | | | 1 | 1 | | | 1 | | | 1 | | | |
| 27 | Paulovich (2011) | 1 | | | | 1 | 1 | | 1 | | 1 | | | | |
| 28 | Drieger (2012) | 1 | | | | 1 | 1 | | | | | 1 | | | |
| 29 | Portesdos (2012) | 1 | | | | 1 | 1 | | | | 1 | | | | 1 |
| 30 | Joeng (2009) | 1 | | 1 | 1 | | | | | | 1 | 1 | | | |
| 31 | Viau (2010) | 1 | | 1 | 1 | | | | | | 1 | 1 | | | |
| 32 | Gou (2009) | 1 | | | 1 | 1 | | | 1 | 1 | | 1 | | | |
| 33 | Choo (2010) | 1 | | | 1 | 1 | | | | 1 | 1 | | | | |
| 34 | Joia (2011) | | | | 1 | 1 | 1 | | | | 1 | | | | |
| 35 | Endert (2011) | | | | 1 | 1 | 1 | | 1 | | 1 | | | | |
| 36 | Cao (2011) | | 1 | | 1 | | 1 | | 1 | | 1 | 1 | | | |
| 37 | Paiva (2011) | | 1 | | 1 | | 1 | | 1 | 1 | 1 | | | | 1 |
| 38 | Garg (2010) | | 1 | | 1 | 1 | | | | 1 | 1 | 1 | | | |
| 39 | Cheng (2015) | | 1 | | 1 | 1 | | | | | | 1 | | | 1 |
| 40 | Mao (2007) | | | | | 1 | | 1 | | | | 1 | | | |
| 41 | Rieck (2015) | | | | | 1 | | 1 | | | | | 1 | | 1 |
| 42 | Schreck (2008) | 1 | | 1 | | 1 | 1 | | 1 | | 1 | 1 | | | |
| 43 | Crnovrsanin (2009) | 1 | 1 | 1 | | | 1 | | | | 1 | | | | 1 |
| 44 | Nam (2012) | | 1 | | 1 | 1 | | 1 | 1 | | 1 | 1 | | | |
| 45 | Bradel (2014) | 1 | 1 | | 1 | | 1 | | | | 1 | 1 | | | |
| 46 | Kidwell (2008) | 1 | 1 | | 1 | | | | 1 | | 1 | 1 | | | |
| 47 | Von Landesberger (2009) | | | 1 | 1 | 1 | | | 1 | | 1 | 1 | | | |
| 48 | Mamani (2013) | | | 1 | 1 | | 1 | | 1 | 1 | 1 | | | | |
| 49 | Poco (2012) | 1 | | 1 | | | 1 | | 1 | | 1 | | | | 1 |
| 50 | Paulovich (2008) | 1 | 1 | | | | | | 1 | | 1 | | | | |
| 51 | Dywer (2008) | 1 | | | | | 1 | | | | 1 | | | | |
| 52 | Chen (2009) | 1 | | | | 1 | | | | | | 1 | | | |
| 53 | Choo (2013) | | 1 | | | 1 | | | 1 | 1 | 1 | | | | |
| 54 | Liu (2014) | 1 | | | | | | 1 | 1 | | 1 | 1 | | | |
| 55 | Molchanov (2014) | | | | 1 | | 1 | | 1 | 1 | 1 | 1 | | | |
| 56 | Dywer (2006) | | | | | | 1 | | | | | | 1 | | 1 |
| 57 | Heine (2007) | | 1 | | | | | | 1 | | 1 | | | | |
| 58 | Ferdosi (2010) | | | | | 1 | | | 1 | | | | | | |
| | SUM | 26 | 15 | 7 | 37 | 20 | 15 | 4 | 28 | 12 | 36 | 33 | 2 | 1 | 12 |

projection. Together with Table 5.1, we can investigate combinations of interaction scenarios. In total, we identified 29 different combinations. We found a maximum of 4 scenarios per paper (in 4 papers, colored in red). Papers colored blue only cover *S4 Feature Selection & Emphasis*. This was the most frequent "setup" and appeared in 9 papers. The five orange dots denote papers that only include *S1*

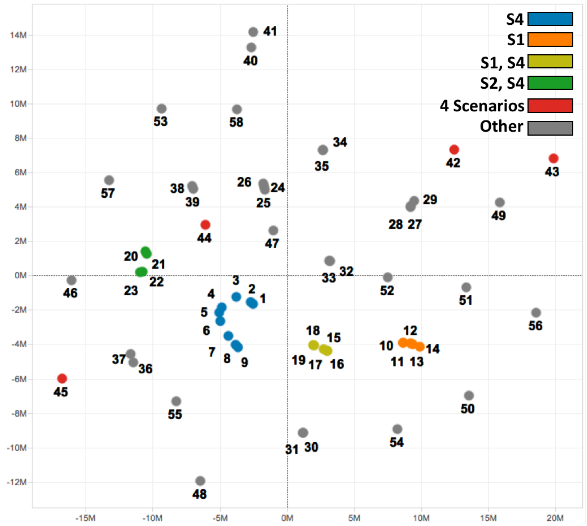



Figure 5.4: Embedding of 58 papers based on interaction scenarios. The plot shows a diverse set of interaction combinations. The main interaction scenarios are *S4 Feature Selection & Emphasis* (blue cluster), *S1 Data Selection & Emphasis* (orange cluster), the combination of S1 & S4 (yellow cluster), and S4 combined with *S2 Data Manipulation* (green cluster). The red cluster contains papers that combine 4 different interaction scenarios.

*Data Selection & Emphasis*, and the five yellow dots represent papers with combinations of *S1* and *S4*. Work applying *S2 Annotation & Labeling* and *S4 Feature Selection & Emphasis* occurred 4 times (green dots). We color the rest of the papers gray, as their combinations of interactions occur less frequently. These gray dots generally appear further away from the center of the view. For example, papers including *S5 DR Parameter Tuning* are placed in the upper area, or *S2. Annotation & Labeling* papers are placed near the upper left corner.

We further observe from Table 5.1 that some interaction scenarios appear more frequently than others. This applies to *S4*, *S1*, and *S5* maybe because they are more general or convenient than others. *S3 Data Manipulation* is used least. One reason might be that manipulating observations—often considered "ground truth"—is not common practice in many domains (e.g., machine learning). Also, note that *S3* only appears in combination with other interaction scenarios.

In this respect, it would be interesting to investigate in more detail *why* some interaction scenarios appear more or less frequently. This naturally raises the question about the *effectiveness* of certain interaction scenarios. In-depth investigation of effectiveness, however, goes beyond the scope of this study. Previous work has shown that assessing the effectiveness of interactive DR solutions depends heavily on context factors, such as users, data, domain, and tasks at hand [257, 260]. A generic comparison of the scenarios' usefulness and effectiveness is thus a non-trivial endeavor, and further work is needed. The study in this chapter is descriptive with the goal to characterize existing interaction scenarios and can be used as a starting point for such endeavors.

### 5.1.5 Further Insights

In this section, we analyze the interaction scenarios in different contexts, such as the interaction paradigm, the combined DR algorithms or other machine learning methods, as well as a temporal perspective.

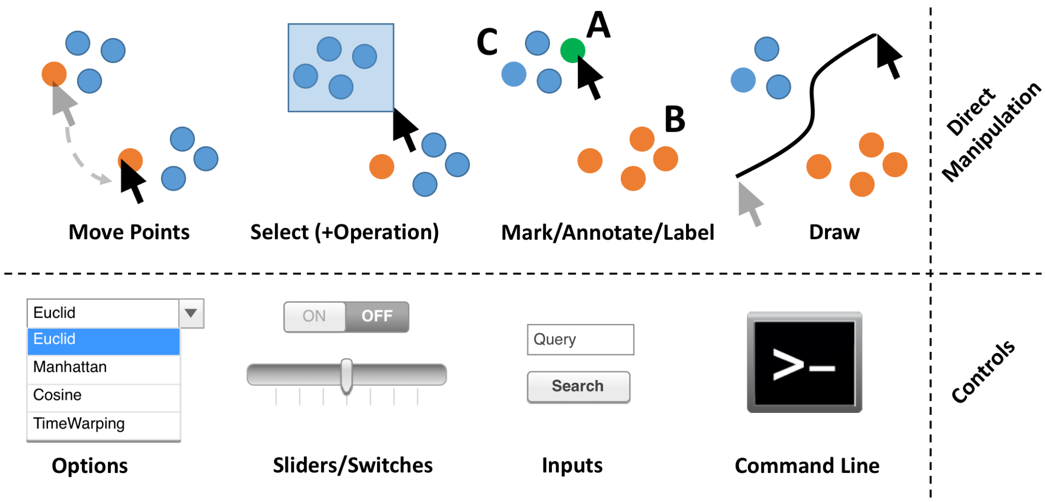



Figure 5.5: Different interaction paradigms: Typical *Direct Manipulation* interactions are shown in the upper half. On the bottom, control elements are shown. DR-Interfaces are usually composed of both.

### Interaction Paradigm & Usability

Each interaction scenario can be realized in multiple ways. Therefore, our analysis also encoded interaction paradigms, including *Direct Manipulation* of visual elements, *Controls* (sliders, buttons, etc.), *Command Line Interface (CLI)*, *Other* (such as gestures or speech input), or *NA* (if interaction was not described in the paper). The results (see Table 5.1-right side columns) reveal balanced usage of *Direct Manipulation* (36) and *Controls* (33). However, novel interaction paradigms (*Other*) only appeared once (multi-touch in [321]), and another set of papers omits details of how interaction is performed (*NA*, 12). It is also worth mentioning that the amount of provided information about the realization and implementation, as well as discussions about usability of interactions strongly varies between the analyzed papers.

Our results show that analysts interact with DR either directly in the visualization, or using control elements. During our study we noticed, especially in *Direct Manipulation*, similar actions may have different meanings or implementations (see upper half of Figure 5.5 as an example). An analyst can move points, select data records (followed by an operation such as deletion), mark (label, or annotate) points, or draw borders in a plot. However, the meaning of an action may vary. Data movement can be "translated" to *S2 Annotation & Labeling* if a point is moved outside a cluster, or to *S3 Data Manipulation* if the data value is changed. Alternatively, the movement can be "translated" to *S4 Feature Selection & Emphasis* by deriving (dis)similarities from user defined distances between data points, or *S6 Defining Constraints* if a data point being moved is interpreted as an anchor point. In such cases, visualization has to act as a "mediator" between human and machine and translate the interactions to appropriate DR pipeline components. In contrast, control elements (Figure 5.5-bottom) are usually directly coupled to specific DR pipeline components. The UI provides, for example, sliders to directly control dimension loading or DR parameters, drop-down menus to select metrics, or buttons to trigger specific operations. There are also cases where natural language text inputs are accepted. On the other hand, *Command Line Interfaces* offer a powerful, well-specified language for programmers, but they are not always convenient or even accessible to analysts.

The final implementation determines the "complexity" of performing an interaction scenario, which depends on user and task characteristics though. DR experts, for instance, might require a large set of directly steerable parameters, and accept a more complex interface. Other users, however, might require less flexibility, but simple ways to provide feedback based on their domain knowledge.

### DR & Machine Learning Algorithms

The interaction scenarios appeared with several different DR algorithms. Each algorithm was assigned to a higher-level category of *Distance Based* (DB), *Linear Projection* (LP), *Graph/Force-Directed* (FD), *Neural Network* (NN), *General*, or *Other* (one approach did not match any others – "data driven feature selection"). Table 5.2 lists these DR algorithms as columns and interaction scenarios as rows. We see that *Distance Based* methods (mainly MDS) were used alone most frequently (17), followed by *Linear Projections* (mainly PCA) alone (12) and *Graph/Force-Directed* methods alone (10). *General* approaches appear in 5 papers, whereas *Neural Networks* were used 3 times (all self-organizing maps). The other columns show examples where various DR algorithms are used in combination. Note that only one of the mixed approaches (other than *General* approaches) lets the analyst switch between DR algorithms. Interestingly, *S4 Feature Selection & Emphasis* was used in all DR algorithm combinations. Table 5.1 further shows that *Clustering* and *Classification* appeared in 31 papers. *Clustering* was used 28 and *Classification* 12 times, while in 9 papers both are used in combination.

Table 5.2: Identified DR techniques shown with interaction scenarios.

| | Distance Based | Linear Projection | Graph/Force-Dir. | General | LP & DB | Neural Network | DB & Graph/FD | LP & Graph/FD | General, NN | Other | SUM |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **# Papers** | 17 | 12 | 10 | 5 | 4 | 3 | 3 | 2 | 1 | 1 | 58 |
| **Data Selection & Emphasis** | 9 | 5 | 6 | 2 | 1 | 2 | | 1 | | | 26 |
| **Annotation/Labeling** | 4 | 5 | 4 | | | | 2 | | | | 15 |
| **Data Manipulation** | 2 | 2 | | | | 2 | | 1 | | | 7 |
| **Feature Selection & Emphasis** | 11 | 11 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 37 |
| **DR Parameter Tuning** | 5 | 3 | 3 | 2 | 1 | 3 | 1 | 1 | | 1 | 20 |
| **Defining Constraints** | 5 | 2 | 4 | 1 | | 1 | 2 | | | | 15 |
| **DR Type Selection** | | 1 | | 2 | 1 | | | | | | 4 |

Table 5.3: Temporal statistics of interaction and DR Techniques.

| | 2006 | 2007 | 2008 | 2009 | 2010 | 2011 | 2012 | 2013 | 2014 | 2015 |
|---|---|---|---|---|---|---|---|---|---|---|
| **#Papers** | 1 | 5 | 5 | 9 | 6 | 10 | 5 | 8 | 3 | 5 |
| **Data Selection & Emphasis** | 1 | 1 | 4 | 6 | 2 | 4 | 3 | 1 | 2 | 1 |
| **Annotation/Labeling** | | 3 | 2 | 1 | 1 | 2 | 1 | 3 | 1 | 1 |
| **Data Manipulation** | | | 1 | 3 | 1 | | 1 | 1 | | |
| **Feature Selection & Emphasis** | | 2 | 2 | 4 | 5 | 8 | 2 | 7 | 2 | 4 |
| **DR Parameter Tuning** | | 1 | 1 | 4 | 4 | 4 | 3 | 1 | | 2 |
| **Defining Constraints** | | | 2 | 1 | | 5 | 3 | 1 | 2 | |
| **DR Type Selection** | | 1 | | | | | 1 | | 1 | 1 |
| **SUM Interaction Types** | 1 | 8 | 12 | 19 | 13 | 23 | 14 | 14 | 8 | 9 |
| **AVG Interactions/Paper** | 1 | 1,6 | 2,4 | 2,11 | 2,17 | 2,3 | 2,8 | 1,75 | 2,67 | 1,8 |

### Temporal Perspective

We did not find any relevant papers on interactive DR in 2005. As shown in Table 5.3, in the corpus we studied, published work on visual interactive DR first appeared in 2006, with one paper that reported work in *S1 Data Selection & Emphasis*. This was followed by 5 related papers published in 2007, where a wider range of interaction techniques such as *S4 Feature Selection & Emphasis*, *S2 Annotation & Labeling*, and *S5 DR Parameter Tuning* were included. These four interaction scenarios appear to be more "established" than the others, as work was continuously reported in these areas in the following

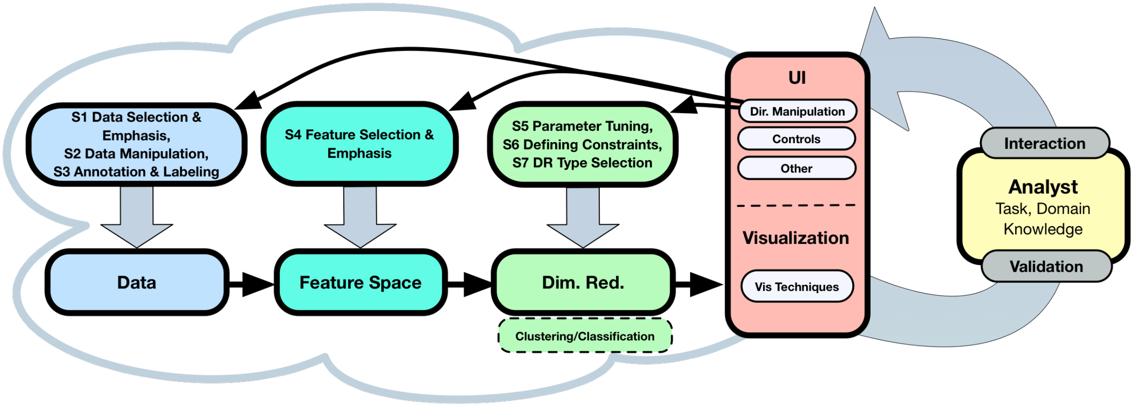



Figure 5.6: Proposed "human in the loop" process for interactive DR. The analyst can iteratively refine the analysis by interacting with the DR pipeline. The visualization interface serves as a "lens" that interactively mediates between the DR pipeline and the analyst, presenting DR results or updates and accepting feedback.

years, whereas the development of other interaction techniques has breaks in between. For example, *S7 DR Type Selection* first appeared in 2007, but then there is a gap until 2012, after which it appears consistently. We admit these trends may not be fully representative due to the limited number of papers and scenarios in our study. One obvious pattern is the number of papers published by year. Years 2009 (9) and 2011 (10) are peaks. Of course, a larger number of papers does not necessarily describe a richer set of interaction scenarios (as shown in the "AVG Interaction/Paper" row where large paper counts do not strongly correspond to large average interaction counts).

### 5.1.6 The Interactive DR Process

With the goal of making our study more broadly applicable, we summarize our findings in a general process for interactive DR in VA. This process is shown in Figure 5.6. It depicts an expanded version of the basic pipeline in Figure 5.1 and is a specialized model of our general pipeline for human-centered machine learning [245] (see Chapter 4.1). Note that this general pipeline is a superset of the process shown in Figure 5.6 and was needed to arrive at a more specialized version for interactive DR, which contains specific steps, knowledge, and details tailored to interactive DR, and is therefore much more actionable. At the top, we add the seven scenarios of interacting with DR techniques and arrange them along the analysis pipeline. *S1-S3* operate on the data, such as by changing data values or annotating labels (blue); *S4* operates on the feature space, such as by changing distance functions or the projection matrix (cyan); and *S5-S7* directly affect the DR algorithms (or additional ML models) (green). At the bottom, the results of the DR process are propagated back to the analyst (yellow).

The core of our process model is the **interactive visual interface** (red), which connects these two streams and serves as a lens for the human analyst on the algorithmic building blocks. While our work focused primarily on characterizing the forms of interaction shown by the top arrows, it is also interesting to consider how DR results can be visually presented to the user. We found dimensionally-reduced data is typically presented in scatterplots or node-link diagrams, confirming previous empirical findings [257]. Yet, our model also draws attention to the fact that other aspects of the process model can be visually represented. For instance, the dimensions (or eigenvector) can be mapped to a parallel coordinate plot [138]. Furthermore, the quality of the DR pipeline can potentially be visualized, either separately, or embedded in the low dimensional representation. Some DR types calculate or identify errors, and in combination with other machine learning methods, additional quality information might be obtained (e.g., the precision of a classifier [190]). Furthermore, different DR pipeline variants (e.g., pre-defined DR configurations or automatically built recommendations) can be visualized [129]. These different perspectives on the DR pipeline support the analyst's interpretation and validation process.

In many VA tools, the analyst has not only the ability to visually inspect and validate the data, but also the ability to provide interactive feedback to control the analysis through the interface. As discussed previously, this feedback is usually in the form of controls and direct manipulation interactions, such as setting positions, selecting, or grouping data items; other interaction paradigms such as command line scripts, gestures and speech input are also possible. The VA system maps user inputs to the specified interaction scenario(s), providing an instance of a typical continuous and iterative process, as it is usually targeted in VA [150, 245]. Note that the ability of the analyst to provide useful feedback depends on the interpretability of visual observations but also on the accessibility (implementation) of the interaction. These aspects further depend on both, the technical competence (DR expertise) and domain knowledge of the analyst, as well as the analysis task (e.g., analyzing data records vs. dimensions [257]). Especially novice analysts with less mathematical skills face problems of interpreting different DR concepts (e.g., linear vs. non-linear models) in a 2D-representation where the actual meaning of the axis is lost.

