

Human-Oriented Statistical Modeling: Making Algorithms Accessible through Interactive Visualization

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Dipl.Ing Thomas Mühlbacher

Matrikelnummer 0625075

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.Ing. Dr.techn Eduard Gröller
Zweitbetreuung: Dr. Harald Piringner

Diese Dissertation haben begutachtet:

Cagatay Turkey

Helwig Hauser

Wien, 30. August 2018

Thomas Mühlbacher

Human-Oriented Statistical Modeling: Making Algorithms Accessible through Interactive Visualization

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.Ing Thomas Mühlbacher

Registration Number 0625075

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.Ing. Dr.techn Eduard Gröller

Second advisor: Dr. Harald Piringner

The dissertation has been reviewed by:

Cagatay Turkey

Helwig Hauser

Vienna, 30th August, 2018

Thomas Mühlbacher

Erklärung zur Verfassung der Arbeit

Dipl.Ing Thomas Mühlbacher
Donau-City Straße 12/1/83, 1220 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 30. August 2018

Thomas Mühlbacher

Acknowledgements

This work would not have been possible without the help of several magnificent people. First and foremost, I thank *Harald Piringer* for his close mentorship throughout my entire dissertation project. Without Harald's encouragement, I would not have started a PhD in the first place, and most likely, not seen it through. As my team coordinator at VRVis¹, he taught me many important lessons on paper writing and work management in general. Besides, Harald sacrificed many hours and nights of his own time to help shape the ideas and papers described in this thesis, which I will never forget.

Second, I wish to thank my official supervisor *Edi Gröller* from TU Wien for his great support. I highly appreciated his encouraging feedback and friendly advice during my endeavours of shaping and finishing this thesis. Moreover, it was his remarkable teaching that sparked my enthusiasm for the field of visualization years ago. Thank you, Meister!

Furthermore, I wish to thank my other great co-workers at VRVis for their encouragement and support. Most notably, I thank my friend *Clemens Arbesser*, who has helped a lot with several papers, even when not credited as co-author. He also supplied my brain with fuel in the form of tea and regular lunch ($\leq 12:30$ h, give or take), as well as lots of good advice. I also thank my other co-workers, most notably *Stephan Pajer*, *Florian Spechtenhauser*, *David Pfahler*, *Lisa Deckert*, and *Oliver Rafelsberger*, as well as the many colleagues I had the pleasure of collaborating with over the years. Finally, I thank the admins of VRVis for being so uncomplicated and supportive when it comes to traveling, and any other aspect regarding work at this institution. This is not taken for granted.

On a personal level, I wish to thank my wonderful parents, *Sylvia* and *Christian Mühlbacher* for their big support, mentally and financially, without which my studies and scientific work would not have been possible. Finally, my biggest thanks go out to my lovely, brilliant wife *Kathleen Jimenez-Mühlbacher*. Kat has not only endured my absences and unrest during nights and morning hours, but always supported me with all her heart. She made this project possible by keeping the rest of my life running, made sure I ate from time to time, and stayed healthy and sane. Moreover, she read drafts of my work, provided helpful feedback until the red thread was clear, and encouraged me over and over to go on. I could not have made this without you! Thanks for everything!

¹VRVis is funded by BMVIT, BMDW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174) which is managed by FFG.

Abstract

Statistical modeling is a key technology for generating business value from data. While the number of available algorithms and the need for them is growing, the number of people with the skills to effectively use such methods lags behind. Many application domain experts find it hard to use and trust algorithms that come as black boxes with insufficient interfaces to adapt. The field of *Visual Analytics* aims to solve this problem by a human-oriented approach that puts users in control of algorithms through interactive visual interfaces. However, designing accessible solutions for a broad set of users while re-using existing, proven algorithms poses significant challenges for the design of analytical infrastructures, visualizations, and interactions.

This thesis provides multiple contributions towards a more human-oriented modeling process: As a theoretical basis, it investigates how user involvement during the execution of algorithms can be realized from a technical perspective. Based on a characterization of needs regarding intermediate feedback and control, a set of formal strategies to realize user involvement in algorithms with different characteristics is presented. Guidelines for the design of algorithmic APIs are identified, and requirements for the re-use of algorithms are discussed. From a survey of frequently used algorithms within R, the thesis concludes that a range of pragmatic options for enabling user involvement in new and existing algorithms exist and should be used.

After these conceptual considerations, the thesis presents two methodological contributions that demonstrate how even inexperienced modelers can be effectively involved in the modeling process. First, a new technique called TreePOD guides the selection of decision trees along trade-offs between accuracy and other objectives, such as interpretability. Users can interactively explore a diverse set of candidate models generated by sampling the parameters of tree construction algorithms. Visualizations provide an overview of possible tree characteristics and guide model selection, while details on the underlying machine learning process are only exposed on demand. Real-world evaluation with domain experts in the energy sector suggests that TreePOD enables users with and without statistical background a confident identification of suitable decision trees.

As the second methodological contribution, the thesis presents a framework for interactive building and validation of regression models. The framework addresses limitations of automated regression algorithms regarding the incorporation of domain knowledge, identifying local dependencies, and building trust in the models. Candidate variables for

model refinement are ranked, and their relationship with the target variable is visualized to support an interactive workflow of building regression models. A real-world case study and feedback from domain experts in the energy sector indicate a significant effort reduction and increased transparency of the modeling process.

All methodological contributions of this work were implemented as part of a commercially distributed Visual Analytics software called Visplore. As the last contribution, this thesis reflects upon years of experience in deploying Visplore for modeling-related tasks in the energy sector. Dissemination and adoption are important aspects of making statistical models more accessible for domain experts, making this work relevant for practitioners and application-oriented researchers alike.

Kurzfassung

Statistische Modellierung ist eine Schlüsseltechnologie, um Daten effizient nutzen und verwerten zu können. Noch nie war das Angebot und auch die Nachfrage nach entsprechenden Methoden und Algorithmen so groß wie heute. Die Zahl der Personen, die diese Methoden effektiv einsetzen können, ist jedoch gering. Oft stehen Algorithmen nur als *Black Boxes* ohne benutzerfreundliche Schnittstellen zur Verfügung, was deren Einsatz für FachexpertInnen ohne statistischen Hintergrund schwierig macht. Der Bereich der *Visual Analytics* verfolgt einen menschenorientierten Lösungsansatz für dieses Problem, der AnwenderInnen durch interaktive, visuelle Schnittstellen in die Lage versetzt, Algorithmen intuitiv zu steuern. Jedoch stellt die Öffnung von bestehenden, erprobten Algorithmen für verschiedene Arten von Usern gravierende Herausforderungen für die Gestaltung von Software-Infrastrukturen, Visualisierungen und Interaktionsmöglichkeiten dar.

Diese Doktorarbeit trägt in mehrerer Hinsicht zur Schaffung eines menschengerechteren Modellierungsprozesses bei. Als theoretische Grundlage wird untersucht, wie die Einbeziehung von Menschen in laufende Algorithmen aus technischer Sicht realisiert werden kann. Basierend auf einer Charakterisierung von Arten der frühzeitigen Kommunikation mit Algorithmen werden formale Strategien erarbeitet, um diese Kommunikation für verschiedene Klassen von Algorithmen zu realisieren. Richtlinien für das Design von algorithmischen APIs werden identifiziert, und Voraussetzungen für die Wiederverwendung von existierenden Algorithmen werden diskutiert. Mittels einer Studie von häufig verwendeten Algorithmen in der Umgebung R kommt die Arbeit zu dem Schluss, dass eine Reihe von pragmatischen Optionen zur Einbeziehung von Menschen in neuen und existierenden Algorithmen vorliegt, und genutzt werden sollte.

Nach diesen konzeptuellen Überlegungen stellt die Arbeit zwei methodische Ansätze vor, die zeigen, wie selbst Menschen mit wenig Modellierungserfahrung effizient in den Modellierungsprozess einbezogen werden können. Zuerst wird eine Technik namens TreePOD beschrieben, die die Auswahl von Entscheidungsbäumen unter widersprüchlichen Zielvorgaben unterstützt, wie etwa Genauigkeit und Interpretierbarkeit. BenutzerInnen können hierbei aus einer Palette von verschiedenartigen Modellen wählen, die durch Variation von algorithmischen Parametern vorberechnet werden. Visualisierungen bieten einen Überblick über erreichbare Modellcharakteristiken und helfen bei der Modellauswahl, während Details über die zugrunde liegenden algorithmischen Prozesse nur bei Bedarf kommuniziert werden. Eine Evaluierung von TreePOD mit FachexpertInnen aus der

Energiewirtschaft hat ergeben, dass der Ansatz selbst Menschen ohne tiefe statistische Ausbildung eine effiziente, selbstsichere Auswahl von Entscheidungsbäumen ermöglicht.

Der zweite methodische Beitrag der Arbeit beschreibt ein Rahmenkonzept für die interaktive Erstellung und Validierung von Regressionsmodellen. Im Gegensatz zu früheren Regressionstechniken erlaubt der Ansatz, Domänenwissen in die Algorithmen einzubringen, lokale Abhängigkeiten zu identifizieren und Vertrauen in die Modelle aufzubauen. Mögliche Variablen zur Modellverfeinerung werden automatisch gereiht, und ihr Zusammenhang mit der Zielgröße der Regression wird visuell dargestellt. Dadurch unterstützt der Ansatz einen interaktiven Arbeitsablauf zur Erstellung von Regressionsmodellen. Eine Fallstudie mit echten Daten aus der Energiewirtschaft, sowie Rückmeldungen von einigen FachexpertInnen legen nahe, dass der Ansatz eine signifikante Aufwandsreduktion und Transparenzerhöhung für den Modellierungsprozess darstellt.

Alle methodischen Beiträge der Arbeit wurden als Teil einer kommerziell vertriebenen Visual-Analytics-Applikation namens *Visplore* implementiert. Als letzte Beitragsleistung reflektiert diese Arbeit über jahrelange Erfahrung mit dem Vertrieb und Einsatz von *Visplore* für modellbezogene Aufgaben in der Energiewirtschaft. Verbreitung und Akzeptanz sind wichtige Aspekte der Bestrebung, statistische Modelle zugänglicher zu machen, wodurch diese Arbeit sowohl für Anwender als auch anwendungsnahe Forscher an Relevanz gewinnt.

Contents

Abstract	ix
Kurzfassung	xi
Contents	xiii
I Overview	1
1 Motivation and Overview	3
1.1 The Problem: A Human-Unaware Modeling Process	5
1.2 Empowering Humans by Interactive Visualization	6
1.3 Thesis Goals and Methodology	9
1.4 Thesis Contributions	11
1.4.1 User Involvement in Ongoing Computations	11
1.4.2 Guided Selection of Decision Trees along Trade-Offs	11
1.4.3 Domain Knowledge-Driven Building of Regression Models	12
1.4.4 Lessons Learned from Real-World Deployments	13
1.4.5 Contribution Overview	14
1.5 Thesis Structure and Authorship Statement	15
2 Background and Related Work	19
2.1 Background: Short Overview of Statistical Modeling	19
2.2 Towards A Human-Oriented Modeling Process	20
2.2.1 Process Models and other Conceptual Efforts	21
2.2.2 Technical Integration of Modeling and Visualization	24
2.2.3 Human-Oriented Interface Design	27
2.3 Visual Analytics Solutions for Modeling Tasks	41
2.3.1 Classification	42
2.3.2 Regression	46
2.3.3 Clustering and Dimension Reduction	51
3 Discussion and Conclusions	63
	xiii

II Publications	69
A Opening The Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations	71
B TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees	83
C A Partition-Based Framework for Building and Validating Regression Models	95
D Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System	107
E Statistical Forecasting in the Energy Sector: Task Analysis and Lessons Learned from Deploying a Dashboard Solution	111
Bibliography	119
Curriculum Vitae	136

Part I
Overview

Motivation and Overview

We live in a data-driven world. With the ability to collect and store data about any digital process at low cost, an unprecedented demand for information has swept into virtually every sector of business and industry. Pouring in from millions of sensors, online services, and scientific simulations, the amount of collected data is said to double every two years [TGRM14]. At the cost of increasing resource consumption [SSS⁺16] and certain sacrifices regarding privacy [Mar17], *Big Data* bears an enormous potential for creating business value and fostering innovation [MCB⁺11]. Opportunities range from massive cost savings by automation, to obtaining a better understanding of the underlying processes, leading to new discoveries, higher efficiency, and better decisions. According to recent business reports, exploiting this potential will be essential for staying competitive in the years to come, putting significant pressure on companies to swim with the tide as not to stay behind [MCB⁺11, HBC⁺16]. However, the size, complexity, and speed at which new data arrives poses significant challenges for the analytical infrastructures of today's organizations. Looking at all of the data is no longer an option, and widespread tools like spreadsheet software have not been designed for the tasks and intricacies of data pouring in at high rates. Turning big data into business value calls for more scalable approaches to knowledge extraction, pushing innovation in computer science, and forcing companies to stock up on data-oriented talent. In the end, all the data in the world becomes worthless without the proper means to analyze it.

In this context, automatic analysis methods like *statistical modeling* have become more relevant than ever. Statistical models are simplified representations of data that can be *learned* by the use of algorithms [HTF09]. A simple example is that of a *linear regression model*, which summarizes an observed relationship between two numeric variables by a single trend line. More complex patterns and relationships can be expressed using a variety of model types that have been proposed over the past decades. In general, modeling exploits the processing power of computers to find and represent patterns in a

consistent, automatized fashion. This supports the generation of value from big data in multiple ways:

First, models support the knowledge generation of humans by breaking down large, complex data into concise, interpretable patterns. Typical examples are looking for untapped potential in business data, or finding causes of a sudden quality loss in a manufacturing process. For knowledge discovery, the model is often nothing more than a means to an end, while the value is generated by human interpretation and action.

A second, increasingly important purpose of models is using them directly as part of an automated process. In a typical automation scenario, models constantly make predictions about ongoing processes based on live data, while human involvement during operation is limited to monitoring. Examples include automated stock trading in finance [VT03] and predictive maintenance in facility management [SSP⁺15]. In such contexts, models are key assets that need to be carefully selected and maintained over time, as their accuracy directly translates to cost savings.

Despite the remarkable progress of modeling algorithms in recent years, their adoption in practice faces a significant lag that limits their usefulness: Economies face an enormous shortage of people with the qualifications and expertise to effectively apply statistical modeling to real-world problems [MCB⁺11]. Domain experts are highly skilled within their fields, but typically lack deep statistical, algorithmic backgrounds required for an effective application of such methods. Data has swept faster into every department and sector of the economy than education and training of human resources could keep pace with. Within one or two decades, *data science*, i.e., the extraction of knowledge from data with tools like statistical modeling, has turned into one of the most sought skills in the job market [DP12]. In fact, the 2011 McKinsey report on big data projected a gap of 140,000 - 190,000 deep analytical talent positions for 2018 in the US alone [MCB⁺11]. According to their 2016 report on data analytics, this *talent gap* persists to be one of the biggest hurdles in exploiting big data, effectively keeping billions of dollars from the economy [HBC⁺16].

Universities and other training institutions have started adding data-oriented analytics programs to improve this situation [Mor15]. However, educating new analytical graduates takes years, and even with importing additional talent, these measures are not regarded as sufficient to fill the ever-growing gap [MCB⁺11]. Moreover, generating value from data is not possible without a thorough understanding of the application domain, which again takes years to obtain. Therefore, these reports stress that a significant amount of talent needs to be re-trained in place, i.e., enabling the existing body of domain experts to join the analytic force instead of gradually replacing them [MCB⁺11, HBC⁺16].

Inspired by this point of view, the human resource gap can be reformulated as a computer science problem: *What are the gaps in current statistical modeling infrastructures that prevent domain experts from adoption? And what can be done on the technological side to enable a better use of existing human and computational resources?* These are the central research questions motivating this thesis. There have never been so many algorithms,

toolkits, and communities dedicated to statistical modeling, machine learning, and data mining. There also have never been so many companies and institutions in desperate need of these techniques, but struggling due to a mismatch between requirements and available skills. This work aims at helping to bridge this gap by making established modeling infrastructures more human-oriented. By developing *visual* interfaces to algorithms that take human creativity and domain knowledge, as well as the educational background of users into account, the thesis strives to enable more of the existing domain experts to master the information age.

1.1 The Problem: A Human-Unaware Modeling Process

The successful use of statistical models relies on both the processing power of computers, and the domain expertise of humans [BL10]. Computers are highly efficient and reliable in performing calculations, trying out possibilities, and delivering consistent results. Humans are able to get the “big picture”, as needed for transforming real-world problems into modeling tasks. They have background knowledge about problem domains, and can assess whether a model is useful for an intended purpose. Moreover, they possess creativity, intuition, and common sense – traits shown to effectively complement the limitations of automated analysis [Kei01]. It is crucial that humans can incorporate all these benefits to refine models, and to steer the process towards results that are useful for particular applications.

As a likely scenario, domain experts would attempt to use modeling algorithms “off the shelf”, as provided by popular computation environments like SPSS, R, or Matlab. Implementations often come with default parameters that make it straightforward to obtain some result as a starting point. However, the underlying computations are often not transparent enough to understand why a particular result was achieved, and how to effectively adapt the process according to particular needs. In most cases, results can only be influenced by low-level parameters, such as termination criteria or statistical optimization details with telling names like ‘gamma’, ‘method’, or ‘C’. While statistical experts with programming skills may find this sufficient, inexperienced modelers may not understand how parameters relate to desired model characteristics like “more interpretable”, “more report-friendly numbers”, or “avoid false positives at all costs”. In other words, algorithms do not speak the language of domain experts and vice versa, making it hard to provide effective feedback. Without better guidance, trial-and-error is often the only option, leading to inefficiency, and possibly frustration if no parameter settings exist to achieve certain goals.

Interactive visualization has been shown to enable an effective discourse between humans and statistical models [KKEM10]. Most computation environments offer some form of static visualizations of model results, which may enable humans to generate ideas for refinement. However, without intuitive interfaces for triggering recalculations, the incorporation of insights remains challenging. Direct manipulation of data and models, for example, has proven effective and straightforward to learn [Shn83], but is all too rare

in standard environments [SSZ⁺17]. Furthermore, visualizations are usually limited to information about the “final” results of an algorithm, such as quality metrics or error plots. Explanations of *why* an algorithm has made certain decisions along the way, e.g., preferring certain explanatory variables over others, are typically not exposed. However, domain experts often find it necessary to learn about a model’s inner workings to provide useful feedback [SSZ⁺17], and to build trust in the models. Without better transparency of the entire modeling process, and more support for intuitive user involvement along the way, human potential may be wasted.

Aggravating this challenge, most algorithm implementations were not designed to provide *any* form of feedback during their execution. Once started for data and parameters, they run in isolation until eventually returning a result [Fek13]. Wrong assumptions such as bad parameter choices, or data quality problems can only be recognized and fixed after an entire computation has finished. For large data or advanced algorithms like clustering or dimension reduction, this may take too long to keep the attention of users [CRM91]. While waiting times hinder the efficiency of any data analysis, they are particularly painful for inexperienced modelers, as their trial-and-error is typically the least guided. With intermediate results that provide an idea of what to expect, and the possibility to cancel and restart the computation, considerable shortcuts could be provided [FPDm12]. But as long as algorithms are not guaranteed to respond after at most a few seconds [CRM91], methods to exploit human strengths by visualization and interaction are hard to realize from a technical perspective.

In summary, many established modeling algorithms come as *black boxes* with insufficient interfaces for domain experts to adapt. They are not human-oriented enough to allow users without programming skills and deep statistical backgrounds to effectively achieve what they want. Aside from inefficiency, possible consequences include limited confidence, settling for suboptimal models, or not using statistical modeling altogether. To improve this situation, modeling infrastructures need to become more transparent, accessible, and inclusive with respect to different educational backgrounds, which is the key motivation for this thesis.

1.2 Empowering Humans by Interactive Visualization

A key requirement for a human-oriented modeling process is an effective communication between users, algorithms, and models. Visualization has proven to be a highly effective medium for conveying information *to* users [FWSN08]. The popular saying “a picture is worth a 1000 words” is true in that humans acquire more information by vision than by all other senses combined [War04]. Together with the processing power of billions of neurons, the human visual system is an enormously effective seeker and interpreter of patterns. Moreover, trends like the growing popularity of infographics in business and journalism suggest that visualization is not just an effective, but also an inclusive language capable of reaching broad audiences from diverse fields [Ber16].

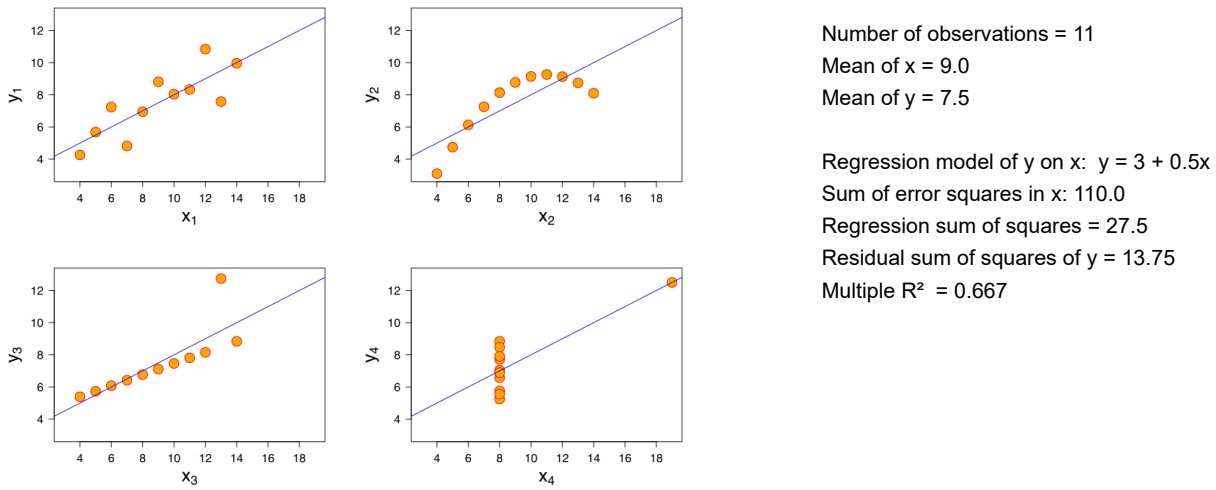


Figure 1.1: Anscombe’s quartet [Ans73]: Four datasets where common statistical properties are exactly the same. Visualization immediately reveals the strong qualitative differences.

In the context of modeling, visualization can answer important questions in an effective and intuitive way. Examples include [WCH15]:

“How does the model look like? How does the model change when its parameters change? How does the shape of the model compare to the shape of the data? Is the model fitting uniformly good, or good in some regions, but poor in other regions? Where might the fit be improved?”

Domain experts must be able to answer such questions to effectively criticize models and to obtain a flawless view of the modeled phenomena [WCH15]. The usefulness of visualization for this purpose is illustrated in Figure 1.1: “Anscombe’s quartet” shows four significantly different datasets that are indistinguishable by common statistical summaries, model parameters, and error metrics [Ans73]. While numbers are clearly misleading in this case, visualization immediately reveals the underlying patterns and severe qualitative differences. More importantly, it does not require a degree in statistics to immediately understand, for example, that a linear approximation fits some of the data distributions in Figure 1.1 better than others. In general, visualization fosters the formation of mental images, enabling humans to recognize wrong assumptions, and to match models against expectations, hypotheses, and analysis goals. Moreover, visualization may spark creativity and intuition for adapting the models to better match the needs of application contexts.

Interactive visualization goes one step further by providing user interfaces to immediately incorporate insights from visualizations, which in turn, updates the visualizations. This enables a dialog between the human brain and the data, where humans can focus on interesting parts or try out different scenarios in an intuitive fashion [War04]. Interac-

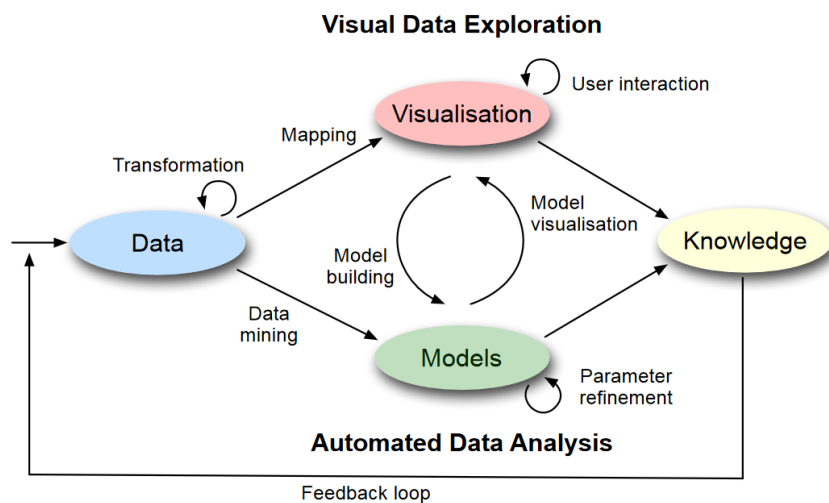


Figure 1.2: By combining automated analysis with interactive visualization, the Visual Analytics process enables a human-centered loop of knowledge discovery [KAF⁺08].

tion paradigms such as zooming, filtering, and linking multiple visualizations enable an exploration of the data from different angles, which fosters the discovery of unexpected patterns and relationships. In this respect, interactive visualization and algorithms for knowledge discovery are two different approaches to the same goal, but with complementary advantages and disadvantages [BL10]: Algorithms complement scalability issues of visualization by automatically focusing on relevant structures, while visualization and interaction enable the incorporation of human strengths and expertise to steer the algorithms towards particular analysis goals. Thus, it seems only logical to strive for a tight integration of the two disciplines.

The respective research direction was termed *Visual Analytics* in 2005 [TC05], and later defined as “*combining automated analytics techniques with interactive visualizations for an effective understanding, reasoning and decision-making on the basis of very large, complex datasets*” [KKEM10]. Visual Analytics aims at enabling an interactive, human-centered process to extract knowledge from the data (see Figure 1.2): After the data is analyzed first to show the important patterns, the Visual Analytics process enters a feedback loop where humans can obtain more knowledge by refining hypotheses, models, and visualizations through interaction [KAF⁺08]. This tight integration of computation and cognition supports tasks like experimenting with data subsets for training and validation, comparing model variants to obtain confidence, and steering computations based on domain knowledge. All of these tasks are difficult for domain experts in current practice, suggesting that visualization may indeed be a suitable approach to reach more users.

This thesis is by no means the first effort to involve humans in statistical modeling through interactive visualization. It builds upon a large body of Visual Analytics research, which shows that human-centric approaches can produce better results than purely automatic,

machine-driven methods (see Chapter 2). However, the field of Visual Analytics is relatively young, and still faces research gaps that have hindered a broader impact in practice. While many approaches exist, there are still important tasks and aspects of modeling, which have not been sufficiently addressed. Moreover, not all Visual Analytics approaches were designed to scale for users with different backgrounds, which may limit the applicability in some cases. Despite all benefits of interactive visualization, it is not trivial to build accessible solutions that enable an efficient modeling process for a broad set of users. In fact, the practical challenges for adoption (Section 1.1) have been acknowledged as difficult open research questions by the Visual Analytics community. In a recent survey of open challenges on the road to human-oriented machine learning [SSZ⁺17], Sacha et al. explicitly called for more work on:

- *Designing interactions for machine learning adaption*, i.e., developing better approaches for understandable interaction with models, such as mapping simple, semantically meaningful user input to complex algorithmic modifications.
- *Providing guidance* for application domain experts, such as recommendations for model refinement in a language that does not require deep statistical backgrounds.
- *Interoperability between algorithms and visualization*, i.e., overcoming technical aspects that preclude a tight integration, such as algorithms running in isolation, inaccessible internal structures, and intransparent algorithmic decisions.

Summarizing the first sections of this thesis, there is a significant need for more work on human-oriented statistical modeling, which has been emphasized by practitioners and researchers alike. Visualization can play an important role, but multiple communities have stressed the importance of interdisciplinary work to overcome challenges like the insufficient interoperability between algorithms and visualization [SSZ⁺17, KMRV15]. By contributing to both the visualization and the algorithmic side of Visual Analytics, this thesis aims at advancing the field towards a more effective, accessible, and inclusive process of statistical modeling.

1.3 Thesis Goals and Methodology

The overarching goal of this work is to advance the *democratization*¹ of statistical models by making established algorithms and infrastructures more human-oriented. The idea is to strive for a tighter integration between computation and interactive visualization, such that human strengths can be made use of, human needs like trust-building are catered to, and educational backgrounds can be seen as an opportunity rather than a limitation. Specifically, this vision comprises three sub-goals:

¹Democratization in the sense of (1) enabling more people to use tools that were previously reserved for a few [Nis17], and (2) being less dependent on preconfigured off-the-shelf solutions [SSZ⁺17].

G1: Develop human-oriented solutions for under-addressed modeling tasks

While much previous work has been dedicated to involving humans in various aspects of the modeling process (see Chapter 2), there are still important tasks that domain experts find hard to accomplish in practice. Regression modeling, for example, has been lacking support for an efficient incorporation of domain knowledge when the work of this thesis started. As a first goal (G1), this thesis seeks to identify aspects and sub-tasks of modeling that are highly relevant for practitioners but still underexplored in the literature, and to design new Visual Analytics approaches to fill these gaps. The identification of gaps is based on problem-driven research collaborations with company partners in the energy sector, and backed up through careful review of the existing Visual Analytics literature. On the one hand, the goal is to support particular workflow gaps for the collaborating domain experts, which is evaluated based on feedback after deployment. On the other hand, the generalizability of the solutions for other tasks, model types, and application scenarios shall be investigated, to maximize the democratization impact of the contributions beyond particular application domains.

G2: Enable users without deep analytical backgrounds

With big data on the rise, the educational and professional backgrounds of people tasked with statistical modeling are becoming more and more diverse. While domain experts are highly skilled within their fields, they may lack a deep background in statistics, programming, and visualization, but still need to build good models. The second thesis goal (G2) refers to investigating how (1) visualization, (2) interaction, and (3) guidance of domain experts can be realized without requiring particular backgrounds. For visualization, G2 makes familiarity and the ease of learning two primary design goals that need to be balanced with traditional objectives like effectiveness. Regarding interaction, algorithmic details should be exposed only on demand, but never necessary to achieve particular modeling goals. Here, the challenge is to design meaningful actions that are simple to perform, but trigger complex algorithmic changes under the hood [EFN11, SSZ⁺17]. For the guidance of inexperienced modelers, G2 seeks to investigate whether a precomputation of possible refinements or alternative models to choose from finds better acceptance than a deep involvement in every tiny decision along the modeling process. G2 is considered as key design goal for all solutions implemented in this thesis (G1). Its fulfillment is evaluated with real domain experts, who claim to have no particular backgrounds in statistics or visualization.

G3: Foster the re-use of existing, proven algorithm infrastructures

Existing systems and languages for data analysis such as R, Python, and MATLAB have widely been used for a long time and offer a variety of proven algorithms. In most cases, algorithms are used as black boxes that run in isolation, which contradicts the requirements of human involvement through interactive visualization. Achieving early communication with the user thus often involves a reimplementations of algorithms by researchers and practitioners in Visual Analytics, leading to a suboptimal use of resources and lots of proprietary code instead of standardized, tested solutions. As

a third goal (G3), this thesis aims at fostering a better re-use of existing algorithmic resources and infrastructures by the visualization community. By studying conceptual and practical possibilities for integrating existing algorithms, the goal is to raise the awareness and understanding of these possibilities within the Visual Analytics community. A second aspect of the goal refers to studying needs and requirements to the design of algorithmic APIs in favor of user involvement and tight integration. Here, the intention is to encourage algorithm developers to regard the supported degree of user involvement as a more conscious design choice for future algorithms.

1.4 Thesis Contributions

This section briefly describes the contributions of this thesis, and relates them to the goals described in the previous section.

1.4.1 User Involvement in Ongoing Computations

An increasing number of interactive visualization tools stress the integration with computational software like MATLAB and R to access a variety of proven algorithms. In most cases, however, the algorithms are used as black boxes that run to completion in isolation, which contradicts the needs of interactive data exploration. This thesis structures, formalizes, and discusses possibilities to enable user involvement in ongoing computations, which is henceforth referred to as **Opening the Black Box**. Based on a characterization of needs regarding intermediate feedback and control, a key contribution is the formalization of strategies for achieving user involvement in algorithms with different characteristics. In the context of integration, considerations for implementing these strategies either as part of the visualization tool or as part of the algorithm are described. Moreover, guidelines for the design of algorithmic APIs are identified. To assess the practical applicability, a survey of frequently used algorithm implementations within R investigates the fulfillment of these guidelines. The thesis concludes that many pragmatic options for enabling user involvement in ongoing computations exist on both the visualization and algorithm side and should be used.

This contribution mainly addresses thesis goal G3, as its key intention is fostering the re-use of existing, proven infrastructures for modeling. Moreover, enabling user involvement in ongoing computations is an important technical requirement for supporting a human-oriented modeling process as envisioned in G1 and G2. For example, possibilities to detect and correct wrong assumptions during computation are particularly important for inexperienced modelers (G2), whose trial-and-error may involve numerous attempts before reaching their goal.

1.4.2 Guided Selection of Decision Trees along Trade-Offs

Decision trees are a common technique for statistical classification. Besides their use for prediction, their understandable model structure makes them useful for human-oriented

applications like hypothesis generation, reporting, and decision support. When building models for such purposes, a high interpretability by humans is often just as important as a high model accuracy. Other objectives may also play a role, such as acquisition costs of variables, or costs associated with particular error types, which may be only vaguely known. Balancing all objectives is hard to automate because it involves know-how about the domain as well as the purpose of the model.

This thesis contributes a new approach to guide model selection along trade-offs, called **TreePOD**. TreePOD is based on exploring a large set of decision trees, generated by sampling the parameters of tree construction algorithms. Based on this set, visualizations of quantitative and qualitative model aspects provide an overview of possible result characteristics. Along trade-offs between two objectives, TreePOD provides selection guidance by focusing on Pareto-optimal tree candidates. TreePOD also conveys the sensitivity of tree characteristics for possible variations, to support what-if analyses and to increase the confidence in selected models. Real-world case studies and feedback from domain experts in the energy sector suggest that TreePOD enables users with and without statistical background a confident identification of suitable decision trees.

TreePOD relates to the goals of this thesis in multiple ways. Model selection along human-oriented trade-offs involving interpretability or subjective costs is an underexplored problem where only partial solutions exist (G1). With its focus on classification problems, TreePOD addresses a ubiquitous and highly relevant problem in the energy sector and beyond. Moreover, the idea of guiding the selection from proactively generated candidates is not limited to any particular model type, which may broaden its impact for democratization. Regarding the inclusion of domain experts without analytical backgrounds (G2), TreePOD makes an effort by exposing parameters of the underlying machine learning process only on demand. Visualizations are deliberately kept simple, and partially use redundant encodings to facilitate an understanding of possibly unfamiliar aspects. Interactions for refining models and creating alternatives rely on semantically meaningful actions like “*show me more like this*”, or “*round coefficients to integer numbers*”, and can be triggered with a single click. In line with re-using computational infrastructures (G3), TreePOD internally connects to existing decision tree algorithms rather than reimplementing them.