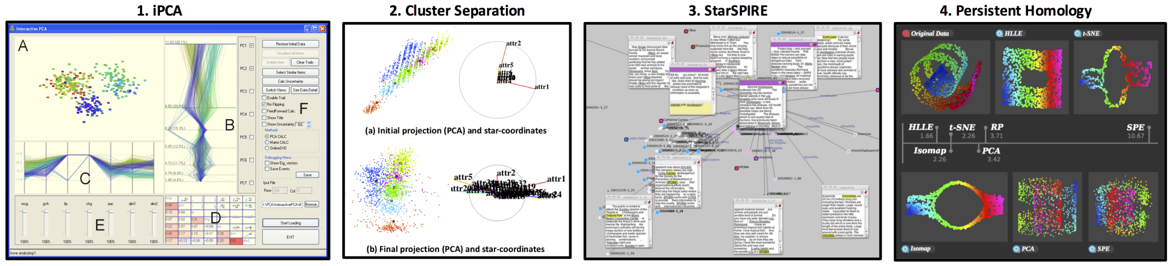
We now demonstrate how the proposed process model can be used for comparative, as well as generative purposes [21]. We first use it to describe and compare four existing examples. We then use it to identify and reason about open research opportunities.

### Descriptive Use of the Process Model – Examples

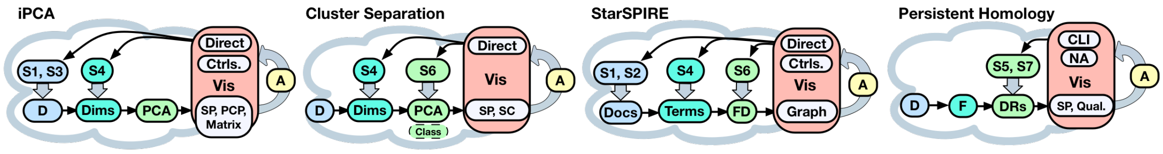
Figure 5.7b instantiates the DR process model on four examples. Their representation in the proposed model provides a consistent way to understand these systems and compare their capabilities for interaction.

**iPCA (S1, S3, S4)** The iPCA system [138] (Figure 5.7a-1) addresses typical data and feature space interactions. Several aspects of PCA are visualized in linked views, including projection, data, eigenvector, and correlation views. Each view supports a wide range of interactions including navigation, selection, and linking & brushing, however, the authors focused on three interactions that require re-computation of PCA. First, for *S1 Data Selection & Emphasis* an analyst can remove data items (e.g., outliers) and observe the resulting changes in data- and eigen-space. Second, the analyst can modify data values in some views or spaces (*S3 Data Manipulation*). Finally, iPCA offers sliders for each dimension for *S4 Feature Selection & Emphasis*, enabling the analyst to modify each dimension's contribution to the final PCA calculation. This lets the analyst test how the DR is affected by removing or "dimming" the importance of certain dimensions.

**Interactive Cluster Separation (S4, S6, other ML)** Molchanov and Linsen [199] present another way to infer feature weights from interactions (Figure 5.7a-2). They invert the process of modifying the projection matrix in a star coordinates widget by allowing the analyst to specify the desired configuration directly in the projection view (by rearranging control points). They show an example where *S4 Feature Selection & Emphasis* is inferred from direct manipulation of data points. In addition, the control points serve as *S6 Defined Constraints* for the projection. To achieve an appropriate DR output, the projection matrix is recalculated "based on an LS solution of an over determined system of linear equations". The control points can be selected by the analyst, however, the authors recommend using cluster medians or centroids for better cluster separation. This implies that the labels must be contained in the data or determined by a classifier beforehand.




(a) Images for each example.




(b) Interactive DR process model instances for each example.

Figure 5.7: Analyzed examples for DR interaction: 1.) iPCA, 2.) Interactive cluster separation, 3.) StarSPIRE, and 4.) Persistent Homology.

**StarSPIRE (S1, S2, S4, S6)** Bradel et al. [37] extend the ForceSPIRE system proposed by Endert et al. [85]. Their extension offers a richer set of interaction scenarios. A modified force-directed layout algorithm visualizes text documents under a computed similarity metric (Figure 5.7a-3). They extend ForceSPIRE with an additional model for relevance-based document retrieval that performs *S1 Data Selection & Emphasis* inferred from user interactions. The analyst can also *S2 Annotate* text documents with further information (terms) that update the similarity calculation and cause a change to the document layout. *S4 Feature Selection & Emphasis* is inferred from user interaction by adjusting the weightings of document terms. This is done in conjunction with annotation, but also by re-sizing elements, searching, highlighting and overlapping documents. In addition, it is possible to rearrange and pin document nodes in the spatialization. The pinned document serves as a *S6 Defined Constraint* for the force-directed layout.

**Persistent Homology (S5, S7)** Rieck and Leitte [233] describe an approach to comparing DR parameter settings across various DR types, such as PCA, t-SNE, HLLE and Isomap. Quality measures are computed to validate and rank the DR setup configurations. The proposed approach visualizes various DR embeddings together with additional quality information (Figure 5.7a-4). Their study does not explain in detail how an analyst would create the different combinations of *S5 Parameter Settings* and *S7 DR Type Selections* (we encoded this work as *NA*). However, several examples illustrate different DR algorithms and parameterizations created by the authors (we assume using *CLI*).

**Comparison** Figure 5.7b shows interactions supported by the above-mentioned systems. For instance, while iPCA offers the ability to *S3 manipulate* data items, StarSPIRE allows the analyst to *S2 annotate* documents with additional terms. iPCA, Cluster Separation, and StarSPIRE allow *S4 Feature Selection & Emphasis*, however, in different ways. iPCA offers slider controls directly coupled to dimension loading. Cluster Separation and StarSPIRE infer *S4 Feature Selection & Emphasis* from direct manipulation, through optimization and term weighting. Cluster Separation and StarSPIRE allow the analyst to *S6 Define Constraints* for the DR process, by positioning and pinning data items. The Persistent Homology approach focuses on the validation and comparison of different DR setups by choosing among several *S7 DR Type Selections* and *S5 Parameter Settings*.

The examples and their comparison illustrate the applicability of the proposed interactive DR process model. It supports evaluating systems with respect to the identified interaction scenarios and their implementations, and can be used to derive further interaction scenarios and implementations not present in current VA systems. We next detail 5 opportunities for research in visual interactive DR systems.

### Generative Use of the Process Model – Opportunities

We can apply our study and process model to better understand and reason about research opportunities. We recommend the following directions:

**Semantic Interaction Design** One challenge in the design of interactive DR systems is the semantic translation of front-end interactions. Section 5.1.5 illustrates that the same front-end interaction can be mapped to several different back-end computations. Ideally, intuitive interactions would direct back-end computation and correctly express the intention of the analyst. For example in StarSPIRE, by moving points closer to each other in the visualization, the analyst can have the similarity measures and the layout updated accordingly. While many systems provide good examples of semantic interaction design, the translation only applies to a subset of interaction scenarios (e.g., feature weighting, similarity computation). Consistently mapping user inputs to more complex actions covering the entire pipeline is an open challenge. Especially in DR, interaction designers have to consider that DR concepts and algorithms are often hard to understand and interpret. Therefore, interaction needs to be accessible and interpretable for end users, enabling them to work with distances and neighborhoods, clusters and class memberships, or importance of dimensions. Scalability of computation will play a crucial role in such interactive systems, as delaying responses hinder usability [49].

**Guidance on DR Type Selection** Our study revealed that *S7 DR Type Selection* has rarely been implemented. Furthermore, semantic interactions derived from direct manipulation interactions are mostly limited to DR pipeline adaptions of the feature space or DR parameters and constraints. We envision future systems that can also infer an appropriate DR algorithm from user inputs. Such VA systems would probably need to implement, calculate and compare various DR types, to identify the “best” results on the fly. Work proposed by Rieck and Leitte [233] shows a promising step in this direction. However, realization and implementation of direct manipulation interactions and translation to DR type selections is still missing. Such techniques, that balance user flexibility with system automation, have great potential for guiding users through complex data analyses, so this is an important area for further investigation. On the algorithmic side, the challenge is to formulate specific DR algorithms as parametric instances that allow smooth transitions between different DR types. For example, continuous model spaces [170, 301] enable analysts to track and interpret model switching and avoid abrupt and confusing transitions.

**Evaluating DR Interactions** As pointed out in Section 5.1.4 we are not aware of studies evaluating the effectiveness of DR interactions in a structured and general setup. It will be a challenge to design and conduct a fair comparative assessment of different interaction scenarios, as they depend on many factors, such as implementation, user experience or tasks. However, it would be useful to gather insights about the effectiveness of the respective interaction scenarios under certain conditions. This would guide researchers and developers in designing interactive DR systems for their specific domains, tasks, and data. Hornbæk provides a comprehensive overview of usability measures from HCI [125] that could be applied to a comparative DR setup.

**Fully Integrated Process** As discussed in Section 5.1.4 and illustrated in Figure 5.7b, existing systems implement only a small subset of possible interactions. While many previous systems have proven useful for specific tasks and problems, more powerful, general-purpose interactive DR tools are needed. An ideal system would provide flexible access to a range of DR algorithms, distance functions, optimization algorithms or quality metrics, and offer many of the interaction types we identified. It will be a challenge to conceptually integrate and steer a wide range of algorithm specific parameters or different combinations of computations. At the same time, the burden of choosing suitable data, features, parameters, and models could be mitigated by tightly integrating the DR pipeline with interactive visualization.

**Analytic Provenance** Given the complex nature of many analysis tasks, the analyst often has to go through many steps and even false starts before reaching sound conclusions. Although analytic provenance has been introduced as a research topic in VA, not much work has been reported on recording interactions to support exploratory data analysis for DR. A major task will be to compare and assess different DR results in a sequence of interactions. For example, when switching between different states, the resulting changes have to be observable and measurable to automatically identify impactful actions within an analysis session. Lehmann and Theisel provide a promising approach to measure the (dis)similarity of projections [173]. However, more research considering a larger set of DR types and interactions is needed.

### 5.1.7 Limitations

Our work comes with certain limitations that result from the approach we adopted. To keep the study focused and manageable, we had to limit our literature analysis to a representative set of examples. After many discussions, we decided to focus on the visualization literature. Our goal was to identify papers that include DR, interaction, and visualization. We find these mainly in the visualization community. We primarily aimed at actionable and extensible results, and with that at transparency and reproducibility by thoroughly describing our method. Nevertheless, we are confident that we analyzed a representative subset of the literature and that our derived model is stable regarding the interaction scenarios. It would be interesting to evaluate the stability of our results by performing an expanded "cross validation" study that also includes/adds papers from machine learning (e.g., KDD) and human-computer interaction (e.g., CHI). Note that we initially started our analysis with landmark publications from all domains and had to limit the set of papers to keep the work manageable.

In our analysis of the literature, we identified several contributions that offer useful interactions to explore and validate DR results, without directly feeding back to a DR calculation. We had long discussions about including these interactions as another scenario but finally decided to exclude these papers to keep the work focused. An example is a system proposed by Stahnke et al. [275] that provides interactions to interpret and interrogate DR results. Their system allows an analyst to investigate approximation errors, examine positions of data points, and "overlay" the influence of specific data dimensions. However, these interactions do not feed back to a subsequent DR calculation.

Similarly, other facets may be involved in interactive DR in specific, and interactive machine learning in general. An important facet is DR quality measures. A framework by Bertini et al. [34] describes an enriched VA pipeline with quality-metric-driven automation. Quality is measured at each stage of the pipeline, with the analyst steering the entire process. Quality measures can augment user interaction at these stages with automatic configurations or recommendations. However, quality measures do not interact with the DR pipeline and can be seen as an add-on to our proposed scenarios. Considering quality measures was a main concern when we began this study, but as the work matured, we decided to focus exclusively on interaction scenarios with DR.

### 5.1.8 Conclusion and Future Work

Giving humans more interactive control over the DR process is a great opportunity for improving exploratory data analysis. It allows the analyst to explore data, feature, parameter and model spaces, taking advantage of their understanding of the data, application domain, and experience in the analysis task at hand. In this study, we systematically analyzed the visualization literature with the goal of identifying common DR operations amenable to interactive control. We summarized our findings in seven guiding scenarios, which we contextualize in a conceptual process model for visual interactive DR. Our analysis revealed several ways that DR can be enriched by user interaction, how these strategies are supported by current VA systems, and points to future research directions in interactive DR. We hope that our contributions help other researchers investigating, designing and evaluating interactive DR systems.

In future work, we plan to develop a system capable of inferring and adapting its settings in a larger design space than current systems for visual interactive DR. We plan to extend our analysis to papers from related domains, such as machine learning and human-computer interaction. Beyond this, we would like to perform a literature analysis and process modeling study focused on interactive clustering, classification, and regression analysis in VA. In the next section, we describe our ongoing work in applying visual interactive DR to criminal intelligence analysis. The work is guided and inspired by the findings of this literature review. It focuses on an interactive cluster analysis of high-dimensional data across different DR types, parameterizations, and feature combinations.

## 5.2 Visual Comparative Case Analytics

Criminal Intelligence Analysis (CIA) faces a challenging task in handling high-dimensional data that needs to be investigated with complex analytical processes. State-of-the-art crime analysis tools do not fully support interactive data exploration and fall short of computational transparency in terms of revealing alternative results. We report our ongoing research into providing the analysts with such a transparent and interactive system for exploring similarities between crime cases. The system implements a computational pipeline together with a visual interface that allows the analysts to interact with each stage of the analysis process and to validate the result. The proposed Visual Analytics (VA) workflow iteratively supports the interpretation of obtained clustering results, the development of alternative models, as well as cluster verification. The visualizations offer a usable way for the analyst to provide feedback to the system and to observe the impact of their interactions.

### 5.2.1 Introduction

Comparative Case Analysis (CCA), also called Similar Fact Analysis (SFA) [226] is an important tool for criminal investigation and crime theory extraction [209]. Given a collection of crime reports, the idea is to analyze the commonalities between crime cases in order to support reasoning and decision making. For example, examining solved crimes that have similar characteristics as an unsolved crime may help the analyst generate a new hypothesis during a criminal investigation, and understanding the uneven distribution of crimes in terms of spaces, types of offenders and victims may help the police to allocate police resources more effectively [64]. CCA starts with the extraction of relevant headings (factors) that are considered to be useful for the understanding of the crime cases. Information is then collated under the headings, resulting in a CCA table where each row is a crime case. A main focus of the heading extraction is the extraction of features and concepts from free text fields such as the Modus Operandi (MO) of crimes. For example given the MO of a burglary case *"offender smashed a window to enter the apartment, untidily searched for money or jewelry, and exited through the main door"*, concept terms such as *"smash"*, *"window"*, *"search"*, *"money"* and *"jewelry"* may be extracted from the text and used as CCA table headings.

The work reported in this chapter addresses some challenges in CCA as part of the EU funded project VALCRI[3] that aims to develop VA tools that improve the effectiveness of current CIA solutions. We design our system in close collaboration with one police officer with data analysis background and receive feedback on a regular basis from several involved police forces across Europe. According to our police partners, traditionally CCA is carried out manually on a spreadsheet. The task becomes increasingly difficult due to the growing volume and complexity of today's crime data, especially in terms of heading extraction and pattern identification and exploration. Existing visual text analytics approaches such as IN-SPIRE [313] (and its predecessors [84, 37]), or recent works described by Ruppert et al. [237] shed light on the possibility of automatically processing textual documents to obtain and explore document clusters. Recent work by Sacha et al.[249] (see Chapter 5.1) surveyed existing visual Dimensionality Reduction (DR) approaches that let the analyst interact with different parts along the DR pipeline (e.g., [138, 199, 40, 233]). Few related works deal with the application of CCA (e.g., Zhang et. al [322]) and generic, visual intelligence data analysis systems such as Jigsaw [276] and a projection based approach presented by Jäckle et al. [134] do not allow police officers to form the customary structured tables.

We present our ongoing research on the development of a VA system to assist crime analysts in conducting CCA more efficiently and effectively. The system design is based on a number of analytical tasks we derived through the discussion with our end users, including:

***Task 1. Understand Cluster Characteristics:*** A major task of CCA is to identify groups of crimes that have similar patterns and to understand the key features that "define" their main characteristics.

***Task 2. Develop Alternative Clusterings:*** The analyst needs to be able to evaluate the clustering result. Therefore, it is essential to enable interactive exploration by letting the analyst provide feedback about important/uninteresting features or groupings.

***Task 3. Verify Cluster Robustness:*** The analyst needs to verify the robustness and stability of the clustering result. This includes examining changes of grouping caused by different feature weightings (e.g., removing or adding features) as well as checking if the clustering result is stable across different computation methods (e.g., using different DR or clustering algorithms).

Driven by these tasks, we designed a VA approach in a user-driven design study with domain experts from CIA. The system instantiates the process model for interactive DR proposed by Sacha et al. [249] (see Figure 5.6) with the aim to provide an interactive visual interface for the analyst to examine groups of similar crimes as well as their main characteristics. Figure 5.8 illustrates the process. The DR pipeline (bottom row) is embedded in an iterative exploration process (right) with several ways to provide interactive feedback to the underlying analytics (top row).

## 5.2.2 Dimensionality Reduction (DR) Pipeline

The DR pipeline takes crime reports as input, transforms the data into a binary feature vector, calculates weighted similarities and applies several DR and clustering techniques to obtain crime clusters.

**Crime Processing:** We apply a natural language processing approach to extract semantically meaningful terms from the unstructured text field ("Modus Operandi"), based on a number of seed word lists. The result is a binary feature vector where each row records the presence or absence of each term in a crime report.

[3] `http://www.valcri.org/`, accessed 03.07.2017

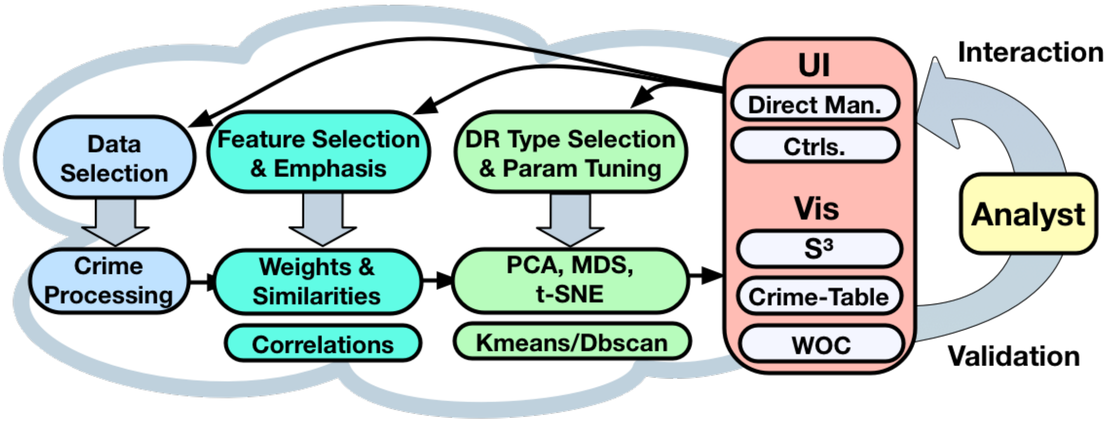



Figure 5.8: The described visual interactive DR system embeds a DR pipeline (bottom) into an iterative exploration process (right) with several user interactions (top).

**Feature Space:** A weighted similarity model multiplies each binary feature value with a weight between zero and 100. Changing the weights triggers a recalculation of the distance matrix and the DR algorithm. Euclidean distance is used to compute the distances serving as input to the distance-based DR algorithms. For linear DR techniques, we normalize the feature values according to weights to adjust the features' variance. The system also calculates Pearson correlations between features to support the analyst in understanding relationships between features.