1.4.3 Domain Knowledge-Driven Building of Regression Models

Regression models are used in many application domains for predicting a quantitative dependent variable based on a set of independent variables, called features. Automated approaches for building regression models are typically limited with respect to incorporating domain knowledge in the process of selecting input variables. Other limitations include the identification of local structures, transformations, and interactions between variables. This thesis contributes a framework for building regression models addressing these limitations. The framework ranks any number of features by their relevance for the target variable, and provides visualizations for a qualitative understanding of relationship structures. A central aspect of both visualization and ranking is the partitioning of

feature domains into disjoint regions, giving the contribution its name **Partition-Based Framework**. Partitioning enables a visual investigation of local patterns and largely avoids structural assumptions for the quantitative ranking. The approach supports different tasks in model building (e.g., validation and comparison), as well as an interactive workflow for feature subset selection. A real-world case study and feedback from domain experts after two months of deployment in the energy sector indicate a significant effort reduction for building and improving regression models.

The Partition-Based Framework contributes to multiple goals of this thesis. Compared to other modeling tasks like classification or clustering, regression had received relatively little attention in Visual Analytics when this thesis started. With incorporating domain knowledge and building trust in the models, the framework supports relevant, yet previously unaddressed tasks for a broad set of users (G1). Moreover, several design choices of the framework are particularly geared towards acceptance by domain experts (G2): Visualizations resemble familiar function graphs, and are deliberately kept low-dimensional to avoid overly steep learning curves. Similar to TreePOD, proactively computed model variants guide possible refinement steps by providing a representative preview of what would happen. Interactions can be triggered with a single click, and are kept track of, such that returning to previous steps to try out different scenarios is possible at any time. Regarding the re-use of algorithms (G3), the framework also integrates existing libraries for regression model identification.

1.4.4 Lessons Learned from Real-World Deployments

During all projects described in this thesis, the author was affiliated with the VRVis Research Center in Vienna. VRVis is a non-university research institution conducting applied research projects in close collaboration with industry partners in various sectors. All technical contributions of this thesis were implemented as part of a Visual Analytics software called *Visplore*, developed at VRVis. Solutions based on *Visplore* are available to industry partners, and are also commercially sold to end customers by distribution partners. This close collaboration enabled two aspects for this thesis: First, all new techniques could be evaluated based on deployments with real-world users addressing real-world problems. Furthermore, the collaboration enabled the author to gather real-world experience and feedback regarding the deployment of Visual Analytics software in general, over a period of several years. Reflecting on these experiences is the fourth contribution of this thesis. One particular result from these reflections is a detailed task analysis of statistical forecasting in the energy sector, which provided a better understanding of prevalent gaps. A second aspect of the contribution is a set of **lessons learned** from the various deployments from a technical and commercial perspective.

This contribution is not as tightly coupled with the methodological and conceptual thesis goals G1-G3 as the other contributions. However, dissemination and adoption, i.e, how to reach a broad set of users, is an important practical aspect of democratization. Moreover, the application-oriented perspective from actual users can be seen as a form of practical validation for many aspects contributed in this thesis.

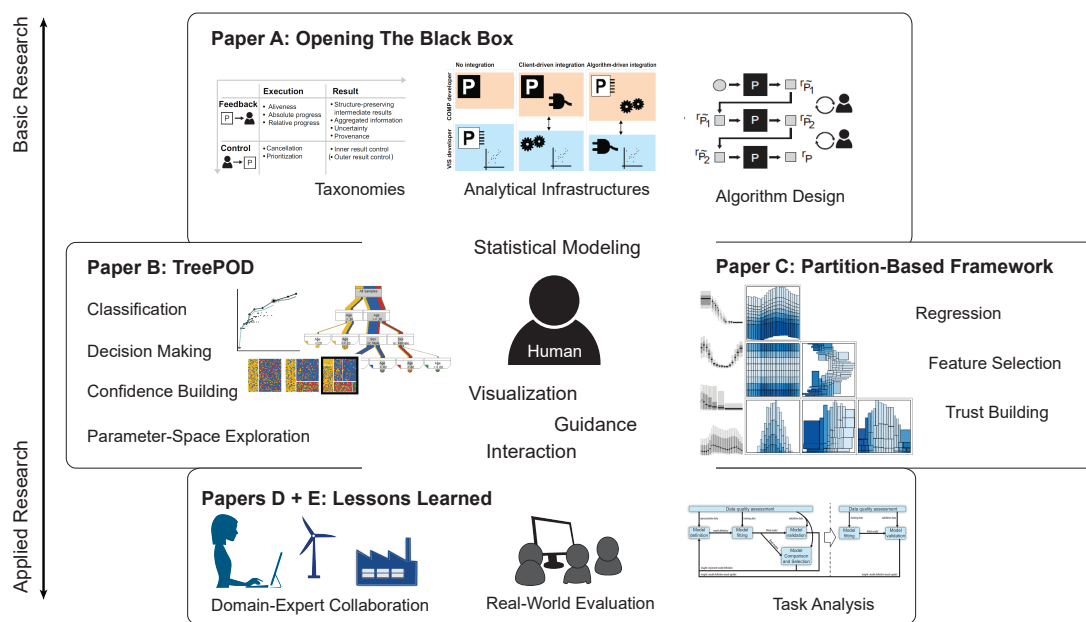


Figure 1.3: The thesis contributions relate to diverse topics of Visual Analytics. Together, they provide a set of measures for bringing statistical modeling closer to domain experts, covering the entire path from basic research to its application in practice.

1.4.5 Contribution Overview

Together, the four contributions aim at providing a comprehensive set of measures to democratize the modeling process on a conceptual, methodological, and practical level. Figure 1.3 shows how the contributions touch diverse topics related to Visual Analytics, and positions them in the spectrum between basic and applied research: On the conceptual side, the work on **Opening the Black Box** presents considerations on how user involvement in algorithms can be realized from a technical point of view. By establishing *what* information should be exchanged with users during computations and *why*, the work can be seen as a technical requirements analysis for human-oriented modeling. Building on this theoretical basis, **TreePOD** and the **Partition-Based Framework** provide methodological contributions that demonstrate *how* even inexperienced users can be effectively involved in the modeling process. To fully cover the path from basic research to its application in practice, the last group of contributions focuses on practice-oriented aspects of democratization like the adoption by real users. Close collaboration with industry partners enabled the author to validate all new methods based on real use-cases, and to reflect on year-long experiences in getting users to adopt Visual Analytics solutions in general. The resulting **lessons learned** are the most practice-oriented contribution of this thesis, intended for practitioners and application-oriented researchers alike.

1.5 Thesis Structure and Authorship Statement

Following the methodology of a cumulative dissertation, this thesis is structured into two parts. The first part provides an overview of the topic at hand, and outlines how the research contributions fit together as parts of a common bigger picture. More specifically, the first part consists of three chapters: The current, first chapter motivates the need for a more human-oriented approach to statistical modeling, and outlines research goals and contributions. Chapter 2 provides an overview of related work, and characterizes gaps regarding the goals of this thesis. Chapter 3 concludes by reflecting on the contributions, discussing their impact, and deriving open research directions for future work. Afterwards, the second part of this thesis presents the main contributions as originally published.

As common in the field of computer science, the research papers constituting this thesis were collaborations of multiple authors. The thesis author, subsequently referred to in first person as “I”, was also the first author of each paper. All papers came out of research projects at the Visual Analysis group at the VRVis Research Center, led by Dr. Harald Piringer. As the head and main supervisor of this group, Harald Piringer had many ideas featured in this thesis, and is also a co-author of all papers. For the individual papers, the contribution of each co-author can be summarized as follows:

- Paper A: Thomas Mühlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair and Marc Streit. Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations, *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643-1652, 2014. [MPG⁺14]

I was responsible for leading this project, and coordinated the collaboration based on regular online meetings. For example, I coordinated joint efforts to survey existing Visual Analytics literature regarding user involvement, and processed the results for the subsequent formulation of common strategies. Moreover, I was deeply involved in shaping the practical contributions of the paper, such as establishing requirements and guidelines for algorithm design, and evaluated their fulfillment in existing algorithm packages. Large parts of the paper were written by me, while Harald Piringer revised my drafts and helped me improve them for submission. Harald Piringer also contributed significantly to the conception and writeup of the theoretical parts of the paper, i.e., characterizing types of user involvement and formulating strategies to achieve them. Marc Streit, Samuel Gratzl, and Michael Sedlmair contributed many ideas to the overall project, and provided valuable feedback on the text. They also helped with literature surveys and the systematic evaluation of existing algorithm packages. Finally, Marc Streit helped to write the related work section of the paper.

- Paper B: Thomas Mühlbacher, Lorenz Linhardt, Torsten Möller and Harald Piringer. TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees, *IEEE Transactions on Visualization and Computer Graphics*, 24(1):174-183, 2018 [MLMP18].

I was the lead person in publishing this work. As such, I was responsible for the writeup of the paper and supplemental materials, as well as the evaluation of TreePOD with users. Moreover, I contributed parts of the implementation, and supervised Lorenz Linhardt in implementing other parts (see below). Harald Piringer contributed many key ideas to the project, including the initial vision of domain expert-friendly model selection along trade-offs. He provided close mentorship throughout the project, and helped significantly with revising all texts for submission. Lorenz Linhardt developed a comprehensive prototype of TreePOD during an internship, which I built upon for the final implementation. He was also involved in design decisions, and helped to prepare one of the evaluation workshops. Torsten Möller introduced Lorenz to our group and provided valuable feedback throughout the project. He also got his entire research group to review early drafts of the paper. Even though not credited as an author, Clemens Arbesser also helped significantly in revising the text in the days before the submission deadline.

- Paper C: Thomas Mühlbacher and Harald Piringer. A Partition-Based Framework for Building and Validating Regression Models, *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962-1971, 2013 [MP13]

Best paper award.

This project was a close collaboration between Harald Piringer and me. I was the lead person in implementing the framework, and in evaluating it with real users. I was also in charge of the writeup, drafted the manuscript, and refined it several times. Harald Piringer provided significant help to improve the text for the submitted version. He also had the initial idea for the partition-based ranking of features, which had sparked the entire project. Subsequently, he was deeply involved in conceptually shaping and refining all aspects of the framework. All design decisions were made in close collaboration between the two of us, building upon feedback from our project partners in the energy sector.

- Paper D: Thomas Mühlbacher and Harald Piringer. Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System. *Proceedings of IEEE Vis Conference 2014, Practitioner Experience Track (poster paper)*, 2014, [MP14]

I wrote the paper and designed the poster for this submission. Harald Piringer is the mastermind behind the Visual Analytics system “Visplore” described in the paper. Together, we reflected on our experiences with its deployment in the energy sector, which enabled me to formulate the “lessons learned”. Harald Piringer also provided valuable feedback to improve the text.

- Paper E: Thomas Mühlbacher, Clemens Arbesser and Harald Piringer. Statistical Forecasting in the Energy Sector: Task Analysis and Lessons Learned from Deploying a Dashboard Solution. *Proceedings of IEEE Vis Conference 2015, Practitioner Experience Track, Short Paper*, 2015, [MAP15]

I wrote this short paper as a significant extension of the poster paper described in the previous paragraph. Based on additional experiences in deploying Visual Analytics solutions in the energy sector, I extended the set of lessons learned by technical and commercial aspects. Harald Piringer helped me to establish a conceptual model of the statistical forecasting process in the energy sector based on a task analysis with domain experts. Clemens Arbesser prepared the figures for the paper, and provided valuable feedback on the writeup.

Background and Related Work

This chapter puts the thesis in perspective of existing efforts for realizing a human-oriented modeling process. First, Section 2.1 provides a brief overview of modeling basics for readers without that background. Afterwards, Section 2.2 describes how various communities have contributed to distinct parts and aspects of a human-centered modeling process. Finally, Section 2.3 describes holistic solutions that successfully wrap up all these parts to support common modeling tasks, in the form of a state of the art report.

2.1 Background: Short Overview of Statistical Modeling

Learning patterns from observations has engaged scientists for centuries [HTF09]. Early examples include the modeling of planetary orbits for navigation [Sti81], predicting the height of children from ancestors [Gal86], and classifying flowers based on properties of their leaves [Fis36]. In those days, all measurements and calculations had to be carried out meticulously by hand. With the introduction of computers, statistical problems increased substantially in their size and complexity [HTF09]. New possibilities of generating, storing, and analyzing data have enabled great opportunities for science and engineering, but also brought many new challenges to the field of statistics.

Given the central role of computation, it is not surprising that computer-science-related fields made significant statistical contributions over the past decades. Challenges in data storage and search, for example, have sparked the field of *data mining*, which has contributed many important algorithms for knowledge discovery in databases. *Machine learning*, a sub-field of artificial intelligence, has brought forth some of the most powerful and flexible learning techniques to date. While these fields may differ regarding terminologies and the communities that brought them into existence, they share a large overlap of methods and goals. Most importantly, they share the idea of *learning from data*, i.e., extracting patterns and interpreting them to understand “what the data says” [HTF09]. To this end, the respective communities have proposed a large number of techniques and algorithms that have been widely used in many application domains.

One way to characterize modeling techniques is based on their learning paradigm, which can either be *supervised*, *unsupervised* or *semi-supervised* [HTF09]: Supervised methods learn a relationship between input and output variables of a system from *training data*, i.e., data instances where the output variables are known. The resulting model can be used to predict the outputs for new, unseen instances. In unsupervised learning, there is no specified output variable, and the goal is to find inherent structure among the inputs, e.g., groups of similar items. Semi-supervised learning falls in-between, with known outputs for some data instances, and a typically large number of instances without.

Each learning paradigm lends itself to particular categories of modeling tasks: On the supervised side, *regression* seeks to predict a quantitative variable based on a number of inputs that can be continuous or categorical. *Classification* on the other hand aims to predict classes of categorical variables, often referred to as “labels”. Among the unsupervised methods, *clustering* aims to group data items into clusters without prior knowledge of explicit class labels. A typical goal is to find clusters that contain similar items, but are very different from other clusters according to a particular similarity metric. As a related unsupervised method, *dimension reduction* tries to find a subset of observed or derived variables that contain the most variation in the data set. Here, the goal is to identify patterns and structures that manifest in high-dimensional subspaces, and are thus not evident on the surface. Semi-supervised variants have been proposed for both supervised and unsupervised methods: Including unlabeled data in predictive tasks may improve the generalization for unseen data, while unsupervised methods like clustering may benefit from indirect labels such as constraints imposed by the user [Zhu06].

Aside from these common categories of modeling tasks, many further methods exist, including association rule mining, anomaly detection, and value imputation. Knowing all these methods and successfully applying them in real-world contexts involves significant expertise. Moreover, real-world analysis often calls for a combination of modeling techniques from multiple categories. For example, predictive modeling based on hundreds of variables may need to be preceded by a dimension reduction step to focus on the most relevant subspaces. Such combinations lead to complex pipelines with many interdependent parameters to be tuned, requiring considerable time to train and validate, even for statistical experts [SSZ⁺17].

2.2 Towards A Human-Oriented Modeling Process

This section summarizes previous efforts to make specific parts and aspects of the modeling process more human-oriented. As a theoretical basis, Section 2.2.1 discusses process models from various communities that emphasize the role of humans in the modeling process. Section 2.2.2 reviews conceptual efforts to realize a tight integration between algorithms and interactive visualization to enable human involvement from a technical perspective. Finally, Section 2.2.3 summarizes efforts in designing user interfaces to the modeling process that account for strengths and weaknesses of humans.

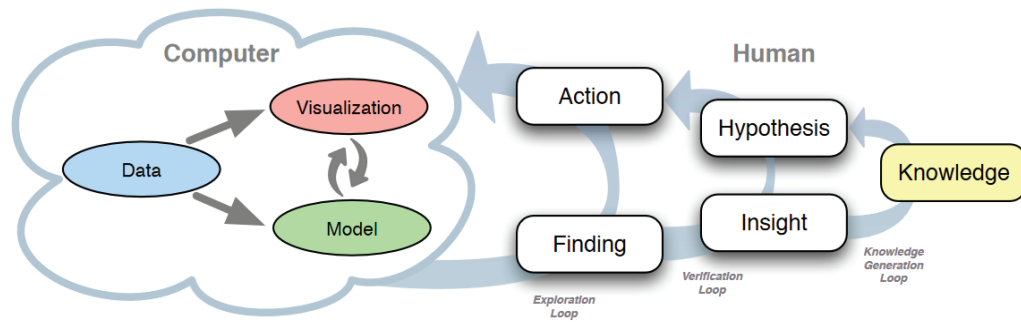


Figure 2.1: The knowledge generation model by Sacha et al. describes the role of computers and humans in Visual Analytics, and stresses a tight integration [SSS⁺14].

2.2.1 Process Models and other Conceptual Efforts

The modeling process is generally understood as a pipeline comprising multiple steps. The need for human involvement between and during these steps has been emphasized by various communities: The Knowledge Discovery Process in Databases [FPSS96], for example, allows user feedback between pipeline steps like data selection, pre-processing, transformation, mining, and interpretation. Similarly, the Reference Model for Information Visualization [CMS99] envisions user interaction with components of the analytical pipeline, and promotes the use of visual interfaces to achieve this goal. The standard model of the Visual Analytics pipeline (see Figure 1.2) characterizes the analysis process as a human-centered loop, where users interact with data, models, and visualizations to generate knowledge [KKEM10]. More recently, Sacha et al. extended this model by a more detailed perspective on human knowledge generation [SSS⁺14]. Their model also clarifies the role of humans and computers in the analysis process, and emphasizes the importance of a tight integration between the two (see Figure 2.1).

While these pipelines describe modeling as a means for human-driven knowledge discovery, other conceptual models involve humans for the purpose of model building as such: In 2003, Fails and Olsen introduced the term *Interactive Machine Learning* (IML) for a process, where users actively influence the decisions made by modeling algorithms [FOJ03]. In contrast to traditional machine learning, where inputs are specified in advance, model updates in IML are rapid and immediately triggered by interaction (see Figure 2.2). The higher responsiveness and shorter design cycles make this approach particularly attractive for domain experts without modeling backgrounds.

Building upon the initial work of Fails and Olsen, several papers have structured the IML process in more detail. Porter et al. propose a design space to characterize user interaction with IML approaches along three dimensions (see Figure 2.3) [PTH13]: (1) *task decomposition*, i.e., the granularity in which labour is subdivided between machine and human to collaborate on tasks; (2) *training vocabulary*, i.e., what types of input users may provide during model training, ranging from low-level labels to higher-level information like matches or constraints; (3) *training dialog*, i.e., at which level and frequency users may interact during the learning process. This design space provides a

2. BACKGROUND AND RELATED WORK

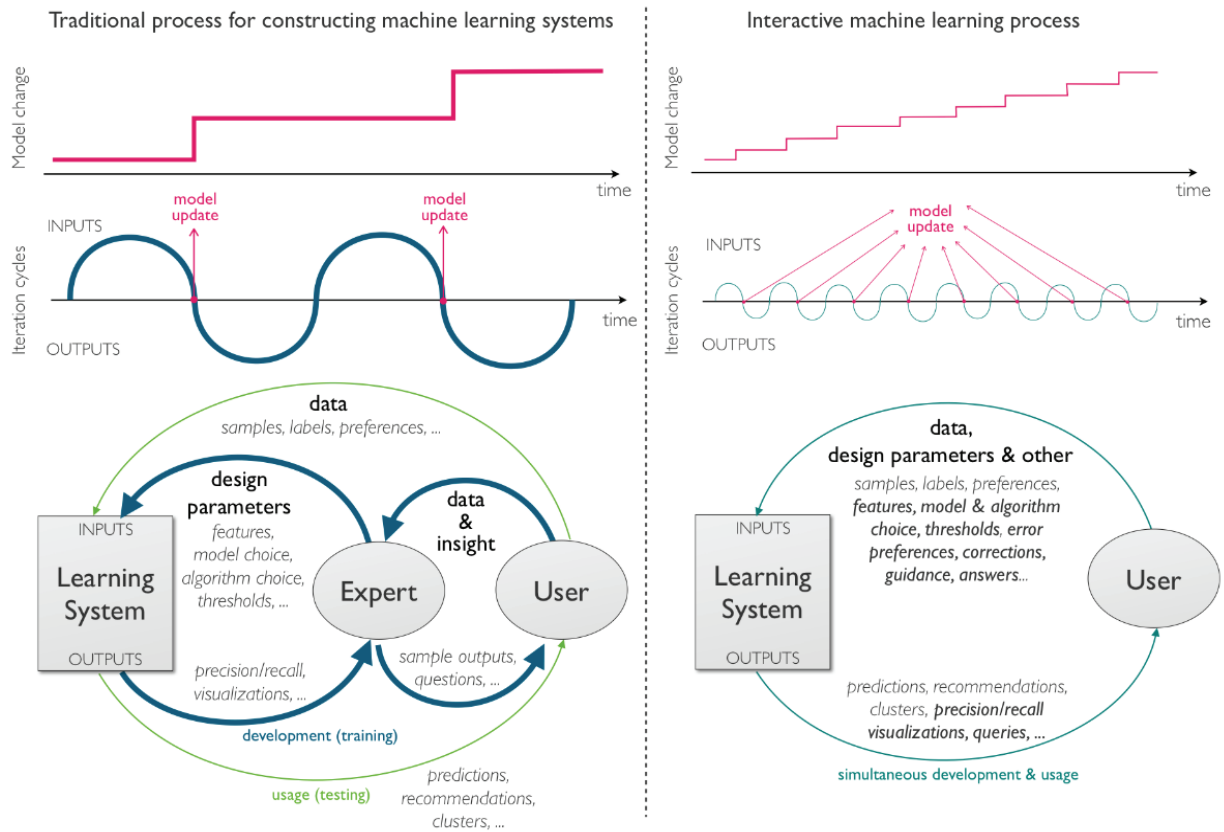


Figure 2.2: In contrast to traditional machine learning, interactive machine learning (IML) involves users in algorithms through rapid cycles of interaction [ACKK14].

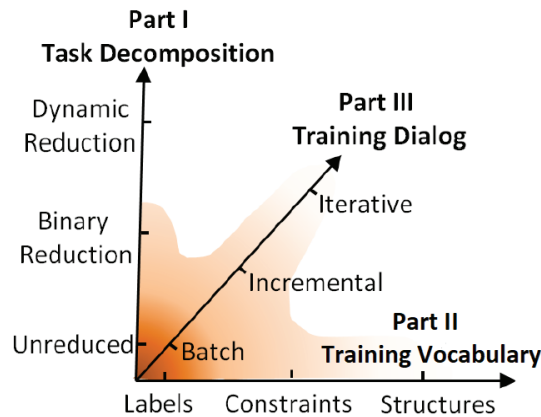


Figure 2.3: Porter et al. characterize interaction with machine learning along three dimensions [PTH13].

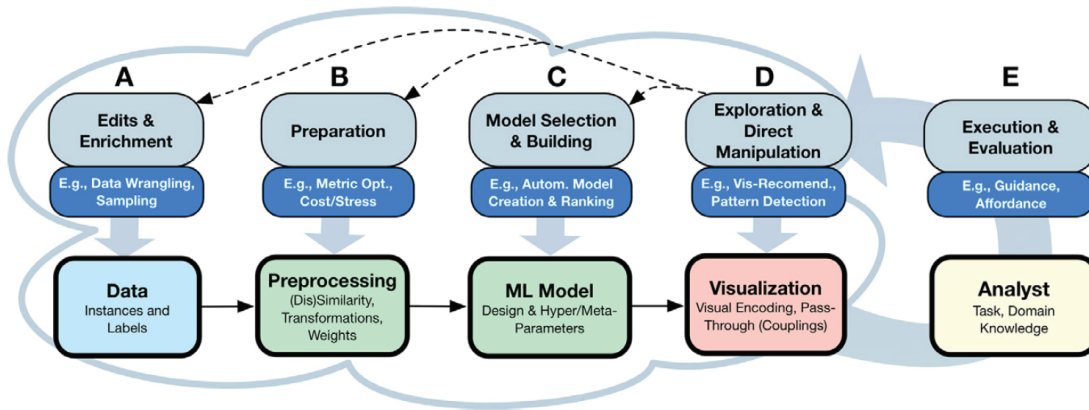


Figure 2.4: Sacha et al. identify opportunities for user involvement for each step of the modeling process [SSZ⁺17].

useful framework for comparing IML approaches to each other, as well as to traditional machine learning, which is situated in the origin of the three-dimensional frame.

Taking a different perspective on IML, Sacha et al. investigate how different components of the modeling pipeline benefit from human involvement (see Figure 2.4)[SSZ⁺17]. They propose a conceptual framework that models human interaction with machine learning in stages like data editing, preparation, model building, and exploration. During all stages, the authors promote interactive visualization as a “lens” between models and the human analyst. Moreover, they stress the importance of meaningful, direct interactions that adapt the machine learning process under the hood. The authors conclude with identifying five research gaps between current practice and IML as envisioned by their framework – three of which are key goals addressed by this thesis (see Chapter 1).

Yet another perspective on IML is provided by Amershi et al., who study the behaviour of users in IML systems based on case studies [ACKK14]. Their findings show that (1) people may violate assumptions of traditional machine learners, e.g. by a tendency to give more positive than negative feedback to learners; (2) people may wish to interact with machine learning in richer ways than anticipated; and that (3) transparent IML systems can lead to better user experience and better models. From this, they derive high level strategies to improve human interaction with machine learning systems, and stress potential benefits of a closer collaboration between the machine learning and human-computer interaction communities.

Summarizing this section, several communities have stressed the importance of user involvement in the modeling process, and the role of visualization as the enabling interface. Process models help to develop a common language across communities [PTH13], and provide high-level guidance on the design of future systems. As common for model papers, many of the discussed works identify gaps and challenges that must be overcome before the envisioned process models can be fully realized. However, they mostly do not provide

	Observed	Suggested
V++	Projection Intelligent Data reduction Pattern Disclosure	Visual Model Building Verification and Refinement Prediction
M++	Model Presentation Patterns Exploration & Filtering	Visualizing Parameter Space & Alternatives Model-Data Linking
VM	White-Box Integration Black-Box Integration	Mixed Initiative KDD

Figure 2.5: Various prevalent integration scenarios and suggested extensions [BL10].

practical guidance for implementing a human-centered modeling process; for example, how a tight integration between visualization and algorithms can be realized, or how user interfaces should be designed. These topics will be discussed in the next two sections.

2.2.2 Technical Integration of Modeling and Visualization

Integrating algorithms and visualization to leverage the benefits of both is the key idea of Visual Analytics [KKEM10]. From a technical perspective, this integration can be achieved in various ways. In their pioneering work from 2010, Bertini and Lalanne identified three recurring integration scenarios in the Visual Analytics literature [BL10]:

1. *Computationally enhanced Visualization (V++)*: techniques that are fundamentally visual, but employ some automatic analysis to complement and improve the visualization. Example: using a projection algorithm to produce relevant views on high-dimensional data (see Section 2.3.3).
2. *Visually enhanced Mining (M++)*: techniques with automatic algorithms as the primary analysis means, while visualization supports an understanding and validation of results. Example: visualizing the structure or results of a model.
3. *Integrated Visualization and Mining (VM)*: tight integrations where neither visualization nor algorithm is predominant. Here, the authors distinguish between *black box approaches* providing a tight feedback loop to restart algorithms for new parameters, and *white box approaches*, where users interact with ongoing computations to affect algorithmic decisions.

In their paper, Bertini and Lalanne also suggest possibilities to improve contributions of each category beyond their surveyed examples (see Figure 2.5). According to this scheme, three contributions of this thesis each implement one of their envisioned suggestions: The Partition-Based Framework supports *Visual Model Building* (V++), TreePOD visualizes the *Parameter Space and Alternatives* (M++), and the work on Opening the Black Box fosters a *Mixed-Initiative KDD* process (VM).

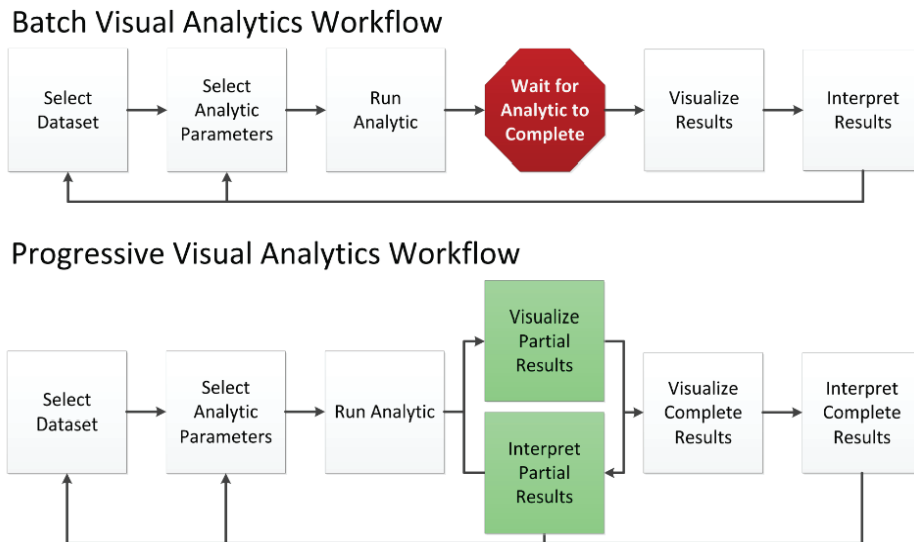


Figure 2.6: The “Progressive Visual Analytics” (PVA) workflow lets users interact with ongoing computations based on partial results [SPG14]

Having surveyed dozens of papers, Bertini and Lalanne conclude that pursuing a mixed-initiative KDD process is one of the most promising directions for future research, and call for more work on realizing tight integrations. Researchers in various communities have echoed this call since, and emphasized the need for interdisciplinary collaboration to achieve this goal. For example, Puolamäki et al. called for *Visually Controllable Data Mining* with goals like (1) providing meaningful visualizations of the model structure; (2) controlling models through visual interaction; and (3) making computations fast enough for visual interaction [PPL10]. While the first two goals are aimed at visualization and interaction designers, the third goal is directed at the designers of algorithmic infrastructures.

The authors of the VisMaster book take a similar line and call for algorithms that better suit the needs of interactive exploration [KKEM10, p. 97f]. Here, major goals are (1) receiving fast initial responses from algorithms with progressive refinement, (2) providing means for triggering recomputations after small changes, and (3) allowing analysts to steer the computation. Fekete, however, points out that these goals are hard to realize with existing infrastructures, as analytical environments were not designed for exploration, and algorithm designers often make no effort to provide early communication with users [Fek13]. As a result, many visualization developers have re-implemented algorithms as part of the visualization tool to enable early user involvement. Typical examples include user-involving variants of expensive algorithms like clustering or dimension reduction (see Section 2.3). The obvious disadvantages of re-implementation include a suboptimal use of resources, and lots of proprietary code instead of standardized and tested solutions.

In 2014, the thesis contribution on Opening the Black Box (Paper A) was among the first

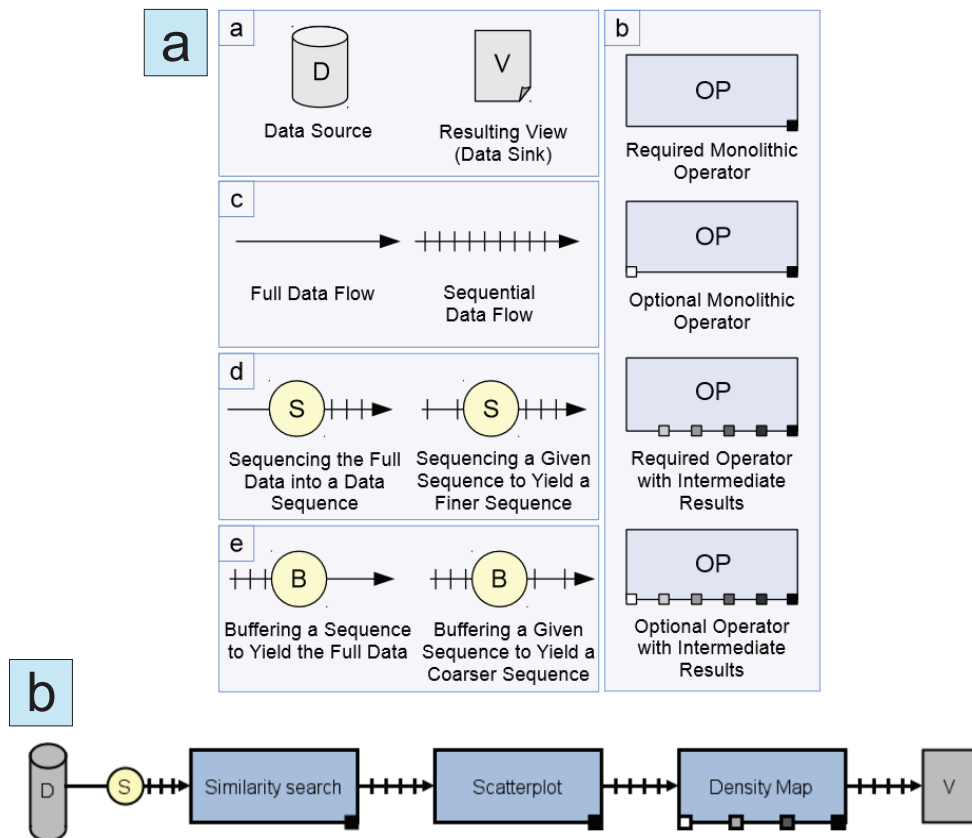


Figure 2.7: (a) Schulz et al. propose a graphical notation to model PVA pipelines as illustrated in (b) [SASS16].

works to investigate how integration can be realized for algorithms that were not designed for exploration. The paper also takes the requirements analysis further, and provides a detailed discussion of how different types of early information exchange support user involvement. Strategies for realizing them in algorithms with different characteristics are presented, and guidelines for algorithm design in favor of user involvement are established. The paper thus provides the needed interdisciplinary perspective on integration that has been stressed in the literature many times [SSZ⁺17, KKEM10]. At the same conference, Stolper et al. presented a conceptual paradigm for realizing early user involvement called “Progressive Visual Analytics” (see Fig. 2.6) [SPG14]. The authors identify very similar goals regarding intermediate feedback and control as Paper A. Moreover, they discuss implications of progressive computation for visualization design, and derive guidelines such as avoiding excessive view changes whenever new results arrive. Building on these papers, Schulz et al. unified and extended the proposed concepts in an “Enhanced Visualization Process Model for Incremental Visualization” [SASS16]. The authors propose a graphical notation for building analytic pipelines using blocks of progressive analytics

and incremental visualizations (see Fig. 2.7). Moreover, they provide guidelines for implementing systems based on their process model using a multi-threading architecture.

More recently, Turkay et al. presented a set of design considerations for visualization and interaction techniques in progressive analytics scenarios [TKBH17]. Specifically, the authors discuss (1) how to design analytics that respect the temporal capabilities of humans; (2) how to integrate computations that can be evaluated progressively; (3) how to design user interaction with progressive processes; and (4) how to inform users about aspects of progressiveness. From these discussions, the authors derive ten practical design recommendations to facilitate the implementation of Progressive Visual Analytics systems. As one example, the authors suggest letting users pause, resume, and navigate through progressions instead of constantly updating views as new results arrive.

In summary, tight integrations of algorithms and visualization have come a long way in the past decade, and research interest in the topic is still growing. Still, there are numerous open challenges that need further research. Conveying the uncertainty of incomplete results, for example, is a multi-faceted topic that has only been partially solved. While approaches exist for certain types of results, such as confidence bounds for aggregates [FPDm12], or low-resolution previews for 2D displays [TKBH17], more work on generalizable approaches is needed. As a related aspect, the instability of progressive results, such as an interesting result suddenly vanishing after an iteration, may need dedicated treatment to avoid the confusion of users [TKBH17].

2.2.3 Human-Oriented Interface Design

A crucial aspect of designing human-oriented modeling tools are user interfaces that account for the strengths and weaknesses of the human user. This section summarizes existing efforts on four aspects of interface design, namely (1) how to visually communicate aspects of models; (2) how to provide means for interacting with models; (3) how to guide users through the modeling process, and (4) human factors regarding machine interfaces such as familiarity, trust, and cognitive biases.

Visualizing Aspects of Models

Visualization has been shown to be an effective medium for deepening the human understanding of statistical models [SSZ⁺17, WCH15]. Answering questions like “how does the model look like”, or “how well does the model fit the data”, is essential to obtaining a flawless view of the modeled phenomena, and to building trust in the models [WCH15]. Moreover, the ability to effectively criticise and improve models relies on an understanding of how good a model already is, and under which circumstances it fails. This understanding can be established by visualizing *quantitative* and *qualitative* aspects of the models.

On the quantitative side, the most commonly inspected type of information is summary statistics of the model, also known as key performance indicators (KPIs). Typical examples include error metrics for predictors such as RMSE, MAPE, or R^2 [HTF09],

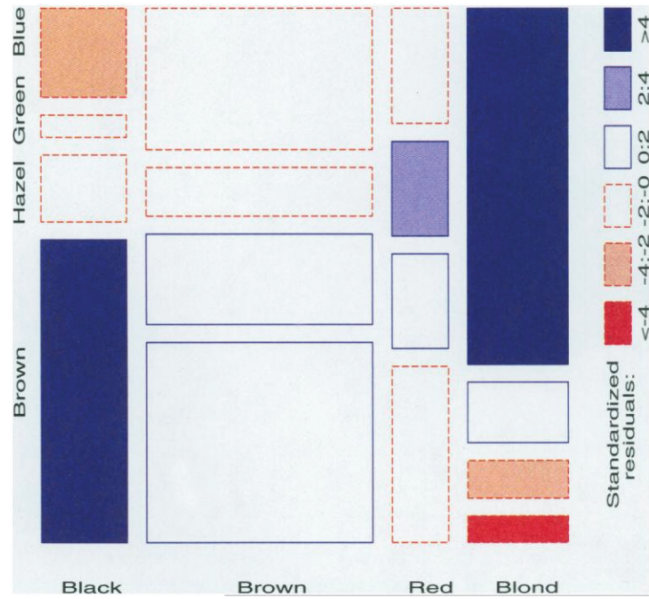


Figure 2.8: Mosaic displays show aggregated residuals for data categories to reveal systematic errors in the fit [Fri99].

or performance metrics for unsupervised techniques such as cluster purity or rand index [Ran71]. As metrics provide a compact summary, they are a suitable abstraction for comparing large numbers of models, e.g., in a ranked list [ZWRH14] or a scatter plot [PSMD14]. However, computing a global score is often insufficient, as it does not account for local differences in the data.

A common approach to account for local phenomena is breaking down KPIs for different data subsets in one display, e.g., categories or partitioned continuous features. Such views indicate circumstances where the model fits the data better than others. Mosaic displays, for example, show the prediction bias of regression models for different data categories in a heatmap (see Figure 2.8) [Fri99]. Similarly, confusion matrices for classification models show which classes are predicted correctly, and which classes are likely to be “confused” with others [HTF09]. Showing these local variations of error may provide clues for model refinement, e.g., to add features that exhibit systematic error patterns to the model [MP13], or to emphasize difficult classes in the learning process [KLTH10].

Model error metrics summarize aspects of the *model behaviour* rather than its form or definition. Thus, they are largely independent of the model type or algorithm used, and may for example be used to compare decision trees to neural networks or SVMs [HTF09]. However, scalable and general as they may be, metrics may hide ambiguities and diversity in the data, and do not provide an intuition of how the model looks like.

Visualizing *qualitative* aspects of models, on the other hand, aims at providing a deeper understanding of the model itself. Here, approaches can roughly be grouped into two

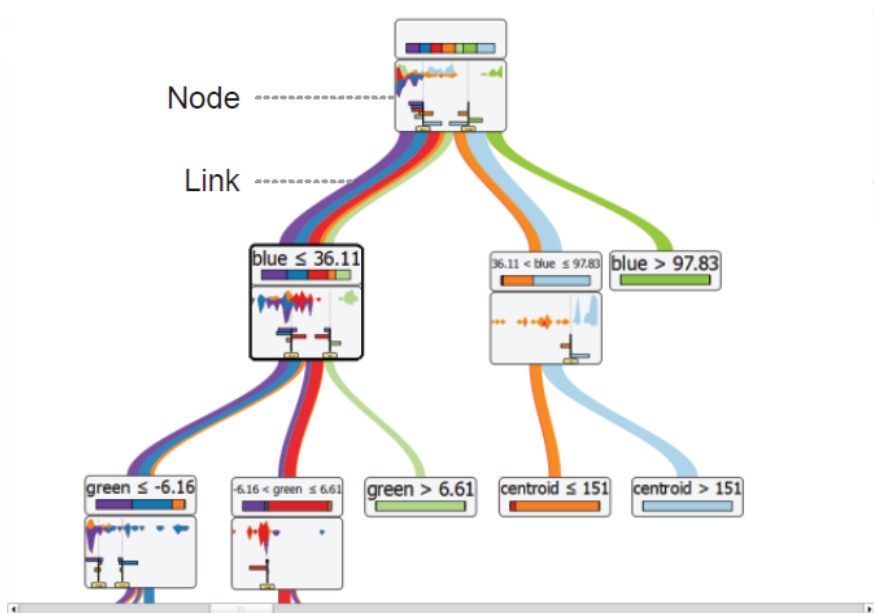


Figure 2.9: Showing the structure of a decision tree reveals the importance of features in separating the data. [vdEvW11].

categories: (1) showing the model definition and structure as such; and (2) showing the model in the context of data.

Looking at the definition of a model may provide important information about relationships in the data. The coefficients of a linear regression model, for example, provide a quantitative notion of sensitivity and feature importance for the target variable [HTF09]. The structure of a decision tree, on the other hand, qualitatively conveys feature importance, as features near the root are selected for the most important splits (see Figure 2.9) [vdEvW11]. Aside from feature importance, showing the model structure may also be helpful to deepen the understanding of the underlying machine learning process. Wonsuphasawat et al. recently proposed a scalable graph-based visualization of deep neural networks [WSW⁺18]. Here, visualization helps to debug the complex structure, and also plays a crucial role for building trust in such models.

The second group of qualitative model visualizations refers to showing the model in the context of data. Visualizing these aspects together can help interpret the typically less familiar model in context of the usually much better understood data space. Here, Wickham distinguishes between two approaches: (1) showing data in the model space “*d-in-ms*”, and (2) showing the model in data space “*m-in-ds*” [WCH15].

Approaches of the first group, “*d-in-ms*” project the data into a subspace generated by the model. A commonly used example is plotting fitted values versus the residuals of a model in a scatter plot to detect non-linearity or outliers [WCH15, HTF09]. Other examples include projecting data points onto principal components (PCA) [Pea01], or visualizing how data points are separated by the nodes of a decision tree [AEK00]. On the

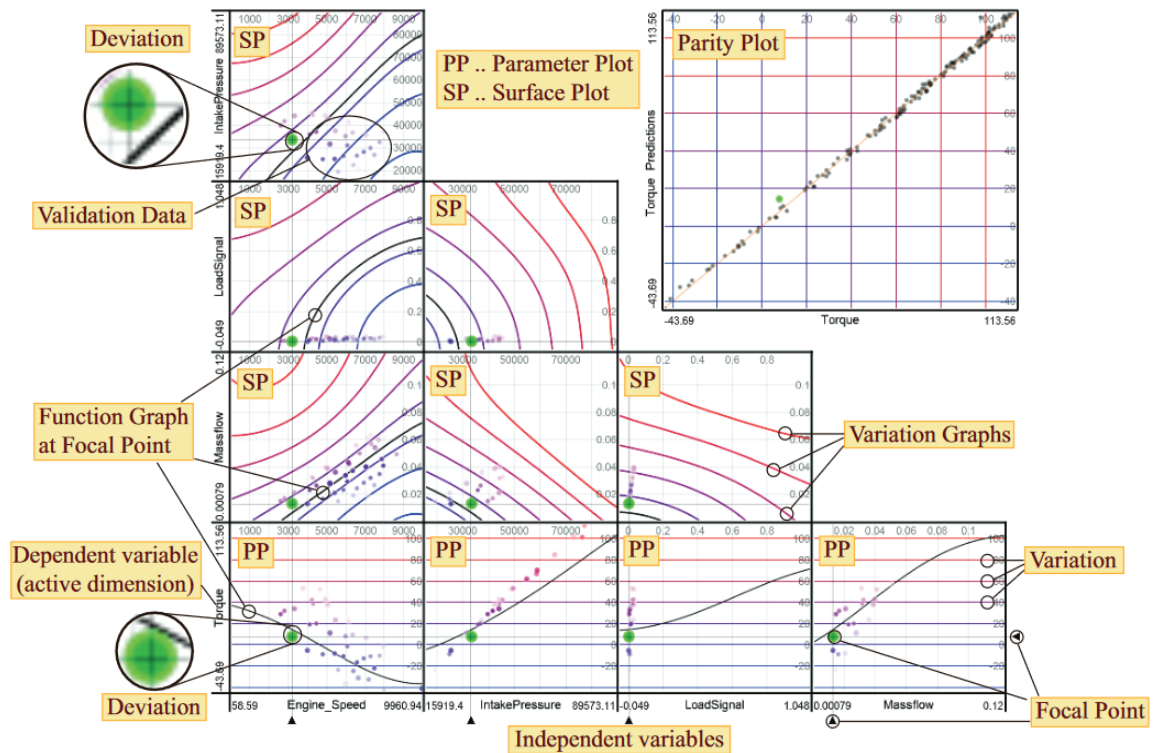


Figure 2.10: HyperMoVal shows the graph of a four-dimensional regression model and nearby data, by slicing the graph around a focal point [PBK10].

level of single observations, “d-in-ms” approaches may be used to explain *why* predictions were made for particular data points. Examples include showing the activation patterns of Bayesian networks [CCH01] and neural networks [TM05] for selected data items, to reveal the responsible features and decisions.

The second group, “m-in-ds” refers to visualizing a representation of the model itself in the data space. For low-dimensional regression models, this is straightforward, as the predicted surface can be shown as a graph in data space. The R packages `visreg` and `ggobi`, for example, plot one- or two-dimensional regression surfaces along with the data points [BB13, SLBC03]. “HyperMoVal” by Piringer et al. extends this idea to higher-dimensional models by slicing the data space, and plotting data points that are near to the surface in the respective slice (see Figure 2.10) [PBK10]. For classification models, Wickham proposes showing a sampled approximation of multi-dimensional classification boundaries in the data space (Figure 2.11a) [WCH15]. While this works well for low numbers of dimensions, he admits that finding informative views in higher-dimensional embeddings may take some time. In the same paper, Wickham also suggests “m-in-ds” approaches for unsupervised techniques: For agglomerative clustering, he proposes connecting data points in the sequence they have been clustered, to convey similarity in 2D views of higher-dimensional spaces. Similarly, he suggests plotting the representative

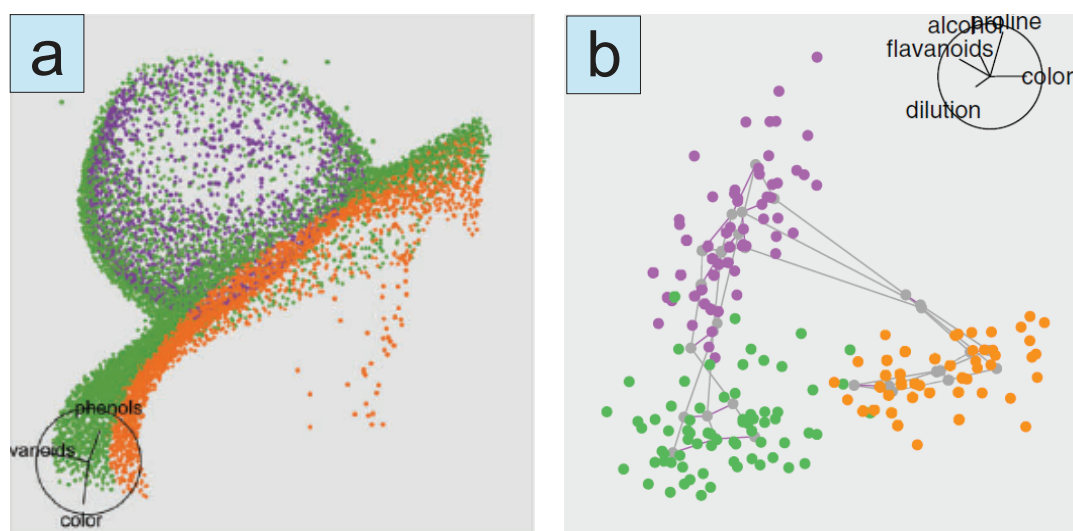


Figure 2.11: Plotting a point-wise approximation of high-dimensional decision boundaries (a) and cluster centers of a Self-Organizing Map (b) in 2D projections conveys the local quality of the models [WCH15].

nodes (cluster centers) of self-organizing maps in the data space, and connect them by lines (see Figure 2.11b). Overlaid with the data points, the resulting net-like structure reveals how well the cluster centers fit the data. In general, Wickham and other authors point out that it requires creativity, and may not always be possible to find suitable visualizations of models in data space - especially as the dimensionality increases [WCH15, ACH15]. However, in cases where it is possible, it is a very direct way of conveying the local quality of the fit.

In addition to all these efforts for visualizing a single model, the literature has emphasized several orthogonal aspects of model visualization. On the one hand, multiple works stress the importance of building and inspecting more than one candidate model [WCH15, PSMD14, SJS⁺17]. Considering diverse alternatives minimizes the risk of getting stuck in local optima and fosters confidence in the models. Dedicated views for comparing two or more models have been proposed [PBK10, KPB14], as well as techniques for exploring entire model spaces based on abstractions of the models [MLMP18, PSMD14, SJS⁺17]. On the other hand, there have been multiple arguments for visualizing the modeling process itself, and not just its end result [WCH15, MPG⁺14, SPG14]. This includes visualizing the progress of learning algorithms to reveal pitfalls early [ZF14, TKBH17], and, in a wider sense, visualizing the analytic provenance during the modeling process to facilitate efficient workflows [GZ09, BCD⁺09, MPR18]. These topics will be illustrated with examples, and discussed in more detail as part of the following sections.

Interaction with Models

Just as important as visualizing aspects of models is the ability to act upon the obtained insights. This section reviews efforts to support an effective communication *from* users *to* models and algorithms.

A historical, yet still very commonly used interface to statistical modeling are command line interfaces that operate by the exchange of text. Data scientists with the respective experience can be highly efficient with such interfaces, and appreciate benefits like its inherent exactness and high reproducibility [Wic18]. Unfamiliar users, however, find significant hurdles in memorizing all the commands and options, and are challenged by the high cognitive load of indirect manipulation [Com17].

The visualization community has promoted the idea of interacting more directly with the visualizations [Shn83, KKEM10, EFN11]. To better understand the requirements of visual interaction, several efforts have been made to taxonomize low-level interactions with visual representations: Shneiderman, for example, proposed the categories *Overview*, *Zoom*, *Filter*, *Details-on-Demand*, *Relate*, *History* and *Extract* [Shn96]. Several authors have proposed similar taxonomies at slightly different granularities and scopes, including Dix and Ellis [DE98], Keim [Kei02], and Wilkinson [Wil06]. Particularly interesting from the human-oriented perspective is the taxonomy by Yi et al., who formulate their categories in terms of user objectives [YKSJ07], for example: “*Mark something as interesting (Select)*”, “*Show me less/more detail (Abstract/Elaborate)*” or “*Show me related items (Connect)*”. While these taxonomies were created from the wider scope of data exploration, many of them can be directly transferred to interaction with statistical models, such as “Show me a less/more complex variant of the model”, or “Show me similar models”.

The field of Human-Computer-Interaction (HCI) describes user interaction as a process with two distinct phases (“Stages of Action”) [Nor13]: *Execution*, and *Evaluation*. Execution refers to performing an action to reach a goal, while Evaluation refers to observing results and evaluating the fulfillment of the goal. Norman also describes that users may not know how to perform an action to reach a goal, or how to evaluate its fulfillment. He calls these gaps (or “gulfs”) between the human goal and the actual state the *Gulf of Execution*, and *Gulf of Evaluation*. It is the goal of human-oriented interface design to avoid or to bridge these gulfs, and to guide users towards and during their interaction.

Many approaches have been proposed in the HCI and visualization communities to implement human-friendly interaction. There is a consensus that *direct manipulation* of graphical user interfaces (GUI) usually enables a more intuitive dialog than command line interfaces. Originally proposed by Shneiderman in 1983, the idea is to allow users to change the state of an object via a graphical handle directly in-place or via an intuitive metaphor [Shn83]. Contemporary examples include sliders for changing quantities, drag-and-drop interfaces for rearranging items, or scrollbars for changing a viewed fraction of a whole. While the use of such controls is easy to master, the challenge for interface designers is finding an appropriate abstraction of the object state to expose. For example,

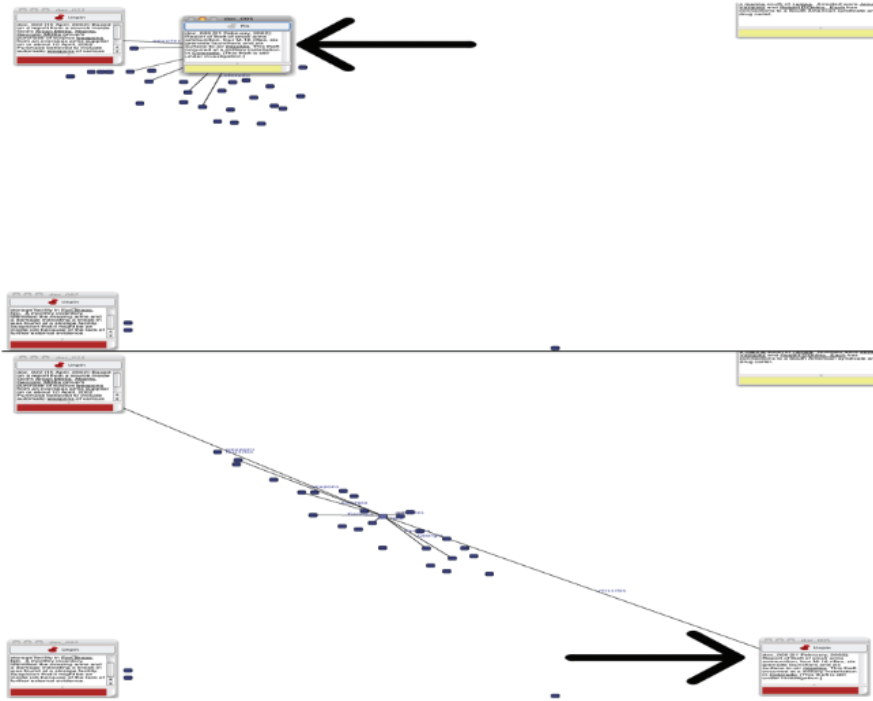


Figure 2.12: Rearranging data items by drag and drop is a common metaphor for guiding similarity-based projection algorithms. [EFN11].

simply exposing the parameters of a statistical model via sliders instead of text does not free unfamiliar users from trial-and-error.

To amend this, Endert et al. propose *semantic interactions*, i.e., simple, meaningful metaphors that trigger complex model changes under the hood [EFN11]. Their semantic interaction pipeline binds model steering techniques to interactive affordances provided by the model visualization itself. In their example implementation, ForceSpire, users can move around document icons in a force-directed layout to express their desired similarity in the model (see Figure 2.12). Under the hood, this simple interaction triggers formal updates of the underlying term weight model or adds constraints to the layout. Similar dragging metaphors for changing distance functions under the hood is provided by the works of Brown et al. [BLBC12] and Endert et al. [EHM⁺11] (see Section 2.3.3).

More generally, the metaphor of specifying a desired target state via direct manipulation is found in several other human-oriented systems: Kapoor et al. allow users to express their dissatisfaction with misclassifications in a confusion matrix to update the underlying weightings (see Figure 2.13a) [KLTH10]. Expressing a like or dislike of items to change models is also the key idea of recommender systems found in entertainment and sales [JWK14, MS11]. Similarly, the Crayons system described by Fails and Olsen lets users modify image classifiers by “painting” over important regions to emphasize them in

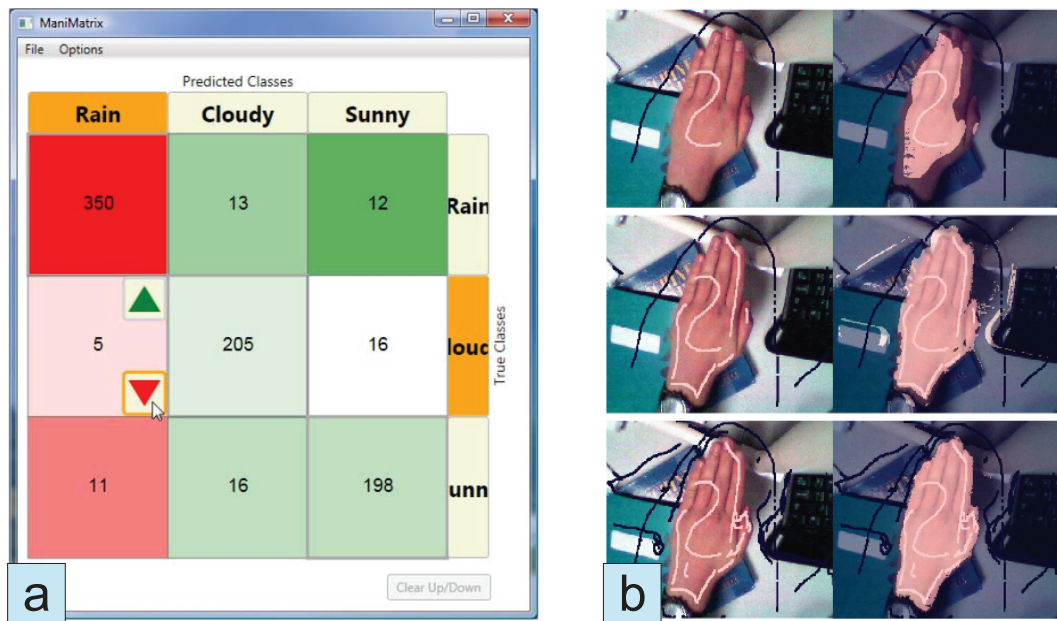


Figure 2.13: (a) ManiMatrix allows users to express dissatisfaction with misclassifications in a confusion matrix to update an underlying classifier [KLTH10]. (b) In the Crayons system, painting over important regions of images allows emphasizing them for classification [FOJ03].

the classification (see Figure 2.13b) [FOJ03]. Designing such intuitive interfaces is often highly model-specific, and thus associated with high implementation effort. This may be one of the reasons why semantic interaction is still relatively rare compared to more conventional approaches to model interaction.

The previously described approaches mostly focus on changing the status quo of *one* particular model according to some goals. A different approach to navigating the model space is the *proactive* creation of alternatives for the user to explore. By using models as key entities of the data model, “conventional” techniques for visualization and interaction can be applied, such as linking and brushing [Hea99] or multiple coordinated views [Rob07, WBWK00]. Schneider et al., for example, represent classifiers by metrics in a scatter plot to support model selection and the building of classifier ensembles [SJS⁺17]. By linking the scatter plot with views of the data space, selection of a model immediately highlights misclassified points, while selecting data points shows the metrics only for the selection. Similarly, Padua et al. support model selection from a set of decision trees generated by a full-factorial sampling of algorithm parameters [PSMD14]. Here, a secondary goal is to provide a deeper understanding of the effect of each algorithm parameter on the model space (see Figure 2.14). Gleicher employs a similar methodology to craft projections of data items that are meaningful to users, so-called “Explainers” [Gle13]. Seeking to create diverse projection functions that represent various trade-

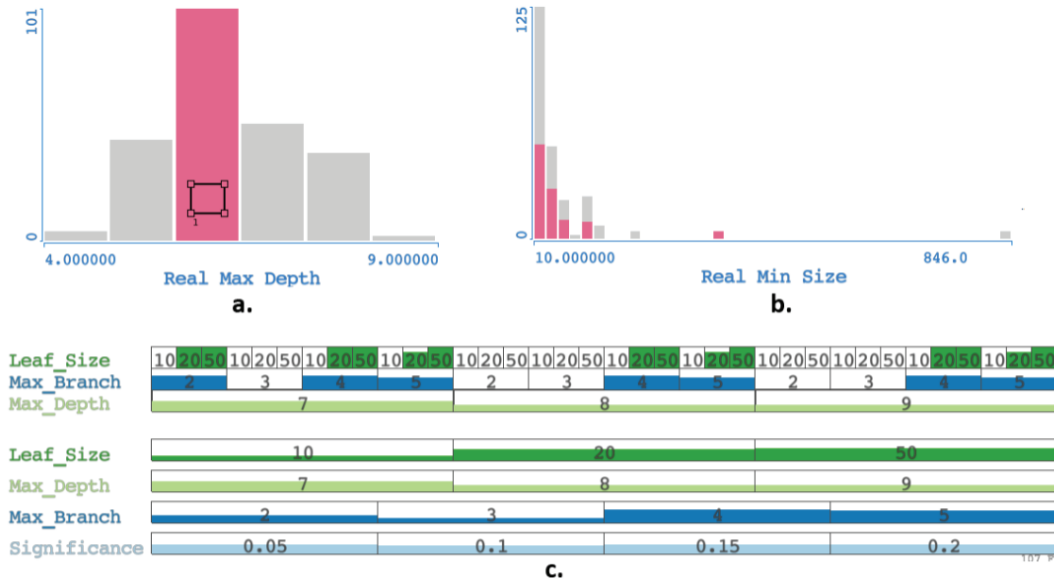


Figure 2.14: Linking and brushing of model candidates in an ensemble conveys the effect of algorithmic creation parameters on decision trees [PSMD14].

offs to choose from, Gleicher proposes varying the parameters of creation procedures. Visualizations of the resulting models then support users to perform model selection based on qualitative inspection. In general, model space exploration techniques free users from specifying their goals in advance to a certain extent. However, most existing solutions do not provide sufficient guidance along trade-offs, and offer limited interactions to refine the model space in interesting regions in an intuitive manner. These shortcomings were two key motivations for the thesis contribution TreePOD (Paper B), which guides model selection under conflicting objectives by focusing on Pareto-optimal candidates [MLMP18]. Moreover, simple actions such as “*show me more like this*”, or “*what would happen if...*” allow users to create variations of interesting candidates with a single click.

Regardless of how interaction with the model space is realized, users often need to try out different scenarios and variations before they reach their goal. Multiple sources thus stress the importance of *history* features to navigate back and forth in the action sequence [NCE⁺11, MP13]. Approaches may reach from simple undo/redo support [ZBB⁺13], timelines of interaction [SIH00], to sophisticated graphical history interfaces that allow for branching, editing, and searching [HMSA08].

In summary, many techniques for interaction have been proposed in the HCI, visualization, and machine learning communities. Designing interfaces that effectively support a broad set of users is a challenging task. Simple interactions that subsume complex model changes under the hood may reduce the hurdles for inexperienced modelers. However, experienced users with a deep understanding of algorithmic details should not be prevented from

incorporating their knowledge. In TreePOD (Paper B), the approach is to offer both: simple interactions on the surface, while algorithmic details are exposed on demand. While this might be a general guideline to enable both groups of users, interfaces should always be designed and evaluated in close collaboration with actual users when possible. Employing a design study methodology, for example, may increase the probability that new metaphors and techniques will be adopted by the intended audiences [SMM12].

Guiding the Modeling Process

While the previous section described how users can guide the machine via interaction, this section focuses on the machine *guiding the user* throughout the modeling process. In other words, it reviews efforts on assisting the user in performing steps to accomplish a modeling task at hand.

Guidance may be realized in many forms. To enable an effective discussion of the subject, Schulz et al. characterize guidance along four dimensions [SSMT13]:

1. *Guidance context*: how much prior knowledge are users expected to have.
2. *Guidance domain*: which entity of interest are users guided towards, e.g., interesting data subsets, optimal models, effective views, etc.
3. *Guidance target*: how users are taken to their goal, which can either be *direct* (“Take me to X”) in case users have an idea what they want, *indirect* (“Take me to Y that are like X”) similar to query-by-example [Zlo77], or *inverse indirect* (“Take me to Z that deviate from X”), to discover new paths.
4. *Guidance degree*: to which extent a suggested path is constrained, and how much freedom is given to deviate from suggestions.

As one example of this scheme, TreePOD (Paper B) provides *direct* guidance by suggesting concrete variations of a selected model, as well as *indirect* guidance towards similar models by offering a button “Show me more like this”. As *guidance context*, prior knowledge is expected to the extent of understanding what the suggested actions mean (e.g., “round all numbers to integers”). The *guidance domain* are Pareto-optimal decision trees. The *guidance degree* is constrained when applying one of the predefined variations, but quite unrestricted when creating a new bunch of similar candidates to choose from.

More recently, Ceneda et al. developed the conceptual model of Schulz et al. further [CGM⁺17]. They summarized *guidance context* and *domain* under the term “knowledge gap”, and introduced a distinction between the *input* and the *output* of the guidance generation process (see Figure 2.15). Inputs refer to elements informing the guidance mechanism, such as data, knowledge, or the analytic history. Outputs are the resulting elements to guide the user, such as visual cues or dedicated views (see below).

Model building and selection involve several non-trivial *choices* that can benefit from guidance. Common examples are the choice of algorithmic parameters and model

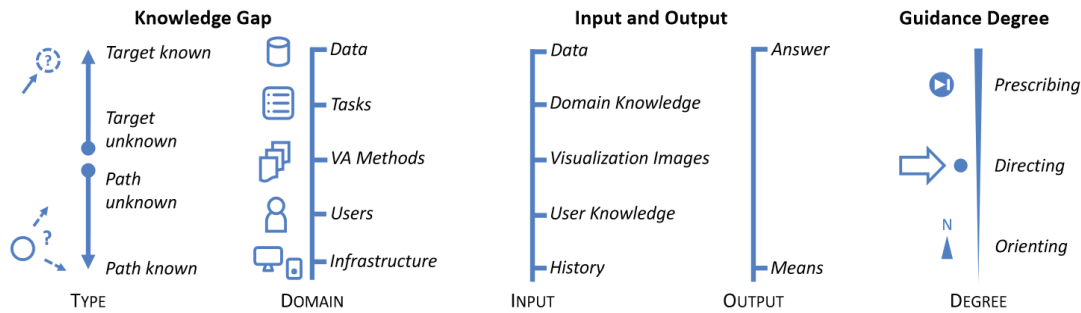


Figure 2.15: Guidance can be characterized with respect to the knowledge gap of users in achieving a particular goal, the input and output of a guidance mechanism, and the degree to which users are constrained by this mechanism [CGM⁺17].

types [SHB⁺14, PSMD14, MLMP18], and the selection of data records and features as model inputs [MBD⁺11, vdEvW11, MP13, KPB14]. A common approach to guiding such choices is to precompute several different possibilities, and conveying their implications to the user. Model refinement, for example, may be guided by trying out model extensions based on different features under the hood, and showing achievable accuracy gains to the user [MP13, vdEvW11]. As a more constrained example of guidance, model space exploration techniques precompute an entire set of ready-made models to choose from, while users skip the building process altogether [PSMD14, MLMP18].

The previous examples guide model identification workflows by using *data* as the primary input for guidance. Another approach is to extract information from the user’s interaction patterns. Mouse and keyboard events, as well as higher-level interactions may implicitly hint at the user’s intentions [CGM⁺17]. Gotz and Wen, for example, demonstrate how patterns of ongoing user interactions can be matched to previous ones to identify and guide the task at hand [GW09]. While such implicit methods are unobtrusive and applicable to any workflow, they are also prone to errors by misinterpreting the activities of the user [OK98].

Several approaches have been proposed to represent the *output* of a guidance generation process visually. One possibility is explicitly showing the results of precomputation-based guidance in a dedicated view. Van den Elzen and van Wijk, for example, show possible results of refining decision trees by additional features in a ranked list (see Figure 2.16) [vdEvW11]. The Partition-Based Framework (Paper C) provides similar list and matrix views to guide the refinement of regression models [MP13]. Having a dedicated view for suggestions allows tool designers to show many details, and is a reasonable choice if the guidance mechanism is a central part of the workflow.

A more light-weight alternative is to provide *visual cues* as part of another visualization, e.g., as overlay. This consumes less space and potentially reduces the cognitive load of switching focus between views. TreePOD (Paper B), for example, shows the Pareto front as an overlay in a scatter plot representing decision trees by two objectives [MLMP18].



Figure 2.16: A ranked list of possible features for model refinement guides the building of decision trees [vdEvW11].

May et al. propose the use of signpost glyphs as a focus-and-context technique for exploring large graphs (see Figure 2.17a) [MSDK12]. While only a part of the graph fits the screen, signposts point towards important nodes that are currently not displayed. Luboschik et al. guide the top-down exploration of multi-scale data by visual cues of where to expect interesting details in finer levels [LMS⁺12]. This is achieved by downsampling the finer levels and comparing the result to the coarser level data, to identify features that will emerge when drilling down. Finally, Willett et al. enhance user interface widgets by visual cues to enable a more informed usage [WHA07]. Examples include enriching selection widgets with visualizations of other users' choices, or augmenting a slider with a histogram of the data (see Figure 2.17b).

In summary, guidance is a multi-faceted topic with high relevance for creating user-friendly modeling solutions. Without proper assistance, users may easily be overwhelmed by the broad range of options for model types, parameters, and data subsets. In the evaluation of Papers B+C, the proposed guidance mechanisms were among the most appreciated aspects for the collaborators. Reaching good models faster, with the confidence of having seen many options and alternatives, added the most value over previous approaches, according to the experts.

Human Factors

The last block of related work on interface design focuses on characteristics of the human user. Humans have unique strengths and weaknesses that the design of human-oriented solutions must take into account. Moreover, humans bring requirements to the table that go beyond typical goals of statistical modeling such as maximizing accuracy.

One such aspect is the need of humans to *trust* the models and the insights that result from the modeling process. At the end of every analysis, it is a human who needs to act with confidence upon the potential findings. Even in automation scenarios, it is humans who have to stand up for mistakes that a model may have made. On the one hand, this can be seen as an argument for integrating and re-using proven, thoroughly tested algorithms, as advocated by this thesis. On the other hand, it is an argument for visualization: In contrast to black-box modeling, one can actually *see* and *understand* the mechanics of the models, which can generate a great amount of trust [ERT⁺17].