**Dimensionality Reduction:** Three DR algorithms for generating 2D embeddings of the data are implemented, including the widely used linear approach *PCA* [214], the distance-based approaches *MDS* [162] that tries to preserve large distances in the data, and *t-SNE* [295] that aims to preserve neighborhoods.

**Clustering:** The final step of our pipeline applies *DBSCAN* clustering to the obtained embedding. The parameters can be tuned by the analyst if the clustering does not provide useful groupings. Alternatively, the analyst can set the number of desired clusters (k) and apply the *k-means* algorithm.

Note that the data processing techniques, DR/clustering algorithms, distance measures and correlation coefficient described above are only a subset of possible choices selected based on their popularity and suitability for the analysis tasks. We are still working with domain experts to evaluate and refine the selection. The computational results of the entire pipeline are passed to the visualization components. In the next section, we describe how they can be used in an interactive and iterative exploration process.

### 5.2.3 Visual Interactive Crime Case Exploration

Our work focuses on the development of a crime cluster table (CCT) that tightly integrates with different interactive visualizations of the DR pipeline (Similarity Space Selector – $S^3$). The presented components are part of a web-based framework that includes further tools to analyze crimes from different perspectives. It is possible to apply data selections based on terms, as well as, spatial, and temporal constraints. All components are linked to enable interactive data exploration (linking & brushing).


Figure 5.9: Example use case: The system can be used to understand and refine the feature weights (top) as well as the used DR and clustering configurations (bottom). Clusters can be explored and interpreted using the crime table. Changes to the similarity model and DR configuration can be tracked using animated transitions and cluster distortions.

### CCT – Crime Cluster Table

We adopted a spreadsheet based approach that comes close to the mental models of the domain experts to visualize detailed crime and cluster characteristics. The analyst is presented with an aggregated cluster representation that encodes feature frequencies in each cell of the table (clusters are represented as rows and features in columns, see Figure 5.9-steps 4,8,11). Sorting the feature columns results in comparable feature histograms for each cluster. We developed this visualization as an essential component for investigating and understanding crime clusters (Task 1). The analysts can further expand any cluster representation to reveal the detailed crimes as columns listing the contained concept terms (see Figure 5.9-step 4). Outliers without a cluster label will be listed in separate rows below the clusters. Feature weights are mapped to font size, and the user can directly adjust them within the table (by clicking on a term and changing the weight using a slider, see Figure 5.9-step 4). Updated results are then obtained from the DR pipeline (Tasks 2 and 3).

### $S^3$ – Similarity Space Selector

$S^3$ combines several visualizations of the underlying DR pipeline and allows the analyst to interactively explore and steer the computations to develop a task-driven similarity model (or spatialization) of crimes. It includes: a) a scatterplot visualizing the crimes (dots) and cluster boundaries (convex hulls), with the most frequent features of each cluster shown as labels on top of the cluster (see Figure 5.9-step 1); b) a correlation matrix for identifying highly correlated (often redundant) or mutually exclusive features (see Figure 5.9-steps 2,3) and c) an interactive bar chart that shows weights of features used for the current configuration (see Figure 5.9-step 5). The aim is to help the analyst understand characteristics of the data and the clusters (e.g., cluster sizes and shapes as well as feature weightings, Task 1). Dragging the feature bars will change the weights and trigger a recalculation of the pipeline (similar to the weight slider in the CCT). Alternatively, the analyst can click on cells in the correlation matrix to remove redundant (highly correlated) features. The analyst can switch between different DR algorithms in the control panel (top panel in Figure 5.9-step 5, Task 2). A re-computation of the embedding will be triggered each time when the analyst changes the feature weight or the DR algorithm. Animated transition of the dots is used to transform the old embedding to the new one in order to preserve users mental map and highlight the changes. Importantly, changes of weight and DR algorithm does not automatically trigger re-clustering of the data. This allows the analyst to track if clusters get distorted or repositioned in the embedding enabling the cluster verification task (Task 3). Based on the observations, the analyst can go ahead with the new clustering.

### Example Use Case

This section describes an exemplary use case that emerged from our initial requirements analysis and was refined based on several rounds of user feedback provided by our domain experts. Figure 5.9 illustrates a simple use case where the analyst starts with feature selection & emphasis (top) and changes over to DR type selection & parameter tuning (bottom) to generate clusters. In step 1, the analyst is presented with the result computed with the default configuration (*equal weights, PCA, DBSCAN*) that does not provide much to go on. Therefore, the analyst switches to the correlation plot and investigates the correlation matrix (step 2) as well as a version with sorted correlation cells (step 3). First, the analyst detects interesting relations, such as the positive correlation between *"smash"* and *"window"* and the negative correlation between *"door"* and *"window"*. The analyst also spots some redundant features (*correlation = 1*, e.g., *"door"* occurs twice in the feature vector) and removes the redundancy by clicking at the respective cells to dis-select. Subsequently, the analyst investigates the CCT to understand the clustering result and decides to increase the weight of the features *"window"* and *"door"*. After the feature selection and emphasis step, the analyst notices that the cluster gets vertically distorted (step 5). In order to double check, the analyst generates a new embedding using MDS and noticed that the crimes are re-grouped in four clusters (step 6). Continuing the investigation, the analyst tunes the parameters of the clustering algorithm and reruns the clustering. The resulting clusters are shown in step 7 and propagated back to the CCT (step 8) where the analyst spots that the clusters are mainly distinguished by two features, *"door"* and *"window"* (as intended). To verify these clusters the analyst updates the embedding by running t-SNE instead of MDS (step 9) and observes that the clusters are similar (valid), however, some sub-clusters seem to emerge. The analyst then develops a new clustering by tuning the parameters (step 10), and the results are automatically updated in the CCT (step 11). The $S^3$ projection and the CCT can now be used in combination to analyze cluster characteristics (e.g., features) and spatial properties (e.g., shape, size, and distance). These clusters can be further tested and verified by going back to other DR types. In MDS (step 12) the three main clusters (colored in light blue, blue, and purple) are still separated with the remaining clusters as subsets. Switching to PCA also reveals that these clusters overlap (from a feature perspective) and some objects are plotted on top of each other. In this way, the analyst gets a feeling about different DR types without much expertise in the computational aspects of the algorithms. By investigating the final clustering result in more detail (steps 10 and 11) the analyst finds that the three main clusters (colored in light blue, blue, and purple) cover multiple features and mainly differ in terms of the containments of *"door"* and/or *"window"*, while the other clusters represent crimes that cover only a single concept term of interest (e.g., orange – *"insecure"*, green – *"force"*, red – *"door"*). The analyst may explore these crimes further with other widgets (e.g., map) or apply other data filters to investigate clusters in more detail. Note that the feature characteristics (such as the dominance of *"window"* and *"door"* in our example) vary depending on the selected input data and analysis task (e.g., a specific region or time interval).

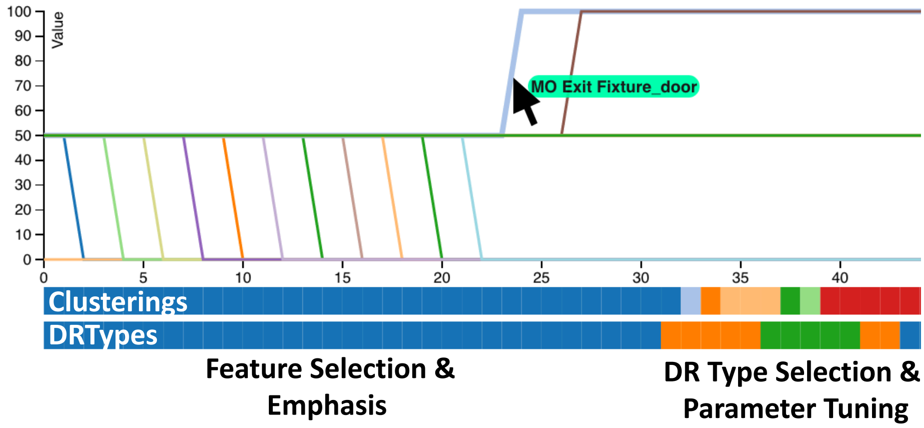



Figure 5.10: The component (WOC) tracks user defined changes to the similarity model (line chart) and DR pipeline configurations (bars).

#### WOC – Weight Observer Component

We are in the process of developing a weight observer component (Figure 5.10) that records analytic provenance [317] with the aim of capturing and evaluating user interactions [82]. The feature weights are visualized as line charts, and the bars below represent the used clustering (top bar) and DR configuration (bottom bar). Hovering over the lines or bars will reveal the tracked information (e.g., feature identifier, clustering technique with parameters, and DR type). The component can be used to understand and observe what the analyst did and which functionalities of the pipeline were used. For example, in Figure 5.10 we can track the interaction of our example use case and clearly identify the two phases of the analysis. In the beginning, the analyst changed some feature weights (line chart) before the analyst developed cluster alternatives and tested different DR types (blue bars change over to different configurations). We can use this approach to further investigate how different analyst use our tool and which interactions are used to solve particular analysis tasks. This can also be used as a history tool to recalculate the saved configurations on demand.

### 5.2.4 Discussion

The system was developed in collaboration with domain experts who provided us with feedback over a period of 1.5 years and hence, we are able to enumerate observations and lessons learned.

Our initial user interface comprised multiple scatterplots that show visual embeddings of crimes generated using different configurations (DR types, feature subsets, etc.). Without much training, our end users reported that it was difficult to understand the different results and settings. They considered the concept of DR to be very abstract and found it hard to interpret and trust the result shown in scatterplots where the “meaning of axis” is missing. Our experts reported more positive feedback after we added the crime table and focused our visual interface on a single plot that can be interactively explored. Interacting with the system and observing the changes helped the analysts to understand how the methods work and how they can interpret the obtained results. There might be a training effect, however, we also learned that it is essential to provide the analysts with tools they are familiar with (e.g., the spreadsheets) and the interpretability of the results is the key to build trust in the system and to provide useful interactive feedback. It is also worth mentioning that the system helped us (as developers) to understand the extracted data. We realized that some features occur with high frequency while others are very sparse. We will continue to refine the seed lists and introduce a threshold to “cut off” sparse features. The cut will also speed up the pipeline calculations.

Like many VA tools, the scalability of our system is limited. Our domain experts suggested a typical "targeted" analysis task (e.g., looking at crimes happened in last three months in a specific region) involves no more than 500 crimes. For our use cases, the tool worked reasonably well on 1000 crimes with 200 features. However, calculating the distances and sorting is bounded by computational complexity. We plan to improve this by applying sampling [165] or progressive approaches [93, 92] to improve the scalability.

For future work, we aim to enrich the table interactions with semantic mappings to DR pipeline adaption (inspired by Endert et al.'s work on semantic interaction [84, 85, 82]). For example, we want to allow the analyst to re-arrange columns or rows to derive feature weights. Similarly, we want to automatically derive which DR type is closest to the analyst's feedback (e.g., when the analyst declares two clusters as similar). Furthermore, the VALCRI project will move into its final phase that will focus on the deployment and integration of all partner's components, fine-tuning the data preparation, and the evaluation of the VALCRI system. Our plan is to measure quantitatively which interactions are used, to capture the analysis processes of different analysts, and to collect qualitative feedback.

### 5.2.5 Conclusions

We introduced our research in designing an interactive CCA system in collaboration with domain experts. Our DR pipeline implementation supports a variety of interactions, but we observed and learned that analysts might be overwhelmed by many visual alternatives and configuration options. To tackle this problem, we allow the users to interpret the obtained results and interact directly with the crime table (the tool that they are familiar with) that helped them to understand and importantly, build trust in the computations. Our visual interaction design is generalizable to other data types and applications. To this end, we now include additional structured metadata, such as the weekday or known offender properties (e.g., gender) in our analysis.

This example application has further demonstrated that conceptual process models can be used to design novel VA methods for specific application areas. Our goal was to instantiate the visual interactive process for DR while designing a system for real world data and analysis problems. The next chapter will summarize and discuss these contributions in a broader context.

# 6 Conclusions and Perspectives

*"Belief triggers the power to do"* – David J. Schwartz

**Contents**



Tightly intertwined visual analytics solutions that effectively combine human and machine strengths via interactive visualizations are the ultimate goal of this dissertation. To achieve this, we contribute conceptual and methodological steps towards more integrated solutions that support the entire knowledge generation process. This chapter summarizes and discusses these contributions in a broader context, describes an ontological framework for knowledge generation workflows in visual analytics, and outlines future research directions.

## 6.1 Summary of Contributions and Discussion

This dissertation demonstrates that a model-driven approach to visual analytics can lead to conceptual as well as methodological advances.

### 6.1.1 Conceptual Contributions

The conceptual research branch of this dissertation describes a series of four interdependent conceptual process models that contribute to our understanding of visual analytics processes during knowledge generation at the intersection of human and machine activities (shown in the center of Figure 6.1). The knowledge generation model for visual analytics (see Chapter 2.1) describes the core concepts of this duality and represents the foundation for the following conceptual contributions. Firstly, the knowledge generation model was specialized for uncertainty propagation and human analytic activities (see Chapter 3.1). The next iteration focused on the human-machine interplay (i.e., how humans can provide feedback along the ML pipeline) and resulted in a human-centered machine learning model (see Chapter 4.1). Finally, the latter was specialized for visual interactive dimensionality reduction (see Chapter 5.1) – a specific machine learning problem.

While these conceptual process models helped us to shape and advance our understanding of these phenomena in visual analytics, which resulted in discussions about research gaps and novel methods, it is not easy to evaluate and measure their impact and usefulness in a broader context. We envisage that these conceptual contributions will help to guide the following aspects of future visual analytics research (shown on top of Figure 6.1):

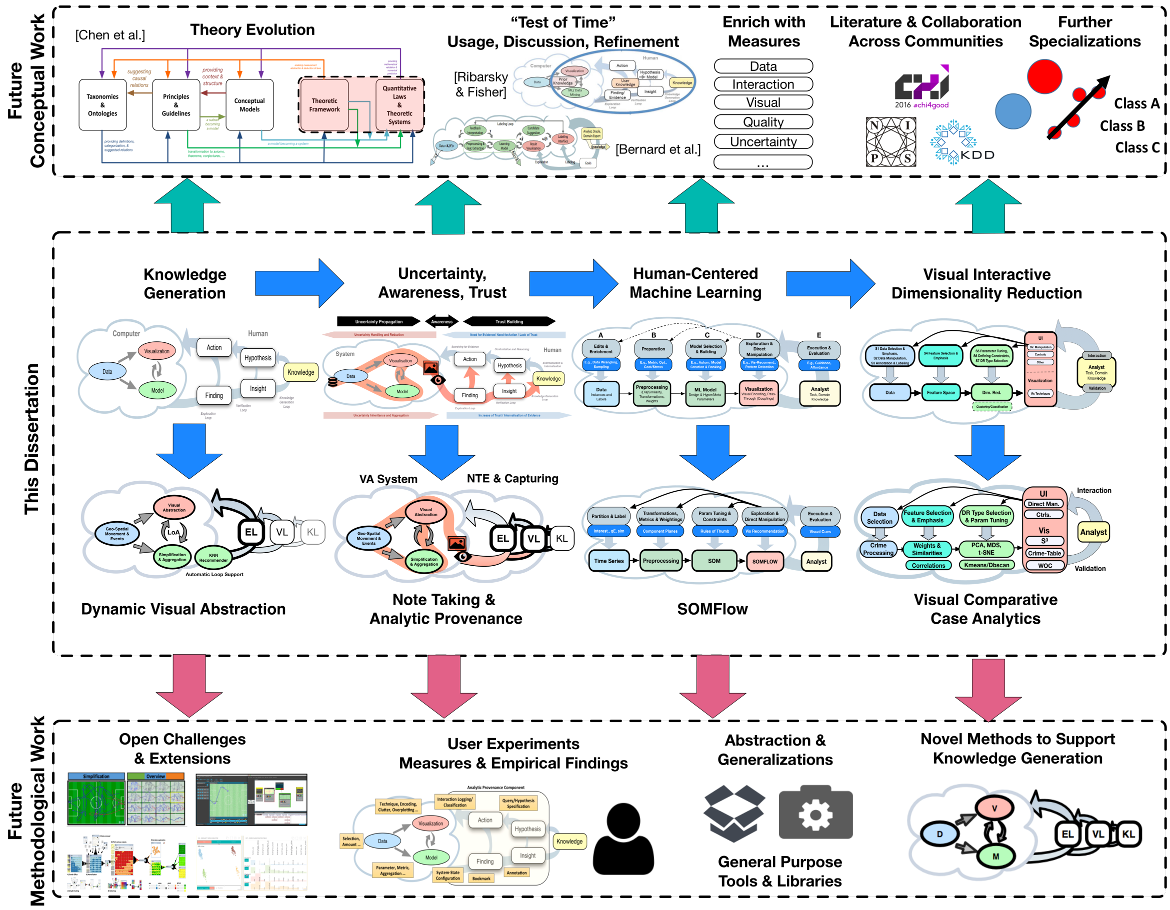



Figure 6.1: Summary of conceptual and methodological contributions (center) augmented with future work (top and bottom).

**Theory Evolution**: Considering the major aspects of a theory [57], this dissertation provides guidelines and conceptual process models including basic concepts of an ontology for visual analytics. However, the focus is still on the left-hand side of this theory evolution framework (see Figure 1.2), and more research effort is needed to investigate and create theoretic frameworks, quantitative laws, and theoretic systems. While our conceptual process models suggest ways to improve the design of visual analytics systems, such design guidelines have to stand the test of time and have to be proven mathematically and observed in real world analysis settings under the conditions of quantitative laws [57].

**Test of Time**: A comprehensive evaluation of the usefulness and impact of our conceptual work is not possible in the scope of this dissertation. The proposed process models and guidelines have to stand the test of time to repeatedly demonstrate their validity [224] and we hope that future evaluations of VA systems (which are informed by our conceptual work) will add evidence to prove their worth [71]. At this point, we can mention that our conceptual work is already well recognized in the visual analytics community. It has been included in several state of the art reports (e.g., visual analytics pipelines [308], predictive visual analytics [180], integrating machine learning into visual analytics [89], or visualization task classifications [154]) by other researchers. In addition, our conceptual works are used, discussed, and refined by other researchers. In this respect, Ribarsky and Fisher [231] discuss the human part of the original knowledge generation model and add more details with explicit reasoning steps and cognitive aspects, and the work of Bernard et al. [31] uses and refines the human-centered machine learning and visual interactive dimensionality reduction process for the case of visual interactive labeling.

**Embedding Measures**: To transition our conceptual work to theoretic frameworks or quantitative laws within theoretic systems, it will be necessary to implement measures within the entire knowledge generation process with the aim to quantitatively explore complex analytical processes. This affects technical as well as human aspects: on the computer side, we can measure data, model, or visual properties such as quality metrics [34], while on the human side, we can measure analytic effort such as interactions (see Chapter 3.2), human behavior using modern tracking devices such as eye tracking, or other cognitive aspects. The definition, collection, and comparison of such measures will be the basis for the discovery of relationships that can ultimately lead to quantitative laws within a theoretical foundation. The process models will help in connecting a variety of existing empirical and quantitative research.

**Interdisciplinary Research:** This dissertation illustrates that visual analytics involves a variety of disciplines ranging from statistics to cognitive sciences. Therefore, we found many landmark publications in different research communities, such as in data mining (e.g., SIGKDD[1]), machine learning (e.g., ESANN[2], NIPS[3]) or human-computer interaction (e.g., ACM CHI[4], HCI[5]). Whilst the major focus of this dissertation addresses the visualization and visual analytics literature (e,g, VIS/TVCG[6], EuroVis[7]), we also included relevant work from further domains, such as the cognitive sciences and psychology (e.g., reasoning and trust building in Chapters 2.1 and 3.1). Hence, the considered literature has to be extended to include all of the aforementioned research fields in order to test, extend, and validate the proposed conceptual works of this dissertation. We can, for example, further extend our structured literature analysis (see Chapter 5.1) to include all papers of the machine learning and human-computer interaction literature. We also identified many open research problems at the intersection of these research areas and argue that a closer collaboration between all involved communities will be vital.

**Further Specializations:** By iteratively refining the basic knowledge generation model for visual analytics, we have derived more specialized conceptual work (knowledge generation – uncertainty, awareness, and trust building – human-centered machine learning – visual interactive dimensionality reduction). However, we have not addressed further specializations for additional machine learning or data mining problems. We can conduct a similar structured literature analysis for clustering, regression, classification, or association rules with the aim to arrive at novel conceptual process models. Furthermore, other researchers (e.g., Ribarsky and Fisher [231]) have shown that human reasoning aspects can be refined as well. However, it will always be an important task to abstract and generalize these specializations in order to validate, integrate, and adapt more general conceptual works.

[1]`http://www.kdd.org/`, accessed on 23.08.17
[2]`https://www.elen.ucl.ac.be/esann/`, accessed on 23.08.17
[3]`https://nips.cc/`, accessed on 23.08.17
[4]`https://chi2018.acm.org/`, accessed on 23.08.17
[5]`http://2017.hci.international/`, accessed on 23.08.17
[6]`http://ieeevis.org/`, accessed on 23.08.17
[7]`http://eurovis2017.virvig.es/`, accessed on 23.08.17

### 6.1.2 Methodological Contributions

The methodological research branch of this dissertation contributes four novel methods to support knowledge generation processes. These methods are grounded in the respective conceptual process models, and they directly address identified open research areas with the aim to overcome problems during the analytical process and to support knowledge generation. Figure 6.1 illustrates the methodological contributions in the middle, each derived from a conceptual process model. The dynamic visual abstraction of soccer movement (Chapter 2.2) contributes a novel way to explore the space of possible abstractions. It achieves a tight integration of abstraction and interaction and learns the desired level of abstraction from explicit user feedback. The described note-taking environment (Chapter 3.2) supports verification processes and provides a novel way to investigate user behavior during analysis. The SOMFlow system (Chapter 4.2) is an example that tightly integrates a machine learning technique with interactive visualizations including guidance and analytic provenance. Finally, our visual comparative case analytics system for criminal analysts (Chapter 5.2) instantiates our process model for visual interactive dimensionality reduction and offers the analyst the ability to interactively develop alternative clusterings, understand the cluster characteristics, and verify cluster robustness across different computational alternatives. In the following, we discuss four major aspects of these methodological contributions (shown in the bottom part of Figure 6.1):

**Open Challenges and Extensions:** Novel methods have been demonstrated to fill specific research gaps and remaining open unsolved challenges and promising extensions have been discussed. Common open issues address the extension and evaluation of visual design alternatives which have not been considered in the respective design iteration (e.g., the trajectory thickness of the visual movement abstraction). In addition, extending the methods with further functionality and data (types) that enable the analyst to solve additional analysis tasks remain open. We also recognized and enumerated different ways to further guide and support the analyst during the analysis (such as improving the recommender system, quality metrics or incorporating semantic interactions). To verify and validate the findings, it would be useful to engage a broader set of users, data types, and application domains. Open technical challenges concern scalability (data records, features, visual, response time) and the integration of components. Finally, all methods can be further generalized and extended with alternative or additional technical methods to enhance their analytical capabilities.

**Empirical Studies and Findings:** It will be necessary to support the conceptual process models with empirical observations and findings. This requires data to be collected using appropriate measures along the entire knowledge generation process within real-world analysis scenarios. Such investigations will improve our understanding of such analytical processes and reveal relations between human and machine aspects. For example, Chapter 3.2 has shown that different user group characteristics can be identified by collecting and analyzing user interaction and trust building measures. However, we need much more comprehensive studies to advance in the field. Such empirical findings will confirm or influence our conceptual work and provide the foundation for the investigation of novel methods, with the ultimate goal to build enhanced visual analytics systems that support the analysts adaptively according to their needs.

**Abstraction and Generalization:** The investigated methods of this dissertation are tailored to specific application areas (e.g., soccer movement analysis, intonation research, crime intelligence analysis). We observed that the basic approach and methods can be transferred to other problem areas. For example, the SOMFlow system has been generalized for time series analysis or the visual crime case analytics system is capable to handle any high-dimensional binary data. The note-taking environment can be connected to any visual analytics system and the soccer analytics system could be generalized for other movement types (e.g., animal movement). Future work can further abstract and generalize the major ideas of this dissertation with the ultimate goal to arrive at general purpose tools and reusable software libraries. A closer collaboration between experts of all involved research areas (machine learning, visualization, human-computer interaction, cognitive sciences) will be necessary to address technical challenges (such as steerable and progressive computations [92, 150]) in the design and dissemination of such software components.

**Further Methods to Support Knowledge Generation:** We hope that many more novel methods can be derived from the enumerated research opportunities. Examples include methods that automatically adapt to the human‘s analytic behavior and support exploration and verification processes according to the analysts‘ needs and the current analysis state. We demonstrated that our conceptual approach leads to novel methods to support and investigate knowledge generation processes in visual analytics and suggest that our research could beneficially influence the design of future visual analytics methods in a variety of application domains.

## 6.2 An Ontological Framework for Knowledge Generation Workflows

The major parts of this dissertation focused on process modeling with the aim to evaluate existing data analysis systems and to investigate novel methods to support knowledge generation processes. Such conceptual process models rely on concepts and relations that are part of a higher-level ontological framework (see the aspects of a theoretical foundation in Figures 1.2 and 6.1–top left). Such an ontology provides a more general framework for the investigated process models that encompasses a broader set of possible data analysis processes. It is therefore more general and extensible than conceptual process models and tries to connect all the concepts (like a dictionary that can be used to describe all the processes). Other researchers have proposed (visualization) ontologies before (e.g., [75, 76]), however, this section will derive a "basic ontology" based on the scope of this dissertation and illustrate how these process models and example systems relate to each other. Please note that our intention is to provide a starting point, an extensible ontology, that can be extended and mapped out by other researchers. It can be refined with additional concepts and instances (e.g., concrete data, machine learning, visualization, or human processes and properties) or even related to other ontologies [76].

**Notation Syntax and Rules:** The elements of our ontology are illustrated in Figure 6.2. It contains the two basic concepts *Property* (*Pt*) and *Process* (*Ps*). A property represents a system state or artifact (input or output) that can be handled by processes. These elements can be connected by a *Predecessor-Successor* relationship and a hierarchical *Sub-Category/Process of* relation. Furthermore, our notation contains a simplified indication for *Knowledge* inputs (*K*) that are injected by human processes (machine-centered processes may include an optional input of knowledge if the human is involved). These simplified indications represent the human feedback loops in typical visual analytics scenarios. We introduce this simplified notation to avoid many arrow crossings that are caused by the connections from knowledge to the respective processes. We also add labels to indicate whether a process is *Human* (*H*) and/or *Machine*-centric (*M*). The same labels can be appended to properties to indicate if they are mainly stored/represented at the human or machine level.

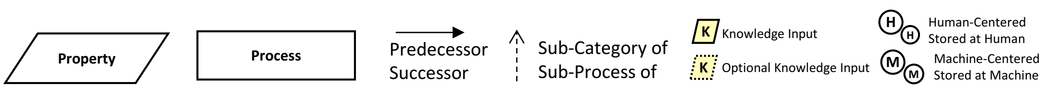



Figure 6.2: Notation elements contain properties and processes that can be connected by predecessor-successor or hierarchical relationships. Further labels indicate knowledge inputs and human or machine-centered processes.

## 6.2.1 Basic Ontology

The basic ontology (shown in Figure 6.3) is arranged in a diamond shape, similar to the visual analytics process model [150]. *Data* (*Pt*) is passed to the *Visual Data Exploration* (*Ps*) as well as to the *Data Mining/Machine Learning* (*Ps*) process. Both processes contribute to a final *Knowledge* (*Pt*) product. *Data* is mostly stored at the *Machine* and encapsulates data processing processes, while the knowledge is stored at the human and encapsulates all human cognitive processes. The *Data Mining/Machine Learning* process is a machine-centered process that can be controlled/steered by human feedback (optional knowledge). The *Visual Data Exploration* process is a human-centric process, where knowledge is injected by user interaction. In visual analytics, these processes are tightly coupled (indicated by two arrows in between). Results/structures of the *Data Mining/Machine Learning* can be passed to the *Visual Data Exploration* (bottom–up arrow), while user interactions within the *Visual Data Exploration* process can be coupled to *Data Mining/Machine Learning* adaptions (top-down arrow).

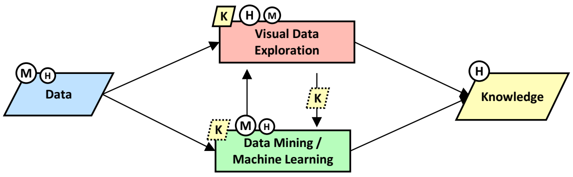



Figure 6.3: Basic ontology for knowledge generation workflows in visual analytics. Data is passed to the visual data exploration and data mining/machine learning processes, which are tightly coupled with the aim to generate knowledge.

**Mapping Out the Basic Ontology:** More details can be covered by extending the four main elements of the basic ontology. Figure 6.4 adds sub-category/processes to these elements. *Data* (*Pt*) can be described as *Data* (*Pt*) that is passed to a *Data Processing* (*Ps*) process that delivers *Data Structures* (*Pt*), while the *Data Processing* (*Ps*) is coupled to a human-centered *Edits & Enrichment* (*Ps*) process that allows the human to inject knowledge (e.g., by adding labels or data selections). The *Data Structures* (*Pt*) are passed to *Data Mining/Machine Learning* (*Ps*) which can be further described with two sub-processes (bottom part of Figure 6.4). A *Pre-Processing* (*Ps*) delivers *Prepared Data* (*Pt*) (e.g., distances) for the subsequent *DM/ML Model* (*Ps*) process that generates *DM/ML Structures* (*Pt*) (e.g., clusters, outliers, etc.). The human analyst can interact by *Preparation* (*Ps*) (e.g., feature selection & emphasis) and *Model Building & Selection* (*Ps*) (e.g., parameter tuning) processes in order to steer the *Data Mining/Machine Learning* (*Ps*) process. *Data Structures* (*Pt*) and *DM/ML Structures* (*Pt*) are passed to the *Visual Data Exploration* (*Ps*) process, which composes of a *Visual Mapping* (*Ps*) providing *Visual Structures* (*Pt*) and a *Rendering* (*Pt*) process to produce *Visual Imagery* (*Pt*). The human is using the visual interface to interact within this process by *Encoding & Configuration* (*Ps*), *Navigation & Transformation* (*Ps*), and *Direct Manipulation* (*Ps*) activities. All human feedback processes within *Visual Data Exploration* (*Ps*) can be coupled to adaptions within the *Data Mining/Machine Learning* (*Ps*) or *Data Processing* (*Ps*) processes. This concept is also known as semantic interaction [82] and indicated by the top-down arrow between *Visual Data Exploration* (*Ps*) and *Data Mining/Machine Learning* (*Ps*). For example, the (semantic) zooming (*Navigation & Transformation*) within the dynamic visual soccer movement abstraction system (Section 2.2) is directly coupled to parameter tuning and abstraction type selections (*Model Building & Selection*). Finally, the *Knowledge* (*Pt*) property encapsulates the human cognitive processes that are involved to generate knowledge from data (right-hand side in Figure 6.4, shown within a circle). It encompasses the human processes that are described on the human side within the knowledge generation model in Sections 2.1 and 3.1. We decided to arrange them within a circle to indicate that these processes are unstructured and can not be described with *Predecessor-Successor* relationships (they may take place in parallel and influence each other, allowing connections everywhere). However, we arranged the human concepts of the knowledge generation model at the circle border to indicate intermediate stages/properties of knowledge within the human cognitive process.

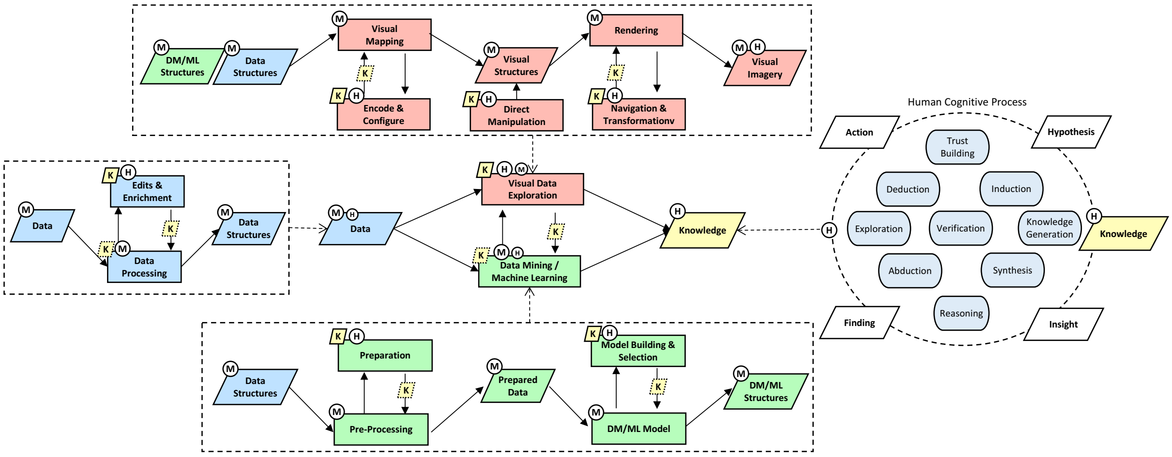



Figure 6.4: Detailed sub-processes and properties are added (outside) for the four major elements (inside) of the basic ontology.

**Referring to the Conceptual Process Models:** This ontological framework can be used to illustrate how the chapters and described conceptual process models relate to each other: Chapter 2.1 provides the basic context of combining human and machine concepts along the knowledge generation process in visual analytics (see, e.g., Figure 2.3 compared to Figure 6.4). Chapter 3.1 puts a sharper

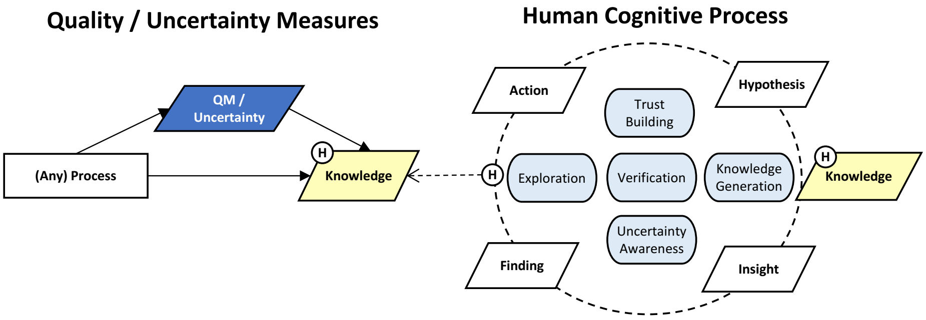



Figure 6.5: Uncertainty, awareness, and trust building within the ontology.

focus on uncertainties and human trust building processes. Figure 6.5 adds potential quality measures *QM/Uncertainty* $(Pt)$ as additional outputs of any processes $(Pt)$ that are propagated to the *Knowledge* $(Pt)$ property that encapsulates the human cognition process with uncertainty awareness and trust building activities. Chapter 4.1 focuses on visual interactive machine learning, especially on how the human-centered processes feed back to machine-centered processes along the *Data Processing* $(Ps)$ and *Data Mining/Machine Learning* $(Ps)$ pipeline (it integrates the top and bottom part within Figure 6.4). Chapter 5.1 specializes the previous chapter for visual interactive dimensionality reduction in providing more detailed building blocks of the dimensionality reduction pipeline with more specific interaction scenarios (see Figure 6.6).

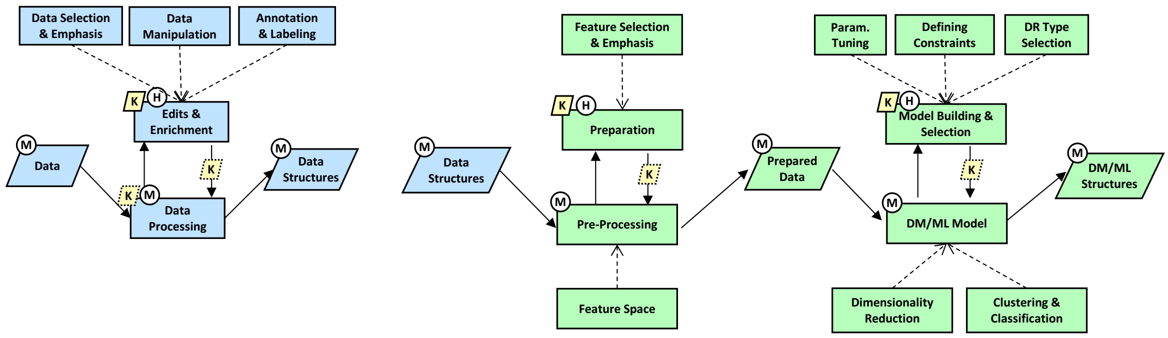



Figure 6.6: The *Data Processing* and *Data Mining/Machine Learning* parts are specialized for visual interactive dimensionality reduction.

### 6.2.2 More Details and Example Workflows

In this section, we illustrate, within the scope of this dissertation, that it is possible to iteratively continue the extension and refinement process of the ontology.

A first iteration has been carried out to revisit all the previous chapters of this dissertation and to add the identified processes to the ontology. The result is shown in the appendix in Figure A.4. It encompasses all the *Data Processing*, *Data Mining/Machine Learning*, or *Visual Data Exploration* processes in more detail and can be used to identify specific pathways within this ontology. In the following, we will briefly illustrate such workflows for each visual analytics system of this dissertation (the visual abstraction system, SOMFlow, and the visual comparative case analytics system). We will focus on the description of how the human-machine interplay (coupling) is realized and where human knowledge can be injected.

**Dynamic Visual Abstraction of Soccer Movement (Figure 6.7, Chapter 2.2):** Movement data is processed, passed to the weighting/similarity calculation, and then simplified and/or clustered. The *Visual Data Exploration* process allows the analyst to navigate within the space of possible abstractions via semantic zooming (*Level of Abstraction* is coupled to abstraction *Type Selections* and *Parameter Tuning*). Further configuration options can be performed via controls in the user interface. It is further possible to directly select trajectories or time intervals to be visualized. Another explicit user feedback (“Learn from Current View”) trains a recommender system (based on the current configuration options and state measures, such as visual crossings or data size) to automatically pre-configure the visual abstraction.

**SOMFlow (Figure 6.8, Chapter 4.2):** Time series are processed, passed to the preprocessing, and then used to train the self-organizing maps algorithm (SOM). The resulting SOM grid with the cells and the artificially generated prototypes are used to classify the data records (time series). Further quality/interestingness measures are computed and passed to the *Visual Data Exploration* process which allows to analyze the SOM result and to reflect the analytic process (e.g., compare previous states/results). The analyst can adapt the computations (SOM parameters, constraints, or preparation) via direct manipulation interactions within the SOM visualization or controls in the user interface. The system explicitly supports different data partitioning tasks supported by visual recommendation cues. Each data partitioning action results in a re-computation of the pipeline, which is embedded into a SOMFlow graph (analytic provenance). The analyst can further assign user-defined data labels.