However, the use of visualization alone is by no means a guarantee for trust. This is especially true if users are confronted with novel, unfamiliar techniques. Visualizations

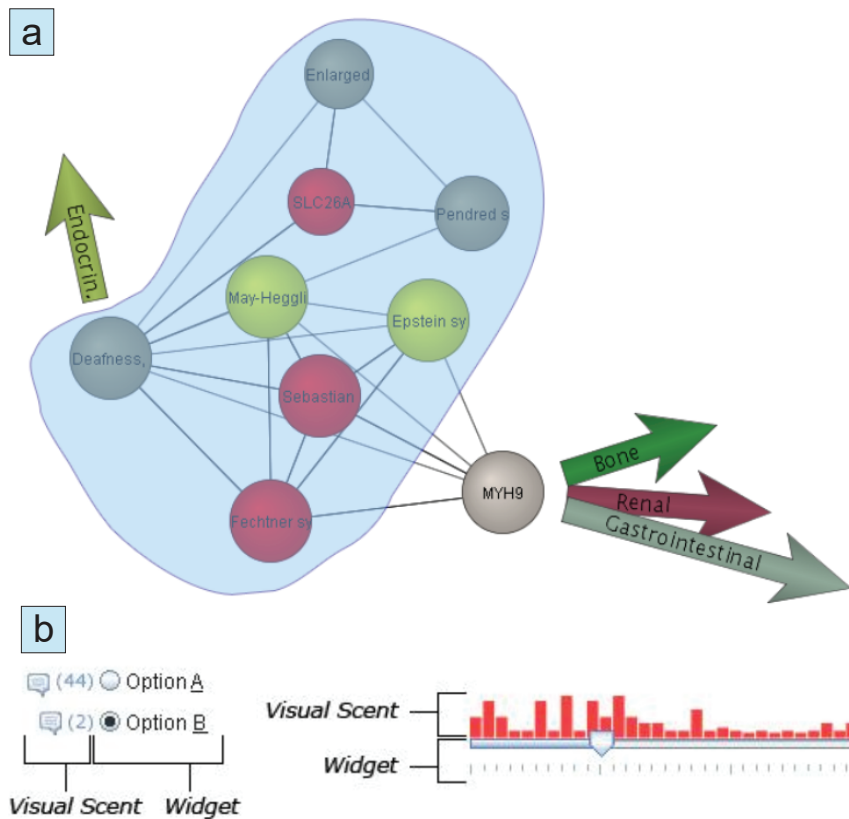


Figure 2.17: (a) Signposts guide users towards interesting nodes in a large network while zoomed in to a small part [MSDK12]. (b) “Scented” user interface widgets enable a more informed usage [WHA07].

need to be understood to an extent that allows for confident interpretation before users can start to trust it. In this respect, Gulati argues, when it comes to human interactions with machines, “familiarity breeds trust” [Gul95]. And yet, even when a new visualization has been understood, domain experts may still remain skeptical and express a preference for known, yet possibly less effective visualizations [SZ88]. Cognitive scientists attribute this skepticism to a human cognitive bias called the *familiarity heuristic* [Ash06], which is associated with the *bias of availability*: People estimate the likelihood of events based on how many examples of such events come to mind [TK73].

Several authors have investigated the implications of familiarity biases for visualization design. Studies by Dasgupta with climate scientists confirmed the tendency of preferring familiar visualizations [Das17]. However, he found that familiarity as such does not necessarily lead to better task performance, while the subjective preference of a visualization does. He also found that participatory design sessions and long-term collaboration can mitigate the effect of familiarity biases. This matches findings of Brehmer et al., who furthermore establish that partial redundancy among visual encodings can facilitate

an understanding of unfamiliar parts [BNTM16]. Takayama et al. also report cases where participative design eventually led to the preference of unfamiliar designs over less effective familiar ones [TK06]. Another study by Dasgupta et al. compared user performance on tasks with a conventional, script-based approach to a Visual Analytics (VA) approach [DLW⁺17]. Despite the users' familiarity with scripting, the authors observed comparable, and even larger levels of trust in the VA approach when it came to complex sensemaking tasks. All these results suggest that familiarity *does* affect experts' preferences, but respective biases can be overcome.

In conclusion, familiarity is a design parameter, which may have a considerable impact on trust. New interactive visualizations can be disruptive, and may need significant time for adoption [Das17]. Participatory design can help, but when designing visualizations for a larger audience, convincing a few experts does not guarantee acceptance on a broader scale. In such practice-oriented cases, it may be a guideline to not deviate more from widely accepted norms than necessary. TreePOD (Paper B) follows this guideline, as well as Brehmer et al.'s suggestion of using partially redundant encodings for anticipated unfamiliar parts [BNTM16]. Both aspects, familiarity and redundancy, received positive feedback from domain experts in the evaluation sessions of TreePOD (see Paper B).

The last aspect discussed in this section are characteristics of the human cognitive system. Benefits and strengths of human cognition are the key motivation for visualization research. However, its weaknesses and limitations also have implications for interface design that human-oriented approaches need to consider.

On the one hand, there are limits to human perception that have been extensively studied in the cognitive sciences, such as color blindness [Gor98] or change blindness [ROC97]. Guidelines to account for such shortcomings have been proposed in the visualization literature. Examples include avoiding color-maps that rely on the distinction of red and green hues [Mun14], or emphasizing potentially overlooked differences [HE12, GAW⁺11]. On the other hand, there are multiple cognitive *biases* that influence how humans behave when interacting with interfaces. Cognitive science describes bias as an error, where the cognitive system unconsciously deviates from seemingly rational behaviour [TK74]. The aforementioned availability bias and the related familiarity heuristic are examples of this phenomenon. Also relevant for analysis is the so-called *confirmation bias*, which describes a tendency to accept confirmatory evidence of an existing hypothesis and to dismiss contrary information [Nic98]. This early fixation may be aggravated by *anchoring bias*, which states that people lean towards a pre-existing goal, or "anchor", once set [TK74]. In general, previous experiences unconsciously shape the approaches of users to the analysis process. Expertise and backgrounds may shape implicit attitudes that may influence how hypotheses and analysis questions are formulated and addressed [WBP⁺17]. As an example, Wall et al. refer to forensic experts who may be more conservative in their judgments than others due to their understanding of the consequences of their decisions.

Interactive interfaces can lead to humans unconsciously feeding these biases back into the analytic process [WBP⁺17]. Information overload [MP77] and confirmation bias, for example, may lead to an early restriction of the information space, while possibly relevant

aspects are overlooked. In model building and selection, this can lead to accepting a local optimum while missing better models based on completely different assumptions. Human-oriented systems should account for such possibilities, and point out biases to the users where possible. As one example in that direction, Nussbaumer et al. suggest using the interaction history to warn users about bias during high-dimensional data exploration [NVH⁺16]. By creating awareness of bias, they argue, users can adapt and proceed their exploration in a less biased way. Law and Basole go one step further, and try to counter selection bias before it arises [LB17]. Their “Breadth-first Exploration” concept actively confronts users with parts of the information space they have not looked at. “Voyager” is an example of a system implementing this strategy for tabular data, constantly augmenting the exploration path with charts of unseen dimensions [WMA⁺16]. Also, TreePOD (Paper B) can be considered an example of this approach, as it provides unbiased alternatives to the selected models throughout the entire workflow.

In summary, humans come with many unique characteristics that human-oriented interface design needs to consider. While basic research on human factors is not the contribution of this thesis, this brief overview completes the discussion on human-oriented design. Moreover, selected aspects like bias avoidance and familiarity were considered as design goals, which was hinted at throughout this section, and is elaborated in the papers.

2.3 Visual Analytics Solutions for Modeling Tasks

While the previous section described previous work on specific *parts* of the modeling process, this section describes *holistic solutions* that successfully combine these parts. It can be seen as a state of the art report on solutions supporting a human-oriented modeling process. The primary goal is to motivate the need for the thesis contributions with respect to previous efforts, including work published after the own papers. As a secondary goal, the section aims to provide an overview of the various approaches as a basis for drawing conclusions and identifying open research directions in Chapter 3.

This section is not the first state of the art report on Visual Analytics approaches to statistical modeling. Here, the focus is on aspects related to the thesis contributions, while other reports assume overlapping, yet slightly different scopes. Moreover, the other reports structure their literature reviews differently. Yafeng Lu et al., for example, recently reviewed the state of the art in Predictive Visual Analytics [LGH⁺17]. Their categorization of papers is structured by the addressed step of the modeling pipeline (pre-processing, modeling, validation,..), as well as the interaction techniques used. Similarly, Junhua Lu et al. reviewed progress and trends in Predictive Visual Analytics structured by pipeline steps and applicable data types [LCM⁺17]. Widening the scope to modeling tasks beyond prediction, Endert et al. reviewed the state of the art in integrating Visual Analytics with machine learning [ERT⁺17]. Their literature review is primarily structured by “user intent”, where they distinguish (1) methods to modify parameters and the computation domain of algorithms, and (2) methods to define analytical expectations about the results. In contrast, the literature review at hand uses the *modeling task* as

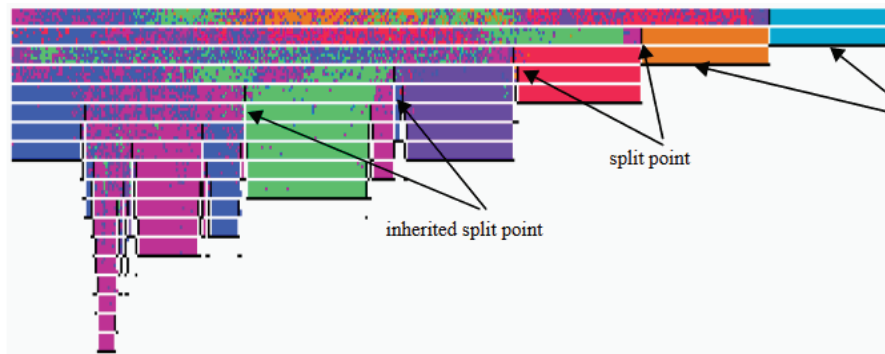


Figure 2.18: Ankerst et al. combine a pixel-based visualization of the the tree structure with algorithmic support for the cooperative building of decision trees [AEK00].

the primary structural element, i.e., *classification*, *regression*, *clustering*, and *dimension reduction*. This enables a direct comparison of the thesis contributions to solutions for the same task, and provides a feeling for how much attention each task has received by the VA community over the past years.

2.3.1 Classification

Classification refers to modeling the relationship between *classes* of a categorical dependent variable and a set of numerical or categorical features [HTF09]. While many different types of classification models have been proposed, some are arguably more suitable for human-oriented applications than others. *Decision trees* for example, are based on interpretable rules like numerical thresholds, which allows humans to understand how the model works. This has made them the subject of some of the most human-centered approaches to classification over the years: Ware, for example, proposed to let users interactively draw decision boundaries as polygons in data space to build decision trees [WFH⁺01]. While manual construction leads to high levels of comprehension, the lack of algorithmic support limits such approaches to classification problems of moderate difficulty. Similarly, Ankerst et al. introduced a manual tree construction approach called “Perception-Based Classification” [AEEK99]. A pixel-based visualization of the training data conveys the distribution of class labels conditioned on the value ranges of features. Users can interactively select features and split points, which updates the visualization to emphasize the remaining variance per feature. While the method fosters comprehension and exploits perceptual capabilities, it also lacks algorithmic support that, for example, could prevent users from over-fitting their data.

To make use of both human and machine capabilities, Ankerst et al. proposed a cooperative approach to decision tree building [AEK00]. Users may still perform manual refinement, while algorithmic support can be triggered at any point to suggest further splits, or train selected branches further. Domain knowledge can be incorporated in the form of constraints for the algorithmic optimization. A pixel-based visualization of the decision tree shows the purity gained by every split, as well as the significance of the

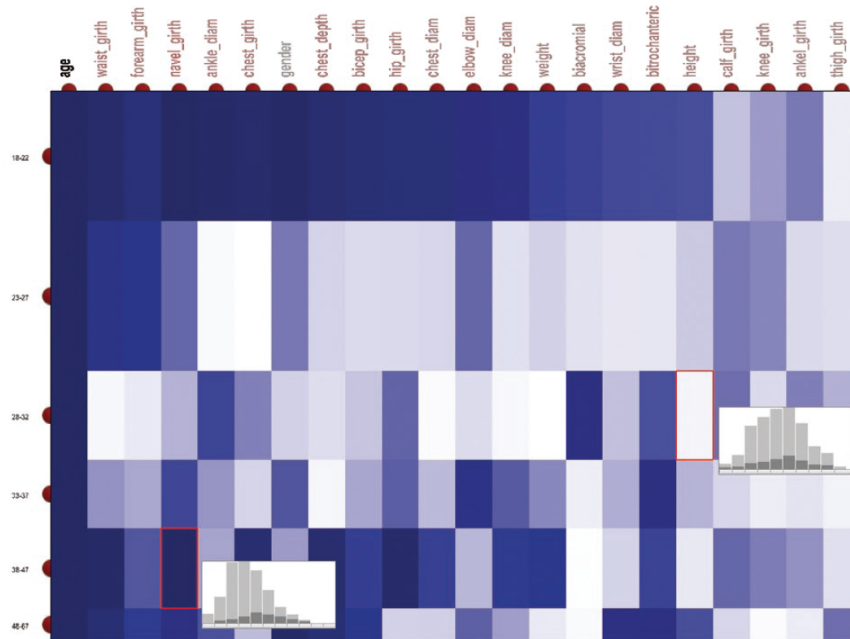


Figure 2.20: Visualizing mutual information of partitioned features with a target variable reveals local dependencies for feature selection [MBD⁺11].

algorithms [KPB14]. Their approach, called “INFUSE”, represents features as glyphs that show how important each feature is for a number of classifiers and cross-validation runs (see Figure 2.21a). Users can interactively select feature subsets, build classifiers, and compare the results to automatic selection algorithms. In general, such interactive “white box” approaches to feature selection allow for an incorporation of domain knowledge and have the potential to foster high levels of trust.

However, not all types of data and classification problems are suitable for relying on a human recognition of patterns. Höferlin et al., for example, argue that interactive feature selection is difficult for classifying high-dimensional video data based on complex features in image space [HNN⁺12]. Instead, they propose “Inter-Active Learning”, a technique to involve users through a “query-by-example” approach: As in traditional active learning [Set12], the system asks users to label difficult data instances in order to improve an automatically learned model. Additionally, users can speed up the process by suggesting data subsets to label instead, based on their understanding of the problem domain and the constructed model. Following the classification of Endert et al. [ERT⁺17], “Inter-Active Learning” differs from the aforementioned approaches as it allows users to define *expectations about the results*, rather than the *inputs of an algorithm*.

Several other “black box” approaches to modeling pursue a similar, result-oriented approach to user involvement: “ManiMatrix” by Kapoor et al., for instance, does not expose inner workings, but presents the user with a classifier right away [KLTH10]. Users

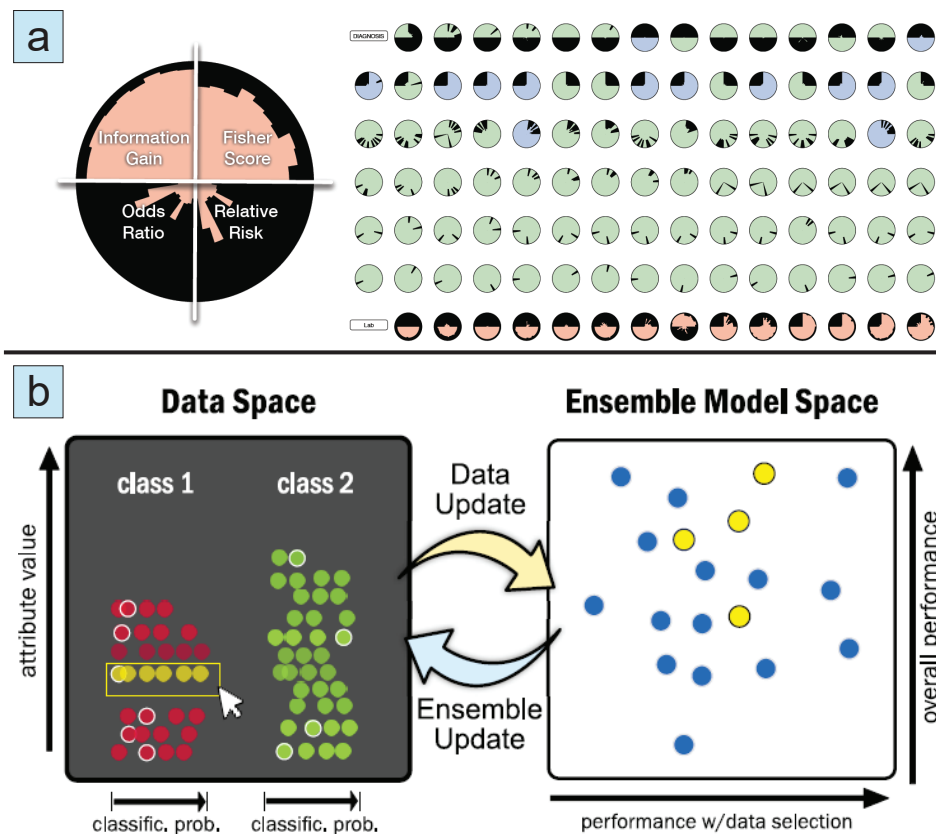


Figure 2.21: (a) Krause et al. use glyphs to encode the results of ranking features by multiple criteria [KPB14]. (b) Linking visualizations of data space and model space supports the creation of accurate ensemble classifiers [SJS⁺17].

may then interact with the confusion matrix to express their dissatisfaction with the misclassifications of particular classes, which updates weights of the classifier under the hood. While this is a nice example of “semantic interaction” as described by Endert et al. [EFN11], tweaking error weights may be time-consuming, especially if their importance is only vaguely known. Instead of tuning a single model, Talbot et al. propose an approach to interactively build an ensemble of classifiers to maximize accuracy [TLKT09]. Similar to “ManiMatrix”, their “EnsembleMatrix” approach lets users interact with confusion matrices, and supports them in finding an optimal combination strategy for the ensemble. Schneider et al. support highly detailed ensemble refinement based on the importance of individual data records [SJS⁺17]. With linked visualizations of model space and data space, users can identify ensemble parts that misclassify important cases, and replace them by models with better performance (see Figure 2.21b). In contrast to these approaches, Padua et al. automatically build an ensemble of decision trees, and let users explore the models with multiple linked views [PSMD14]. Here, the goal of creating multiple models is to support an informed selection from alternatives, instead of

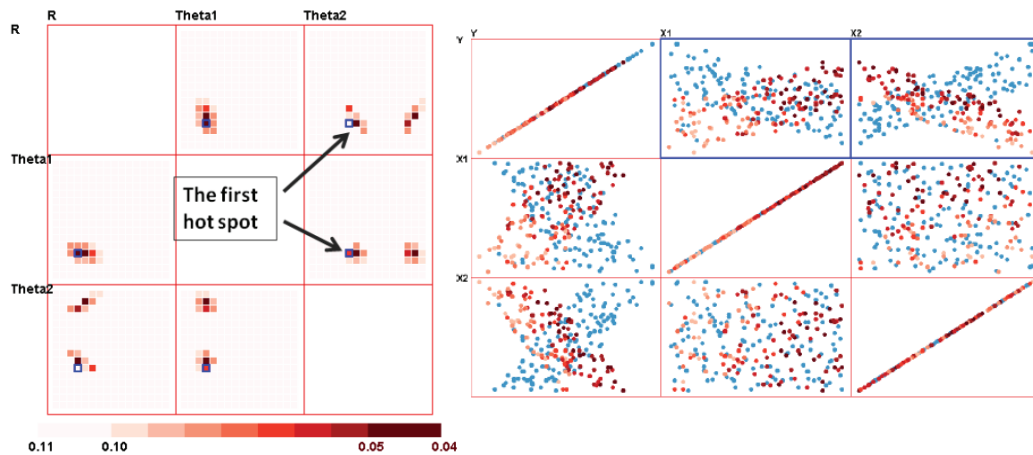


Figure 2.22: Visualizing the parameter space of linear models reveals multiple coexisting trends [GWR09].

combining them into a single, more accurate predictor. The ensemble is systematically built by parameter variation, such that exploring the models may help to understand the parameter spaces of tree construction algorithms.

In summary, a variety of approaches have been proposed to involve users in classification tasks through visualization. Many approaches enable the incorporation of domain knowledge and human strengths in the iterative improvement of a single model. A few others focus on selection and combination tasks based on multiple models, and allow users to provide feedback in terms of desired result characteristics. Building on lessons learned from both approaches, the thesis contribution TreePOD (Paper B) tries to combine benefits of both into a holistic workflow: Based on an automatically created ensemble of diverse tree candidates, guidance along trade-offs in terms of meaningful tree characteristics enables a confident selection even for non-experts in statistics. At any point, users may trigger the computation of new trees, and modify single trees for what-if analyses and a flexible incorporation of domain knowledge.

2.3.2 Regression

Regression refers to the task of predicting a continuous target variable from numerical or categorical features [HTF09]. Compared to the large number of interactive classification approaches, regression has received relatively little attention in Visual Analytics so far. In particular, dedicated support for incorporating domain knowledge in building, validating, and comparing regression models was lacking when this thesis started.

Numerous approaches employ regression and correlation analysis as a means for knowledge discovery. Guo et al., for example, propose a framework for the discovery of multivariate

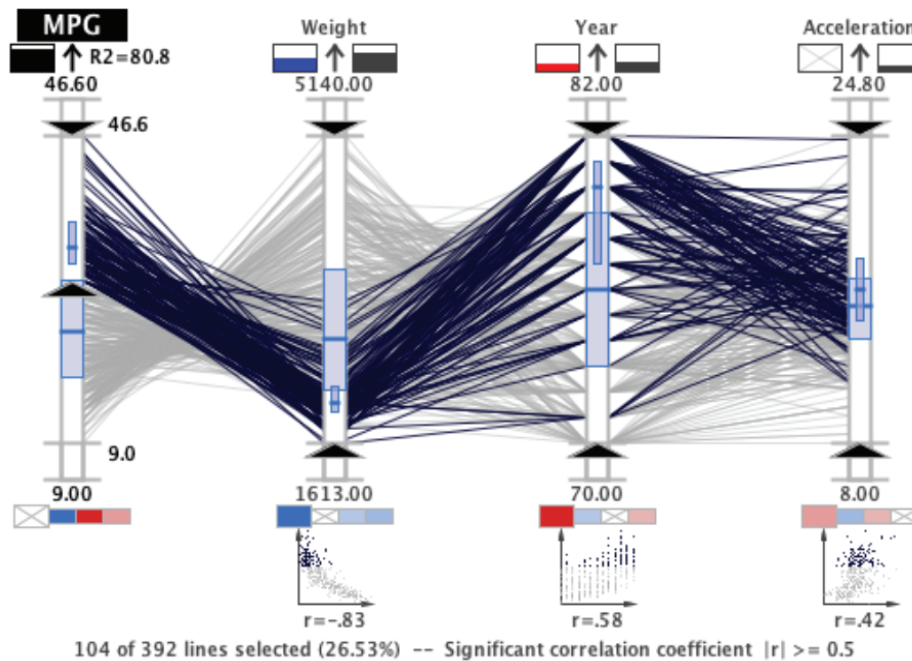


Figure 2.23: Steed et al. use colored rectangle glyphs below the axes of parallel coordinates to convey the correlation with other features. Small thermometer glyphs above the axes show the importance of each feature for a regression model. [SSFJK14].

trends [GWR09]. Interactive visualizations of the model parameter space enable the detection of multiple, coexisting trends in the data (see Figure 2.22). With this system, users are able to interactively select patterns and to extract data subsets that fit a model well. However, the approach is limited to linear models and supports no incorporation of domain knowledge in the model building process. Approaches like “INFUSE” and “Smartstripes” (see previous section) also guide the user towards data subsets that explain a target variable well [KPB14, MBD⁺11]. Originally intended for classification, however, these approaches require a discretization of the target variable, which may incur a problematic loss of detail for regression.

Among the few Visual Analytics approaches to regression model building, Steed et al. propose an interactive system based on augmented parallel coordinates [SSFJK14]. Small glyphs near each axis indicate the correlation with every other axis, guiding the user towards highly correlated features (see Figure 2.23). The system is linked with Matlab to integrate algorithms for stepwise feature selection and multiple linear regression model fitting. The algorithms are integrated as black boxes with limited means of interaction. Users can, however, interactively experiment with the subset of used features, which is guided by a thermometer glyph encoding the relevance of each feature for the target.

A similar black box integration of modeling algorithms with Visual Analytics has been

2. BACKGROUND AND RELATED WORK

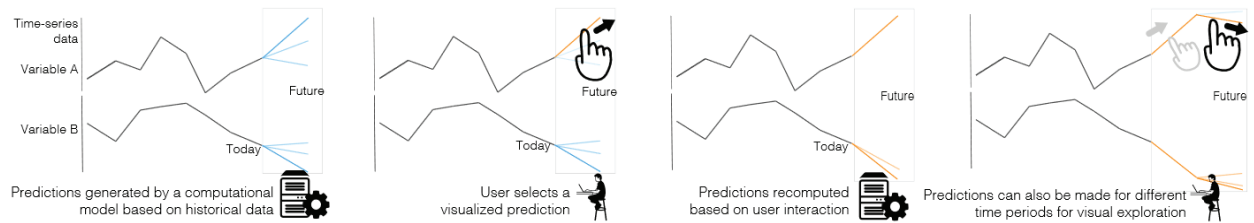


Figure 2.24: “TimeFork” lets users explore different futures predicted by automatically built time series models for decision making [BZS⁺16].

proposed for predicting movie sales from social media data [LKT⁺14]. The presented system ranks features by their relevance for the target, and supports building various model types based on a user-defined feature subset. Models can be compared using different metrics, and validated by inspecting point-wise residuals. While this allows users to experiment with features and model parameters in a feedback loop, the system does not provide dedicated guidance for model refinement and parameter choices.

Several solutions focus on building prediction models for quantitative time series data. Hao et al. propose a Visual Analytics approach to predict large seasonal time series with a focus on preserving local peaks [HJM⁺11]. Their system integrates multiple autoregressive model types that can be built, which may be preceded by a peak-preserving smoothing algorithm. Users may interact with sliders to try out parameters for the smoothing and the models, and view the results in multiple linked views. Bögl et al. go one step further and provide dedicated visual guidance for the refinement and selection of autoregressive ARIMA models [BAF⁺13]. Based on various model diagnostic plots, users can interactively tune models provided by an integration with the statistical environment *R*. The visual interface follows the workflow of the Box-Jenkins methodology for model selection, which makes the method highly effective for statistical experts familiar with this approach. In contrast, the time series prediction system “TimeFork” by Badam et al. is geared towards decision makers without deep statistical backgrounds [BZS⁺16]. Their idea is to show possible future developments for multiple time series by slightly varying the input values of a predictor. Users can focus on a particular path by interaction, and investigate how this development would affect the predictions of the other time series (see Figure 2.24). In contrast to Hao and Bögl’s approaches, however, modeling is not the primary objective but rather a means to the end of decision support. As such, users are not deeply involved in the modeling process itself, but interact with its results.

Other works focus on the *validation* of regression models. Based on “HyperSlice” [vWvL93], “HyperMoVal” by Piringer et al. shows slices of a multidimensional regression model together with nearby data points [PBK10] (see Figure 2.25). Users can interactively adjust the slice position to identify regions with a bad fit, and compare multiple models by the use of different line stipple patterns. Linked multivariate views deepen the understanding of training and validation data for model refinement, and allow users to focus

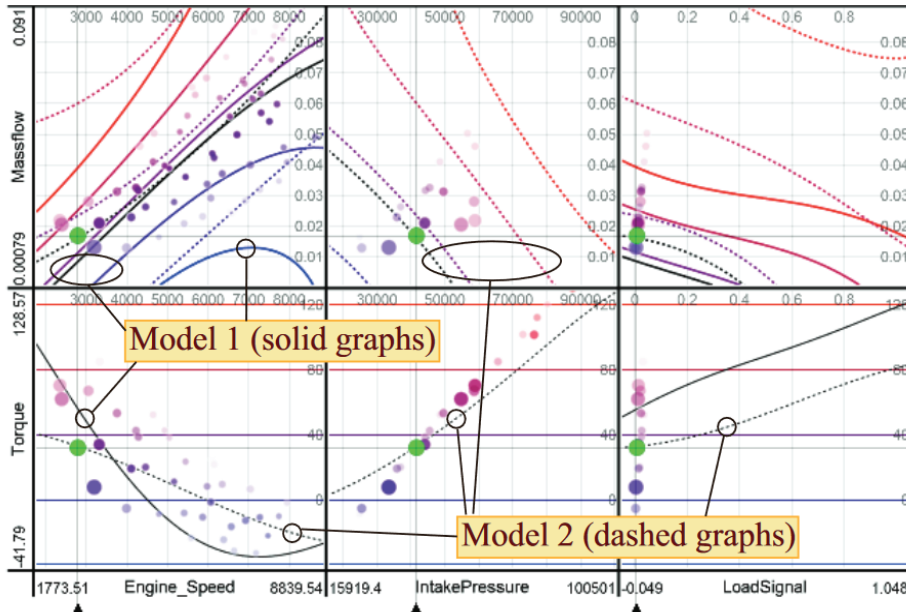


Figure 2.25: “HyperMoVal” supports the validation and comparison of multidimensional regression models in the context of nearby data points [PBK10].

on data subsets for validation via brushing. Regression models (SVM) can be trained by a black box integration of a library, but there is no dedicated support or guidance for the building process as such. Moreover, the point-wise approach of slicing the model may lead to important local dependencies to be missed, especially if the number of dimensions increases. The Partition-Based Framework (Paper C) was the first regression approach to support guided interactive workflows for both building and validation, while allowing users to incorporate domain knowledge. Here, partitioning the feature space is the key idea to provide an overview of local dependencies, and to avoid structural assumptions in the ranking of features by relevance. Moreover, the framework goes beyond other feature ranking approaches [MBD⁺11, KPB14], as it ranks and visualizes *pairs* of features to support an identification of 2D-feature interactions explaining the target.

Since its publication, several follow-up works have built on the idea of the Partition-Based Framework and extended it in various ways. Klemm et al. extend the approach beyond features and feature pairs to also consider three-dimensional subsets of the feature space [KLG⁺16]. Their solution adds a voxel-based visualization that shows the relevance of feature triplets for the target, to identify relevant interactions of three features (see Figure 2.26). In “LoVis”, Zhao et al. also rank features by relevance, and visualize feature dependencies very similar to Paper C [ZWRH14]. Additional views support the discovery of local patterns, and reveal complementary features that explain different parts of the data well in combination (see Figure 2.27). Users can incrementally add local models to build composite predictors, while dedicated views and metrics prevent the

2. BACKGROUND AND RELATED WORK

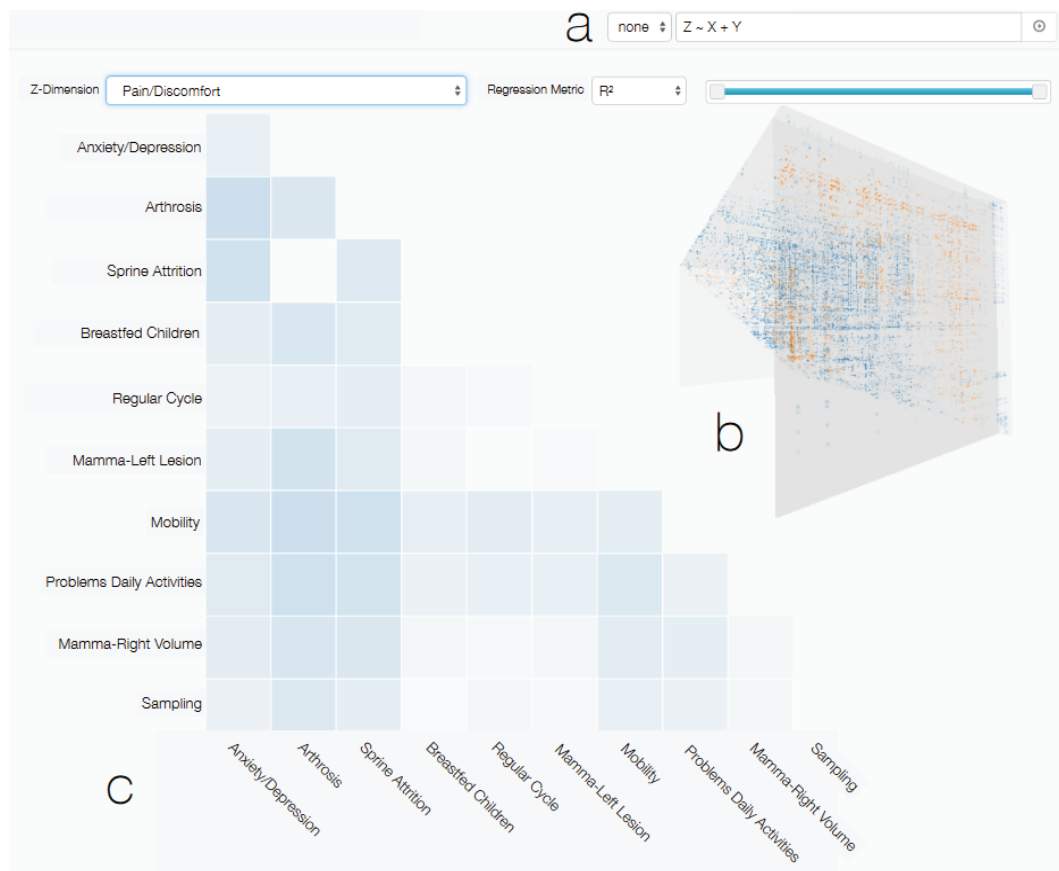


Figure 2.26: Matrix- and voxel-based displays support the discovery of dependencies of a target on pairs and triplets of features [KLG⁺16].

model from getting unnecessarily complex. More loosely related to Paper C, Andrienko et al. support regression modeling for predicting real estate prices from geographic and demographic attributes [AAR⁺16]. Treed regression models are built using automatic feature selection algorithms, while users can interactively define the candidate features and data records. Related to Paper C, the prediction bias is visualized over a geographic map, which may inform refinements like modeling regions separately from each other. Additionally, the geographical variance of feature importance is visualized on a map to support an understanding of how determining factors differ geographically.

In conclusion, interactive regression has started to receive increasing attention over the past years. Compared to classification or clustering (see next section), however, the number and diversity of approaches are limited. Most existing techniques, including Paper C, focus on incremental model refinement by guiding the selection of relevant features. Result-oriented approaches to model building and selection, where users can trade-off expectations in terms of result characteristics, are surprisingly rare for regression.

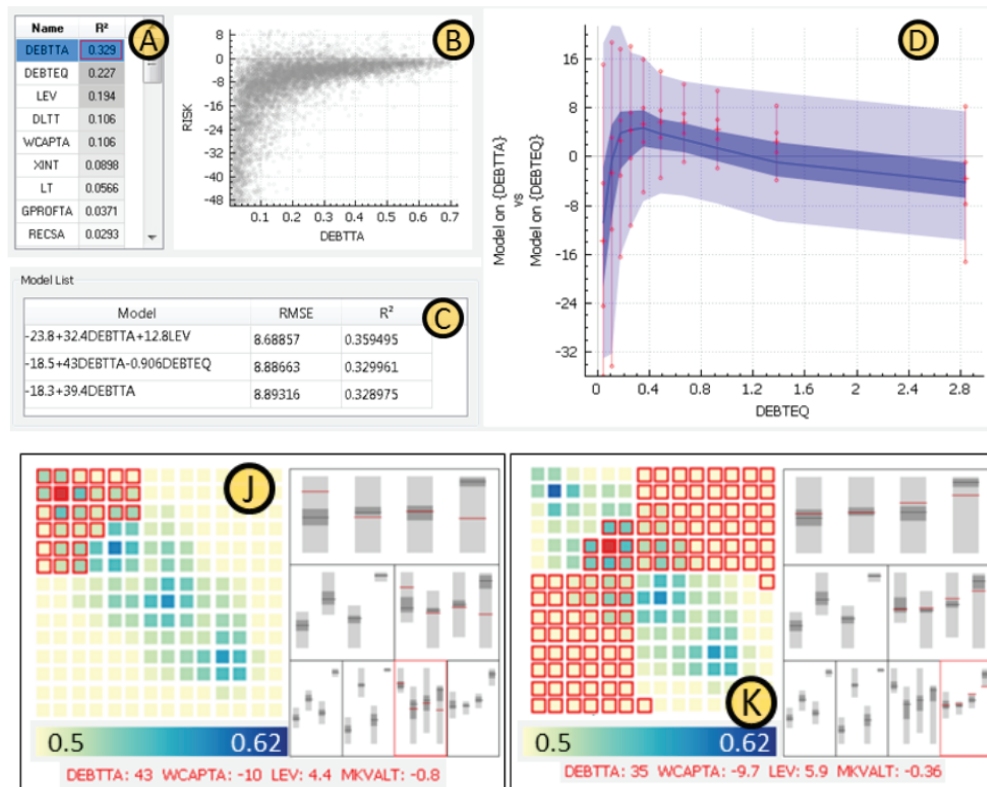


Figure 2.27: “LoVis” ranks and visualizes relationships to support the building of multiple models for local effects. Partitioning of the data space at varying coarseness, as shown in the lower row, guides the trade-off between over- and underfitting. [ZWRH14].