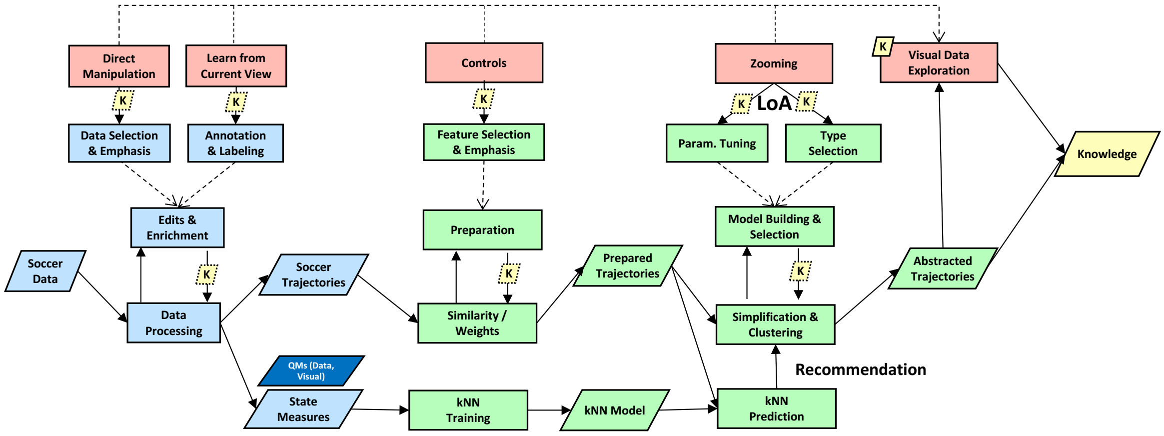



Figure 6.7: Dynamic Visual Abstraction of Soccer Movement Workflow: Human feedback stems from the *Visual Data Exploration* process. A tight human-machine coupling is achieved by the *Level of Abstraction* (LoA) and a recommender system to learn the degree of abstraction on-the-fly.

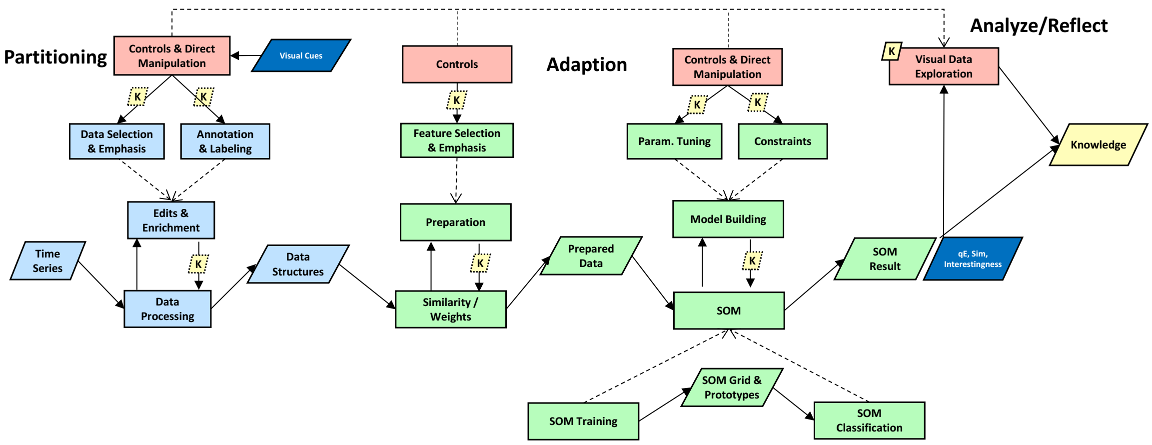



Figure 6.8: SOMFlow Workflow: The analyst can adapt the SOM-preprocessing, the SOM-parameters, or provide constraints. Data partitioning interactions are explicitly supported by visual recommendation cues. Each feedback iteration is embedded into a SOMFlow graph.

**Visual Comparative Case Analytics (Figure 6.9, Chapter 5.2):** Features are extracted from crime reports, passed to the preprocessing (weighted similarities, Pearson correlations), and then used to project and cluster the data. The analyst can switch between the dimensionality reduction and clustering algorithms, tune respective parameters, and apply feature weightings in order to develop and evaluate alternative clustering results. Additional components (such as a search bar, timeline, or map) allow the analyst to select crime reports of interest. This tightly integrated *Visual Data Exploration* process supports criminal analysts in developing alternative clusterings, understanding the cluster characteristics, and evaluating the cluster robustness across algorithmic configuration options.

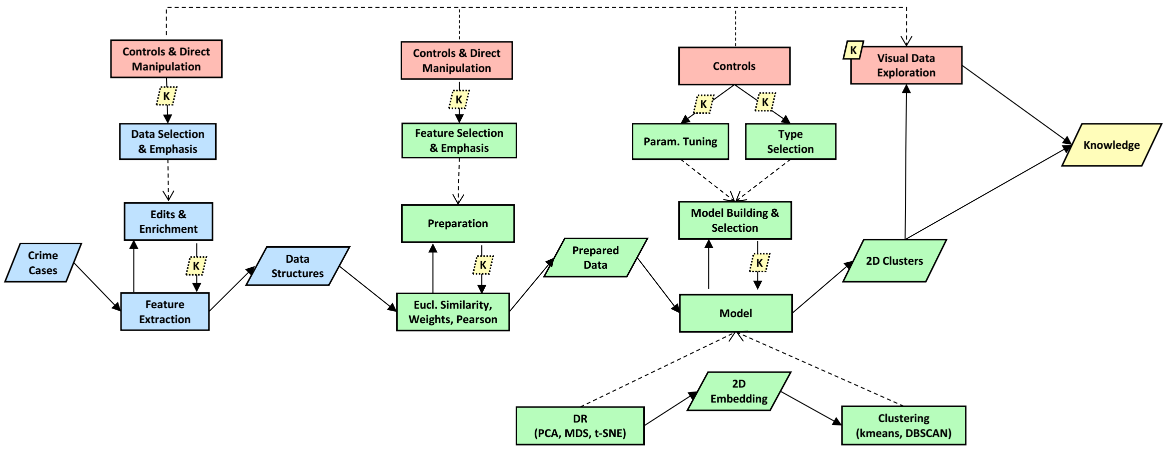



Figure 6.9: Visual Comparative Case Analytics Workflow: The analyst can switch between different dimensionality reduction and clustering configurations and apply feature weightings.

This subsection illustrates that the ontology can be iteratively extended and instantiated for many concrete data analysis scenarios and systems. We limited our endeavor of mapping out the concepts and of applying examples to the scope of this dissertation. It will be an interesting task in the future to conduct a more comprehensive iteration by applying more example systems and by considering more literature. We have successfully used this ontological framework to illustrate how the conceptual process models and example systems relate and build upon each other.

## 6.3 Concluding Remarks

This dissertation emphasized that visual analytics is not just about computer applications with visual data representations. It is a collaborative process that comprises of human and machine analytical activities with the aim to generate knowledge from data. The major research question of **"How to achieve a tight integration between automated analysis and visual interaction to better support human knowledge generation in VA?"** has been addressed with conceptual process models that identify ways or "handles" that are amenable to interactive control in order to effectively involve the human (and expert knowledge) into complex analytical processes. The conceptual process models have been used to highlight specific research areas that aim to support human knowledge generation processes, such as semi-automation and recommendation, analytic behavior and trust building, and visual interaction with machine learning. This dissertation provides exemplary solutions to these research areas and proposes directions for further investigations. We hope that our perspective and model-driven approach to visual analytics will guide and inspire other researchers of all involved research communities, with the ultimate goal to establish a joint interdisciplinary theoretical foundation for visual analytics that will lead to more tightly integrated visual analytics methods that support the analysts adaptively according to their needs.

# List of Figures

# List of Tables

# A
# Appendix

## A.1 Expert Drawings Produced During the Evaluation

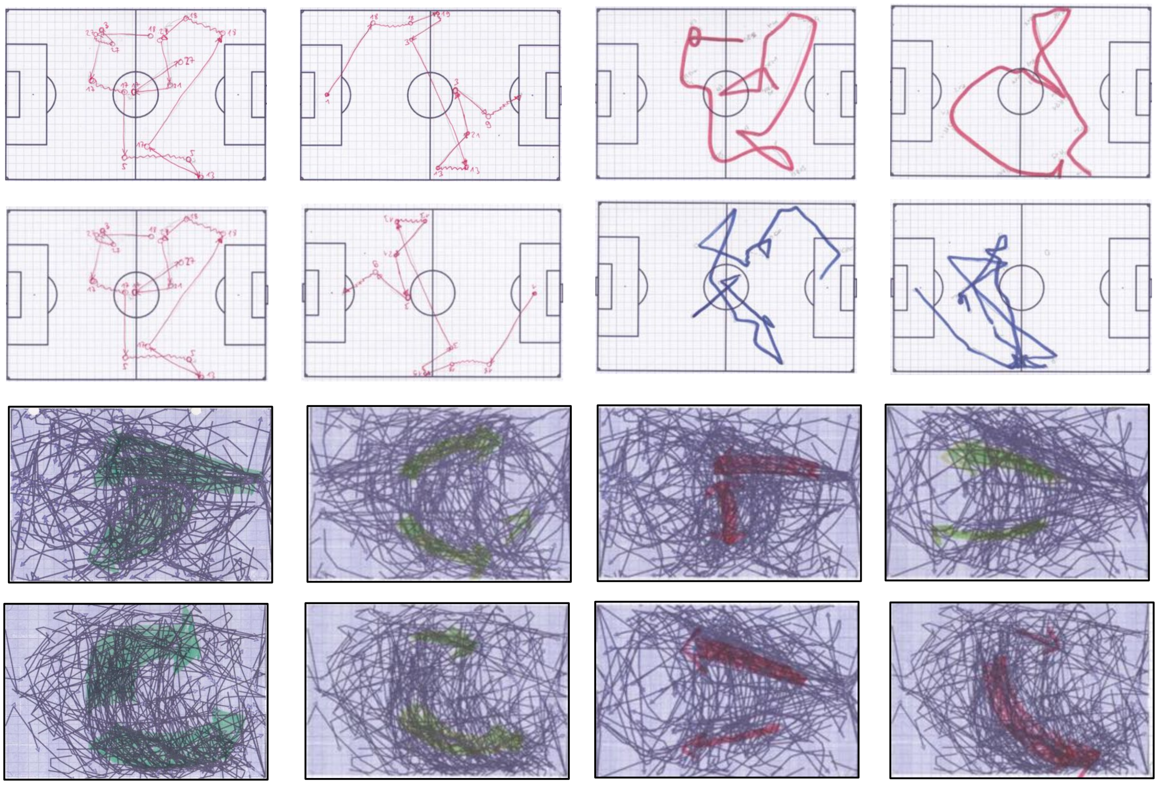


Figure A.1: Expert drawings comparable to the detail, simplification, and aggregation layer.

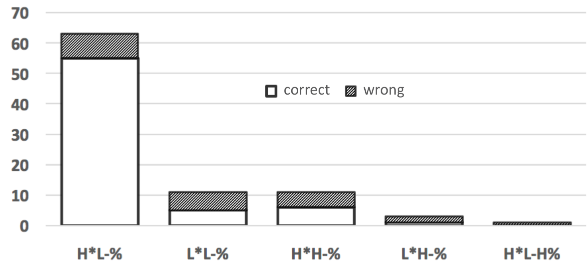



Figure A.2: The annotation results of the second task. 75% of the data have been labeled correctly without listening to any recordings (what the SME usually does for each). The whole task was done within 15 minutes instead of 3 hours.

## A.2 SOMFlow – Annotation Study

This study aimed to test if our SOMFlow system can be used for annotation tasks (assuming we have recorded utterances without metadata).

*Dataset:* We used the "sumimasen" utterances of the first dataset (89 items, see study 1) to keep the amount of manual work manageable.

*Tasks and Procedure:* The preparation task for the SME was to create a manually annotated "ground truth" (or gold standard) for the pitch contours using the ToBI annotation system [22] provided with a visual representation and a recording of the pitch contour. The SME had to assign one of the ToBI-labels $l \in \{$H*L-%, L*L-%, H*H-%, L*H-%, H*L-H%$\}$, each label describing the pitch progression of the utterance (H = high, L = low, * = accented syllable, % = utterance end). E.g., H*L-% is used for utterances that start with a high pitch and include a steep fall in the end. The time needed for the manual annotation task was 3 hours. The actual task in our study was to create the same labels using our SOMFlow system. We started with an initial standard SOM and provided the SME with all the functionality and recommendations that were needed to validate the quality of cells/clusters (e.g., *qe*, u-matrix, recommendations). The task for the SME was to apply a class label to "good" or "homogeneous" cells (with a clear annotation category, indicated by the prototype and bandwidth overlay) and to create a new SOM iteration for "imprecise" or "heterogeneous" cells until all items have been labeled. We measured the time and exported the dataset containing the assigned labels. The SME was not able to listen to the recordings in order to purely evaluate the visual representations of our system.

***Results**:* The whole task has been accomplished within 15 minutes and the overall accuracy of the annotation task was 75% as compared to the ground truth. An overview of the labels and errors is shown in Figure A.2. We can see that most of the contours have been labeled H*L-% and annotation errors have been made in every category. Figure A.3 shows the resulting annotation graph (right) and the overlay of correct (green) and wrongly (red) annotated pitch contours for the first SOM. The graph shows that the SME produced sub SOMs for the cells that needed to be investigated in more detail. We observed that the SME used the *qe* together with the bandwidth overlay to judge about the cell quality. The recommended cell extensions supported the SME to identify similar cells, however, the SME did not always extend the current selection. From the left SOM, we can derive that the SME was able to annotate the pitch contours of the first and second rows without any errors. However, the last row contains a lot of errors. The SME also reported that for some (hard/noisy) contours it is impossible to assign class labels without the ability to listen to the recordings. Therefore, the SME was very satisfied with the result, also because the errors concentrate on specific cells within the initial SOM. The SME

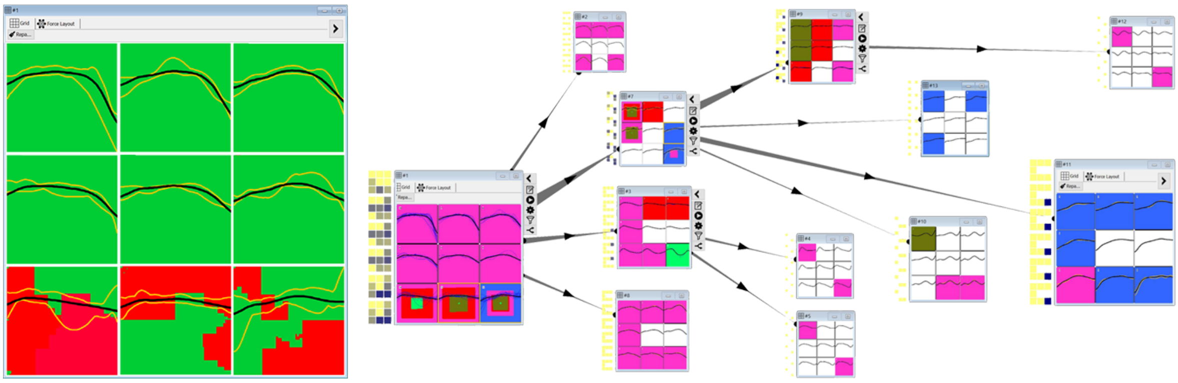


Figure A.3: Result of the annotation task: Left – correct (green) and wrong (red) values are shown as a SOM meta-overlay, right – final annotation graph that was produced during the study with the class labels as meta-overlays. We can clearly see that the SME was able to annotate 2/3 of the SOM without any problems (upper two rows). However, many errors occurred within the last row.

was optimistic that these errors could be mitigated by providing the SME with the ability to listen to the audio recordings within our system what would still save the SME a lot of time for their annotation tasks.

## A.3 Mapping Out the Proposed Ontology

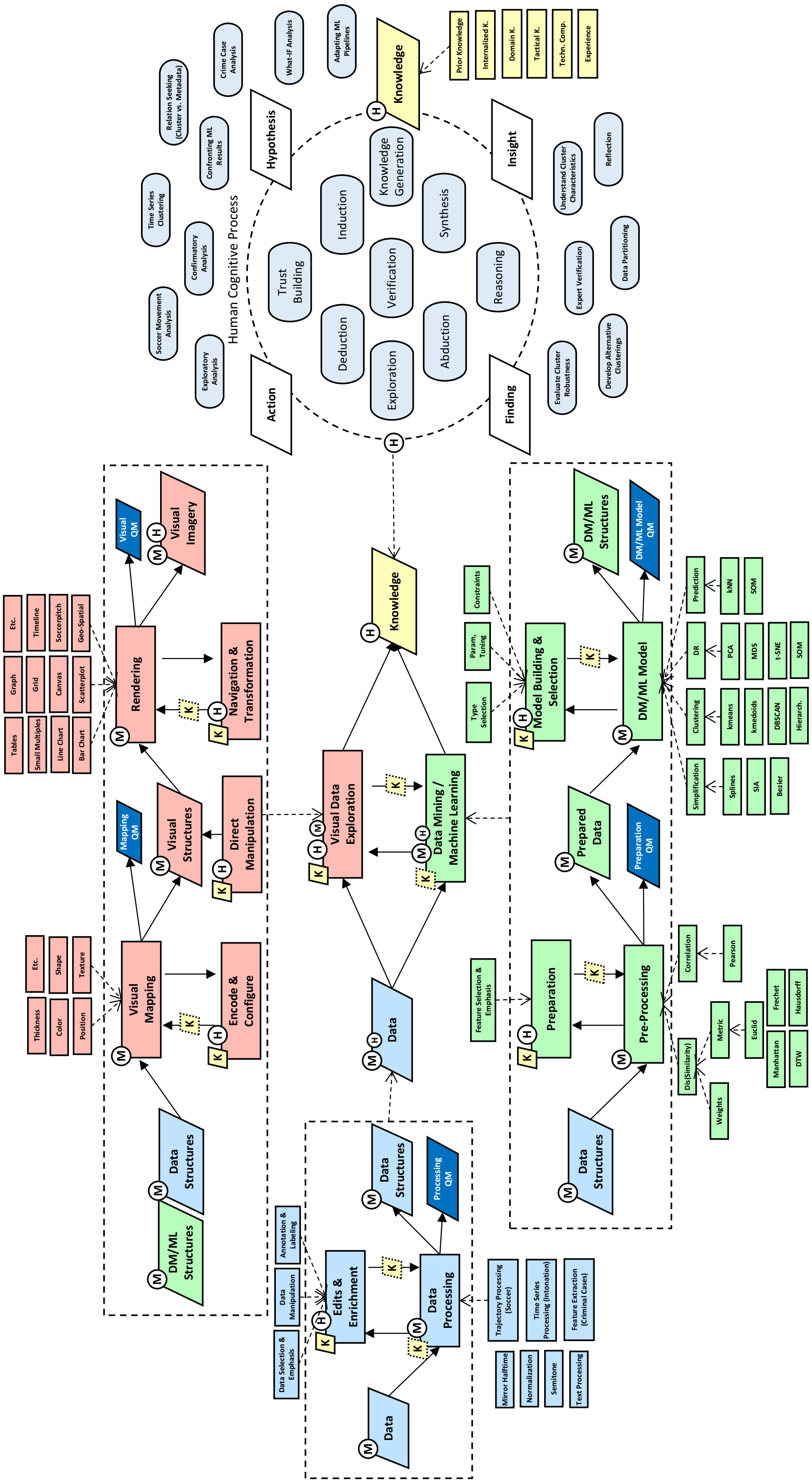


Figure A.4: The proposed ontology is further mapped out for more detailed processes that have been identified within this dissertation.

# Bibliography


[1] Aigner, W., Miksch, S., Schumann, H., and Tominski, C. *Visualization of time-oriented data*. Springer Science & Business Media, 2011.

[2] Al-Dohuki, S., Wu, Y., Kamw, F., Yang, J., Li, X., Zhao, Y., Ye, X., Chen, W., Ma, C., and Wang, F. Semantictraj: A new approach to interacting with massive taxi trajectories. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 11–20.

[3] Alpaydin, E. *Introduction to Machine Learning, 3rd Edition*. MIT Press, 2014.

[4] Amershi, S., Cakmak, M., Knox, W., and Kulesza, T. Power to the people: The role of humans in interactive machine learning. *AI Magazine 35*, 4 (2014), 105–120.

[5] Amershi, S., Fogarty, J., and Weld, D. ReGroup: Interactive machine learning for on-demand group creation in social networks. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2012), ACM, pp. 21–30.