A reason might be that regression is mostly used for predictive tasks, where automatable objectives like accuracy are typically more important than, for example, interpretability. Nonetheless, it might be promising to investigate building ensembles of diverse regression models similar to TreePOD (Paper B), and to guide users in exploring this space of possibilities. Seeing multiple alternatives for explaining a target may be interesting information on its own, and might also reduce the danger of getting stuck in a local optimum as opposed to the refinement of single models [WCH15].

2.3.3 Clustering and Dimension Reduction

The last section of this state of the art report discusses approaches from unsupervised learning. Specifically, it focuses on two of the most common unsupervised methods, namely *clustering* and *dimension reduction*.

Clustering refers to the task of finding inherent groups in the data according to some notion of similarity [HTF09]. In contrast to supervised learning, there is usually no ground truth that the results of clustering methods can be compared to. Thus, it is

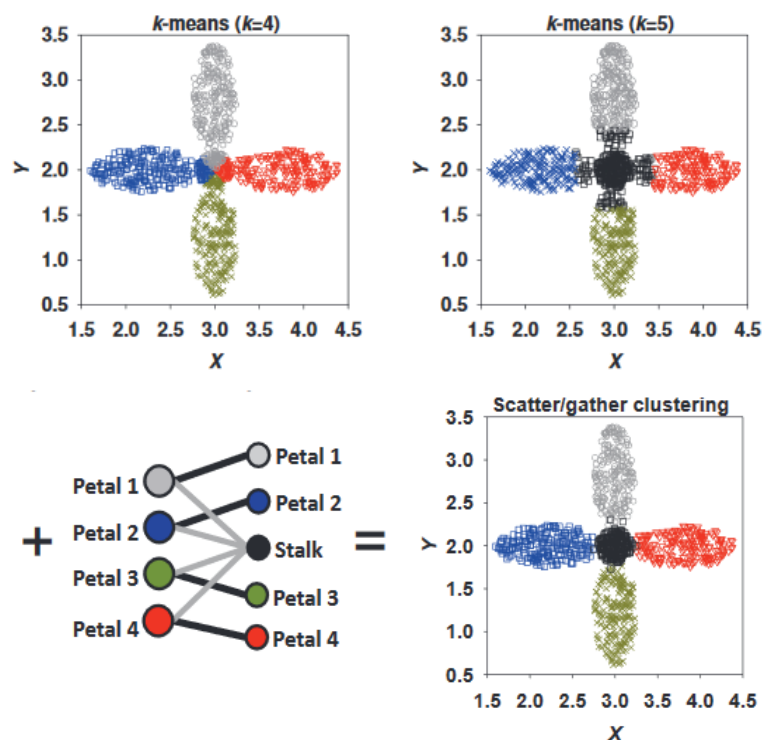


Figure 2.28: Relating clusters across different parametrizations enables algorithms to produce results that match user expectations [HOG⁺12].

not as straightforward to measure the quality of clusterings in crisp, quantitative terms like accuracy or squared error [CENS06]. While several metrics exist, a comprehensive assessment of cluster adequacy often involves qualitative judgments and know-how about the purpose of the clustering. This may explain why clustering has received relatively much attention by the visualization community.

Several Visual Analytics approaches involve the user in the building and refinement of clusters. Cohn et al. envision a process where users guide a clustering algorithm by interactively specifying constraints [CCM03]. Based on a given clustering, users may indicate certain items that should not be grouped together, and move them to different clusters. While such constraints may not be known in advance, the authors argue that “critiquing is easier than constructing”, and users typically “know it when they see it”. In line with this principle, several approaches offer the modification of spatial arrangements to express constraints for cluster optimization [BDW08]. Lee et al., for example, allow users to shift text documents to interact with clusters obtained by topic modeling [LKC⁺12]. Des Jardins et al. let users drag items in a force-directed layout to modify clusters [DMF07], and Kumar et al. allow users to specify constraints on triplets of data items (X is more similar to Y than to Z) [KK08]. In contrast to these observation-level techniques for steering, Hossain et al. let users define constraints on entire clusters [HOG⁺12]. Based on such input, their “ScatterGather” algorithm tries to produce a clustering that better reflects the expectations of the analyst (see

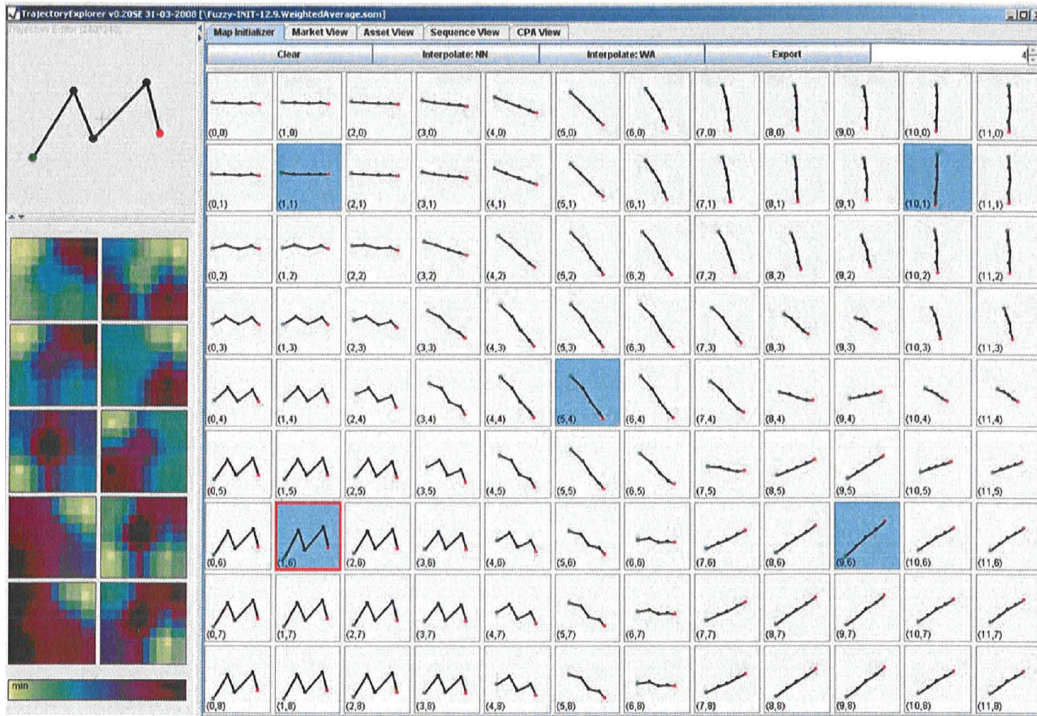


Figure 2.29: Sketching a few anchor trajectories (shown in blue) allows users to steer the clustering of a Self-Organizing Map [SBvLK08].

Figure 2.28). Relying more on user interaction than algorithmic updates, Bruneau et al. let users modify cluster labels similar to a painting application [BPBO15]. Based on a 2D embedding of clustered points, convenient user interactions with algorithms under the hood support re-labeling many points at once. For example, a “diffusion” of class labels to nearby structures is triggered with one click instead of manual brushing and labeling.

In contrast to these result-oriented approaches, other works involve the user in the clustering process as such. Schreck et al. let users visually monitor and control the training of a self-organizing map clustering algorithm for trajectory data [SBvLK08]. By interactive drawing of trajectories, users may initialize the algorithm based on domain knowledge and expectations (see Figure 2.29). During training iterations, the clustering result is constantly visualized, while users can modify clusters or parameters, and terminate the algorithm when sufficiently converged. Rinzivillo et al. also rely on human control in building clusters of trajectory data [RPN⁺08]. In their approach, users are presented with clusterings that are generated step by step, and may decide at any point to stop, modify, or terminate ongoing calculations. In such tight steering applications, it is crucial to deliver results in at most a few seconds as not to lose the user’s attention [CRM91]. Ahmed and Weaver suggest a pre-computation scheme based on expected user interactions to provide clustering results at interactive rates [AW12]. Alternatively, strategies for providing approximate results can be applied to keep users focused on the analytical process (see Paper A) [MPG⁺14].

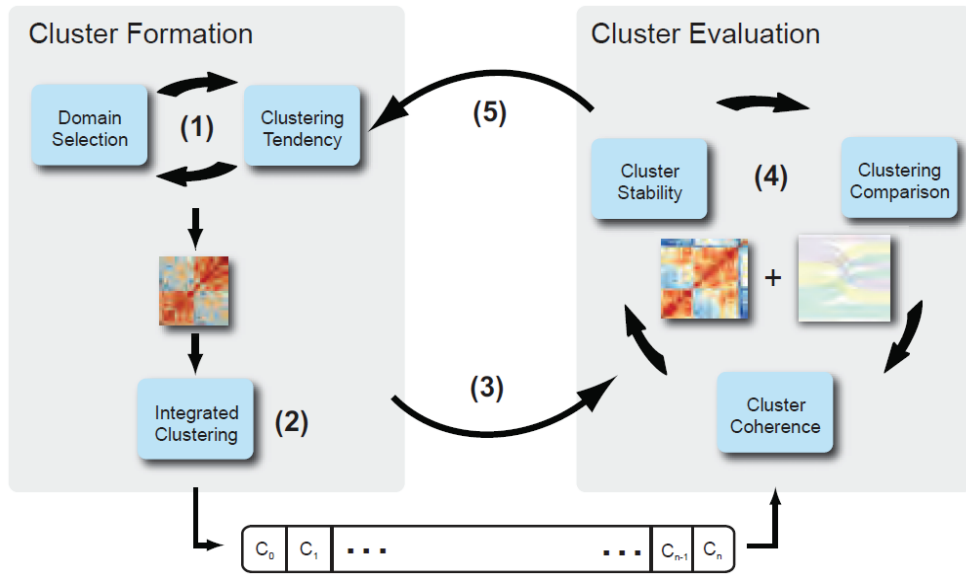


Figure 2.30: A tight integration of cluster formation and evaluation allows users to select clusterings using multiple views [TPRH11a].

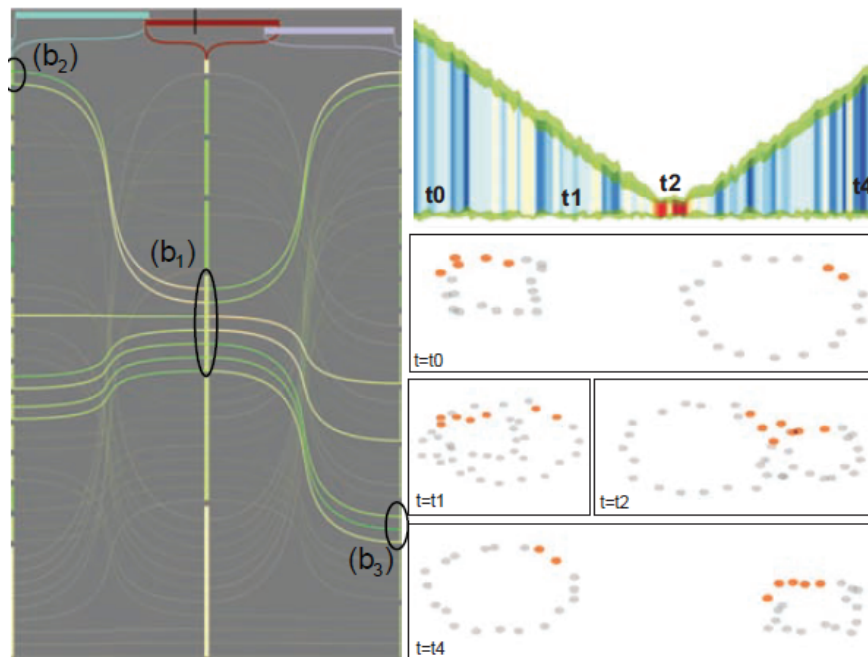


Figure 2.31: Visualizing cluster structure over time reveals system stability and topological changes [TPRH11b].

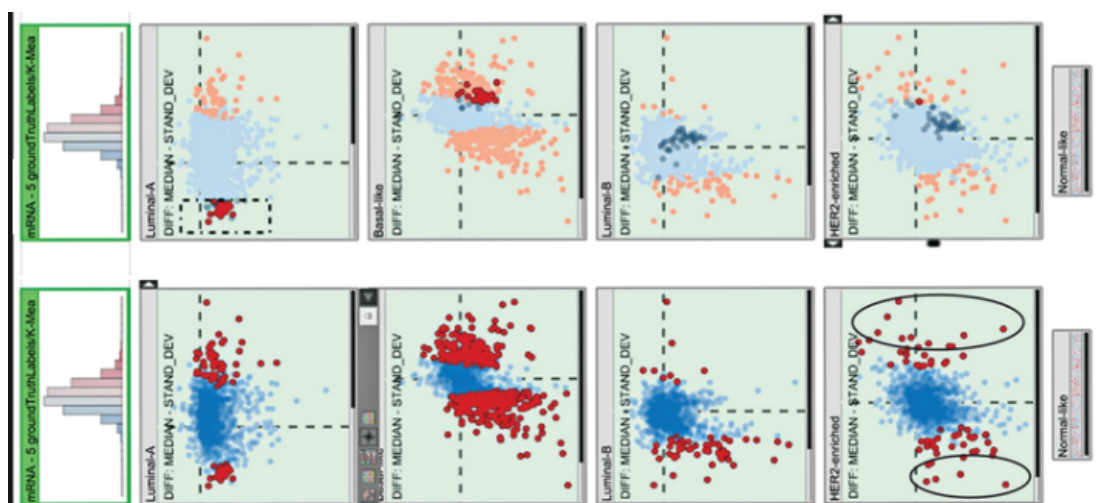


Figure 2.32: Linked scatterplots showing features as points support comparisons of relevant features across clusterings [TLS⁺14].

While the previously discussed works support cluster building and refinement, several approaches focus on validation, comparison, and selection tasks. Turkay et al. propose a tight integration of cluster formation and evaluation to enable an effective steering of algorithms [TPRH11a]. Multiple coordinated views convey whether (1) the data tends to form clusters, (2) clusters remain stable across alternative clusterings, and whether (3) clusters are cohesive, i.e., highly similar within but highly different from other clusters (see Figure 2.30). Based on these views, clusterings can be effectively compared and selected for a particular application. In the same year, Turkay et al. extended the approach to account for aspects of time-dependent data [TPRH11b]. Here, the idea is to visualize the stability of a clustering across multiple time steps, and to reveal structural changes like suddenly merging or splitting clusters (see Figure 2.31). In an application with biomolecular simulations, this allowed analysts to effectively identify vanishing clusters, and to determine the time when the simulated system stabilizes. In more recent follow-up work, Turkay et al. contributed interactive visualizations to compare clusterings with respect to their most discriminative features [TLS⁺14]. For each of multiple clusterings, statistical measures are computed per feature, and shown as points in coordinated scatter plots (see Figure 2.32). Selecting important features for one clustering highlights them in the representations of the others, revealing stable explanations as well as alternatives.

The “Hierarchical Cluster Explorer” by Seo and Shneiderman enables cluster comparisons for choosing an adequate model complexity [SS02]. Their system provides multiple linked views to provide an overview of hierarchically clustered data from the context of genomics. While users can not influence the clustering algorithm as such, a significant benefit of the approach is that no parameters need to be specified in advance. Once the clustering is computed, users can adjust the “cutting” position through the hierarchy with a slider to select a clustering with adequate level of detail (see Figure 2.33). Caruana et al.

2. BACKGROUND AND RELATED WORK

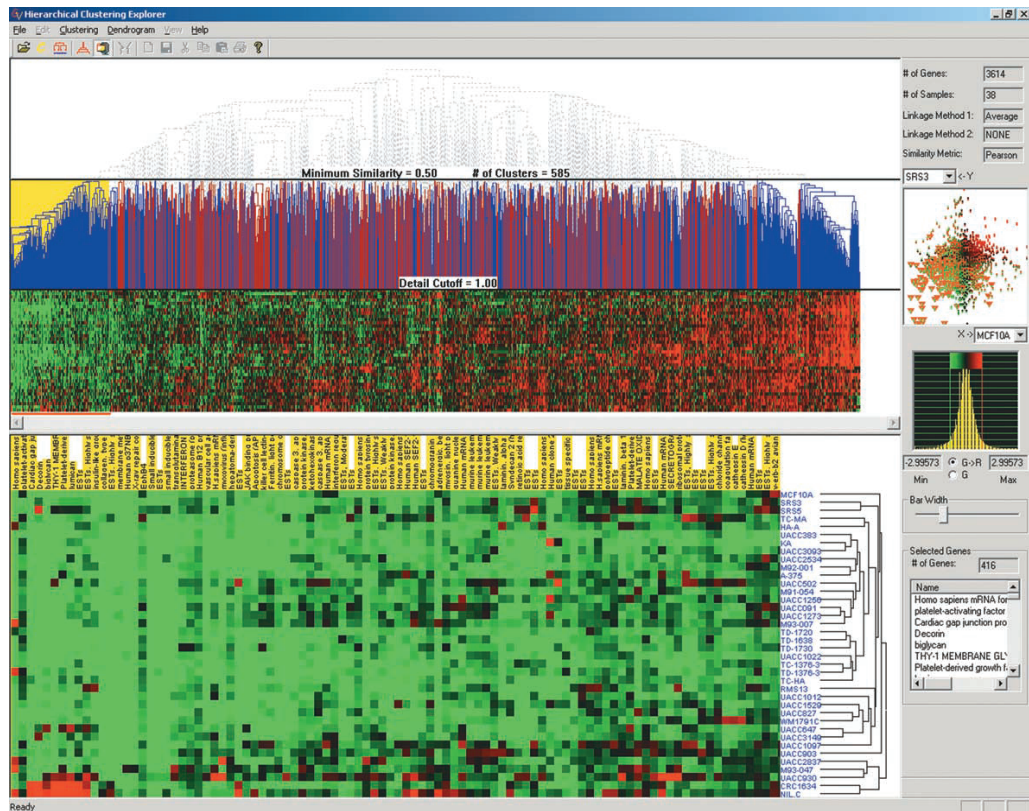


Figure 2.33: Interactively defining a cut through a dendrogram supports choosing an adequate model complexity for hierarchical clusterings [SS02].

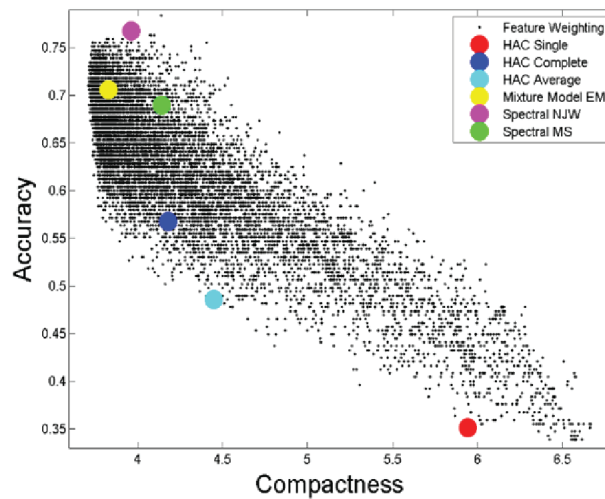


Figure 2.34: Meta-clustering a large number of clusterings by their properties guides users towards distinct groups of achievable model characteristics, and supports trade-offs between them [CENS06].

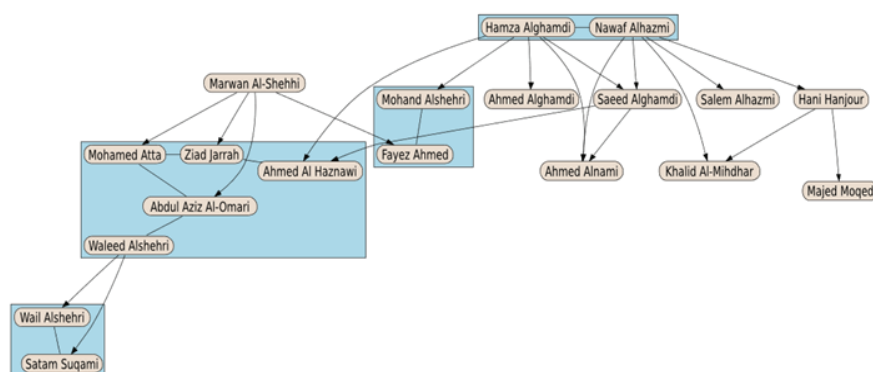


Figure 2.35: Constraining items into inseparable groups allows users to align similarity embeddings with mental models [DKM06].

take this idea of proactive computation and result-oriented model selection one step further [CENS06]. Their “Meta-Clustering” approach first produces a large, diverse set of alternative clusterings by repeating the k-means algorithm for different input conditions. Then, these clusterings are clustered by their characteristics, so that users only need to inspect a small set of qualitatively different candidates (see Figure 2.34). In this respect, this approach is very similar to TreePOD (Paper B), as it provides scalable guidance by focusing on a manageable subset of decision-relevant alternatives. Moreover, Meta-Clustering shares TreePOD’s inclusivity for inexperienced modelers (G2), as no parameters have to be specified or understood in advance.

Concluding the part on clustering, note that many more approaches exist that have not been discussed in this section. Given that the thesis’ contributions do not focus on clustering, the intention was to provide an overview of representative solutions with a clear human-oriented focus. Please refer to other state of the art reports such as the one by Endert et al. [ERT⁺17] for a more complete survey.

The second part of this section deals with dimension reduction, another highly common technique for unsupervised learning. Dimension reduction refers to finding, deriving, or synthesizing new dimensions from the existing ones, where most of the interesting variation happens. Combined with visualization, the typical goal is to embed high-dimensional data into two or three dimensions, such that salient structures like clusters, outliers, or manifolds become visible [SZS⁺17].

Several Visual Analytics solutions make use of human expertise in identifying and refining such embeddings. As for clustering, a common approach is letting users specify constraints for embedding algorithms to incorporate mental models. Endert et al., as well as Buja et al. enable users a specification of “anchor points” that remain in fixed positions, while algorithms such as PCA or MDS arrange the data points around them [EHM⁺11, BSL⁺08]. Similarly, “ForceSpire” by Endert et al. supports the anchoring of documents in a force-directed layout [EFN11]. As a higher-level form of constraints, Dwyer et al. enable users to define regions that must not overlap, or that may not be torn apart [DKM06] (see Figure 2.35).

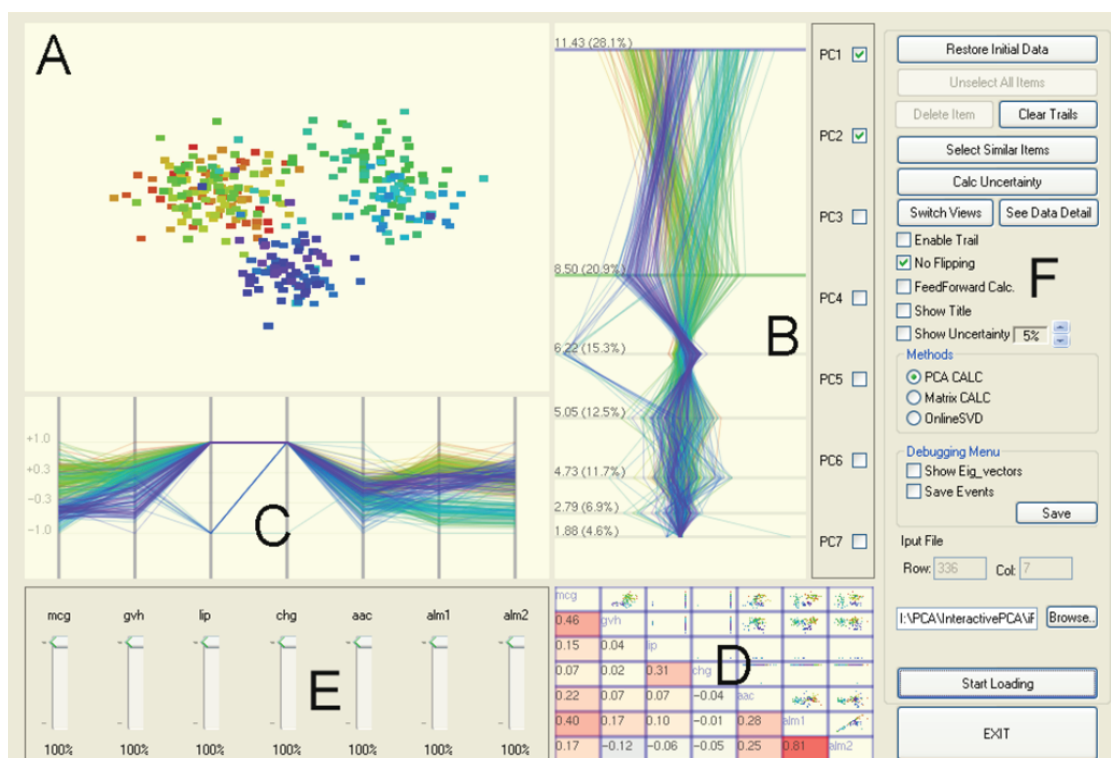


Figure 2.36: The “iPCA” system allows users to weight features by importance for principal component analysis with a slider interface [JZF⁺09].

Another group of solutions makes use of domain knowledge to adjust the notion of similarity between items that underlies most embedding algorithms. In the simplest case, users can modify explicit weights assigned to the data dimensions to stress their importance for similarity. The “iPCA” approach by Jeong et al. provides sliders to adjust these weights, also called “dimension loadings” (see Figure 2.36) [JZF⁺09]. Johansson and Johansson employ a more output-driven approach, and let users modify weights for quality metrics that embeddings may optimize [JJ09]. As for most interactive weighting approaches, however, weights are often only vaguely known and may involve trial-and-error until a particular result is achieved [PSTW⁺17]. To incorporate mental models more intuitively, Endert et al. propose direct, “semantic interactions” with items in the 2D embedding to define similarities [EFN11]: By dragging an item closer to another one, users may attribute higher similarity to this pair, which updates the parameters of the embedding under the hood. As a result, similar items to either of them will also be moved to reflect the updated similarity function. Similar spatial metaphors have been employed by various other solutions, including “DisFunction” by Brown et al. (see Figure 2.37a) [BLBC12], and an interactive projection approach by Molchanov and Linsen [ML14]. Buja et al. let users move around entire groups of items to escape from local minima of stress optimization, and to explore the stability of the embedding [BSL⁺08]. Other solutions allow users to manipulate the actual data

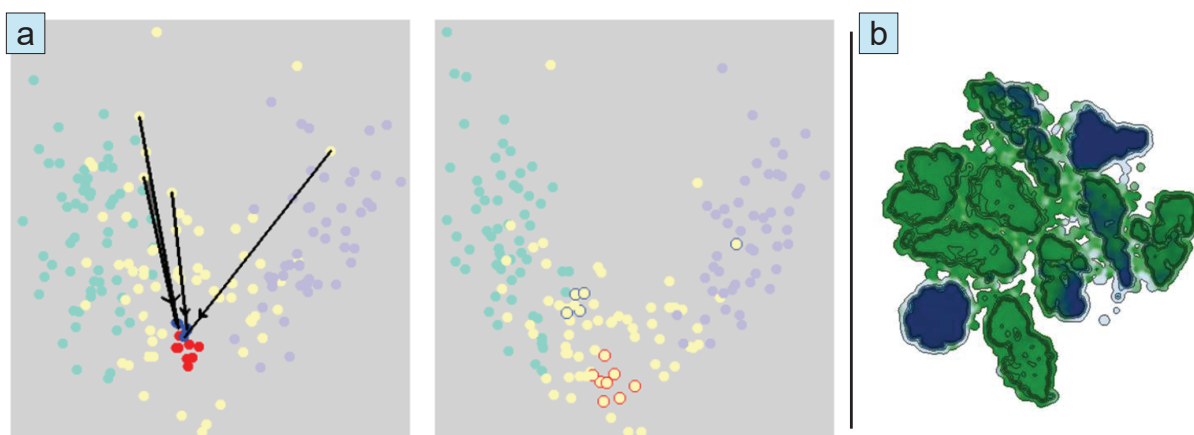


Figure 2.37: (a) In “Dis-Function”, users may drag points (shown in blue) closer to others (red) to update the underlying distance function for these preferences (middle image) [BLBC12]. (b) Pezzotti et al. allow users to prioritize important regions in a progressive t-SNE projection algorithm [PLvdM⁺17]. Green color indicates regions that are still coarsely approximated.

values underlying the embedding to explore its stability. In the “iPCA” approach, for example, users may remove data items such as outliers, or edit their values to see what happens to the embedding [JZF⁺09]. Trying out different what-if scenarios helps users in building confidence, and conveys the sensitivity of the underlying algorithmic functions. “StarSPIRE” [BNH14] and “ForceSPIRE” [EFN11], on the other hand, let users annotate data items to enrich them with additional semantics for the similarity computations.

The previously discussed works seek a semantic dialog with the user based on results. Another approach is to let users directly steer the parameters and process of the embedding algorithm as such. Choo et al., for example, expose the parameters of a supervised dimension reduction algorithm by a slider interface [CLKP10]. Here, the idea is to let users visually explore the parameter space to reveal structures such as clusters. “DimStiller” by Ingram et al. allows users to build dimension reduction pipelines by concatenating different algorithmic operators [IMI⁺10]. Using individual controls, the parameters of each operator can be tuned, while immediate visual feedback is provided to guide the exploration. Other approaches rely on the user to prioritize the work of expensive dimension reduction algorithms. In “MDSteer”, Williams and Munzner propose a progressive, steerable version of multidimensional scaling (MDS) for very large datasets [WM04]. To shortcut the high execution times of MDS, the approach shows partial results based on subsets of the data after a few seconds. Users may then indicate regions, or data subsets, that should be computed next, to obtain an understanding of the most important parts after shorter time. Pezzotti et al. employ a similar methodology for the t-SNE algorithm [PLvdM⁺17]. Based on a roughly approximated notion of similarity, intermediate results are shown that become incrementally more accurate over time. Users may prioritize interesting data subsets, while the current degree of approximation is color-coded to convey the uncertainty of the incremental results (see Figure 2.37b).

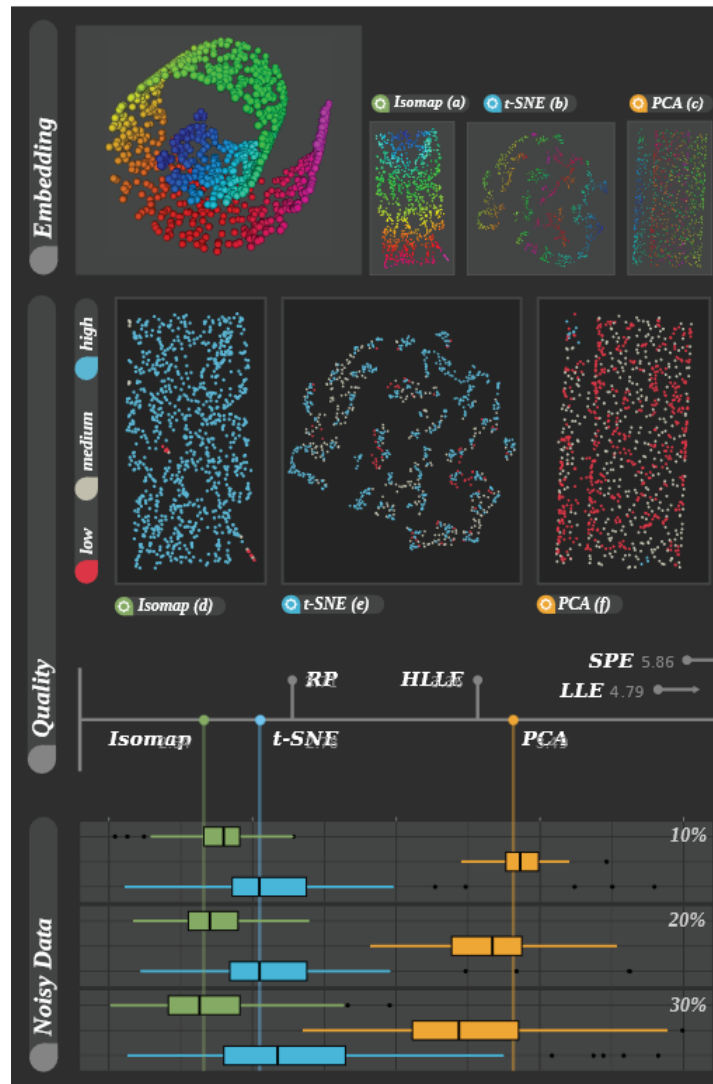


Figure 2.38: Ranking and visual comparison allow users to select appropriate embeddings [RL15].

The last block of solutions focuses on supporting the evaluation and comparison of multiple dimension reduction algorithms. Embeddings of high-dimensional data can be generated in various ways by different algorithms and parametrizations. A study by Lewis et al. has shown that especially inexperienced modelers often disagree on the quality of dimension reduction methods [LVdMds12]. To support effective comparisons, Rieck and Leitte propose a system to visualize and rank embeddings by quality measures (see Figure 2.38) [RL15]. Being based on general topological properties of the embeddings, their framework supports comparisons between results of different algorithm types, like PCA, MDS, t-SNE or IsoMap. Similarly, Liu et al. propose the use of distortion-

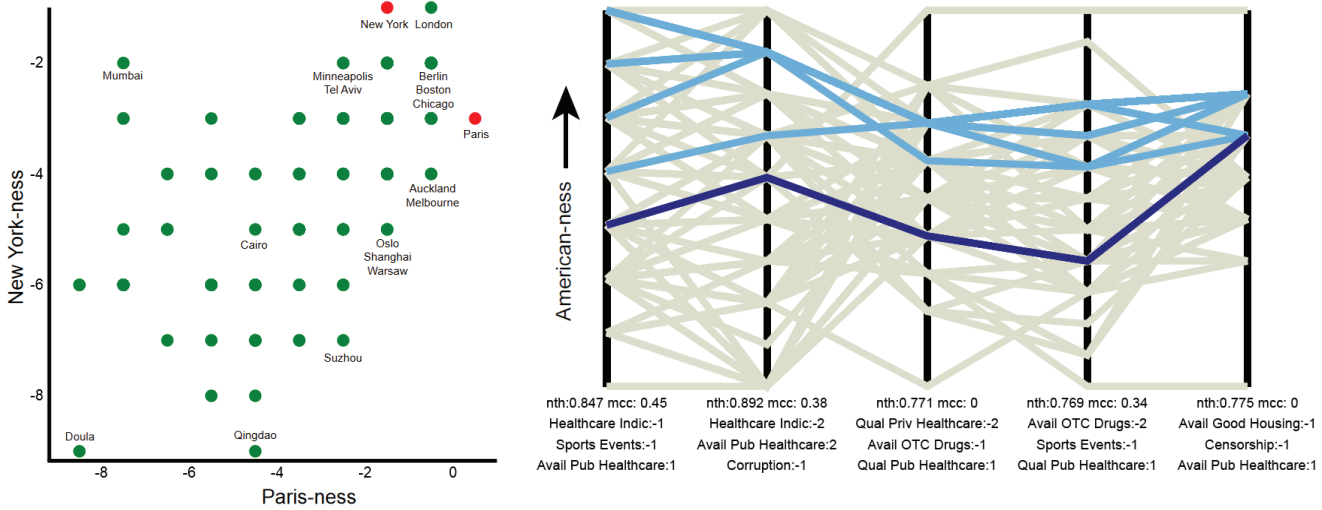


Figure 2.39: Annotating objects with semantic concepts like “Paris-ness” for cities allows users to craft projections with meaningful axes [Gle13]. Projections can be compared regarding their alignment with prior knowledge, simplicity, and other objectives (right image).

based quality metrics to compare different types of embeddings [LWBP14]. As a less general, but highly human-oriented approach to model selection, Gleicher proposes an integrated approach for crafting and comparing projections [Gle13]. In contrast to statistical dimension reduction, the idea of the proposed “Explainers” system is to create projections that align with semantic, user-specified annotations. In a dataset of cities, for instance, users may annotate cities with vague notions like “American-ness” or “Paris-ness” based on intuition. The system would then create a diverse set of projections onto these fictional, yet meaningful axes, which users can explore (see Figure 2.39). Similar to TreePOD (Paper B), the Explainers system considers trade-offs in choosing from these projections, including their simplicity, expressive power, alignment with prior knowledge, and diversity. This human-oriented approach to model selection, and the understandable quality of the embedding sets this system somewhat apart from other dimension reduction techniques, and makes it particularly noteworthy in the context of this thesis.

In conclusion, the Visual Analytics community has brought forth various ways of involving humans in dimension reduction tasks. Again, this section only provided a coarse overview, as no thesis contribution focuses on unsupervised learning techniques. Please refer to Sacha et al. [SZS⁺17] or Endert et al. [ERT⁺17] for more complete surveys on the topic.

Discussion and Conclusions

This thesis has made multiple contributions to Visual Analytics in favor of a more democratic, human-oriented modeling process. With the Partition-Based Framework and TreePOD (Papers B+C), it introduced new human-centered solutions for previously under-addressed modeling tasks (G1). TreePOD has outlined a methodology to effectively involve users without deep statistical backgrounds in the modeling process through visualization (G2). The work on Opening the Black Box (Paper A) has shown a large potential for re-using existing algorithms to implement human-oriented solutions (G3), and paved the way for more user-involving algorithm design in the future. Finally, evaluations with real users, and reflections on real-world deployments (Papers D+E) suggest a high relevance of the thesis contributions for Visual Analytics in practice.

Throughout the papers, the thesis makes multiple cases for the benefits of human involvement in the modeling process. A key lesson learned, however, is that not all forms of user involvement are equally suitable for all types of user and application scenarios. Effective steering of complex algorithms, for example, requires a certain understanding of the involved mechanisms, which may preclude inexperienced modelers in some cases. On the other hand, experts with the respective knowledge of inner workings should not be limited by superficial, overly simplistic interfaces. Choosing an adequate degree, or *scope* of user involvement is thus crucial for the adoption of new techniques by particular user groups. The three main contributions (Papers A,B,C) promote three radically different scopes, by providing feedback loops of varying tightness (see Figure 3.1): On one side of the design space, Opening the Black Box (Paper A) argues for involving users in the tightest possible loop. At any time during an ongoing computation, users can incorporate control signals to cancel, prioritize, or steer the algorithm's work (Figure 3.1a). While the ability to cancel early is helpful for any user, the usefulness of steering for inexperienced modelers relies on intuitive interfaces to do so. Designing such interfaces requires effort and creativity, especially if the gap between algorithm complexity and a user's expertise is large. The Partition-Based Framework for regression (Paper C) employs a less tight

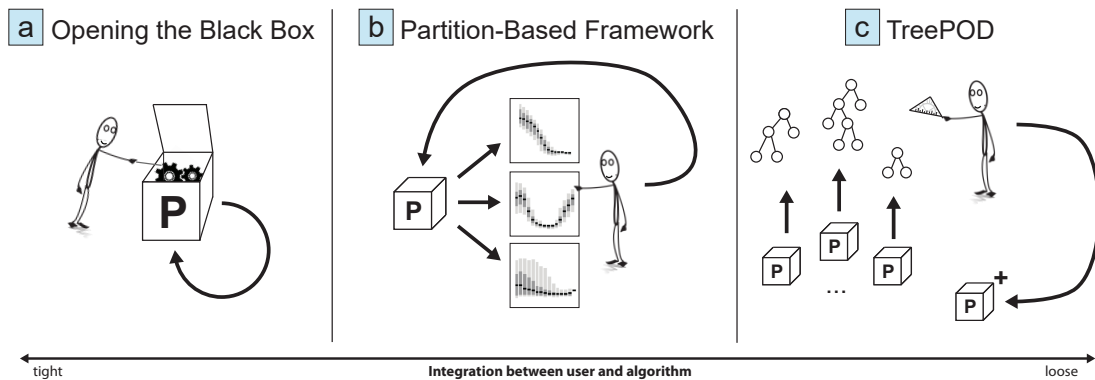


Figure 3.1: The three main contributions provide feedback loops for user involvement at varying scopes: (a) with ongoing computations, (b) with algorithmically suggested refinements of a single model, and (c) with an extendable ensemble of models.

feedback loop, and does not require users to bother with algorithmic details. In contrast to the real-time interaction of Paper A, the cooperation between algorithm and user can be described as “turn-based”: After algorithms provide a guided list of possible model refinements, the user selects the desired model change, which in turn, triggers new algorithmic suggestions (Figure 3.1b). While the required algorithmic knowledge is low in this process, users still need a certain understanding of how selected features affect the model to make good choices. On the most novice-friendly end of the spectrum, TreePOD (Paper B) promotes the loosest integration between algorithm and user. Here, model candidates are built in advance, while users perform judgments and make decisions purely based on the results (Figure 3.1c). On demand, users may trigger refinements or compute new candidates to extend the candidate set using controls that hide algorithmic details per default.

Of all contributed concepts, TreePOD was appreciated as the least background-assuming approach by our collaborating domain experts. The result-oriented approach allowed them to identify good models in much shorter time, at much higher confidence than previously used tools. As another lesson learned, however, they warned us from hiding algorithm parameters entirely, as this would remove familiar ground for modelers with the respective expertise. Despite its more time-consuming approach to cooperative model building, the Partition-Based Framework was also considered a huge step forward by our collaborators. Still much faster than previously used tools, the framework provided them with an unprecedented transparency and confidence in the built models.

In addition to positive feedback from our collaborators, the thesis contributions also had a measurable impact on the scientific community. The Partition-Based Framework, for example, received the best paper award at IEEE VAST 2013, and seems to have inspired several follow-up works as described in Section 2.3.2. By August 2018, the paper has

been cited 77 times according to Google Scholar¹. Opening the Black Box, which was published one year later, has been cited 64 times, while other model papers have built on the proposed concepts. The “Enhanced Visualization Process Model for Incremental Visualization” by Schulz et al., for example, explicitly relates the building blocks of their model to the proposed strategies for user involvement [SASS16]. TreePOD, on the other hand, is the newest of the main contributions, and was cited 3 times since its publication in 2018.

Before concluding this thesis, the author wants to stress that this work does not regard human involvement as unconditionally beneficial in all scenarios. Parts of the modeling process that can be automated well, *should* be automated, to free up as much human resources for more important high-level tasks as possible. Yet, there is ample evidence that human involvement *can* lead to better results due to the additional knowledge and cognitive abilities that automatic approaches simply do not have [ACKK14, KKEM10]. Moreover, human-oriented requirements arising from applications, as well as needs regarding trust and accountability [KPB14] often make human involvement attractive, and sometimes inevitable. Nevertheless, it is important to note that human involvement incurs costs that need to be balanced with the benefits.

One such cost is the fact that human intervention may affect the reproducibility of results. Blessing and curse at the same time, interactive specifications like brushing or dragging items around the screen are much less exact than textual specification in scripting environments. Also, the mere fact that humans may deviate from well-defined algorithmic paths at all can be seen as counterproductive in certain analytical scenarios, and may require additional justification in reports.

As a second type of cost, model inspection and manual improvement may take up considerable amounts of human time. This is particularly the case for approaches that promote deep human involvement to optimize or override algorithmic decisions. Studies from human-computer interaction have shown that humans are only willing to spend their attention and time on complex tasks, if the expected benefits outweigh the expected costs [Bla02]. In cases, where only a few models need to be built to serve as key assets, fighting for every percent of accuracy and increased confidence may certainly be worthwhile. With increasing digitization, however, trends indicate growing numbers of devices and processes that need to be modeled and monitored in many sectors. Power grid operators, for example, predict the production from hundreds of power plants on a regular basis, while industries constantly add new devices that need predictive maintenance. In such scenarios, the time invested per model is an important criterion to minimize. Thus, the need to produce good models without deep involvement in every algorithmic decision may become increasingly pressing in the years to come.

Result-oriented approaches like TreePOD can be seen as a first step in this direction. However, to the author’s knowledge, no solutions exist that support modeling hundreds or

¹https://scholar.google.at/scholar?hl=en&as_sdt=0%2C5&q=m%C3%BChlbacher+piringner&btnG=, last accessed 2018, Aug 24.

thousands of targets in acceptable time, while still considering knowledge and requirements from the domain. Designing more scalable approaches to model building and validation that account for the increasingly strict time constraints of experts remains an important challenge for future work. In line with the growing desire for automation, a possible direction could be to shift human involvement from the building phase to (1) the high-level specification of objectives for entire model batches, and (2) the posterior inspection and remedy of problem cases.

As data volumes and velocity keep increasing, a second topic for future research is extending the work on scalable infrastructures for online and progressive computation. This includes adding more algorithms and libraries that provide early communication with users, as well as making visualization and interaction modules ready for the implications. While a few promising examples exist [FPDm12, TKBH17, PLvdM⁺17], it remains a challenge to deal with the incompleteness, instability, and uncertainty of incremental results in general analysis scenarios. It may also take time for analysts to get used to working with incomplete results as opposed to conventional, offline analytics. Thus, it is crucial that new technical solutions are oriented towards the cognitive capabilities and characteristics of users, and not the other way around [TKBH17].

Finally, it remains a challenge to guide the modeling process while keeping the exploration path and result as unbiased as possible. Information overload, as well as algorithmic suggestions to conquer it, may lead users to an early pruning of eventually better solutions. Step-wise techniques for model refinement, such as the Partition-Based Framework (Paper C) are particular prone to such biases. As a possible direction, free computational resources could be used much more often to proactively explore more possibilities than the user explicitly asked for [LB17]. The importance of showing analytic provenance to make the explored path tangible has been recognized by the community. It can be argued, however, that showing the *unseen* paths as context is equally important to complete the picture of provenance, to avoid potential bias, and to increase confidence in the conclusions finally drawn.

In conclusion, human-oriented modeling has come a long way since this thesis started in 2013. Practical works on integration, better support for regression, and improved guidance for model selection are only three examples of the many significant contributions made by the community over the years. And yet, most of these useful works have still not made it into the mainstream technology of modeling in practice so far. Today, the large majority of analysts uses standard scripting environments for modeling, and off-the-shelf visualization software for the results, which is far from the state of the art in interactive machine learning. However, the first commercial tools with a focus on human-oriented modeling have recently started to emerge. *Exploratory* [Nis17], and *Dataiku DSS* [dat13], for example, provide interactive visual interfaces to proven algorithms from R or Python, that can be used without programming skills. According to these sources, as well as our own experiences with Visplore, the need for more accessible statistical modeling is real, and growing. And the more interactive approaches become part of the mainstream, the easier it will get to translate the newest techniques and findings from

Visual Analytics science into practice. Until then, I hope, that the described efforts for empowering inexperienced modelers, and for making algorithms ready for visualization have contributed an important part to this joint movement of democratization.

Part II

Publications

PAPER

A

Opening The Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations

Published in *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643-1652, 2014. [MPG⁺14]

Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations

Thomas Mühlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair and Marc Streit

Abstract—An increasing number of interactive visualization tools stress the integration with computational software like MATLAB and R to access a variety of proven algorithms. In many cases, however, the algorithms are used as black boxes that run to completion in isolation which contradicts the needs of interactive data exploration. This paper structures, formalizes, and discusses possibilities to enable user involvement in ongoing computations. Based on a structured characterization of needs regarding intermediate feedback and control, the main contribution is a formalization and comparison of strategies for achieving user involvement for algorithms with different characteristics. In the context of integration, we describe considerations for implementing these strategies either as part of the visualization tool or as part of the algorithm, and we identify requirements and guidelines for the design of algorithmic APIs. To assess the practical applicability, we provide a survey of frequently used algorithm implementations within R regarding the fulfillment of these guidelines. While echoing previous calls for analysis modules which support data exploration more directly, we conclude that a range of pragmatic options for enabling user involvement in ongoing computations exists on both the visualization and algorithm side and should be used.

Index Terms—Visual analytics infrastructures, integration, interactive algorithms, user involvement, problem subdivision

1 INTRODUCTION

A tight interplay between visualization, interaction, and analytical computation is the core aspect of Visual Analytics [25, 45]. The motivation is to combine cognitive and perceptual capabilities of human analysts with computational capabilities for tasks like statistical modeling, planning, and decision making [43]. In addition to intelligent visualization and interaction concepts, involving the user in the analysis process implies delivering results and visual feedback within at most a few seconds [9] and ideally less than 100 ms [42]. Particularly for large data, this requirement contradicts the computational effort of many advanced algorithms like clustering or dimension reduction.

As a compromise, a growing number of systems apply strategies like early visual feedback of partial results [17, 23], cancellation on arrival of new input [36], or active steering of a computation in progress [10]. In current practice, however, this type of application-specific fine-tuning often involves a reimplementation of algorithms by researchers and practitioners in Visual Analytics [15]. The obvious disadvantages include a sub-optimal use of skills and resources and an explosion of proprietary implementations rather than standardized and tested solutions.

In contrast, existing systems and languages for data analysis have widely been used for a long time and offer a variety of proven algorithms. In fact, an increasing number of academic and commercial visualization tools stress the integration with software like MATLAB and R (e.g., the R integration in Tableau¹ or [38]). Two key goals are to offer the algorithmic functionality within the visualization tool and to increase the acceptance by data analysts who have been working with

these script-based environments for years. Packages like RServe² or the MATLAB API³ make the integration reasonably easy from a software engineering point of view. However, as pointed out by Fekete, “*computation and analyses are often seen as black boxes that take tables as input and output, along with set of parameters, and run to completion or error without interruption*” [15]. In general, it is commonly stated in the Visual Analytics community that exploration is not taken into account in most infrastructures for analysis computation [25], explaining “*calls for more research [...] on designing analysis modules that can repair computations when data changes, provide continuous feedback during the computation, and be steered by user interaction when possible*” [15].

Motivated by and echoing these calls, we structure requirements and formalize strategies to achieve them. Using this formalization, we argue that a range of possibilities for implementing these strategies already exists based on currently available computation infrastructures. Specifically, the focus of this paper is on studying conceptual possibilities for tightly integrating analytical algorithms of existing computation software into interactive visualization tools. A main goal of the paper is to increase the awareness and the understanding of these possibilities within the Visual Analytics community. Another goal is to improve the understanding of the needs of Visual Analytics applications within communities focusing on algorithm design like Knowledge Discovery and Data Mining. To this extent, the contributions of the paper can be summarized as follows:

- A structured characterization of visual exploration needs concerning user involvement in ongoing computations.
- A formal characterization and comparison of strategies for achieving user involvement in different types of algorithms
- Considerations for implementing these strategies either as part of the visualization tool or as part of the algorithm, including an identification of requirements and guidelines for the design of algorithmic APIs in favor of a tight integration.
- A survey of frequently used algorithms for knowledge extraction and model building of multivariate data regarding the fulfillment of these guidelines as a case study based on the software R.

2 RELATED WORK

Over the last decades, the interplay between computational environments and visualization has been addressed in numerous research papers and commercial systems. On the one hand, visualization systems

-
- Thomas Mühlbacher is with VRVis Research Center, Vienna, Austria.
E-mail: tm@vrvis.at.
 - Harald Piringer is with VRVis Research Center, Vienna, Austria.
E-mail: hp@vrvis.at.
 - Samuel Gratzl is with Johannes Kepler University Linz, Austria.
E-mail: samuel.gratzl@jku.at.
 - Michael Sedlmair is with University of Vienna, Austria.
E-mail: michael.sedlmair@univie.ac.at.
 - Marc Streit is with Johannes Kepler University Linz, Austria.
E-mail: marc.streit@jku.at.

Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014; date of publication xx xxx 2014; date of current version xx xxx 2014.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

¹<http://www.tableausoftware.com>

²<http://rforge.net/Rserve>

³<http://www.mathworks.com/products/matlab>

integrate computational tools to perform calculations, as found in research [24, 38, 47] as well as in commercial products like Tableau, JMP⁴, or Spotfire⁵. However, these implementations often boil down to a black box integration that is insufficient for realizing interactive exploration of large datasets as envisioned in this paper. On the other hand, there are graphical libraries developed for extending computational environments by visualization capabilities, such as the Gobi[44] package for R. However, these extensions are usually not designed for dealing with large datasets and do not allow users to actively interact with ongoing computations.

The visualization community has already identified the need for intermediate results, which state-of-the-art computational environments cannot provide in most of the cases. According to Fekete [15], analytical environments are not designed for exploration and algorithm designers often make no effort to provide such early results during computation. In the VisMaster book [25, p. 97f], the authors take the same line by explicitly identifying needs and goals for realizing interactive visual analysis. The major goals are to get fast initial response with progressive refinement, to provide means for triggering recomputation following small changes, and to allow analysts to steer the computation. The work by Fisher et al. [17] confirms the need for early feedback during computations by a user study on incremental visualization. A more general discussion of user involvement in online algorithms together with a description of example implementations has been provided by the CONTROL project [23]. We take this requirements analysis one step further, and provide a detailed discussion of how different types of early information exchange support user involvement in interactive exploration.

User interaction in a more general sense was investigated by Yi et al. [52], who proposed seven interaction categories for visualization based on the user’s intent. Card et al.’s venerable work [9] on three levels of time constraints in interaction, or in Nielsen’s book on usability engineering [32]. While these works cover important needs of user involvement in general, we focus on bidirectional user involvement in ongoing computations (Sec. 3), and we derive strategies for achieving the desired involvement in practice (Sec. 4).

To enable earlier user involvement, strategies to accelerate result availability have been proposed in various contexts. Examples include pre-aggregation strategies for databases such as OLAP and data cubes [28], as well as sampling and filtering techniques for progressive refinement in online aggregation [16, 17, 23], enumerative queries [23], or data mining [51]. For subdividable problems, Divide-and-recombine (D&R) approaches split up a problem into multiple parts, solve the parts individually, and finally recombine the partial results. Examples include MapReduce [12] or RHIFE [21]. However, these examples focus more on speeding up the computation of large data by parallelization, rather than actively involving the user.

The need of visualizing incremental results can be found in many different application contexts beyond multivariate data analysis. Progressive drawing is a well-known approach in volume rendering [8], map rendering applications such as Google or Bing Maps, or the drawing of function graphs [35]. Particularly interesting in this respect is the work by Angelini and Santucci [1], as it provides a formal model that allows characterizing and evaluating incremental visualizations regardless of the application context. Furthermore, much research has gone into visually representing the uncertainty of incomplete results [17, 20, 34]. While this is important, the focus of this paper lies on achieving intermediate feedback in the first place, while particular visualization techniques are out of scope.

In summary, many approaches have been proposed to achieve user involvement in ongoing computations. Building on these possibilities, our primary goal is to provide a more formal characterization and comparison of strategies for achieving user involvement for different types of algorithms, together with a discussion in the context of integration with existing computational environments.

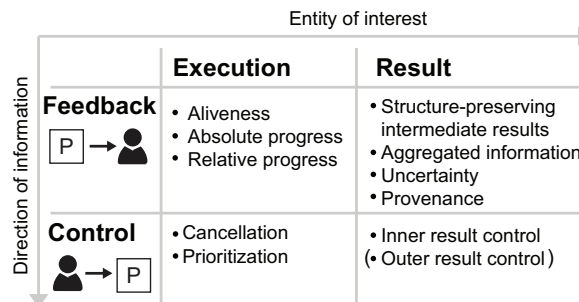


Fig. 1. Types of User Involvement (TUI) structure the needs of visual exploration concerning user involvement in ongoing computations along two directions: the direction of information and the entity of interest.

3 TYPES OF USER INVOLVEMENT

As a starting point for discussing integration concepts, this section describes four different *Types of User Involvement* (TUI) that an interactive visualization may support for an ongoing computation of an algorithm P . In the context of this paper, the main goal of the TUI is to discuss the needs of visual exploration regarding the integration with ongoing computations more specifically.

The scope of the TUI is limited to the time between the start and the end of a computation of P , i.e., it does neither include a priori parameterization, nor any further application of the final result r_P . Accordingly, we define the scope of P such as to include any bounded computation or algorithm that has a well-defined end. Note that this explicitly excludes problems that can change over time, such as accounting for new data that concurrently arrives via streaming. We will refer to a variety of algorithms from multivariate analysis for illustrating the TUI and other concepts. In particular, we will relate to the well-known algorithms k -means clustering (as in the R method `kmeans`) and model-based feature subset selection (as in the R method `regsubsets`) as recurring examples whenever reasonable, and refer to them using the abbreviations `KMEANS` and `SUBSETS`.

We define the TUI based on two orthogonal dimensions (see Fig. 1):

1. **The direction of information.** We distinguish between *feedback* and *control*. Feedback comprises information which is passed from the computation to the user and requires an appropriate visual representation to enable an efficient and correct interpretation by the user. Control is information passed from the user to the computation and requires appropriate interaction techniques.
2. **The entity of interest.** We distinguish between information concerning the *execution* of the computation, and information concerning final or intermediate *results* of P .

The four TUI are defined as the Cartesian product of these dimensions and will be discussed in the following sections: execution feedback (Sec. 3.1), result feedback (Sec. 3.2), execution control (Sec. 3.3), and result control (Sec. 3.4). This classification of TUI is independent of any specific algorithm or the structure of its result as well as any particular implementation and strategy *how* a certain involvement has been realized. We emphasize that our focus is on the question *what* can be visualized and controlled and *why* from a user’s perspective rather than on the issue *how*, which depends on the particular algorithm P . We also stress that this paper does not assume every type of user involvement to be indiscriminately beneficial for each situation. User involvement may incur costs and complexities on multiple levels including the implementation, the computation, and the application by users. Identifying the most appropriate degree of user involvement is a key topic of Visual Analytics [3, 11, 27, 46] and depends on the algorithm and the application context. Our focus is on the classification of known types of user involvement and on general strategies to accomplish it on a technical level rather than on the assessment, when specific TUI are appropriate.

⁴<http://www.jmp.com>

⁵<http://spotfire.tibco.com>

3.1 Execution Feedback

This TUI comprises any kind of feedback about the ongoing computation of P as such. Common types of information include:

- **Aliveness** confirms that the computation is in progress and no event has occurred that may cause failure to eventually deliver the final result, e.g., a crash, a deadlock, or a lost connection.
- **Absolute progress** includes information about the current execution phase of P which may be qualitative (e.g., “computing distance matrix...”) or quantitative (e.g., “iteration 12” for KMEANS or the number of processed data items [1]).
- **Relative progress** includes information about the degree of completeness of P which is frequently provided as percentage or as an estimate of the remaining time.

From the user point of view, execution feedback should mainly answer two questions: First, can any result be expected at all? This may not be the case either due to the occurrence of a failure or an unacceptably long time required for computation. Second, does it make sense to wait for the result or do something else in between?

3.2 Result Feedback

This TUI involves any kind of intermediate feedback regarding the result r_P of the ongoing computation of P . We distinguish between four common classes of result feedback:

- **Structure-preserving intermediate results** $r_{\tilde{P}_i}$ are structurally equivalent to the final result r_P in the sense that the same techniques for visualization and data processing can be applied to them as surrogates while r_P is not yet available. This is typically the case for iterative and anytime algorithms. For example, the intermediate object positions after each iteration of a multi-dimensional scaling algorithm are structurally equivalent to the final positions. In case of the SUBSETS example, the best subset found so far has the same structure as the eventually best subset. Further examples from literature include non-negative matrix factorization [10], self-organizing maps [40], and data aggregation [17]. The structure of r_P may be multi-faceted, however, and consist of multiple parts of which only a subset is provided as feedback during computation. In the KMEANS example, r_P comprises both the cluster centers and the cluster assignments of all data points. To limit data transfer, showing only the intermediate centers during computation could be a reasonable option.
- **Aggregated information** provides a certain aspect of intermediate results $r_{\tilde{P}_i}$ without preserving the structure in full detail. Common examples are quality measures of $r_{\tilde{P}_i}$ [5], e.g., the goodness of fit for the best subset so far with respect to a certain type of model (in the SUBSETS example) or the overall stress of the current solution in case of multidimensional scaling.
- **Uncertainty** concerning the final result r_P as estimated based on the available intermediate information. An example are confidence bounds for r_P [17].
- **Provenance** includes any type of meta-information concerning simplifications made for generating $r_{\tilde{P}_i}$. Depending on the strategy to enable user involvement (see Sec. 4), this class may involve information about the considered data or the settings of complexity parameters. In this respect, provenance information is related to execution feedback about the absolute progress, but always refers to a particular intermediate result $r_{\tilde{P}_i}$.

An appropriate visual representation depends on many aspects like the type and structure of the intermediate results $r_{\tilde{P}_i}$, the update rate of $r_{\tilde{P}_i}$, the involved amount of transferred data, and the intended goal of the visualization. It should, however, ensure that the result is perceived as intermediate. One option for doing so is to explicitly represent the change of the intermediate results over time. Examples include techniques of comparative visualization [19] to represent the difference between $r_{\tilde{P}_i}$ and $r_{\tilde{P}_{i-1}}$, or line graphs to visualize the convergence of aggregated information or uncertainty over time [17].

The key benefit of intermediate result feedback for the user is to enable an earlier continuation of the analysis based on preliminary information. Moreover, result feedback supports the decision whether the ongoing computation should be cancelled. This may be the case

if the current intermediate result is already good enough or if the final result is not likely to be good enough. Finally, access to intermediate results is a key requirement for result control (see Sec. 3.4).

3.3 Execution Control

This TUI involves any kind of control of the execution of the ongoing computation of P as such. The most important type of execution control is **cancellation**, i.e., an explicit or implicit request to cancel the execution prematurely. Explicit requests are issued by the user if control feedback or result feedback suggests that either intermediate results are good enough, or the final result is unlikely to be good, or no result can be expected in acceptable time at all. Implicit requests are typically triggered by updated dependencies of the algorithms like changed input data and algorithm parameters. Such requests often entail a subsequent restart of the computation, a paradigm described in the context of multi-threaded visual analysis [36].

Another type of execution control is the **prioritization** of the remaining work. While the final result r_P is not affected, the purpose is to alter the sequence of intermediate results in order to generate presumably more interesting ones earlier. In this respect, prioritization can be regarded as borderline case between execution and result control. Examples include algorithms involving spatial partitioning or hierarchical structures where users may want to process more interesting parts first [50]. As another example, algorithms processing search spaces may benefit from looking into more promising regions first.

3.4 Result Control

This TUI refers to user interaction with the ongoing computation of P in order to steer the final result r_P . This enables users to take advantage from human perception and domain knowledge [3, 25, 45], e.g., for early validation of intermediate results, guided feature selection, weighting, and for avoidance of being stuck in local extrema. In the widest sense, this TUI corresponds to the common understanding of the Visual Analytics process as defined by Keim et al. [26]. Consequently, a significant share of the Visual Analytics literature addresses this TUI, e.g., clustering [31], classification [48], regression [30], dimension reduction [14], distance functions [7], and many others.

In the context of this paper, it is helpful to distinguish between inner and outer result control. The difference is whether the steering is based on intermediate results of a single execution of P , or on final results of multiple individual executions. **Inner result control** thus refers to the ability of controlling a single ongoing computation of P before it eventually returns a final result. Typical examples are partial modifications of the computation state between two consecutive iterations of P . In the KMEANS example, users could be allowed to shift, merge, or split cluster centers between iterations.

Outer result control involves multiple consecutive executions of P that do not directly re-use previous results. It imposes no requirements on the algorithm, but relies on the visualization tool to enable the discourse between the user and the computation. As stated above, our scope of the TUI is limited to the time between the start and the end of a single computation of P . Therefore, outer result control is not relevant for this paper from the point of view of algorithm design.

4 STRATEGIES FOR ACHIEVING USER INVOLVEMENT

The previous section defined types of user involvement in ongoing computations. This section describes four strategies S1 – S4 to achieve user involvement for algorithms with different characteristics. We note that our focus is on the technical applicability of these strategies for enabling *any* type of user involvement, not on the discussion when specific TUI are appropriate from an application point of view. The motivation of these strategies within this paper is achieving a tighter user involvement in integrations of interactive visualization software with computational environments, such as R or MATLAB. However, their formulation does not rely on this application context, but can be regarded as a contribution to general algorithm design regarding early user involvement.

The common key idea of the four strategies is to replace the execution of an algorithm P by a series of smaller steps $\{\tilde{P}_1, \dots, \tilde{P}_n\}$ in order to allow feedback and control between any subsequent steps \tilde{P}_i

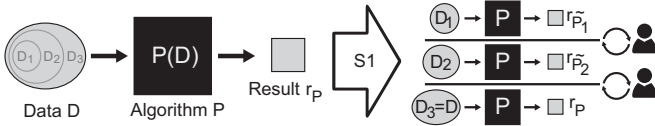


Fig. 2. S1: Data Subsetting. Additional passes of P for increasing subsets of data are computed to allow user involvement after a shorter time.

and \tilde{P}_{i+1} . The simplification of P to steps \tilde{P}_i can be achieved in several ways, which can be characterized based on two orthogonal aspects: (1) The *dimension* of simplification of P can either be the *input data*, the *parameters*, or the *algorithm* itself. (2) The *approach* of simplification along these dimensions can be realized by *subdivision* of P for divisible problems, or based on *simplified extra passes* if a subdivision is not possible. The Cartesian product of these two aspects yields six combinations which are entirely covered by our four strategies: S1 computes extra passes for simplified data, S2 computes extra passes for simplified parameters or a simplified algorithm. S3 subdivides P into subproblems with respect to data or parameters, while S4 subdivides the control flow of the algorithm as such. In this sense, we argue that our set of strategies is complete in the context of our scope, i.e., enabling user involvement during the computation of operations with bounded effort.

Motivated by the structured approach of describing and comparing design patterns in software engineering [18], we characterize each strategy in terms of the name, definition, scope, examples, considerations regarding user involvement, and computational implications.

4.1 S1: Data Subsetting

Definition. Perform computations of P for increasingly larger subsets $D_i \subseteq D_{i+1} \subseteq D$ of data records or dimensions of a data table D in *additional passes* and enable user involvement after completing every pass $\tilde{P}_i = P(D_i)$ (see Fig. 2).

Scope. S1 operates solely in the data space. As a consequence, S1 is structurally applicable to any algorithm that operates on a data table or data vector D . From an application point of view, S1 requires that intermediate results provide a meaningful approximation of the final result r_p . Specifically, this is the case for algorithms inferring a global structure like clusters, trends, and aggregation.

In contrast to strategies subdividing the workload into disjoint segments (i.e., S3), the subsetting of D does not rely on reusing results between passes. As a consequence, S1 is in particular applicable in situations where algorithms cannot reasonably be subdivided for inherent structural reasons or due to constraints imposed by their programming interface. This makes S1 the most generally applicable strategy that requires little knowledge about the inner structure of P .

However, the type of P determines whether subsetting is possible and reasonable in terms of *data records* (i.e., rows of D) or in terms of *data dimensions* (i.e., columns of D). Another important consideration is the method for defining the subsets of D , which significantly affects how representative the intermediate results are. This issue is directly related to sampling, which is discussed extensively in the literature also in context of visualization [4, 13, 28, 33].

Examples. As stated above, S1 is generally suitable for algorithms inferring a global structure or information. This includes, for example, most algorithms from unsupervised statistical learning [22]. In this case, subsetting data records may enable an early detection and potentially correction of wrong assumptions or inadequate parameters, e.g., a wrong number of clusters in the KMEANS example. Dimension reduction techniques like PCA and MDS may also benefit from subsetting of data records just as most descriptive statistics like statistical moments (e.g., mean, variance, skewness), percentiles, etc.

While the purpose of subsetting in S1 is achieving early user involvement, techniques from supervised learning may already include record-based subsetting for the purpose of model validation [22]. Depending on the purpose of the model, however, applying S1 for speedup may still be applicable, e.g., when visually indicating linear trends in a scatter plot. More care must be taken with subsetting of *dimensions* in machine learning, as the selection of features is very critical for the quality and representativity of results.

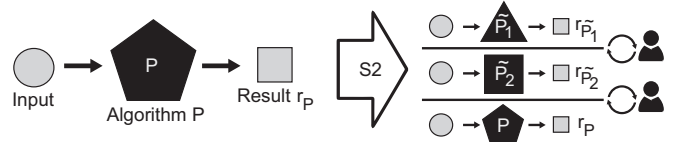


Fig. 3. S2: Complexity Selection. Additional passes of P for simplified parameters are computed to allow user involvement after a shorter time.

User involvement. The key parameters for enabling user involvement are the number and the size of the data subsets D_i . These parameters enable a tradeoff between frequency of user involvement, quality in terms of completeness of intermediate results, and computational overhead. The typically known size of D_i relative to D enables a direct quantification of the completeness in terms of the considered data size. This information should especially be conveyed as feedback regarding result provenance and can also be seen as absolute progress. Feedback concerning the relative progress with respect to the overall time, however, requires intimate knowledge of the computational complexity of P . Especially for client-driven implementations (Sec. 5.1), this may also be a major challenge for enabling user control mechanisms at an approximately equal rate.

Computational implications. The computational overhead of S1 is the sum of all $P(D_i)$, which can be very significant. On the other hand, the execution of the $P(D_i)$ and the $P(D)$ is easily parallelizable. In so far, S1 does not necessarily incur a latency for receiving the final result. In an extreme case, all $P(D_i)$ as well as $P(D)$ are scheduled independently, loosely relying on the increasing effort for computing increasing percentages of D to arrive in order. The memory consumption, however, typically also increases with the degree of parallelization due to a multiplication of algorithm-internal structures. Regarding the storage of D_i , indexing of D should be used whenever possible to avoid data duplication and reduce data transfer.

4.2 S2: Complexity Selection

Definition. Perform computations of P for less complex parameter configurations \tilde{P}_i in *additional passes* before computing P itself, and enable user involvement after completing every pass \tilde{P}_i (see Fig. 3).