[6] Anderson, B. B., Kirwan, C. B., Jenkins, J. L., Eargle, D., Howard, S., and Vance, A. How polymorphic warnings reduce habituation in the brain—insights from an fmri study. *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2015).

[7] Andrienko, G. L., Andrienko, N. V., Bak, P., Keim, D. A., and Wrobel, S. *Visual Analytics of Movement*. Springer, 2013.

[8] Andrienko, G. L., Andrienko, N. V., Budziak, G., von Landesberger, T., and Weber, H. Coordinate transformations for characterization and cluster analysis of spatial configurations in football. In *Machine Learning and Knowledge Discovery in Databases - European Conf., ECML PKDD* (2016), pp. 27–31.

[9] Andrienko, G. L., Andrienko, N. V., Rinzivillo, S., Nanni, M., Pedreschi, D., and Giannotti, F. Interactive visual clustering of large collections of trajectories. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2009), pp. 3–10.

[10] Andrienko, N., and Andrienko, G. Spatial generalization and aggregation of massive movement data. *IEEE Trans. on Visualization and Computer Graphics 17*, 2 (2011), 205–219.

[11] Andrienko, N., Andrienko, G., Barrett, L., Dostie, M., and Henzi, P. Space transformation for understanding group movement. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2169–2178.

[12] Arias-Hernández, R., Kaastra, L. T., Green, T. M., and Fisher, B. D. Pair analytics: Capturing reasoning processes in collaborative visual analytics. In *Hawaii Intern. Conf. on Systems Science (HICSS)* (2011), pp. 1–10.

[13] Arnott, D. Cognitive biases and decision support systems development: a design science approach. *Inf. Syst. J. 16*, 1 (2006), 55–78.

[14] Asano, Y., Gubian, M., and Sacha, D. Cutting down on manual pitch contour annotation using data modeling. In *Proc. of the 8th Intern. Conf. on Speech Prosody* (2016).

[15] Atkeson, C. G., Moore, A. W., and Schaal, S. Locally weighted learning for control. *Artif. Intell. Rev. 11*, 1-5 (1997), 75–113.

[16] Aupetit, M., and Sedlmair, M. Sepme: 2002 new visual separation measures. In *IEEE Pacific Visualization Symp.* (2016), pp. 1–8.

[17] Bach, B., Pietriga, E., Liccardi, I., and Legostaev, G. Ontotrix: a hybrid visualization for populated ontologies. In *Proc. of the 20th Intern. Conf. companion on World wide web* (2011), ACM, pp. 177–180.

[18] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval*. ACM Press. ACM Press, 2011. Second Edition.

[19] Barber, B. *The logic and limits of trust*, vol. 96. Rutgers University Press New Brunswick, NJ, 1983.

[20] Bass, E. J., Baumgart, L. A., and Shepley, K. K. The effect of information analysis automation display content on human judgment performance in noisy environments. *Journal of cognitive engineering and decision making 7*, 1 (2013), 49–65.

[21] Beaudouin-Lafon, M. Designing interaction, not interfaces. In *Conf. on Advanced Visual Interfaces (AVI)* (2004), ACM, pp. 15–22.

[22] Beckman, M. E., Hirschberg, J., and Shattuck-Hufnagel, S. The original ToBI system and the evolution of the ToBI framework. In *Prosodic Typology – The Phonology of Intonation and Phrasing*, S.-A. Jun, Ed. Oxford University Press, 2005.

[23] Behrisch, M., Korkmaz, F., Shao, L., and Schreck, T. Feedback-driven interactive exploration of large multidimensional data supported by visual classifier. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2014), pp. 43–52.

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

[25] Bernard, J., Daberkow, D., Fellner, D. W., Fischer, K., Koepler, O., Kohlhammer, J., Runnwerth, M., Ruppert, T., Schreck, T., and Sens, I. Visinfo: a digital library system for time series research data based on exploratory search - a user-centered design approach. *Int. Journal on Digital Libraries 16*, 1 (2015), 37–59.

[26] Bernard, J., Ruppert, T., Scherer, M., Kohlhammer, J., and Schreck, T. Content-based layouts for exploratory metadata search in scientific research data. In *Proc. of the 12th ACM/IEEE-CS Joint Conf. on Digital Libraries, JCDL* (2012), pp. 139–148.

[27] Bernard, J., Ruppert, T., Scherer, M., Schreck, T., and Kohlhammer, J. Guided discovery of interesting relationships between time series clusters and metadata properties. In *12th Intern. Conf. on Knowledge Management and Knowledge Technologies, I-KNOW* (2012), p. 22.

[28] Bernard, J., Steiger, M., Mittelstädt, S., Thum, S., Keim, D., and Kohlhammer, J. A survey and task-based quality assessment of static 2d colormaps. vol. 9397, SPIE Press, pp. 93970M–93970M–16.

[29] Bernard, J., von Landesberger, T., Bremm, S., and Schreck, T. Multiscale visual quality assessment for cluster analysis with self-organizing maps. In *Visualization and Data Analysis* (2011), p. 78680N.

[30] Bernard, J., Wilhelm, N., Krüger, B., May, T., Schreck, T., and Kohlhammer, J. Motionexplorer: Exploratory search in human motion capture data based on hierarchical aggregation. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2257–2266.

[31] Bernard, J., Zeppelzauer, M., Sedlmair, M., and Aigner, W. A Unified Process for Visual-Interactive Labeling. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2017), M. Sedlmair and C. Tominski, Eds., The Eurographics Association.

[32] Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., and Wiswedel, B. KNIME - the konstanz information miner: version 2.0 and beyond. *SIGKDD Explorations 11*, 1 (2009), 26–31.

[33] Bertini, E., and Lalanne, D. Surveying the complementary role of automatic data analysis and visualization in knowledge discovery. In *Proc. of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration* (2009), pp. 12–20.

[34] Bertini, E., Tatu, A., and Keim, D. A. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Trans. on Visualization and Computer Graphics 17*, 12 (2011), 2203–2212.

[35] Beyer, H. R., and Holtzblatt, K. *Contextual design - defining customer-centered systems*. Kaufmann, 1998.

[36] Boy, J., Eveillard, L., Détienne, F., and Fekete, J. Suggested interactivity: Seeking perceived affordances for information visualization. *IEEE Trans. on Visualization and Computer Graphics 22*, 1 (2016), 639–648.

[37] Bradel, L., North, C., House, L., and Leman, S. Multi-model semantic interaction for text analytics. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2014), 163–172.

[38] Brehmer, M., and Munzner, T. A multi-level typology of abstract visualization tasks. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2376–2385.

[39] Brodlie, K., Osorio, R. A., and Lopes, A. A review of uncertainty in data visualization. In *Expanding the Frontiers of Visual Analytics and Visualization*, J. Dill, R. Earnshaw, D. Kasik, J. Vince, and P. C. Wong, Eds. Springer, 2012, pp. 81–109.

[40] Brown, E., Liu, J., Brodley, C., and Chang, R. Dis-function: Learning distance functions interactively. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2012), pp. 83–92.

[41] Brown, E. T., Ottley, A., Zhao, H., Lin, Q., Souvenir, R., Endert, A., and Chang, R. Finding waldo: Learning about users from their interactions. *IEEE Trans. on Visualization and Computer Graphics 20*, 12 (2014), 1663–1672.

[42] Buja, A., Cook, D., and Swayne, D. F. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics 5* (1996), 78–99.

[43] Bunte, K., Schneider, P., Hammer, B., Schleif, F.-M., Villmann, T., and Biehl, M. Limited rank matrix learning, discriminative dimension reduction and visualization. *Neural Networks 26*, 1 (2012), 159–173.

[44] Buttenfield, B., and Weibel, R. Visualizing the quality of cartographic data. In *Third International Geographic Information Systems Symposium (GIS/LIS)* (1988).

[45] Cai, H., and Lin, Y. Tuning trust using cognitive cues for better human-machine collaboration. In *Proc. of the Human Factors and Ergonomics Society Annual Meeting* (2010), vol. 54, SAGE Publications, pp. 2437–2441.

[46] Callahan, S. P., Freire, J., Santos, E., Scheidegger, C. E., Silva, C. T., and Vo, H. T. Managing the evolution of dataflows with vistrails. In *Proc. of the 22nd Intern. Conf. on Data Engineering Workshops, ICDE* (2006), p. 71.

[47] Cao, N., Gotz, D., Sun, J., and Qu, H. DICON: interactive visual analysis of multidimensional clusters. *IEEE Trans. on Visualization and Computer Graphics 17*, 12 (2011), 2581–2590.

[48] Card, S. K., Mackinlay, J. D., and Shneiderman, B. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[49] Card, S. K., Robertson, G. G., and Mackinlay, J. D. The information visualizer, an information workspace. *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (1991), 181–186.

[50] Castelfranchi, C. Trust mediation in knowledge management and sharing. In *Proc. of the 2nd Intern. Conf. on Trust Management (iTrust)* (2004), pp. 304–318.

[51] Cedilnik, A., and Rheingans, P. Procedural annotation of uncertain information. In *Proc. of the IEEE Visualization 2000* (2000), pp. 77–83.

[52] Ceneda, D., Gschwandtner, T., May, T., Miksch, S., Schulz, H., Streit, M., and Tominski, C. Characterizing guidance in visual analytics. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 111–120.

[53] Chang, R., Ziemkiewicz, C., Green, T. M., and Ribarsky, W. Defining insight for visual analytics. *IEEE Computer Graphics and Applications 29*, 2 (2009), 14–17.

[54] Charmaz, K. *Constructing grounded theory*. Sage, 2014.

[55] Chatfield, C. Model uncertainty. *Encyclopedia of Environmetrics* (2006).

[56] Chen, M., and Golan, A. What may visualization processes optimize? *IEEE Trans. on Visualization and Computer Graphics 22*, 12 (2016), 2619–2632.

[57] Chen, M., Grinstein, G., Johnson, C. R., Kennedy, J., and Tory, M. Pathways for theoretical advances in visualization. *IEEE Computer Graphics and Applications* (2017).

[58] Chien, L., Tat, A., Proulx, P., Khamisa, A., and Wright, W. Grand challenge award 2008: Support for diverse analytic techniques - nspace2 and geotime visual analytics. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2008), pp. 199–200.

[59] Choo, J., Bohn, S., and Park, H. Two-stage framework for visualization of clustered high dimensional data. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2009), 67–74.

[60] Choo, J., Lee, H., Kihm, J., and Park, H. ivisclassifier: An interactive visual analytics system for classification based on supervised dimension reduction. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2010), pp. 27–34.

[61] Chuang, J., Ramage, D., Manning, C. D., and Heer, J. Interpretation and trust: designing model-driven visualizations for text analysis. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2012), pp. 443–452.

[62] Cohn, D., Atlas, L., and Ladner, R. Improving generalization with active learning. *Machine Learning 15*, 2 (1994), 201–221.

[63] Cook, K. A., Cramer, N., Israel, D., Wolverton, M., Bruce, J., Burtner, R., and Endert, A. Mixed-initiative visual analytics using task-driven recommendations. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2015), pp. 9–16.

[64] Cope, N. Intelligence led policing or policing led intelligence?: Integrating volume crime analysis into policing. *Br. J. Criminol. 44* (2004), 188–203.

[65] Correa, C. D., Chan, Y., and Ma, K. A framework for uncertainty-aware visual analytics. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2009), pp. 51–58.

[66] Crouser, R. J., and Chang, R. An affordance-based framework for human computation and human-computer collaboration. *IEEE Trans. on Visualization and Computer Graphics 18*, 12 (2012), 2859–2868.

[67] Cullen, A. C., and Frey, H. C. *Probabilistic techniques in exposure assessment: a handbook for dealing with variability and uncertainty in models and inputs*. Springer Science & Business Media, 1999.

[68] De Boor, C. *A practical guide to splines*. Springer, New York, 2001.

[69] de Visser, E. J., Cohen, M. S., Freedy, A., and Parasuraman, R. A design methodology for trust cue calibration in cognitive agents. In *Proc. of Conf. on the Virtual, Augmented and Mixed Reality VAMR, Part I* (2014), pp. 251–262.

[70] Demšar, U., Buchin, K., Cagnacci, F., Safi, K., Speckmann, B., Van de Weghe, N., Weiskopf, D., and Weibel, R. Analysis and visualisation of movement: an interdisciplinary review. *Movement Ecology 3*, 1 (2015), 5.

[71] Dix, A. Theoretical analysis and theory creation. *Research Methods in Human-Computer Interaction* (2008), 175–195.

[72] Donoho, D. High-Dimensional Data Analysis: The Curse and Blessings of Dimensionality. Aide-mémoire for a lecture for the American Math. Society "Math. Challenges of the 21st Century", 2000.

[73] Dou, W., Ribarsky, W., and Chang, R. Capturing reasoning process through user interaction. *Proc. IEEE EuroVAST 2* (2010).

[74] Drieger, P. Visual Text Analytics using Semantic Networks and Interactive 3D Visualization. In *EuroVA 2012: International Workshop on Visual Analytics* (2012), K. Matkovic and G. Santucci, Eds., The Eurographics Association.

[75] Duke, D. J., Brodlie, K. W., and Duce, D. A. Building an ontology of visualization. In *IEEE Visualization Conference, VIS* (2004), p. 7.

[76] Duke, D. J., Brodlie, K. W., Duce, D. A., and Herman, I. Do you see what I mean? *IEEE Computer Graphics and Applications 25*, 3 (2005), 6–9.

[77] Dunne, C., Riche, N. H., Lee, B., Metoyer, R. A., and Robertson, G. G. Graphtrail: analyzing large multivariate, heterogeneous networks while supporting exploration history. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2012), pp. 1663–1672.

[78] Dwyer, T., Koren, Y., and Marriott, K. Ipsep-cola: An incremental procedure for separation constraint layout of graphs. *IEEE Trans. on Visualization and Computer Graphics 12*, 5 (2006), 821–828.

[79] Dzindolet, M. T., Peterson, S. A., Pomranky, R. A., Pierce, L. G., and Beck, H. P. The role of trust in automation reliance. *Int. J. Hum.-Comput. Stud. 58*, 6 (2003), 697–718.

[80] Efron, B., and Tibshirani, R. Improvements on cross-validation: The .632+ bootstrap method. *Journal of the American Statistical Association 92*, 438 (1997), 548–560.

[81] Elmqvist, N., Stasko, J. T., and Tsigas, P. Datameadow: a visual canvas for analysis of large-scale multivariate data. *Information Visualization 7*, 1 (2008), 18–33.

[82] Endert, A. *Semantic Interaction for Visual Analytics: Inferring Analytical Reasoning for Model Steering*. Synthesis Lectures on Visualization. Morgan & Claypool Publishers, 2016.

[83] Endert, A., Chang, R., North, C., and Zhou, M. X. Semantic interaction: Coupling cognition and computation through usable interactive analytics. *IEEE Computer Graphics and Applications 35*, 4 (2015), 94–99.

[84] Endert, A., Fiaux, P., and North, C. Semantic interaction for sensemaking: Inferring analytical reasoning for model steering. *IEEE Trans. on Visualization and Computer Graphics 18*, 12 (2012), 2879–2888.

[85] Endert, A., Fiaux, P., and North, C. Semantic interaction for visual text analytics. *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2012), 473–482.

[86] Endert, A., Han, C., Maiti, D., House, L., Leman, S., and North, C. Observation-level interaction with statistical models for visual analytics. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2011), 121–130.

[87] Endert, A., Hossain, M. S., Ramakrishnan, N., North, C., Fiaux, P., and Andrews, C. The human is the loop: new directions for visual analytics. *Journal of Intelligent Information Systems 43*, 3 (2014), 411–435.

[88] Endert, A., North, C., Chang, R., and Zhou, M. Toward usable interactive analytics: Coupling cognition and computation. In *Proc. ACM SIGKDD Workshop Interactive Data Exploration and Analytics (IDEA)* (2014).

[89] Endert, A., Ribarsky, W., Turkay, C., Wong, B. W., Nabney, I., Blanco, I. D., and Rossi, F. The state of the art in integrating machine learning into visual analytics. *Computer Graphics Forum* (2017).

[90] Ericsson, K. A., and Simon, H. A. *Protocol analysis*. MIT-press, 1984.

[91] Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. From data mining to knowledge discovery in databases. *AI Magazine 17*, 3 (1996), 37–54.

[92] Fekete, J., and Primet, R. Progressive analytics: A computation paradigm for exploratory data analysis. *CoRR abs/1607.05162* (2016).

[93] Fekete, J.-D. ProgressiVis: a Toolkit for Steerable Progressive Analytics and Visualization. In *1st Workshop on Data Systems for Interactive Analysis* (2015), p. 5.

[94] Fernandez, C., Ley, E., and Steel, M. F. Model uncertainty in cross-country growth regressions. *Journal of applied Econometrics 16*, 5 (2001), 563–576.

[95] Fisher, D., Drucker, S. M., and König, A. C. Exploratory visualization involving incremental, approximate database queries and uncertainty. *IEEE Computer Graphics and Applications 32*, 4 (2012), 55–62.

[96] Flanagin, A., and Metzger, M. The credibility of volunteered geographic information. *GeoJournal 72*, 3 (2008), 137–148.

[97] Fröhlich, H., Chapelle, O., and Schölkopf, B. Feature selection for support vector machines by means of genetic algorithm. In *Proc. of the 15th IEEE Intern. Conf. on Tools with Artificial Intelligence* (2003), pp. 142–148.

[98] Gahegan, M. Beyond tools: Visual support for the entire process of giscience. *Exploring geovisualization*, 4 (2005), 83–99.

[99] Garg, S., Ramakrishnan, I. V., and Mueller, K. A visual analytics approach to model learning. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2010), 67–74.

[100] Goddard, K., Roudsari, A. V., and Wyatt, J. C. Automation bias: Empirical results assessing influencing factors. *I. J. Medical Informatics 83*, 5 (2014), 368–375.

[101] Görg, C., Liu, Z., Kihm, J., Choo, J., Park, H., and Stasko, J. T. Combining Computational Analyses and Interactive Visualization for Document Exploration and Sensemaking in Jigsaw. *IEEE Trans. on Visualization and Computer Graphics 19*, 10 (2013), 1646–1663.

[102] Gotz, D., and Wen, Z. Behavior-driven visualization recommendation. In *Proc. of the 14th Intern. Conf. on Intelligent User Interfaces* (2009), ACM, p. 315–324.

[103] Gotz, D., When, Z., Lu, J., Kissa, P., Cao, N., Qian, W. H., Liu, S. X., and Zhou, M. X. Harvest: An intelligent visual analytic tool for the masses. In *Proc. of the 1st Intern. Workshop on Intelligent Visual Interfaces for Text Analysis* (2010), IVITA, ACM, pp. 1–4.

[104] Gotz, D., and Zhou, M. X. Characterizing users' visual analytic activity for insight provenance. *Information Visualization 8*, 1 (2009), 42–55.

[105] Gove, R., Saxe, J., Gold, S., Long, A., and Bergamo, G. SEEM: A scalable visualization for comparing multiple large sets of attributes for malware analysis. In *Proc. Workshop on Visualization for Cyber Security, VizSec* (2014), ACM.

[106] Gratzl, S., Gehlenborg, N., Lex, A., Pfister, H., and Streit, M. Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Trans. on Visualization and Computer Graphics 20*, 12 (2014), 2023–2032.

[107] Green, T., Ribarsky, W., and Fisher, B. Visual analytics for complex concepts using a human cognition model. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2008), pp. 91–98.

[108] Green, T. M., and Maciejewski, R. A role for reasoning in visual analytics. In *Hawaii Intern. Conf. on System Sciences, HICSS* (2013), pp. 1495–1504.

[109] Green, T. M., Ribarsky, W., and Fisher, B. D. Building and applying a human cognition model for visual analytics. *Information Visualization 8*, 1 (2009), 1–13.

[110] Griethe, H., and Schumann, H. The visualization of uncertain data: Methods and problems. In *SimVis* (2006), pp. 143–156.

[111] Groce, A., Kulesza, T., Zhang, C., Shamasunder, S., Burnett, M. M., Wong, W., Stumpf, S., Das, S., Shinsel, A., Bice, F., and McIntosh, K. You are the only possible oracle: Effective test selection for end users of interactive machine learning systems. *IEEE Trans. Software Engineering 40*, 3 (2014), 307–323.

[112] Gudmundsson, J., and Wolle, T. Football analysis using spatio-temporal tools. *Computers, Environment and Urban Systems 47* (2014), 16–27.

[113] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (2003), 1157–1182.

[114] Haber, R. B., and McNabb, D. A. Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing 74* (1990), 93.

[115] Halkidi, M., Batistakis, Y., and Vazirgiannis, M. Clustering validity checking methods: Part ii. *SIGMOD Rec. 31*, 3 (Sept. 2002), 19–27.

[116] Han, J., Kamber, M., and Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011.

[117] Harrison, L., Dou, W., Lu, A., Ribarsky, W., and Wang, X. Analysts aren't machines: Inferring frustration through visualization interaction. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2011), pp. 279–280.

[118] Healey, C. G., and Dennis, B. M. Interest driven navigation in visualization. *IEEE Trans. Vis. Comput. Graph. 18*, 10 (2012), 1744–1756.

[119] Heine, C., and Scheuermann, G. Manual clustering refinement using interaction with blobs. *Computer Graphics Forum* (2007), 59–66.

[120] Hengl, T. Visualisation of uncertainty using the hsi colour model: computations with colours. In *Proc. of the 7th Intern. Conf. on GeoComputation* (2003), pp. 8–17.

[121] Henry, N., Fekete, J., and McGuffin, M. J. Nodetrix: a hybrid visualization of social networks. *IEEE Trans. on Visualization and Computer Graphics 13*, 6 (2007), 1302–1309.

[122] Heuvelink, G. B. M., and Burrough, P. A. Developments in statistical approaches to spatial uncertainty and its propagation. *International Journal of Geographical Information Science 16*, 2 (2002), 111–113.

[123] Höferlin, B., Netzel, R., Höferlin, M., Weiskopf, D., and Heidemann, G. Inter-active learning of ad-hoc classifiers for video visual analytics. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2012), pp. 23–32.

[124] Hoffman, P. E., and Grinstein, G. G. Information visualization in data mining and knowledge discovery. Morgan Kaufmann, San Francisco, CA, USA, 2002, ch. A Survey of Visualizations for High-dimensional Data Mining, pp. 47–82.

[125] Hornbæk, K. Current practice in measuring usability: Challenges to usability studies and research. *International Journal of Man-Machine Studies 64*, 2 (2006), 79–102.

[126] Howard, D., and MacEachren, A. M. Interface design for geographic visualization: Tools for representing reliability. *Cartography and Geographic Information Systems 23*, 2 (1996), 59–77.

[127] Hurter, C., Tissoires, B., and Conversy, S. Fromdady: Spreading aircraft trajectories across views to support iterative queries. *IEEE Trans. on Visualization and Computer Graphics 15*, 6 (2009), 1017–1024.

[128] Hyvärinen, A., Karhunen, J., and Oja, E. *Independent Component Analysis*. Wiley, 2001.

[129] Ingram, S., Munzner, T., Irvine, V., Tory, M., Bergner, S., and Möller, T. Dimstiller: Workflows for dimensional analysis and reduction. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2010), pp. 3–10.

[130] Iqbal, S. T., and Horvitz, E. Disruption and recovery of computing tasks: field study, analysis, and directions. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2007), pp. 677–686.

[131] Isenberg, T., Isenberg, P., Chen, J., Sedlmair, M., and Möller, T. A systematic review on the practice of evaluating visualization. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2818–2827.

[132] Ivanov, Y., Wren, C., Sorokin, A., and Kaur, I. Visualizing the history of living spaces. *IEEE Trans. on Visualization and Computer Graphics 13*, 6 (2007), 1153–1160.

[133] Jäckle, D., Fischer, F., Schreck, T., and Keim, D. A. Temporal MDS plots for analysis of multivariate data. *IEEE Trans. on Visualization and Computer Graphics 22*, 1 (2016), 141–150.

[134] Jäckle, D., Stoffel, F., Mittelstädt, S., Keim, D. A., and Reiterer, H. Interpretation of Dimensionally-Reduced Crime Data: A Study with Untrained Domain Experts. *Proc. Int. Conf. on Information Visualization Theory and Applications* (2017).

[135] Janetzko, H., Sacha, D., Stein, M., Schreck, T., Keim, D. A., and Deussen, O. Feature-driven visual analytics of soccer data. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2014), pp. 13–22.

[136] Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. *Recommender Systems: An Introduction*, 1st ed. Cambridge University Press, New York, NY, USA, 2010.

[137] Jawaheer, G., Weller, P., and Kostkova, P. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interactive Intelligent Systems 4*, 2 (2014), 8:1–8:26.

[138] Jeong, D. H., Ziemkiewicz, C., Fisher, B. D., Ribarsky, W., and Chang, R. iPCA: An interactive system for pca-based visual analytics. *Computer Graphics Forum 28*, 3 (2009), 767–774.

[139] Johansson, S., and Johansson, J. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Trans. on Visualization and Computer Graphics 15*, 6 (2009), 993–1000.

[140] Joia, P., Petronetto, F., and Nonato, L. G. Uncovering representative groups in multidimensional projections. *Computer Graphics Forum 34*, 3 (2015), 281–290.

[141] Kaden, M., Lange, M., Nebel, D., Riedel, M., Geweniger, T., and Villmann, T. Aspects in classification learning - Review of recent developments in Learning Vector Quantization. *Foundations of Computing and Decision Sciences 39*, 2 (2014), 79–105.

[142] Kadivar, N., Chen, V. Y., Dunsmuir, D., Lee, E., Qian, C. Z., Dill, J., Shaw, C. D., and Woodbury, R. F. Capturing and supporting the analysis process. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2009), pp. 131–138.

[143] Kahneman, D. *Thinking, fast and slow*. Macmillan, 2011.

[144] Kahneman, D., and Tversky, A. Subjective probability: A judgment of representativeness. In *The Concept of Probability in Psychological Experiments*. Springer, 1974, pp. 25–48.

[145] Kandel, S., Heer, J., Plaisant, C., Kennedy, J., van Ham, F., Riche, N., Weaver, C., Lee, B., Brodbeck, D., and Buono, P. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization 10*, 4 (2011), 271–288.

[146] Kandel, S., Paepcke, A., Hellerstein, J. M., and Heer, J. Wrangler: interactive visual specification of data transformation scripts. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2011), pp. 3363–3372.

[147] Kang, Y., Görg, C., and Stasko, J. T. How can visual analytics assist investigative analysis? design implications from an evaluation. *IEEE Trans. on Visualization and Computer Graphics 17*, 5 (2011), 570–583.

[148] Kapoor, A., Lee, B., Tan, D. S., and Horvitz, E. Learning to learn: Algorithmic inspirations from human problem solving. In *Proc. of the AAAI Conf. on Artificial Intelligence* (2012).

[149] Keim, D. A. Information visualization and visual data mining. *IEEE Trans. on Visualization and Computer Graphics 8*, 1 (2002), 1–8.

[150] Keim, D. A., Kohlhammer, J., Ellis, G. P., and Mansmann, F. *Mastering the Information Age - Solving Problems with Visual Analytics*. Eurographics Association, 2010.

[151] Keim, D. A., Mansmann, F., Schneidewind, J., Thomas, J., and Ziegler, H. *Visual analytics: Scope and challenges*. Springer, 2008.

[152] Keim, D. A., Munzner, T., Rossi, F., and Verleysen, M. Bridging Information Visualization with Machine Learning (Dagstuhl Seminar 15101). *Dagstuhl Reports 5*, 3 (2015), 1–27.

[153] Kennedy, M. C., and O'Hagan, A. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*, 3 (2001), 425–464.

[154] Kerracher, N., and Kennedy, J. Constructing and evaluating visualisation task classifications: Process and considerations. *Computer Graphics Forum 36*, 3 (2017), 47–59.

[155] Klemmer, S. R., Thomsen, M., Phelps-Goodman, E., Lee, R., and Landay, J. A. Where do web sites come from?: capturing and interacting with design history. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2002), pp. 1–8.

[156] Klir, G. J., and Wierman, M. J. *Uncertainty-based information: elements of generalized information theory*, vol. 15. Springer Science & Business Media, 1999.

[157] Kohonen, T. Essentials of the self-organizing map. *Neural Netw. 37* (Jan. 2013), 52–65.

[158] Kohonen, T., Hynninen, J., Kangas, J., and Laaksonen, J. Som pak: The self-organizing map program package. *Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science* (1996).

[159] Kohonen, T., Schroeder, M. R., and Huang, T. S., Eds. *Self-Organizing Maps*, 3rd ed. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[160] Kosara, R., and Mackinlay, J. Storytelling: The next step for visualization. *Computer 46*, 5 (2013), 44–50.

[161] Kretz, D. R., Simpson, B., and Graham, C. A game-based experimental protocol for identifying and overcoming judgment biases in forensic decision analysis. In *IEEE Conf. on Technologies for Homeland Security (HST)* (2012), pp. 439–444.

[162] Kruskal, J. B., and Wish, M. *Multidimensional scaling*, vol. 11. Sage, 1978.

[163] Kurzhals, K., Fisher, B. D., Burch, M., and Weiskopf, D. Evaluating visual analytics with eye tracking. In *Proc. of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV)* (2014), pp. 61–69.

[164] Kwon, B. C., Fisher, B., and Yi, J. S. Visual analytic roadblocks for novice investigators. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2011), pp. 3–11.

[165] Kwon, B. C., Verma, J., Haas, P. J., and Demiralp, Ç. Sampling for scalable visual analytics. *IEEE Computer Graphics and Applications 37*, 1 (2017), 100–108.

[166] Lam, H. A framework of interaction costs in information visualization. *IEEE Trans. on Visualization and Computer Graphics 14*, 6 (2008), 1149–1156.

[167] Lam, H., Bertini, E., Isenberg, P., Plaisant, C., and Carpendale, S. Empirical studies in information visualization: Seven scenarios. *IEEE Trans. on Visualization and Computer Graphics 18*, 9 (2012), 1520–1536.

[168] Laube, P., Imfeld, S., and Weibel, R. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science 19*, 6 (2005), 639–668.

[169] Lee, J., and Verleysen, M. *Nonlinear dimensionality reduction*. Springer, 2007.

[170] Lee, J. A., Renard, E., Bernard, G., Dupont, P., and Verleysen, M. Type 1 and 2 mixtures of Kullback-Leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing 112* (2013), 92–108.

[171] Lee, S. H., and Chen, W. A comparative study of uncertainty propagation methods for black-box-type problems. *Structural and Multidisciplinary Optimization 37*, 3 (2009), 239–253.

[172] Legrenzi, P., Girotto, V., and Johnson-Laird, P. N. Focussing in reasoning and decision making. *Cognition 49*, 1 (1993), 37–66.

[173] Lehmann, D. J., and Theisel, H. Optimal sets of projections of high-dimensional data. *IEEE Trans. on Visualization and Computer Graphics 22*, 1 (2016), 609–618.

[174] Liu, S., Maljovec, D., Wang, B., Bremer, P.-T., and Pascucci, V. Visualizing High-Dimensional Data: Advances in the Past Decade. *Computer Graphics Forum* (2015).

[175] Liu, S., Wang, B., Bremer, P., and Pascucci, V. Distortion-guided structure-driven interactive exploration of high-dimensional data. *Computer Graphics Forum 33*, 3 (2014), 101–110.

[176] Liu, S., Wang, B., Thiagarajan, J. J., Bremer, P., and Pascucci, V. Visual exploration of high-dimensional data through subspace analysis and dynamic projections. *Computer Graphics Forum 34*, 3 (2015), 271–280.

[177] Liu, Z., Görg, C., Kihm, J., Lee, H., Choo, J., Park, H., and Stasko, J. T. Data ingestion and evidence marshalling in jigsaw VAST 2010 mini challenge 1 award: Good support for data ingest. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2010), pp. 271–272.

[178] Liu, Z., Nersessian, N. J., and Stasko, J. T. Distributed cognition as a theoretical framework for information visualization. *IEEE Trans. on Visualization and Computer Graphics 14*, 6 (2008), 1173–1180.

[179] Liu, Z., and Stasko, J. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Trans. on Visualization and Computer Graphics 16*, 6 (2010), 999–1008.

[180] Lu, Y., Garcia, R., Hansen, B., Gleicher, M., and Maciejewski, R. The state-of-the-art in predictive visual analytics. *Computer Graphics Forum 36*, 3 (2017), 539–562.

[181] Lush, V., Bastin, L., and Lumsden, J. Developing a geo label: providing the gis community with quality metadata visualisation tools. *Proc. of the GIS Research UK (GISRUK)* (2013), 3–5.

[182] MacEachren, A. M. Visualizing uncertain information. *Cartographic Perspectives 13* (1992), 10–19.

[183] MacEachren, A. M. Visual analytics and uncertainty: Its not about the data. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2015), The Eurographics Association.

[184] MacEachren, A. M., and Ganter, J. H. A pattern identification approach to cartographic visualization. *Cartographica: The International Journal for Geographic Information and Geovisualization 27*, 2 (1990), 64–81.

[185] MacEachren, A. M., Robinson, A., Hopper, S., Gardner, S., Murray, R., Gahegan, M., and Hetzler, E. Visualizing geospatial information uncertainty: What we know and what we need to know. *Cartography and Geographic Information Science 32*, 3 (2005), 139–160.

[186] MacEachren, A. M., Roth, R. E., O'Brien, J., Li, B., Swingley, D., and Gahegan, M. Visual semiotics & uncertainty visualization: An empirical study. *IEEE Trans. on Visualization and Computer Graphics 18*, 12 (2012), 2496–2505.

[187] Mackaness, W. A., Ruas, A., and Sarjakoski, T., Eds. *Generalisation of Geographic Information: Cartographic Modelling and Applications*. Elsevier, 2007.

[188] Magnani, L. *Abduction, Reason and Science: Processes of Discovery and Explanation*. Springer, 2001.

[189] Mahyar, N., Sarvghad, A., and Tory, M. A closer look at note taking in the co-located collaborative visual analytics process. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2010), pp. 171–178.

[190] Mamani, G. M. H., Fatore, F. M., Nonato, L. G., and Paulovich, F. V. User-driven feature space transformation. *Computer Graphics Forum 32*, 3 (2013), 291–299.

[191] Mansmann, F., Fischer, F., and Keim, D. A. Dynamic visual analytics facing the real-time challenge. In *Expanding the Frontiers of Visual Analytics and Visualization*. Springer, 2012, pp. 69–80.

[192] Mansouryar, M., and Hedayati, A. Smoothing via iterative averaging (sia) a basic technique for line smoothing. *International Journal of Computer and Electrical Engineering 4*, 3 (2012), 307.

[193] Manzey, D., Reichenbach, J., and Onnasch, L. Human performance consequences of automated decision aids: The impact of degree of automation and system experience. *Journal of Cognitive Engineering and Decision Making* (2012), 57–87.

[194] Melville, P., and Sindhwani, V. Recommender systems. In *Encyclopedia of Machine Learning*. 2010, pp. 829–838.

[195] Miller, G. A. The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological review 63*, 2 (1956), 81.

[196] Miller, R. B. Response time in man-computer conversational transactions. In *American Federation of Information Processing Societies: Proc. of the AFIPS Joint Computer Conf. - Part I* (1968), pp. 267–277.

[197] Mittelstädt, S., Stoffel, A., and Keim, D. A. Methods for compensating contrast effects in information visualization. *Computer Graphics Forum 33*, 3 (2014), 231–240.

[198] Moehrmann, J., Bernstein, S., Schlegel, T., Werner, G., and Heidemann, G. Improving the usability of hierarchical representations for interactively labeling large image data sets. In *Human-Computer Interaction. HCI International* (2011), pp. 618–627.

[199] Molchanov, V., and Linsen, L. Interactive design of multidimensional data projection layout. In *EuroVis - Short Papers* (2014), N. Elmqvist, M. Hlawitschka, and J. Kennedy, Eds., The Eurographics Association.

[200] Mühlbacher, T., and Piringer, H. A partition-based framework for building and validating regression models. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 1962–1971.

[201] Mühlbacher, T., Piringer, H., Gratzl, S., Sedlmair, M., and Streit, M. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Trans. on Visualization and Computer Graphics 20*, 12 (2014), 1643–1652.

[202] Muir, B. M. Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies 27*, 5-6 (1987), 527–539.

[203] Nam, E. J., Han, Y., Mueller, K., Zelenyuk, A., and Imre, D. Clustersculptor: A visual analytics tool for high-dimensional data. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2007), 75–82.

[204] Nam, J. E., and Mueller, K. Tripadvisor$^{\text{n-d}}$: A tourism-inspired high-dimensional space exploration framework with overview and detail. *IEEE Trans. on Visualization and Computer Graphics 19*, 2 (2013), 291–305.

[205] Nguyen, P. H., Xu, K., and Wong, B. A survey of analytic provenance. *Middlesex University* (2014).

[206] Nobarany, S., Haraty, M., and Fisher, B. Facilitating the reuse process in distributed collaboration: a distributed cognition approach. In *Proc. of the ACM conf. on Computer Supported Cooperative Work* (2012), ACM, pp. 1223–1232.

[207] Norman, D. *The Design of Everyday Things*. Basic Books, 2002.

[208] North, C. Toward measuring visualization insight. *IEEE Computer Graphics and Applications 26*, 3 (2006), 6–9.

[209] NPIA. National policing improvement agency: Professional practice on analysis, 2008.

[210] Oulasvirta, A., and Hornbæk, K. Hci research as problem-solving. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2016), pp. 4956–4967.

[211] Pampalk, E., Rauber, A., and Merkl, D. Using smoothed data histograms for cluster visualization in self-organizing maps. In *Proc. of the Intern. Conf. on Artificial Neural Networks* (2002), ICANN, Springer-Verlag, pp. 871–876.

[212] Pang, A. Visualizing uncertainty in geo-spatial data. In *Proc. of the Workshop on the Intersections between Geospatial Information and Information Technology* (2001).

[213] Pang, A., Wittenbrink, C. M., and Lodha, S. K. Approaches to uncertainty visualization. *The Visual Computer 13*, 8 (1997), 370–390.

[214] Pearson, K. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine 2* (1901), 559–572.

[215] Peirce, C. S. *Collected papers: Elements of Logic*, vol. 2. Belknap Press, 1965.

[216] Pelekis, N., Kopanakis, I., Marketos, G., Ntoutsi, I., Andrienko, G., and Theodoridis, Y. Similarity search in trajectory databases. In *Proc. of the 14th International Symposium on Temporal Representation and Reasoning* (2007), IEEE Computer Society, pp. 129–140.

[217] Perin, C., Vuillemot, R., and Fekete, J. Soccerstories: A kick-off for visual soccer analysis. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2506–2515.

[218] Pezzotti, N., Lelieveldt, B. P. F., van der Maaten, L., Höllt, T., Eisemann, E., and Vilanova, A. Approximated and user steerable tsne for progressive visual analytics. *IEEE Trans. on Visualization and Computer Graphics 23*, 7 (2017), 1739–1752.

[219] Phillips, B. K., Prybutok, V. R., and Peak, D. A. Decision confidence, information usefulness, and information seeking intention in the presence of disconfirming information. *InformingSciJ 17* (2014), 1–24.

[220] Pike, W. A., Stasko, J. T., Chang, R., and O'Connell, T. A. The science of interaction. *Information Visualization 8*, 4 (2009), 263–274.

[221] Pirolli, P., and Card, S. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. of the Intern. Conf. on Intelligence Analysis* (2005), vol. 5, pp. 2–4.

[222] Pohl, M., Smuc, M., and Mayr, E. The user puzzle - explaining the interaction with visual analytics systems. *IEEE Trans. on Visualization and Computer Graphics 18*, 12 (2012), 2908–2916.