Scope. S2 operates in the parameter space of P . Therefore, the applicability of S2 is determined by the existence and accessibility of complexity parameters that enable a speed vs. quality tradeoff. In particular, this applies to approximation algorithms [49] and many heuristics of computationally intractable problems in operations research, but also to many algorithms in other fields like statistics (see below).

In contrast to S1 that operates in data space, the application of S2 is very dependent on P and typically requires structural knowledge of P and the effect of parameter changes in context of the specific data. This is a highly non-trivial issue in general as also shown by the growing importance of parameter space analysis as a topic in visualization literature [41]. In particular, the purpose of many complexity parameters in statistics is to adjust the suitability for particular data and a particular purpose rather than to simply trade off quality versus speed. An example is the bias vs. variance tradeoff of many types of statistical models [22], where additional complexity improves the model quality only to a certain point before degrading generalizability due to over-fitting. As a consequence, S2 should only be considered for algorithms where concluding from intermediate results $r_{\tilde{P}_i}$ to the final result r_p is meaningful.

On the other hand, S2 does not require any structural decomposability of P , as it is the case for S3 and S4. In contrast to S1 which requires vector- or tabular-oriented data, S2 is also applicable to algorithms working on non-decomposable operands like analytical functions.

Examples. In operations research, approximation algorithms for computationally intractable problems are common. They provide a solution that is provably optimal up to a constant – and often definable – factor and have provable run-time bounds [49]. For nearest neighbor search, for example, ϵ -approximate variants exist that enable to trade off the probability of finding the true nearest neighbor versus space and time costs especially in high-dimensional spaces (e.g., Arya et al. [2]).

Further examples of complexity parameters include the refinement

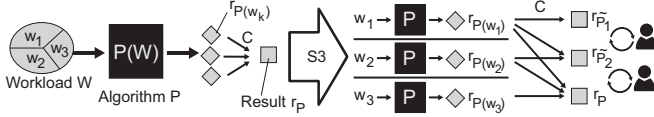


Fig. 4. S3: Divide and Combine involves (1) applying P to independent parts w_k of a workload W , and (2) recombining the results using a combination C to obtain intermediate results or the final result.

of subdivision schemes, the size of search radii, thresholds for stopping criteria, or even the algorithm itself as long as different algorithms yield structurally equivalent results, which includes most heuristics. Many algorithms for feature subset selection (SUBSETS), for example, differ in whether they perform a greedy or an exhaustive search.

User involvement. The key parameters for enabling user involvement are the number of approximation steps and the complexities of the steps. Unlike for S1, the degree of freedom for both parameters is determined by P . Quantitative complexity parameters like thresholds may enable a precise choice of the number of approximations and even a quantification of the completeness or precision (e.g., for estimating relative progress). Conversely, categorical parameters such as available heuristic algorithms may impose a strict limitation on the number and complexities of steps, and complexities may be hard to estimate or even order. This makes a quantification of the progress difficult or impossible in general and only enables qualitative feedback regarding progress and result provenance. While feedback regarding result provenance is essential especially for strategy S2, however, only expert users will often be able to interpret the information. As for S1, another challenge for client-driven implementations (Sec. 5.1) will typically be to enable user control mechanisms at an approximately equal rate.

Computational implications. The computational overhead of S2 is the sum of additional passes for computing the steps \tilde{P}_i . However, these computations are independent and thus easily parallelizable. As discussed for S1, this enables to reduce or avoid the latency for receiving the final result r_P at the cost of increasing memory consumption.

4.3 S3: Divide and Combine

Definition. Subdivide a workload W into n disjoint parts $\{w_1, \dots, w_n\}$, apply P independently to each part w_k to generate *partial results* $\{r_{P(w_1)}, \dots, r_{P(w_n)}\}$, and compute intermediate results or the final result based on combining some or all $r_{P(w_k)}$ (see Fig. 4).

Scope. S3 imposes two requirements on P : First, independent applications of P to parts w_k of a subdivided workload W must be possible in order to generate the partial results $r_{P(w_k)}$. Second, a meaningful combination C must exist to combine subsets of partial results to *intermediate results* $r_{\tilde{P}_i}$ that are structurally equivalent to the final result r_P . In particular, applying C to *all* partial results yields the final result r_P of P . In context of S3, a single step \tilde{P}_i thus comprises the computation of a subset of partial results and the application of C to them.

The subdivision of W can be defined in terms of *data* (i.e., data space-based) or *parameters* (i.e., parameter space-based) that P is applied to. A data space-based subdivision is specifically possible in cases where applying P to a collection of elements (e.g., a set, a vector, a matrix) internally involves applying the same operation to each element. A parameter space-based subdivision is applicable to algorithms that take a specification of a domain (e.g., the extents of a search space) as a parameter, given that disjoint parts of the domain can be processed independently. It should be noted that a disjoint subdivision of the workload W does not in all cases imply a disjoint subdivision of the data or parameter space considered by each $P(w_k)$ for computation. In other words, the disjoint subdivision applies to the *output* of P rather than the *input* of P . Tolerating a certain overlap in the inputs of multiple $P(w_k)$ extends the applicability of S3 to operations that require a specified context around each processed element, e.g., for convolution or pattern search.

The characteristics of the combination C depend on the structure of the result r_P . In many cases, C is a composition of partial results in order to restore their context within W which has been lost due to the initial subdivision. Sometimes, C may also be a simple aggregation (e.g., maximum or mean). In any case, a practical requirement is that

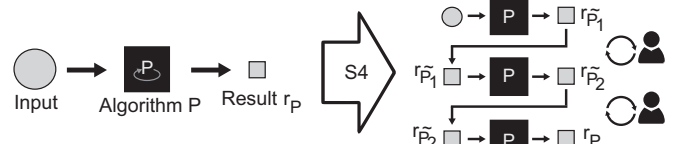


Fig. 5. S4: Dependent subdivision. The idea is to involve the user between sequentially dependent steps of algorithms P , e.g., iterations.

an application of C should be cheap to perform compared to computing the partial results themselves. The reason is that the intermediate results $r_{\tilde{P}_i}$ are composed of arbitrary and non-disjoint subsets of the partial results (see Fig. 4). C may thus be applied to a single partial result multiple times for generating different intermediate results.

In order to avoid confusion, we point out that S3 is related and often applicable yet not equivalent to divide and conquer (D&C) algorithms [6]. For D&C algorithms, the subdivision is an inherent property of the algorithm while S3 refers to a subdivision-based strategy in a broader sense. In particular, S3 does not require that the application of P to a single element $P(w_k)$ becomes trivial. In this respect, S3 is more related to parallelization paradigms like MapReduce [12].

Examples. An example for data space-based S3 is the sampled evaluation of a function, e.g., for progressive rendering of increasingly fine-grained function graphs. In this case, P is the evaluation of the function for a set of positions, W are the positions of all samples, w_k is a certain subset, and C restores the context (i.e., the position and order) of results of P within W . Another example is the progressive computation of aggregates, e.g., the average [17]. In this case, P may involve potentially optimized algorithms for computing the aggregate for blocks of data w_k and C further aggregates multiple $r_{P(w_k)}$ according to their cardinality.

A parameter space-based example is the computation of the autocorrelation of a time series. The parameter refers to the interval (W) of considered lags. This interval can be separated in disjoint parts and C recomposes the autocorrelation as a function of the lag-size.

User involvement. The two key characteristics of S3 are the degree of subdivision (*DIVIDE*, i.e., the number n of w_k), and the strategy to generate the intermediate results $r_{\tilde{P}_i}$ (*COMBINE*). In general, *DIVIDE* is the more decisive factor for execution feedback and control while *COMBINE* determines the frequency and quality of result feedback. An approximately uniform subdivision of W facilitates feedback regarding the relative progress as compared to S1 and S2 and also enables user control like cancellation at a roughly equal rate.

Regarding result feedback, *COMBINE* defines the ordering of the computations for all parts w_k , and the amount of additional completeness for each intermediate result $r_{\tilde{P}_i}$. As the number of intermediate results $r_{\tilde{P}_i}$ is independent of *DIVIDE*, the rates for feedback and control may be different. It is therefore a possible strategy to internally decouple the processing of *COMBINE* from *DIVIDE* (e.g., by multi-threading), and to define the progress for each \tilde{P}_i in terms of additional time rather than W by applying C to all already completed $r_{P(w_k)}$.

Computational implications. Increasing the degree of subdivision *DIVIDE* enables more fine-grained execution feedback and control without inherently incurring higher costs for obtaining the final result $r_{P(W)}$. In practice, however, each application of P may involve a certain overhead for reasons including the internal structure of P , potentially overlapping inputs of multiple $P(w_k)$, and implementation-related issues (e.g., data transfer, initialization, etc.). The latter are typically more significant for client-driven implementations (see Sec. 5.1). In addition, the overhead of S3 includes generating intermediate results from partial results and thus depends on *COMBINE*.

The independence of all $P(w_k)$ makes S3 suitable for performing computations in parallel. In general, parallelization is typically a key motivation for subdivision. In the context of user involvement, however, a certain degree of sequential scheduling is required in order to involve the user between independent subsets of workload parts.

4.4 S4: Dependent Subdivision

Definition. Subdivide P into sequentially dependent steps \tilde{P}_i so that the result $r_{\tilde{P}_i}$ of each step is an input to the next step \tilde{P}_{i+1} , and is structurally equivalent to the final result r_P . Enable user involvement between steps (see Fig. 5).

Scope. S4 poses requirements regarding the decomposability of P and the structural equivalence of the result of each step $r_{\tilde{P}_i}$ to r_P . In particular, this includes iterative algorithms where the result of each iteration serves as input to the next. In addition to inherently iterative problems, multiple problems can directly be transformed to iterative problems (e.g., recursive problems [6]), or an iterative variant exists (e.g., iterative PCA [39]). While iterative algorithms are the by far most important example of S4, the sequential steps could also be defined in terms of an ordered domain that needs to be processed sequentially, e.g., progressive signal reconstruction as described below.

In contrast to S1 and S2, each step \tilde{P}_i can reuse the previous result $r_{\tilde{P}_{i-1}}$ to avoid redundant computation. In contrast to S3, each step \tilde{P}_i depends on the previous step \tilde{P}_{i-1} , i.e., it is not possible to decompose the workload into independent parts.

Examples. S4 is in particular applicable to inherently iterative algorithms. In statistical learning, prominent examples include (1) the training of regression or classification models such as neural networks, (2) dimension reduction algorithms like multi-dimensional scaling, and (3) clustering algorithms such as partitioning around medoids or the recurring example KMEANS. Other examples are force-based algorithms for graph layout [29], as well as algorithms that – potentially recursively – build hierarchical structures (e.g., decision trees), where each recursion adds to the complexity of $r_{\tilde{P}_i}$.

Concerning sequential processing of an ordered domain, consider a progressive reconstruction of a signal (e.g., a time series or an image) from a frequency-based representation, as common for displaying large JPEG images. An implementation of S4 could define \tilde{P}_i as to reconstruct a certain disjoint band of increasingly higher frequencies and to add the result to the already reconstructed part of the signal.

User involvement. The key parameter for enabling user involvement is the step size, denoted by s . Varying s enables to trade off the frequency of feedback and user control against the computational overhead involved with each step. For iterative algorithms, s is typically defined in terms of iterations which enables user involvement at an approximately equal rate. Whether relative feedback can reasonably be provided depends on whether the number of steps is known in advance. As this often does not apply to convergent algorithms, a distance from a termination criterion may be provided instead.

In contrast to all other strategies, each step \tilde{P}_i depends on the result of the previous step \tilde{P}_{i-1} for S4. We argue that this is a requirement for permitting meaningful control of the ultimate result r_P within an ongoing computation of P , i.e., enabling inner result control (see Sec. 3.4). For S1 and S2, changing data or parameters typically requires to restart computing all intermediate results, beginning with the simplest step. For S3, obtaining a homogeneous final result r_P requires that each step $P(w_k)$ is computed in the same way for all independent workload parts w_k . For these reasons, outer result control is more appropriate when applying S1, S2, or S3. In contrast, for iterative algorithms, inner result control can be reasonable to enable domain knowledge for affecting convergence (e.g., avoiding local extrema).

Computational implications. As discussed for the degree of subdivision of S3, the step size s may have a practical effect on the computational overhead imposed by S4. Unlike for S3, however, the sequential dependence of the \tilde{P}_i on each other does not permit parallelization.

5 CLIENT-DRIVEN VS ALGORITHM-DRIVEN IMPLEMENTATION OF STRATEGIES

The previous sections characterized types of user involvement in ongoing computations (Sec. 3) and described four strategies to achieve user involvement for algorithms with different characteristics (Sec. 4). This section discusses possibilities of realizing the strategies when integrating interactive visualization software and computational environments. We will refer to these environments as VIS and COMP, where

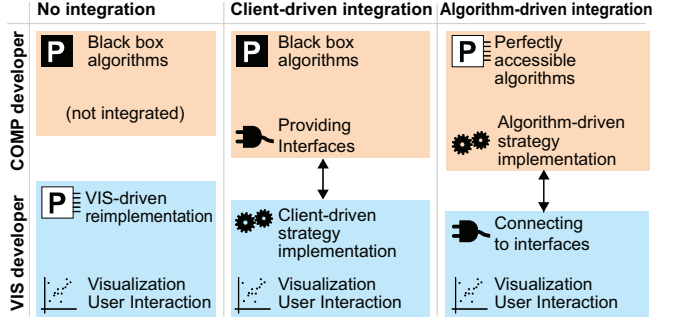


Fig. 6. Reimplementing algorithms P in VIS allows to achieve user involvement but involves substantial effort for VIS (left). In client-driven integration, VIS implements strategies for achieving user involvement by connecting to existing P that provide adequate interfaces (middle). In algorithm-driven integration, VIS connects to algorithms that implement strategies and provide communication directly from within (right).

VIS can be any interactive visualization tool and COMP can be computational environments such as R or MATLAB, as well as any other external computational resource or library.

We discriminate between three responsibilities:

Algorithm refers to performing the computation of P or \tilde{P}_i .

Client refers to visualizing the feedback of *Algorithm* and the handling of user input, i.e., the elements for human-computer interaction.

Flow control refers to implementing the control flow and communication between *Algorithm* and *Client*. This includes the implementation of a strategy for defining and scheduling the steps \tilde{P}_i .

Currently, a frequent situation in Visual Analytics is that all three responsibilities, i.e., *Algorithm*, *Client* and *Flow control* are implemented as part of VIS (see Fig. 6, left column), which has disadvantages as pointed out in related work [15] and in the introduction. In context of integrating VIS and COMP, however, we assume that the role of the *Algorithm* is provided by COMP and the role of the *Client* is taken by VIS. In this case, the *Flow control* can either be a responsibility of VIS (*client-driven integration*, center column in Fig. 6), or a responsibility of COMP (*algorithm-driven integration*, right column in Fig. 6). Characterizing, comparing, and discussing these two scenarios is the purpose of this section. We establish requirements imposed by client- and algorithm-driven integration, and we derive guidelines to the design of the interface of P in favor of flexible client control.

5.1 Client-driven integration

In client-driven integration, the definition and scheduling of the \tilde{P}_i is managed by VIS, while their computation is performed by COMP. Between any steps \tilde{P}_i and \tilde{P}_{i+1} , VIS can realize user involvement, e.g., by visualizing $r_{\tilde{P}_i}$ or by adjusting the call of \tilde{P}_{i+1} according to user input. This *externalization of the control flow* requires that the programming interface of P exposes all parameters that clients need to define \tilde{P}_i . We subsequently analyze these requirements for each strategy.

In S1, the \tilde{P}_i are defined as executions of P on subsets of the input data D . Externally produced subsets can be fed to P instead of the full D , making S1 applicable for client-driven integration without additional requirements. In S2, a client-driven selection of complexity involves calling P for different parameters. This yields the following interface Requirement for a Client-driven integration (RC) for S2:

RC 1. *Expose complexity parameters to trade off speed for quality.*

Considering the central role of most complexity parameters, this criterion is not very limiting in practice (see Sec. 6). For S3, clients need to define a part of the subdivided workload W for each call of P . To support S3, the interface of P must meet the requirement RC2:

RC 2. *Enable a precise specification of the processed workload.*

For data space-based applications of S3, a subdivision into coherent blocks is often possible on the client side. Parameter space-based applications of S3 require the possibility to fully specify boundaries via the interface, e.g., the lower and upper limit of a considered subspace.

Requirement RC2 also holds for S4, as the sequential processing of iterations is also based on a decomposition of workload. The specification can either be explicit, i.e., the number of iterations performed in \tilde{P}_i , or implicit, by means of a stopping criterion.

A second requirement of S4 is the ability to pass the final state of \tilde{P}_i on to \tilde{P}_{i+1} . We identify RC3 for S4 as follows:

RC 3. *Provide access to all parts of the state as output, and accept equivalent information as input, in order to enable resuming the computation with minimal redundancy.*

The form of this state information depends on P . For statistical learning, the state is often the model itself (e.g., a regression model or cluster centers). While the model is typically the output of such P , not all interfaces support accepting a model as an input to proceed with.

It should be noted that not all strategies are appropriate for every algorithm. Given that a particular strategy is appropriate for P , algorithm developers should account for those RC that are required by this strategy in order to enable an appropriate degree of user involvement. Regarding execution feedback, the arrival of steps \tilde{P}_i and their definition can be reported for aliveness and absolute progress. Regarding result feedback, the client may directly visualize intermediate results $\tilde{r}_{\tilde{P}_i}$ or use any output of steps \tilde{P}_i to derive information such as quality metrics as a post-processing step in VIS. Regarding execution control, considering user input in between enables to call a specific \tilde{P}_{i+1} for prioritization, or not at all for premature cancellation. Regarding result control, \tilde{P}_{i+1} can be called for modified inputs.

Client-driven control enables several possibilities of realizing parallelization of steps \tilde{P}_i . For S1, S2, and S3, multiple invocations of P can be parallelized using multiple threads within COMP, multiple instances of COMP, or even different computers in network- or cloud-based environments. In practice, however, the incurred latency may exclude some options regarding responsiveness for user involvement.

As an inherent limitation of client-driven integration, user involvement is only possible between steps \tilde{P}_i , i.e., invocations of P . This may limit the achievable frequency of user involvement as opposed to a reimplementing of P .

5.2 Algorithm-driven integration

In algorithm-driven integration, the *Flow Control* is realized directly within the implementation of P . Specifically, P is responsible for defining simplification steps \tilde{P}_i according to a particular strategy, and for communicating with the *Client* in order to enable user involvement.

Definition of the simplification steps. When defining the steps \tilde{P}_i , four objectives can be identified for different TUI: (1) Execution feedback should be provided as precisely as possible and at approximately equal rates. (2) Result feedback should provide good approximations of r_P as early as possible. (3) Both execution control and result control should have a minimal latency. (4) The computational overhead of user involvement should be minimal.

These objectives are partly contradicting each other. Defining the steps \tilde{P}_i represents a mechanism to control this tradeoff for optimization within a given context. While the four objectives generally apply to client-driven as well as algorithm-driven integration scenarios, typically the client knows their preference in context. For algorithm-driven integration scenarios, it is thus desirable that the client has means of controlling the definition of steps \tilde{P}_i via the interface of P . However, a direct definition of steps often requires an intimate knowledge about the inner structure of P . In this sense, an algorithm-driven *Flow Control* provides two key advantages over client-driven integration:

First, the implementation of P is the more appropriate place to be aware of the inner structure and any implications than the client. Ideally, the client can specify the preference of the objectives and certain constraints (e.g., a minimal frequency of feedback) while the algorithm knows *how* to realize this specification. Based on this consideration, we formulate the following **Guideline** for the design of P 's interface in the context of Algorithm-driven integration (*GA*):

GA 1. *Offer means for specifying preference and constraints by the client regarding desired feedback and control rates.*

A second advantage of algorithm-driven integration is that communication is not limited to the times between structurally equivalent \tilde{P}_i .

For example, execution control can be realized after arbitrary blocks of code, allowing to check for cancellation signals often without having to generate result feedback at the same rate. This *source-code level of granularity* also allows minimizing the overhead of executing multiple steps \tilde{P}_i instead of a single P , as the products of potential common initialization steps can be reused.

Communication with the Client. In algorithm-driven integration, the extent of supported feedback and control is entirely up to P . This makes sense, as appropriate types of user involvement are strongly algorithm-dependent. On that account, it is a key goal of this paper to encourage algorithm developers to acknowledge the degree of supported user involvement as a conscious design choice.

The exchange of information between P and VIS can be implemented in different ways. A simple feedback mechanism commonly found in command line-based computation environments is providing a textual *trace* of the ongoing computation to a console. This one-directional form of communication is usually intended to be read directly by users, not clients like VIS. As a result, parsing the trace may be difficult and highly algorithm-specific. As it is intended for console display, larger amounts of data can not be communicated reasonably.

A more flexible option is the definition of an interface by the algorithm for sending feedback to and querying control information from an unknown client during the computation. Technically, a broad set of communication techniques exists, including the registration of call back procedures, dedicated points of code insertion, registration mechanisms implementing the Observer design pattern [18], message passing and application-layer network protocols.

As a common guideline in software engineering, we argue for a separation of concerns in that algorithm implementations should need to care more about *what* to communicate and *when*, rather than about *how* and to *whom*. Details of the communication such as the number and location of clients, or issues like parsing protocols should be decoupled from the actual implementation of P in order to minimize the implementation effort of algorithm developers and to maximize the reusability of an implementation in various environments. We see two options to achieve this:

The first option is pragmatic in the sense that the algorithm developers should support the communication technique that incurs the minimal effort on their side. Translating one of the aforementioned techniques to another is typically possible and requires an Adapter [18] which can – and should – be realized outside the algorithm, for example by the client developer. In many cases, the most simple technique is providing means for registering *callback methods* in the same programming language as the algorithm implementation. This inversion of control enables a single client to insert code into P that is called at semantically meaningful positions of the control flow for exchanging feedback and control signals, e.g., between iterations. P does not need to know the client but executes callbacks as a black box. A direct use by clients may pose certain challenges, such as different languages of VIS and COMP, or requiring VIS and COMP to be executed on the same machine. However, as argued above, Adapter objects can be defined to address these challenges, e.g., by translating local calls to Remote Procedure Calls (RPC). We thus suggest the following interface guideline as a pragmatic step towards separation of concern:

GA 2. *Provide a callback interface to allow a client-side customization of the communication protocol.*

The second option of the algorithm developer is to rely on an existing communication infrastructure, which may be external or internal. External refers to libraries and middleware outside the environment of COMP, e.g., for message passing. While this typically enables more powerful communication possibilities (e.g., over a network), a disadvantage for clients could be to incur the communication infrastructure as potentially unwanted dependency. In contrast, an internal infrastructure refers to a dedicated extension of the COMP environment itself (e.g., MATLAB or the R core) that algorithms and clients can use for benefit without additional complexity or dependency. However, such extensions are typically not provided today. *We thus recommend to realize powerful and easy-to-use communication mechanisms*

Algorithm (package)	Description	Operates on table	Exposes complexity param. (RC1)	Definable workload (RC2)	State restorability (RC3)	Provides communication from within	Comm. granularity (GA1)	Allows callbacks (GA2)
tsne (tmsne)	T-SNE dimension reduction	✓	✓	✓	✓	✓ (trace+GA2)	✓	✓
neuralnet (neuralnet)	Neural network	✓	✓	✓	✓	✓ (trace)	✓	✗
optim (stats)	Optimization	-	✓	✓	✓	✓ (trace)	✓	✗
sammon (MASS)	Multi-dimensional scaling	✓	✓	✓	✓	✓ (trace)	✗	✗
vegas (R2Cuba)	Monte Carlo Integration	-	✓	✓	✓	✓ (trace)	✗	✗
kmeans (cluster)	K-means clustering	✓	✓	✓	✓	✗	✗	✗
som (kohonen)	Self organizing map	✓	✓	✓	✓	✗	✗	✗
emcluster (EMCluster)	Expectation max. clustering	✓	✓	✓	✓	✗	✗	✗
rpart (rpart)	Recursive tree construction	✓	✓	✓	✓	✗	✗	✗
regsubsets (leaps)	Best subset feature selection	✓	✓	✓	✓	✗	✗	✗
biglm (biglm)	Linear model	✓	✓	✓	✓	✗	✗	✗
pam (cluster)	Partitioning around medoids	✓	✓	✗	✓	✓ (trace)	✗	✗
acf (stats)	Autocorrelation	✓	✓	✗	✗	✗	✗	✗
ksvm (kernlab)	Support vector machine	✓	✓	✗	✗	✗	✗	✗

Table 1. A survey of frequently used R algorithms regarding the fulfillment of the identified requirements and guidelines in favor of a tight integration.

between algorithms and clients as valuable future core extensions of widely-used COMP environments.

A key consideration of respective extensions refers to their level of abstraction. Low-level mechanisms do not directly support any semantics of the communicated information but rely on the end points to do so. Conversely, high-level mechanisms could directly provide support for specific types of user involvement. For execution feedback and control (e.g., cancellation), this seems rather straightforward. For result feedback and control, however, defining standardized means seems highly non-trivial, but would enable benefits like querying intermediate results of different algorithms transparently to the client. While this may be a too demanding step for current computation environments, considerations like these could be a starting point for designing new computation infrastructures, as suggested by Fekete [15].

6 CASE STUDY “R”: APPLICABILITY OF STRATEGIES

In the previous section, we identified a set of requirements and guidelines for the design of algorithmic interfaces in favor of an applicability of the proposed strategies. In this section, we investigate to which degree an exemplary computational environment fulfills the requirements of client-driven integration or even actively supports user involvement in the sense of algorithm-driven integration. Specifically, we surveyed 14 common algorithms for important problems related to multivariate analysis from the scripting environment R. The selection of R was motivated by its broad acceptance in academia and corporate research, the choice of algorithms was inspired by R’s reference list of recommended packages for common topics, *CRAN Task Views*⁶.

Table 1 gives an overview of the survey results. The table suggests that the large majority of the inspected algorithms supports a client-driven application of multiple strategies, while only a few of them directly provide algorithm-driven feedback. This indicates that there is currently a large potential of realizing user involvement at the hands of VIS developers, as well as potential for COMP developers to support user involvement more directly.

The data-based S1 is applicable to all algorithms operating on a data table. This applies to all our examples except for optimization (*optim*) and monte-carlo integration (*vegas*), which take analytic functions as inputs. Also, all investigated algorithms expose some complexity parameter or method selector that influences the runtime of single steps (RC1). For example, the *pamonce* option enables algorithmic short cuts in Partitioning-around-medoids clustering (*pam*), and best-subset feature selection (*regsubsets*) offers a selector of exhaustive vs. stepwise methods. Furthermore, the majority of investigated subdividable algorithms fulfills the interface requirements of strategies S3 and S4 (RC2, RC3).

However, not all surveyed algorithms fulfill RC1 - RC3, which allows us to discuss potential interface improvements for specific real-world examples. Note that this discussion is neither an assessment of the algorithms themselves nor their specific implementations.

⁶<http://cran.r-project.org/web/views>

Example 1: The clustering method *pam* iteratively performs an exhaustive search of medoids, i.e., data records that exhibit a minimal sum of distances to all other records. While *pam* allows the specification of an initial set of medoids (RC3), it is not possible to subdivide iterations into separate calls (RC2). Adding a numeric parameter indicating the number of iterations to perform in each step would enable users to suggest cluster medoids in-between, in order to speed up convergence for large datasets as well as to avoid local minima.

Example 2: The iterative training of support vector machines as provided by *ksvm* does not expose the divisibility of the underlying Sequential Minimal Optimization [37]. However, the usefulness and convergence of SVMs highly depends on the choice of multiple model parameters. We suggest to enable a specification of the number of iterations and the previously trained model as input parameters of *ksvm*. This would allow early previews and cancellation of the model identification for an exploration of model parameters.

Example 3: Computing the autocorrelation function of a time series (*acf*) can be seen as a divide-and-combine approach of computing correlations between a time series T and different lags of T . While *acf* allows a specification of the longest computed lag (*lag.max*) in the sense of RC1, it lacks the counterpart *lag.min* needed for a workload specification according to RC2.

After surveying the examples regarding the client-driven applicability of strategies, we now discuss the direct support of user involvement as provided by algorithms. Several algorithms provide a uni-directional *trace* of textual feedback to a console during their execution (*pam*, *vegas*, *sammon*, *optim*, *neuralnet* and *tsne*). Most of them allow specifying different levels of verbosity, while *tsne*, *neuralnet* and *optim* even allow specifying the interval between messages (GA1). Apart from this trace, one example comes close to the *perfectly accessible algorithm* as outlined by algorithm-driven integration: The iterative *tsne* algorithm for dimension reduction allows clients to define a callback (GA2) that is executed instead of printing the trace at regular, client-definable intervals (GA1). This enables flexible feedback in a consistent way.

However, we found no algorithms that consider *control* signals during their execution. A possible explanation could be that R usually runs in single-threaded, stand-alone command line environments, where the receiving of concurrent control signals is practically not feasible. With callbacks at hand, however, algorithms could incorporate control by considering the return value of callbacks in their control flow. As long as measures like this have not been adopted, providing user control is possible by implementing S1-S4 on the client-side.

This case study shows that very few of the examined R implementations directly provide intermediate feedback in a consistent way, and none of them directly supports intermediate control. This confirms the necessity of external means such as client-driven strategies when integrating VIS with R for visual exploration. On the upside, all surveyed algorithms fulfill the requirements of at least one client-driven strategy. The fact that there are good as well as bad examples shows that integrability lies at the hands of the single COMP developer, even

without the availability of standardized communication protocols.

7 DISCUSSION AND FUTURE WORK

This paper is intended to show individual developers in the VIS and COMP communities practical measures of supporting integrability on their end. We agree with previous work [15] that the development of algorithms that directly provide standardized communication would be highly desirable in this context, as it allows reuse and minimizes effort for the VIS community. However, agreeing on protocol standards and implementing them for existing P is tedious, and putting the full load on the shoulders of COMP developers is not reasonable. The client-driven application of S1-S4 can be seen as a practicable alternative that allows VIS developers to achieve user involvement for a large number of existing implementations. Adhering to interface requirements in favor of client-driven integration is a more manageable first request to COMP developers than providing *perfectly accessible algorithms*.

Fekete has identified two key limitations of current integrations between VIS and COMP for the purpose of exploration [15]: First, “*algorithms provided by analytical environments are not designed for exploration and make no effort in providing early results quickly to the analyst*”. Our paper directly addresses this issue, as the characterized strategies and resulting guidelines pave the way for tighter integrations that support user involvement during computations. As the second issue, Fekete states that “*when data is large [...] transfer time itself exceeds the reactivity requirement*” [15]. This issue is further aggravated by the exchange of intermediate signals. However, many forms of intermediate communication are substantially smaller than the regular inputs or outputs of P , e.g., the cluster centers in KMEANS. Apart from data size, the severity of this limitation in practice depends on infrastructural aspects of the integration that are beyond the scope of this paper. Examples include *network-based vs. memory-based communication, same machine vs. different machine in LAN / Internet, stateless vs. workspace-based COMP, internal data source vs. tertiary database*, as well as *overheads incurred the internal data format of COMP*. As our discussion does not cover these aspects as such, we demonstrate in the following that the presented strategies and integration scenarios can work for moderately large datasets.

As an initial proof of concept, we implemented four common integration scenarios by connecting our VIS environment *Visplore* [30, 35, 36] to R and MATLAB: We integrated *Visplore* with (1) the c-based R-API as part of the *Visplore* process [24], (2) the COM interface of the MATLAB engine in a different process (3) the RServe package via TCP running on the same PC⁷ as *Visplore*, and (4) RServe running on a different PC⁸ via Gigabit LAN. Table 2 reports timings of transferring arrays of randomized double precision values from VIS to COMP. Timings for the other direction, i.e., COMP to VIS, were equivalent in this measurement. As a second experiment, we implemented the client-driven versions of S1 and S4 for the R-method k_{means} based on the local API integration of R. The input of a 20-dimensional table of random data records is transferred to the R-workspace once, while cluster labels for each record are returned to *Visplore* after every step \tilde{P}_i . Table 3 states average timings of early result availability for varying numbers of data records (S1) as well as percentages of the full iteration count (S4), for $k = 20$ clusters. The intention of these tests is to show that data transfer can be sufficiently fast for data sizes commonly found in real world analyses. Especially in local integrations, computation times are often the more limiting factor.

Array size	R API, local	MATLAB, local	RServe, local	RServe, LAN
100 MB	0.017s	0.254s	0.476s	0.883s
1024 MB	0.171s	2.663s	5.018s	8.510s

Table 2. Timings of transferring arrays of double precision random values between *Visplore* and COMP environments using different integration scenarios. Measurements were averaged across 10 repetitions.

While most examples in this paper stem from the field of multivariate analysis, the discussed TUI and strategies are generalizable

⁷Windows PC, Intel Xeon E3-1245 V2 CPU @ 3.4 Ghz, 16GB RAM

⁸Windows Notebook, Intel i7-3612QM CPU @ 2.1 Ghz, 8GB RAM

Number of rows	2 iterations	5 iterations	10 iterations	20 iterations
20,000	0.105s	0.256s	0.490s	0.845s
200,000	1.594s	4.041s	8.728s	19.256s
2,000,000	19.744s	59.244s	128.209s	291.869s

Table 3. Timings of the availability of \tilde{r}_{P_i} in *Visplore* when executing k_{means} on a 20-dim. dataset of random numbers for $k = 20$ clusters in R. *Visplore* and R are executed on the same PC using the c-based API of R. Measurements were averaged across 10 repetitions.

to many algorithms in other disciplines. However, there are contexts where users will not consider all TUI as desirable. While result control might counteract the reproducibility of results in some cases, early result feedback might as well be unfamiliar to users that are accustomed to “seeing precise figures” [17]. In such cases, the potentially large overheads of executing additional steps \tilde{P}_i might be particularly painful if these resources could have been used to execute P as a black box more quickly. Finally, approximate solutions often introduce the need for explicit encoding of incompleteness and uncertainty, which increase the complexity of drawings and may even confuse users unfamiliar with such techniques.

We see multiple directions for future work: (1) We plan to implement client-driven integration strategies for different algorithms within *Visplore*, in order to evaluate them in the context of real-world tasks. (2) While our discussion of realizing client-driven strategies assumed the client to be VIS, we intend to investigate implementing it as an autonomous piece of reusable middleware. (3) To enable a more general assessment of the applicability of strategies, we also intend to extend our survey to include additional COMP environments like MATLAB or Python.

8 CONCLUSION

In this paper we characterized possibilities of achieving a tight integration between computational environments and visualization software. We laid the ground by a structured characterization of needs for user involvement in ongoing computations. Based on this classification, we formalized and described strategies to realize these needs for algorithms of different characteristics. A detailed discussion of considerations for client-driven and algorithm-driven implementations enabled us to identify guidelines to algorithmic interfaces which we evaluated based on a survey of common algorithms of the software R.

The combination of automated analysis techniques with interactive visualization is the key idea of Visual Analytics [25]. In this sense, we see our work as contribution on multiple levels. On a theoretical level, the formalization and comparison of technical strategies to achieve user involvement is a contribution to the theoretical foundations of Visual Analytics. On a practical level, we believe that the described implementation considerations facilitate an adoption for numerous integration scenarios based on existing computation environments. On a community level, we hope that the identification of specific requirements and guidelines for client-driven and algorithm-driven implementations fosters the development of computational infrastructures which are better suited to the needs of visual exploration.