[223] Pölzlbauer, G. Survey and comparison of quality measures for self-organizing maps. In *Proc. of the 5th Workshop on Data Analysis (WDA)* (2004), J. Paralič, G. Pölzlbauer, and A. Rauber, Eds., Elfa Academic Press, pp. 67–82.

[224] Popper, K. *The logic of scientific discovery*. Routledge, 2005.

[225] Pretorius, A. J., and van Wijk, J. J. What does the user want to see? what do the data want to be? *Information Visualization 8*, 3 (2009), 153–166.

[226] Prowse, J., and Bennett, E. *Working Manual of Criminal Law*. Carswell Legal Pubns, 2000.

[227] Ragan, E. D., and Goodall, J. R. Evaluation methodology for comparing memory and communication of analytic processes in visual analytics. In *Proc. of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV)* (2014), pp. 27–34.

[228] Rakotomamonjy, A., Bach, F. R., Canu, S., and Grandvalet, Y. Simplemkl. *Journal of Machine Learning Research 9* (2008), 2491–2521.

[229] Ratanamahatana, C. A., Lin, J., Gunopulos, D., Keogh, E. J., Vlachos, M., and Das, G. Mining time series data. *Data Mining and Knowledge Discovery Handbook, 2nd ed.* (2010), 1049–1077.

[230] Rauber, A., Merkl, D., and Dittenbach, M. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *IEEE Trans. Neural Networks 13*, 6 (2002), 1331–1341.

[231] Ribarsky, W., and Fisher, B. D. The human-computer system: Towards an operational model for problem solving. In *49th Hawaii Intern. Conf. on System Sciences, HICSS* (2016), pp. 1446–1455.

[232] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B. *Recommender Systems Handbook*, 1st ed. Springer-Verlag, 2010.

[233] Rieck, B., and Leitte, H. Persistent homology for the evaluation of dimensionality reduction schemes. *Computer Graphics Forum 34*, 3 (2015), 431–440.

[234] Roberts, J. C., Keim, D. A., Hanratty, T., Rowlingson, R. R., Walker, R., Hall, M., Jacobson, Z., Lavigne, V., Rooney, C., and Varga, M. From Ill-Defined Problems to Informed Decisions. *EuroVis Workshop on Visual Analytics* (2014).

[235] Robinson, A. Collaborative synthesis of visual analytic results. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2008), pp. 67–74.

[236] Rousseeuw, P. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics 20*, 1 (1987), 53–65.

[237] Ruppert, T., Staab, M., Bannach, A., Lücke-Tieke, H., Bernard, J., Kuijper, A., and Kohlhammer, J. Visual interactive creation and validation of text clustering workflows to explore document collections. *Electronic Imaging 2017*, 1 (2017), 46–57.

[238] Sacha, D., Al-Masoudi, F., Stein, M., Schreck, T., Keim, D. A., Andrienko, G., and Janetzko, H. Dynamic Visual Abstraction of Soccer Movement. *Computer Graphics Forum* (2017).

[239] Sacha, D., Asano, Y., Rohrdantz, C., Hamborg, F., Keim, D. A., Braun, B., and Butt, M. Self organizing maps for the visual analysis of pitch contours. In *Proc. of the 20th Nordic Conf. of Computational Linguistics, NODALIDA* (2015), pp. 181–189.

[240] Sacha, D., Boesecke, I., Fuchs, J., and Keim, D. A. Analytic Behavior and Trust Building in Visual Analytics. In *EuroVis 2016 - Short Papers* (2016), E. Bertini, N. Elmqvist, and T. Wischgoll, Eds., The Eurographics Association.

[241] Sacha, D., Jentner, W., Zhang, L., Stoffel, F., and Ellis, G. Visual Comparative Case Analytics. In *EuroVis Workshop on Visual Analytics (EuroVA)* (2017), M. Sedlmair and C. Tominski, Eds., The Eurographics Association.

[242] Sacha, D., Jentner, W., Zhang, L., Stoffel, F., Ellis, G., and Keim, D. A. Applying Visual Interactive Dimensionality Reduction to Criminal Intelligence Analysis. Tech. rep., Universität Konstanz, Middlesex University, 2017. VALCRI White Paper Series.

[243] Sacha, D., Kraus, M., Bernard, J., Behrisch, M., Schreck, T., Asano, Y., and Keim, D. A. Somflow: Guided exploratory cluster analysis with self-organizing maps and analytic provenance. *IEEE Trans. on Visualization and Computer Graphics* (2017).

[244] Sacha, D., Sedlmair, M., Zhang, L., Lee, J. A., Peltonen, J., Weiskopf, D., North, S. C., and Keim, D. A. What you see is what you can change: Human-centered machine learning by interactive visualization. *Neurocomputing* (2017).

[245] Sacha, D., Sedlmair, M., Zhang, L., Lee, J. A., Weiskopf, D., North, S. C., and Keim, D. A. Human-Centered Machine Learning Through Interactive Visualization: Review and Open Challenges. In *Proc. of the 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (2016).

[246] Sacha, D., Senaratne, H., Kwon, B. C., Ellis, G. P., and Keim, D. A. The role of uncertainty, awareness, and trust in visual analytics. *IEEE Trans. on Visualization and Computer Graphics 22*, 1 (2016), 240–249.

[247] Sacha, D., Senaratne, H., Kwon, B. C., and Keim, D. A. Uncertainty Propagation and Trust Building in Visual Analytics. *IEEE VIS 2014 - Provenance for Sensemaking Workshop (poster paper)* (2014).

[248] Sacha, D., Stoffel, A., Stoffel, F., Kwon, B. C., Ellis, G. P., and Keim, D. A. Knowledge generation model for visual analytics. *IEEE Trans. on Visualization and Computer Graphics 20*, 12 (2014), 1604–1613.

[249] Sacha, D., Zhang, L., Sedlmair, M., Lee, J. A., Peltonen, J., Weiskopf, D., North, S. C., and Keim, D. A. Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 241–250.

[250] Saraiya, P., North, C., and Duca, K. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Trans. on Visualization and Computer Graphics 11*, 4 (2005), 443–456.

[251] Scheidegger, C. E., Vo, H. T., Koop, D., Freire, J., and Silva, C. T. Querying and creating visualizations by analogy. *IEEE Trans. on Visualization and Computer Graphics 13*, 6 (2007), 1560–1567.

[252] Schneider, P., Hammer, B., and Biehl, M. Adaptive relevance matrices in learning vector quantization. *Neural Computation 21* (2009), 3532–3561.

[253] Scholtz, J. Beyond usability: Evaluation aspects of visual analytic environments. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2006), pp. 145–150.

[254] Schreck, T., Bernard, J., Tekusova, T., and Kohlhammer, J. Visual cluster analysis of trajectory data with interactive kohonen maps. *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2008), 3–10.

[255] Schreck, T., Bernard, J., Von Landesberger, T., and Kohlhammer, J. Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization 8*, 1 (2009), 14–29.

[256] Schulz, H.-J., Nocke, T., Heitzler, M., and Schumann, H. A design space of visualization tasks. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2366–2375.

[257] Sedlmair, M., Brehmer, M., Ingram, S., and Munzner, T. Dimensionality reduction in the wild: Gaps and guidance. Technical Report TR-2012-03, Dept. of Computer Science, University of British Columbia, 2012.

[258] Sedlmair, M., Heinzl, C., Bruckner, S., Piringer, H., and Moller, T. Visual parameter space analysis: A conceptual framework. *IEEE Trans. on Visualization and Computer Graphics 20*, 12 (2014), 2161–2170.

[259] Sedlmair, M., Meyer, M. D., and Munzner, T. Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans. on Visualization and Computer Graphics 18*, 12 (2012), 2431–2440.

[260] Sedlmair, M., Munzner, T., and Tory, M. Empirical guidance on scatterplot and dimension reduction technique choices. *IEEE Trans. on Visualization and Computer Graphics 19*, 12 (2013), 2634–2643.

[261] Senaratne, H., and Gerharz, L. An assessment and categorisation of quantitative uncertainty visualisation methods for geospatial data. In *AGILE intern. conf. on geographic information science-advancing geoinformation science for a changing world* (2011).

[262] Senaratne, H., Gerharz, L., Pebesma, E., and Schwering, A. Usability of spatio-temporal uncertainty visualisation methods. In *Bridging the Geographic Information Sciences*. Springer, 2012, pp. 3–23.

[263] Settles, B. Active learning literature survey. *University of Wisconsin, Madison 52* (2010), 55–66.

[264] Settles, B. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[265] Shao, L., Sacha, D., Neldner, B., Stein, M., and Schreck, T. Visual-interactive search for soccer trajectories to identify interesting game situations. In *Visualization and Data Analysis VDA* (2016), pp. 1–10.

[266] Shneiderman, B. Direct manipulation: A step beyond programming languages. *IEEE Computer 16*, 8 (1983), 57–69.

[267] Shneiderman, B. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the IEEE Symp. on Visual Languages* (1996), pp. 336–343.

[268] Shrinivasan, Y., and van Wijk, J. Supporting exploration awareness in information visualization. *IEEE Computer Graphics and Applications 29*, 5 (Sept. 2009), 34–43.

[269] Shrinivasan, Y. B., Gotz, D., and Lu, J. Connecting the dots in visual analysis. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2009), pp. 123–130.

[270] Shrinivasan, Y. B., and van Wijk, J. J. Supporting the analytical reasoning process in information visualization. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2008), pp. 1237–1246.

[271] Simmhan, Y. L., Plale, B., and Gannon, D. A survey of data provenance techniques. *Computer Science Department, Indiana University, Bloomington IN 47405* (2005).

[272] Simon, S., Mittelstädt, S., Keim, D. A., and Sedlmair, M. Bridging the Gap of Domain and Visualization Experts with a Liaison. In *Eurographics Conf. on Visualization (EuroVis) - Short Papers* (2015), E. Bertini, J. Kennedy, and E. Puppo, Eds., The Eurographics Association.

[273] Sips, M., Neubert, B., Lewis, J. P., and Hanrahan, P. Selecting good views of high-dimensional data using class consistency. In *Computer Graphics Forum* (2009), pp. 831–838.

[274] Skeels, M. M., Lee, B., Smith, G., and Robertson, G. G. Revealing uncertainty for information visualization. *Information Visualization 9*, 1 (2010), 70–81.

[275] Stahnke, J., Dörk, M., Müller, B., and Thom, A. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Trans. on Visualization and Computer Graphics 22*, 1 (2016), 629–638.

[276] Stasko, J. T., Görg, C., and Liu, Z. Jigsaw: supporting investigative analysis through interactive visualization. *Information Visualization 7*, 2 (2008), 118–132.

[277] Stasko, J. T., Görg, C., Liu, Z., and Singhal, K. Jigsaw: Supporting Investigative Analysis through Interactive Visualization. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2007), pp. 131–138.

[278] Stein, M., Janetzko, H., Breitkreutz, T., Seebacher, D., Schreck, T., Grossniklaus, M., Couzin, I. D., and Keim, D. A. Director's cut: Analysis and annotation of soccer matches. *IEEE Computer Graphics and Applications 36*, 5 (2016), 50–60.

[279] Stitz, H., Luger, S., Streit, M., and Gehlenborg, N. AVOCADO: visualization of workflow-derived data provenance for reproducible biomedical research. *Computer Graphics Forum 35*, 3 (2016), 481–490.

[280] Stoffel, F., Sacha, D., Ellis, G., and Keim, D. A. VAPD - A Visionary System for Uncertainty Aware Decision Making in Crime Analysis. In *Symp. on Visualization for Decision Making Under Uncertainty at IEEE VIS* (2015).

[281] Tak, S., and Toet, A. Color and uncertainty: It is not always black and white. *The Eurographics Association* (2014).

[282] Talbot, J., Lee, B., Kapoor, A., and Tan, D. Ensemblematrix: interactive visualization to support machine learning with multiple classifiers. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2009), pp. 1283–1292.

[283] Tam, G. K. L., Kothari, V., and Chen, M. An analysis of machine- and human-analytics in classification. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 71–80.

[284] Tatu, A., Albuquerque, G., Eisemann, M., Bak, P., Theisel, H., Magnor, M. A., and Keim, D. A. Automated analytical methods to support visual exploration of high-dimensional data. *IEEE Trans. on Visualization and Computer Graphics 17*, 5 (2011), 584–597.

[285] Tenbrink, T. Cognitive discourse analysis: accessing cognitive representations and processes through language data. *Language and Cognition 7* (3 2015), 98–137.

[286] Thomas, J., and Cook, K. A. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE Computer Society Press, 2005.

[287] Thomson, J., Hetzler, E., MacEachren, A., Gahegan, M., and Pavel, M. A typology for visualizing uncertainty. In *Electronic Imaging 2005* (2005), International Society for Optics and Photonics, pp. 146–157.

[288] Tory, M., and Möller, T. Human factors in visualization research. *IEEE Trans. on Visualization and Computer Graphics 10*, 1 (2004), 72–84.

[289] Tukey, J. *Exploratory Data Analysis*. Addison-Wesley series in behavioral science. Addison-Wesley Publishing Company, 1977.

[290] Turkay, C., Kaya, E., Balcisoy, S., and Hauser, H. Designing progressive and interactive analytics processes for high-dimensional data analysis. *IEEE Trans. on Visualization and Computer Graphics 23*, 1 (2017), 131–140.

[291] Uggirala, A., Gramopadhye, A. K., Melloy, B. J., and Toler, J. E. Measurement of trust in complex and dynamic systems using a quantitative approach. *Intern. Journal of Industrial Ergonomics 34*, 3 (2004), 175–186.

[292] Ultsch, A. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In *Kohonen Maps* (1999), Elsevier, pp. 33–46.

[293] van den Elzen, S., and van Wijk, J. Baobabview: Interactive construction and analysis of decision trees. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)* (2011), pp. 151–160.

[294] van den Elzen, S., and van Wijk, J. J. Small multiples, large singles: A new approach for visual data exploration. *Computer Graphics Forum 32*, 3 (2013), 191–200.

[295] van der Maaten, L., and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research 9* (2008), 2579–2605.

[296] van der Maaten, L., Postma, E., and van den Herik, H. Dimensionality reduction: A comparative review. Tech. rep., Tilburg Centre for Creative Computing, Tilburg University, 2009.

[297] van der Wel, F. J., van der Gaag, L. C., and Gorte, B. G. Visual exploration of uncertainty in remote-sensing classification. *Computers & Geosciences 24*, 4 (1998), 335–343.

[298] van Wijk, J. J. The value of visualization. In *16th IEEE Visualization Conference VIS* (2005), pp. 79–86.

[299] Vance, T. J. *An Introduction to Japanese Phonology*. State University of New York Press, 1987.

[300] Vellido, A., Martín-Guerrero, J. D., and Lisboa, P. J. G. Making machine learning models interpretable. In *20th European Symposium on Artificial Neural Networks, ESANN* (2012).

[301] Venna, J., Peltonen, J., Nybo, K., Aidos, H., and Kaski, S. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research 11* (2010), 451–490.

[302] Vesanto, J. Som-based data visualization methods. *Intelligent Data Analysis 3*, 2 (1999), 111–126.

[303] Vesanto, J. Using SOM in data mining. *Doctoral Dissertations, Licentiate's thesis, Helsinki University of Technology, Espoo, Finland* (2000).

[304] von Landesberger, T., Bremm, S., Schreck, T., and Fellner, D. Feature-based automatic identification of interesting data segments in group movement data. *Sage Information Visualization 13*, 3 (2014), 190–212.

[305] von Landesberger, T., Fiebig, S., Bremm, S., Kuijper, A., and Fellner, D. W. Interaction taxonomy for tracking of user actions in visual analytics applications. In *Handbook of Human Centric Visualization*. Springer, New York, 2014, pp. 653–670.

[306] Wagner, M., Rind, A., Thür, N., and Aigner, W. A knowledge-assisted visual malware analysis system: Design, validation, and reflection of KAMAS. *Computers & Security 67* (2017), 1 – 15.

[307] Wagner, M., Sacha, D., Rind, A., Fischer, F., Luh, R., Schrittwieser, S., Keim, D. A., and Aigner, W. Visual analytics: Foundations and experiences in malware analysis. In *Empirical Research for Software Security: Foundations and Experience*, L. ben Othmane, M. Gilje Jaatun, and E. Weippl, Eds. Taylor & Francis Group, LLC, 2017.

[308] Wang, X., Zhang, T., Ma, Y., Xia, J., and Chen, W. A survey of visual analytic pipelines. *Journal of Computer Science and Technology 31*, 4 (2016), 787–804.

[309] Ware, C. *Information visualization: perception for design*. Elsevier, 2012.

[310] Wiese, R. *The Phonology of German*. Oxford University Press, Oxford, 2000.

[311] Willems, N., van de Wetering, H., and van Wijk, J. J. Evaluation of the visibility of vessel movement features in trajectory visualizations. *Computer Graphics Forum 30*, 3 (2011), 801–810.

[312] Winters, K. M., Lach, D., and Cushing, J. B. Considerations for characterizing domain problems. In *Proc. of the Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization (BELIV)* (2014), pp. 16–22.

[313] Wise, J. A. The ecological approach to text visualization. *Journal of the American Society for Information Science (JASIS) 50*, 13 (1999), 1224–1233.

[314] Wismüller, A., Verleysen, M., Aupetit, M., and Lee, J. A. Recent advances in nonlinear dimensionality reduction, manifold and topological learning. In *Proc. 18th European Symposium on Artificial Neural Networks (ESANN)* (2010).

[315] Wongsuphasawat, K., Moritz, D., Anand, A., Mackinlay, J. D., Howe, B., and Heer, J. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Trans. on Visualization and Computer Graphics 22*, 1 (2016), 649–658.

[316] Wright, W., Schroh, D., Proulx, P., Skaburskis, A., and Cort, B. The sandbox for analysis: Concepts and methods. In *ACM SIGCHI Conf. Human Factors in Computing Systems (CHI)* (2006), ACM, pp. 801–810.

[317] Xu, K., Attfield, S., Jankun-Kelly, T. J., Wheat, A., Nguyen, P. H., and Selvaraj, N. Analytic provenance for sensemaking: A research agenda. *IEEE Computer Graphics and Applications 35*, 3 (2015), 56–64.

[318] Yi, J. S., Kang, Y.-a., Stasko, J. T., and Jacko, J. A. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. on Visualization and Computer Graphics 13*, 6 (2007), 1224–1231.

[319] Yi, J. S., Kang, Y.-a., Stasko, J. T., and Jacko, J. A. Understanding and characterizing insights: how do people gain insights using information visualization? In *Proc. of the Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization* (2008), ACM, p. 4.

[320] Young, D., and Shneiderman, B. A graphical filter/flow representation of boolean queries: A prototype implementation and evaluation. *Journal of the American Society for Information Science (JASIS) 44*, 6 (1993), 327–339.

[321] Yu, B., Liu, R., and Yuan, X. MLMD: Multi-Layered Visualization for Multi-Dimensional Data. In *EuroVis - Short Papers* (2013), M. Hlawitschka and T. Weinkauf, Eds., The Eurographics Association.

[322] Zhang, L., Rooney, C., Nachmanson, L., Wong, B. L. W., Kwon, B. C., Stoffel, F., Hund, M., Qazi, N., Singh, U., and Keim, D. A. Spherical similarity explorer for comparative case analysis. In *Visualization and Data Analysis* (2016), pp. 1–10.

[323] Zhou, L., Burgoon, J. K., Nunamaker, J. F., and Twitchell, D. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group decision and negotiation 13*, 1 (2004), 81–106.

[324] Zhuo, W., and Nadjin, Y. MalwareVis: Entity-based visualization of malware network traces. In *Proc. Int. Symp. Visualization for Cyber Security, VizSec* (2012), ACM, pp. 41–47.

[325] Zuk, T., and Carpendale, M. S. T. Visualization of uncertainty and reasoning. In *Intern. Symp. on Smart Graphics SG* (2007), pp. 164–177.