ACKNOWLEDGMENTS

This work has been supported by the Austrian Funding Agency (FFG) within the scope of the programs COMET K1 and ICT of the Future (project nr. 840232), as well as the FWF-funded project P24597-N23 (VISAR). Thanks go to Clemens Arbesser for valuable discussions, and to Johanna Schlereth for help with the figures.

REFERENCES

- [1] M. Angelini and G. Santucci. Modeling Incremental Visualizations. In *Proc. of the EuroVis Workshop on Visual Analytics (EuroVA '13)*, pages 13–17. Eurographics Association, 2013.
- [2] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An Optimal Algorithm for Approximate Nearest Neighbor Searching Fixed Dimensions. *Journal of the ACM*, 45(6):891–923, 1998.
- [3] E. Bertini and D. Lalanne. Investigating and Reflecting on the Integration of Automatic Data Analysis and Visualization in Knowledge Discovery. *ACM SIGKDD Explorations Newsletter*, 11(2):9–18, 2010.

- [4] E. Bertini and G. Santucci. Give Chance a Chance: Modeling Density to Enhance Scatter Plot Quality Through Random Data Sampling. *Information Visualization*, 5(2):95–110, 2006.
- [5] E. Bertini, A. Tatu, and D. Keim. Quality Metrics in High-Dimensional Data Visualization: An Overview and Systematization. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.
- [6] G. Brassard and P. Bratley. *Fundamentals of Algorithmics*. Prentice-Hall, 1996.
- [7] E. Brown, J. Liu, C. Brodley, and R. Chang. Dis-Function: Learning Distance Functions Interactively. In *Proc. of the IEEE Conference on Visual Analytics in Science and Technology*, pages 83–92. IEEE, 2012.
- [8] S. Callahan, L. Bavoil, V. Pascucci, and C. Silva. Progressive Volume Rendering of Large Unstructured Grids. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):1307–1314, 2006.
- [9] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The Information Visualizer, an Information Workspace. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, page 181–186. ACM, 1991.
- [10] J. Choo, C. Lee, C. Reddy, and H. Park. UTOPIAN: User-Driven Topic Modeling Based on Interactive Nonnegative Matrix Factorization. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):1992–2001, 2013.
- [11] R. Crouser and R. Chang. An Affordance-Based Framework for Human Computation and Human-Computer Collaboration. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2859–2868, Dec. 2012.
- [12] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1):107–113, 2004.
- [13] A. Dix and G. Ellis. by chance: enhancing interaction with large data sets through statistical sampling. In *Proc. of the Conference on Advanced Visual Interfaces (AVI '02)*, page 167–176. ACM, 2002.
- [14] A. Endert, C. Han, D. Maiti, L. House, S. Leman, and C. North. Observation-Level Interaction with Statistical Models for Visual Analytics. In *Proc. of the IEEE Conference on Visual Analytics in Science and Technology (VAST '11)*, pages 121–130. IEEE, Oct. 2011.
- [15] J.-D. Fekete. Visual Analytics Infrastructures: From Data Management to Exploration. *Computer*, 46(7):22–29, 2013.
- [16] D. Fisher. Incremental, Approximate Database Queries and Uncertainty for Exploratory Visualization. In *Proc. of the IEEE Symp. on Large Data Analysis and Visualization (LDAV '11)*, pages 73–80. IEEE, 2011.
- [17] D. Fisher, I. Popov, S. Drucker, and mc schraefel. Trust Me, I'm Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*, page 1673–1682. ACM, 2012.
- [18] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [19] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual Comparison for Information Visualization. *Information Visualization*, 10(4):289–309, 2011.
- [20] H. Griethe and H. Schumann. The Visualization of Uncertain Data: Methods and Problems. In *Proc. of Conference Simulation and Visualization (SimVis '06)*, page 143–156, 2006.
- [21] S. Guha, R. Hafen, J. Rounds, J. Xia, J. Li, B. Xi, and W. S. Cleveland. Large Complex Data: Divide and Recombine (D&R) with RHIFE. *Stat*, 1(1):53–67, 2012.
- [22] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*, volume 1. Springer, 2nd edition, 2009.
- [23] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. J. Haas. Interactive Data Analysis: The CONTROL Project. *Computer*, 32(8):51–59, 1999.
- [24] J. Kehler, R. N. Boubela, P. Filzmoser, and H. Piringer. A Generic Model for the Integration of Interactive Visualization and Statistical Computing using R. In *Poster Proc. of the IEEE Conference on Visual Analytics Science and Technology (VAST '12)*, page 233–234. IEEE, 2012.
- [25] D. A. Keim, J. Kohlhammer, G. Ellis, and F. Mansmann, editors. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010.
- [26] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual Analytics: Scope and Challenges. In *Visual Data Mining*, number 4404 in Lecture Notes in Comp. Science, pages 76–90. Springer, 2008.
- [27] D. A. Keim, F. Mansmann, and J. Thomas. Visual Analytics: How Much Visualization and How Much Analytics. *SigKDD Explorations*, 2009.
- [28] Z. Liu, B. Jiang, and J. Heer. imMens: Real-Time Visual Querying of Big Data. *Computer Graphics Forum*, 32(3pt4):421–430, 2013.
- [29] M. J. McGuffin and I. Jurisica. Interaction Techniques for Selecting and Manipulating Subgraphs in Network Visualizations. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):937–944, 2009.
- [30] T. Mühlbacher and H. Piringer. A Partition-Based Framework for Building and Validating Regression Models. *IEEE Trans. on Visualization and Computer Graphics (VAST '13)*, 19(12):1962–1971, 2013.
- [31] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre. ClusterSculptor: A Visual Analytics Tool for High-Dimensional Data. In *Proc. of the IEEE Symp. on Visual Analytics in Science and Technology (VAST '07)*, pages 75–82, Oct. 2007.
- [32] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.
- [33] F. Olken and D. Rotem. Random Sampling from Database Files: A Survey. In *Statistical and Scientific Database Management*, number 420 in Lecture Notes in Comp. Science, pages 92–111. Springer, 1990.
- [34] C. Olston and J. D. Mackinlay. Visualizing Data with Bounded Uncertainty. In *Proc. of the IEEE Symp. on Information Visualization (InfoVis '02)*, page 37–. IEEE, 2002.
- [35] H. Piringer, W. Berger, and J. Krasser. HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation. *Computer Graphics Forum (EuroVis '10)*, 29(3):983–992, 2010.
- [36] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1113–1120, 2009.
- [37] J. Platt. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. *Advances in Kernel Methods - Support Vector Learning*, 1998.
- [38] O. Rübél, S. V. E. Keränen, M. Biggin, D. W. Knowles, G. H. Weber, H. Hagen, B. Hamann, and E. W. Bethel. Linking Advanced Visualization and MATLAB for the Analysis of 3D Gene Expression Data. In *Visualization in Medicine and Life Sciences II*, Mathematics and Visualization, pages 265–283. Springer, 2012.
- [39] S. Roweis. EM Algorithms for PCA and SPCA. In *Proc. of the Conference on Advances in Neural Information Processing Systems*, page 626–632. MIT Press, 1998.
- [40] T. Schreck, J. Bernard, T. Tekusova, and J. Kohlhammer. Visual Cluster Analysis of Trajectory Data with Interactive Kohonen Maps. In *Proc. of the IEEE Symp. on Visual Analytics in Science and Technology (VAST '08)*, pages 3–10. IEEE, 2008.
- [41] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Trans. on Visualization and Computer Graphics (cond. accepted for InfoVis)*, 2014.
- [42] B. Shneiderman. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11(6):70–77, 1994.
- [43] M. Streit and O. Bimber. Visual Analytics: Seeking the Unknown. *Computer*, 46(7):20–21, 2013. Guest Editors' Introduction.
- [44] D. Temple Lang and D. F. Swayne. GGobi Meets R: An Extensible Environment for Interactive Dynamic Data Visualization. In *Proc. of the 2nd International Workshop on Distributed Statistical Computing*, 2001.
- [45] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005.
- [46] C. Turkay, F. Jeanquartier, A. Holzinger, and H. Hauser. On Computationally-Enhanced Visual Analysis of Heterogeneous Data and its Application in Biomedical Informatics. In A. Holzinger and I. Jurisica, editors, *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics*, number 8401 in Lecture Notes in Computer Science, pages 117–140. Springer Berlin Heidelberg, Jan. 2014.
- [47] S. Urbanek. No Need to Talk to Strangers - Cooperation of Interactive Software with R as Moderator. In *Proc. of the Symp. of the Interface of Computing Science and Statistics*, 2002.
- [48] S. van den Elzen and J. van Wijk. BaobabView: Interactive Construction and Analysis of Decision Trees. In *Proc. of the IEEE Symp. on Visual Analytics Science and Technology (VAST '11)*, pages 151–160. IEEE, 2011.
- [49] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.
- [50] M. Williams and T. Munzner. Steerable, Progressive Multidimensional Scaling. In *IEEE Symp. on Information Visualization, 2004. INFOVIS 2004*, pages 57–64, 2004.
- [51] P. C. Wong, H. Foote, D. Adams, W. Cowley, and J. Thomas. Dynamic Visualization of Transient Data Streams. In *Proc. of the IEEE Symp. on Information Visualization (InfoVis '03)*, pages 97–104. IEEE, 2003.
- [52] J. S. Yi, Y. a. Kang, J. Stasko, and J. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Trans. on Visualization and Computer Graphics (InfoVis '07)*, 13(6):1224–1231, 2007.

PAPER

B

TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees

Published in *IEEE Transactions on Visualization and Computer Graphics*, 24(1):174-183,
2018 [MLMP18].

TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees

Thomas Mühlbacher, Lorenz Linhardt, Torsten Möller, and Harald Piringer

Abstract—Balancing accuracy gains with other objectives such as interpretability is a key challenge when building decision trees. However, this process is difficult to automate because it involves know-how about the domain as well as the purpose of the model. This paper presents TreePOD, a new approach for sensitivity-aware model selection along trade-offs. TreePOD is based on exploring a large set of candidate trees generated by sampling the parameters of tree construction algorithms. Based on this set, visualizations of quantitative and qualitative tree aspects provide a comprehensive overview of possible tree characteristics. Along trade-offs between two objectives, TreePOD provides efficient selection guidance by focusing on Pareto-optimal tree candidates. TreePOD also conveys the sensitivities of tree characteristics on variations of selected parameters by extending the tree generation process with a full-factorial sampling. We demonstrate how TreePOD supports a variety of tasks involved in decision tree selection and describe its integration in a holistic workflow for building and selecting decision trees. For evaluation, we illustrate a case study for predicting critical power grid states, and we report qualitative feedback from domain experts in the energy sector. This feedback suggests that TreePOD enables users with and without statistical background a confident and efficient identification of suitable decision trees.

Index Terms—Model selection, classification trees, visual parameter search, sensitivity analysis, Pareto optimality

1 INTRODUCTION

Decision trees are a common technique for statistical classification. Hierarchical decision rules model *classes* of a categorical variable depending on numerical or categorical independent variables, called *features*. The decision rules are typically inferred from *training data* for which the classes are known, which is referred to as supervised learning [14]. Frequent types of rules include thresholds on numerical features and class membership vectors on categorical features. In contrast to other types of classification models such as neural networks, a key advantage of decision trees is the ability of humans to understand how the model works. Experts in many fields such as medical diagnosis, image processing, or fraud detection therefore appreciate decision trees for their *interpretability* [14, 19]. In addition to classifying new data instances, the understandable model structure also supports explaining class dependencies for hypothesis generation and reporting.

The process of building decision trees involves multiple trade-offs. As for other model types, the most well-known trade-off is that between over- and underfitting the data for robust generalization (bias-variance trade-off). Automated techniques exist which adjust the model complexity accordingly, e.g., by using different data for growing and pruning the tree [14]. In addition to *accuracy*, however, aspects regarding model *interpretability* by humans are often equally important for decision trees. Model interpretability has received much attention recently [12, 15, 19] and is a multi-faceted goal by itself. Simple trees with limited depth and comprising only few decision rules based on a small number of features are typically easier to understand. Moreover, decision trees intended for human decision makers benefit from nice, round thresholds [15] (e.g., $x \leq 100$ instead of $x \leq 99.475$).

Balancing accuracy gains, interpretability and other objectives such as feature acquisition costs [10, 22, 45] is a key challenge when building decision trees. However, this process is difficult to automate because it involves know-how about the domain as well as the purpose of the model and often requires a qualitative assessment of the deci-

sion tree by domain experts. Even with a deep understanding of the learning algorithm, obtaining a decision tree that satisfies all objectives takes substantial time for trial-and-error [32]. Aggravating the challenge, many domain experts do not have a background in statistical learning [46], but still need to build decision trees which meet their objectives while reflecting their domain knowledge.

This paper proposes TreePOD, a new Visual Analytics technique for decision tree identification which addresses these challenges. Inspired by work on visual parameter space exploration [40] and in line with recent work in statistics [49], our approach is based on exploring a large set of tree candidates. A key goal is to support a *global-to-local strategy for model selection* (G1) that initially provides the user with a comprehensive overview of possible tree characteristics. A second goal is to *address users with and without deep statistical background* (G2). For this reason, TreePOD takes a result-oriented approach which focuses on characteristics of generated trees such as prediction accuracy, complexity, and interpretability. Details of the machine learning process (e.g., training parameters) are hidden by default and exposed only at request. In order to foster a *quick identification of suitable trees* (G3), TreePOD supports an effective quantitative and qualitative comparison of model alternatives. In order to further *increase the user confidence in the selected model* (G4), TreePOD visualizes the sensitivity of tree candidates on variations of generation parameters. Based on TreePOD as the main contribution of this paper, additional contributions include:

- An outlined workflow for decision tree selection.
- A case study to address a real-world problem in the energy sector.
- Qualitative feedback of domain experts from the energy sector.

2 RELATED WORK

Research in statistical learning has devised many automated algorithms for building decision trees, e.g., CART [6], C4.5 [37], and CHAID [16]. Many of these algorithms use entropy minimization to choose features and split positions when growing the tree. After the growing phase, automated approaches can be used to ensure the generalizability of the model, e.g., by pruning and cross validation [14]. Decision trees have also been extended to ensemble learning techniques such as random forests. Such approaches may further increase the accuracy at the cost of incurring significantly higher complexity compared to single trees. Gleicher [12] notes that accuracy is not the only concern and mentions efficiency, generalizability, robustness, conciseness, verifiability, self-consistency, and comprehensibility as some other qualities that model designers must consider. Gleicher also stresses that these properties form trade-offs where the

-
- Thomas Mühlbacher and Harald Piringer are with the VRVis Research Center. Email: {tm | hp}@vrvis.at
 - Lorenz Linhardt is with ETH Zurich. Email: llorenz@student.ethz.ch
 - Torsten Möller is with the University of Vienna. Email: torsten.moeller@univie.ac.at

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxx/TVCG.201x.xxxxxx

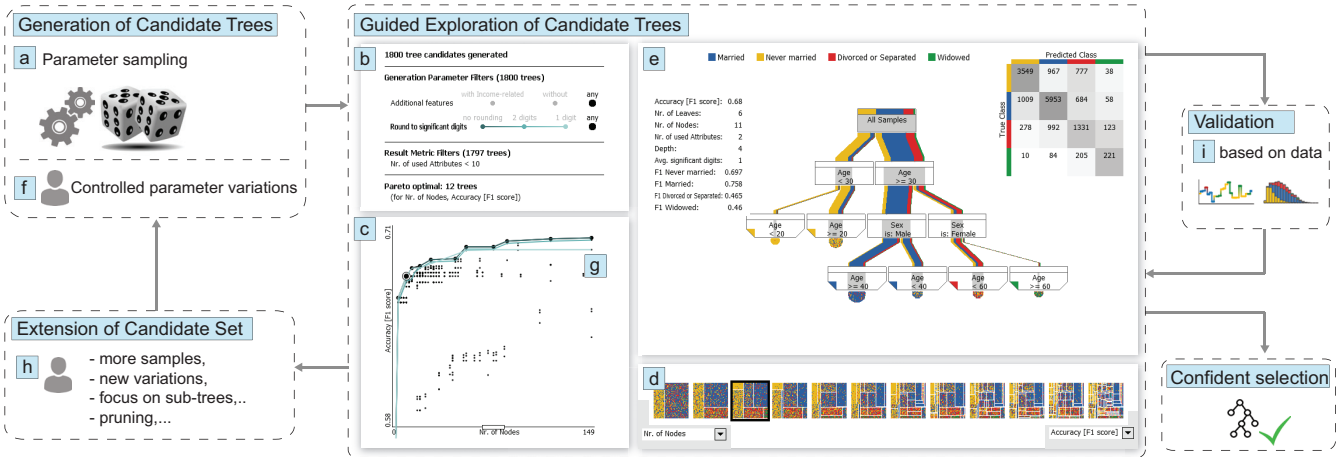


Figure 1. Selection of decision trees explaining *marital status* in the UCI Census Income 1994 dataset [21]. (a) Candidate trees are generated by sampling the parameters of decision tree algorithms. Linked visualizations guide the selection from this set by providing (b) a summary of tree candidates and parameter variations, (c) a sensitivity-aware overview of the trade-off between the conflicting objectives *accuracy* and *number of nodes*, (d) a qualitative comparison of Pareto-optimal trees, and (e) details of a selected decision tree. (f) Applying controlled parameter variations to every tree conveys the effect of parameter changes on tree characteristics, e.g., how rounding of decision boundaries affects accuracy (g). Users can extend the set of candidate trees at any time, (h) and validate trees based on data using linked views.

proper balance depends on the context and needs.

An increasing number of automated approaches take comprehensibility into account as an important goal. Jung et al. [15], for example, perform rounding of model coefficients in logistic regression classifiers in order to make them easier for humans to interpret. Lakkaraju et al. [19] include metrics for interpretability in the objective function for model selection. In many cases, however, assessing comprehensibility requires a qualitative inspection by domain experts.

In contrast to such automated approaches, visualization research has focused on cooperative approaches for decision tree construction which enable users to incorporate their domain knowledge in the generation process. Ankerst et al. [3] let the user evaluate intermediate results of the construction algorithm to specify constraints. This enables the computer to automatically create patterns satisfying these constraints. Van den Elzen and van Wijk [46] support an iterative refinement of a tree during the growing, optimization, and pruning phases. This process is based on BaobabView, a technique for visualizing decision trees which combines advantages of other methods such as node-link diagrams [13, 48] and icicle plots [3, 23]. All these cooperative approaches may improve comprehensibility and user confidence in the model. A study by Liu and Salvendy [23] shows that resulting trees have relatively high classification accuracies and small sizes. However, focusing on the iterative refinement of single trees may not lead to the global optimum. Moreover, such approaches do not communicate the overall achievability of modeling objectives and may require statistical know-how and significant time by the user.

In order to provide a global coverage of possible tree characteristics, some automated approaches obtain multiple decision trees as result. Zhao [50] creates Pareto optimal decision trees to capture the trade-off between different types of misclassification errors. Likewise, Czajkowski and Kretowski [9] use an evolutionary algorithm to generate multiple decision trees which are Pareto optimal for contradictory metrics such as accuracy and the number of nodes. These approaches focus on generating an appropriate set of decision trees, not on exploring this set to facilitate the model selection by a human expert. Czajkowski and Kretowski stress that the comprehensibility of the generated Pareto front is a main issue for future work.

In visualization, an increasing number of systems provide global exploration strategies of parameter spaces [40], e.g., in simulation [1, 7, 25, 35] and image analysis [43]. In many cases, the goal is to identify input parameter values which optimize the output in some sense. Assessing the output often involves both quantitative metrics and qualitative judgments of complex results, for example segmented image data [43]. Statistical model selection is a closely related problem. Understanding the relation between abstract generation parameters and

the resulting model is typically non-intuitive and model selection is usually based on multiple quantitative and qualitative criteria. Related work for exploring model spaces include subspace clustering [28], neural networks [26], and association rules [8].

In the context of decision trees, we regard the work by Padua et al. [32] as most similar. Their system supports the analysis of a large set of candidate trees generated by sampling the parameter space of decision tree algorithms. Linked views visualize this parameter space as well as metrics of the resulting trees and thus enable to relate inputs to outputs by interaction. The trees are shown as node-link diagrams and small icicle plots that convey the structure but not the accuracy. This system provides a global overview of tree characteristics (G1) and guides statistical experts towards useful training parameters. However, their work does not explicitly recognize trade-offs between objectives (G3) and does not visualize their sensitivity on changes of generation parameters or the evaluation data. The analysis focuses on an existing set of trees and does not address the integration in an interactive workflow for decision tree building. Moreover, by exposing many details about generation parameters, the system is primarily designed for users with statistical background which contradicts goal (G2).

3 OVERVIEW OF TREEPOD

TreePOD is a new Visual Analytics technique for sensitivity-aware model selection. The key idea is to create a large set of candidate trees that can be explored with respect to objectives such as prediction accuracy, or interpretability. To this end, the parameter space of tree construction algorithms is sampled to create a diverse set of trees (Fig. 1a, Sec. 4). Visualizing the candidate set at different levels of detail in multiple coordinated views [39] enables a global-to-local strategy for model selection [40] (Sec. 5): A summary panel displays a concise description of the candidate set, and provides various ways of focusing on candidate subsets (Fig. 1b). A quantitative overview shows achievable values for pairs of objectives, and guides selection along trade-offs by identifying the Pareto front, i.e., the set of Pareto-optimal trees (Fig. 1c). Tree maps at the bottom visualize accuracy and complexity of the Pareto-optimal trees in a compact form (Fig. 1d). A detail panel shows the currently selected tree and its characteristics (Fig. 1e).

To investigate local sensitivities of tree characteristics to parameter changes, users can specify a controlled variation of parameters (Fig. 1f, Sec. 6). Visualizing these variations shows how characteristics of single trees, multiple trees, or entire Pareto-fronts are affected by constraints such as rounded decision rules (Fig. 1g). Section 6.3 describes how this approach to *sensitivity-aware trade-off exploration* supports a variety of model selection tasks.

While TreePOD focuses on analyzing and choosing from an existing set of candidates, we also outline its integration in a workflow for

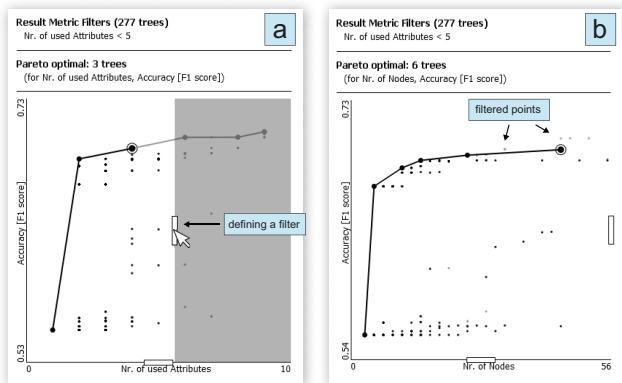


Figure 2. The Pareto front guides tree selection along trade-offs between two result metrics, in this example (a) *accuracy vs. nr. of used attributes*, and (b) *accuracy vs. nr. of nodes*. Hard constraints on metrics filter the set of tree candidates in all views.

building decision trees (Sec. 7). Key steps in this workflow include the incremental extension of the candidate set based on insights from exploration (Fig. 1h), and the validation of trees based on data (Fig. 1i).

As a guiding example illustrating TreePOD, consider the following fictional scenario: *Jane, an analyst working for the ministry of social affairs, aims to predict the multi-class attribute Marital Status in the UCI “Census Income Dataset 1994” [21]. Features comprise 12 demographic attributes like Age, Sex, Income, Occupation, Native Country, and many others¹. Her goal is to obtain an accurate and concise set of rules suitable for reporting or policy-making.*

4 GENERATION OF CANDIDATE TREES

A prerequisite for model selection is the availability of good candidate models. Automatic decision tree algorithms help to identify tree candidates efficiently. Based on a specification of various parameters, they produce a decision tree for pre-classified data by heuristic optimization in two distinct phases:

1) Training: Given a subset of training data and training parameters, the algorithm generates an initial tree description. Training parameters include a set of candidate features and a selection criterion that defines a feature selection strategy (e.g., maximizing information gain [37], Gini impurity [6] or gain ratio [38]). Other parameters include numerical termination criteria for the build process such as a maximal tree depth or a minimal leaf size needed for further splits.

2) Post-processing: In the optional second phase, post-processing such as pruning to avoid overfitting [14], or rounding of numerical decision borders to increase interpretability [42, 15] may be applied.

Training and post-processing involve numerical, categorical and set-typed parameters. For easier readability, we subsequently use *parameter value* as an umbrella term for all types of parameters. Choosing parameter values that result in desirable trees is non-trivial and typically requires substantial effort [32]. Instead of forcing the parameter space upon the user, TreePOD constructs a diverse set of candidates by sampling various parameters in a stochastic or pseudo-random fashion. This may include drawing feature subsets, drawing the maximal tree depth from a range (e.g., [1,...,10]), or randomly choosing a tree pruning method. As a key benefit, stochastic assignment of parameters helps creating diverse and unbiased candidates, which increases the probability of reaching the global optimum during exploration. It also reduces the need to specify parameter values prior to exploration.

Users can also manually assign parameters to incorporate knowledge about algorithms [27] or previously obtained insights. This includes setting parameters to a fixed value for all trees (e.g., max depth = 6), as well as manual adjustment of sampling ranges (e.g., max depth \in [1,...,6]). However, we provide reasonable defaults for all sets and ranges to keep the mandatory user input to a minimum. Data subsets

¹For better demonstration, we intentionally exclude the highly correlated feature *Relationship Status*, as this would yield trivial rules like *Marital Status* ‘Married’ if *Relationship Status* is: *Wife*

for growing, pruning, and evaluation can also be manually specified, but are otherwise automatically determined by splitting the available data into random parts of equal size.

TreePOD also supports various common pruning techniques [14]. As the simplest method, we support collapsing sub-trees if all leaves within produce the same classification. Pruning can also be deactivated to allow for a more detailed analysis of achievable accuracy.

In the guiding example, Jane wants to know how well small models can perform. She generates 300 decision trees by sampling (1) the maximal tree depth between 1 and 6, (2) the minimal leaf size required for further splits, (3) as well as subsets of the 12 available features to obtain different explanations. This generates 300 candidates that are evaluated for an exploration of their results (see Figure 2).

5 GUIDED EXPLORATION OF PARETO-OPTIMAL TREES

This section describes interactive visualizations of the tree candidates at different levels of detail. The goal is to support the selection of suitable trees based on quantitative and qualitative characteristics.

5.1 Candidate summary panel

At the coarsest level, TreePOD provides a concise summary of all tree candidates (see Fig. 1b). This view describes how the set of candidates is successively refined by the user during exploration. Users may define *generation parameter* filters, for example to focus on trees based on particular feature subsets or rounding thresholds. Tree candidates may also be filtered based on their *result metrics* such as accuracy (see Section 5.2). The current set of filters is summarized in this view. Furthermore, the panel states the number of *Pareto-optimal* trees regarding two objective metrics, which is used as central guidance concept in TreePOD. These concepts will be introduced in the following sections.

5.2 Quantitative trade-off overview

The model selection process typically involves quantitative metrics. The metrics in our implementation refer to three types of objectives:

(1) Accuracy, as measured by the F1 score (aka F-measure) [51]. We provide per-class scores (e.g., *F1 “Married”*) as well as the overall score by computing the weighted average of F1 across classes (denoted *Accuracy [F1 score]*).

(2) Complexity, optionally expressed as either the total *number of nodes*, the *number of leaves*, the maximum *tree depth*, the *number of used attributes*, or the total *feature cost*.

(3) Interpretability in terms of human-friendly numbers, computed as the *average num. of significant digits* in numerical rules [31].

We do not intend to make a case for any particular metric. The concepts of TreePOD could be applied to other metrics as well.

For an effective quantitative overview of the tree candidates, TreePOD displays two user-specified metrics in a 2D scatter plot (e.g., *Accuracy vs. Nr. of used attributes* in Fig. 2a). This provides an overview of the candidates in terms of quantitative characteristics and may reveal patterns such as discontinuities or clusters caused by distinct parameter settings, e.g., the inclusion of important features.

Not all candidates are equally relevant for model selection. For example, among all trees of the same size in Fig. 2b, some are substantially more accurate than others. An established concept in multi-criteria decision making is Pareto optimality [18]. In general, a solution is considered Pareto-optimal if no other solution exists that is better for some criteria without being worse for others. The set of all Pareto optimal solutions is called *Pareto front*. In our case, this front comprises all candidate models which are Pareto optimal regarding the two objectives mapped to the axes of the scatter plot.

Pareto-optimal candidates are highlighted using an increased point size and connected with a line to visualize the Pareto front (see Fig. 2). Drawing the front as an interpolated line rather than step-wise is a potentially too optimistic approximation of the real Pareto front. However, we decided to tolerate this as the selection relies on the discrete set of candidates rather than on the continuous shape of the Pareto front. Visually, drawing interpolated lines enables to compare slopes across neighbouring segments. Very steep and very shallow segments indicate transitions that provide high gain of one objective for low additional cost of the other, guiding users towards possible “sweet spots”.

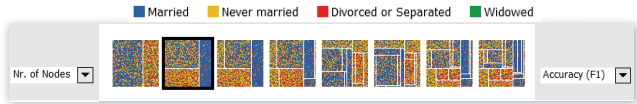


Figure 3. Pixel-based treemaps convey qualitative aspects of accuracy and complexity along a Pareto front.

Any tree can be selected by a click, making it *focal*. In the scatter plot, this *focal tree* is highlighted by a black circle around the point (Fig. 2). Linked views focus on it as well, for example, to show the tree description and parameters which led to that result (see Sec. 5.4).

The view also enables to define range filters for objective values by dragging handles inwards from the plot borders. In Fig. 2, all trees using more than 5 attributes are excluded as indicated by a semi-transparent gray area. Filters persist when changing objectives, which allows investigating a filtered set of candidates with respect to other objectives. This supports a global-to-local workflow for model selection, where the considered set of trees is iteratively refined (G1). Filtered points are not considered when computing the Pareto fronts, but are still displayed in a lower intensity as context. Additionally, a textual representation is shown in the candidate summary (see Fig. 2).

5.3 Qualitative comparison along the Pareto front

The quantitative overview described in the previous section provides effective guidance to trees with high objective values. However, summary metrics hide multiple sources of ambiguity that may be relevant to the decision maker. For example, a high overall accuracy of models can be the result of well-explaining features, or of highly skewed base rates [51]. Likewise, a single accuracy metric does not inform about the distribution of accuracy among the classes.

To visualize such qualitative aspects along a trade-off, we encode the set of Pareto-optimal candidates using small *tree maps* [41] (see Fig. 3). Their sequence represents a linear traversal of the 2D Pareto front, i.e. one objective improves while the other deteriorates from left to right. This arrangement facilitates switching to the next more accurate or next simpler Pareto-optimal tree for an efficient browsing of candidates. Clicking a plot makes the corresponding tree *focal*.

Each partition in a tree-map corresponds to a leaf node, with a relative size proportional to the percentage of data instances classified by that leaf. This enables an effective perception of complexity for the corresponding decision tree (see Fig. 3).

Inspired by perception-based approaches to classification [2, 3, 20], we encode the class distribution within a leaf by a quasi-random placement of pixels according to the class frequencies. The emerging pattern enables an intuitive perception of purity and, for high-purity leaves, easy identification of the predominant class. The selected plot in Figure 3, for example, indicates a first split that isolates *Married* persons very well (mostly blue leaf). The other leaves are much less pure. Discriminability of hue depends on the size of coherent areas [29] and thus on the separability of a data set. We found that, in practice, 5-7 classes can be effectively discriminated also for small pure leaves. For noisy leaves, discrimination of single pixels is typically less important than the overall perception of entropy, which is directly supported by the encoding. This encoding has the advantage that both over- and underfitted trees result in high-frequency patterns. Simple and accurate trees, however, contain large, homogeneous regions. This provides effective qualitative guidance along the trade-off.

Our approach to pixel-based encoding of class distribution is inspired by work of Ankerst et al. [3], but differs with respect to two major aspects: first, their approach shows all levels of the tree next to each other, visualizing the purity gained by every split. Our approach focuses on the leaves to enable an efficient comparison of accuracy and complexity across multiple trees. Second, their pixel arrangement is spatially linked to data items. Our pixel placement is random, which avoids visual structure within the leaves that distracts from the perception of tree complexity. Details on the topology and splits of the tree are shown in a linked visualization (see Sec. 5.4).

Inspecting the tree maps in Fig. 1d, Jane discovers that the more complex Pareto-optimal candidates are refinements of a few simpler ones. She also perceives “Widows” as least frequent Marital Status (green).

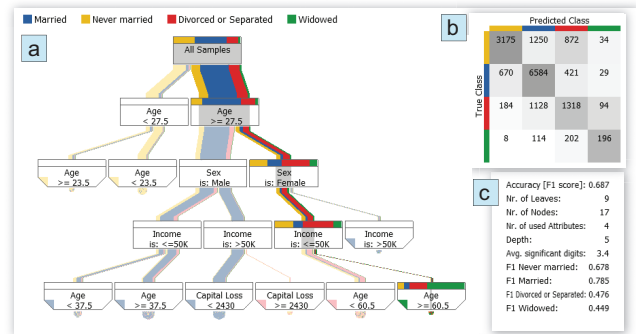


Figure 4. Details for a selected tree evaluation. A node is hovered to focus on the explanation of *Widowed* persons.

5.4 Details for a selected model

Additional views of TreePOD show further details of the focal tree:

1) Structural aspects of the tree: when decision trees are used for explanatory purposes, inspecting the rules is essential. This includes the *names* of the used features, as well as their *depth* in the tree as a notion of their importance. Moreover, the exact split values are often important for explanation or hypothesis generation. The rule definition is also essential for qualitative judgments of interpretability based on domain knowledge. Another structural aspect of trees refers to the topology, e.g., distinguishing deep from wide trees.

To visualize these aspects, we use a node-link diagram inspired by BaobabView [46]. Each node contains its rule definition as text. The width of a link leading into a node is proportional to the number of data items that it applies to. Within links, space is subdivided into stacked, colored bands that convey the proportion of each class [46]. Since we want to emphasize the significance of paths and leaves, we reduce the visual footprint of other aspects. For example, we encode a leaf’s decision as a colored triangle glyph instead of coloring the whole leaf, as this would result in large salient areas that distract from the significance of the links. For the same reason, we show detail information for nodes only on demand: when hovering a node, all nodes between it and the root show horizontally stacked bars conveying the gain of purity along the path (see Fig. 4). Hovering class labels in a coloring legend visually emphasizes leaves yielding that class.

As an indicator for decision confidence, we add a bubble to each leaf node, using the same pixel-based purity encoding as the plots in Sec. 5.3. Their size is proportional to the number of classified data items. Apart from making leaf nodes more salient, these bubbles facilitate visual correspondence of leaves with the tree map visualization.

2) Quantitative properties of the tree: The quantitative metrics listed in the beginning of Section 5 can be inspected in a list. In particular, this includes metrics currently not shown in the trade-off visualizations. As a familiar encoding of accuracy per class, we also provide a confusion matrix. A column-wise encoding of relative frequencies using a linear gray-scale informs the user about systematic misclassifications. On demand, users can switch to a row-wise relative encoding to focus on recall rather than precision. Absolute numbers are stated per cell to support comparisons in any case.

As TreePOD generates its tree candidates by parameter sampling, the particular parameter values that led to a tree can be interesting and are shown on demand. We hide this list by default to focus on the resulting trees, rather than the machine learning process (G2).

Inspecting the details of Pareto-optimal trees, Jane discovers “Age” as an important feature that is often used for the first split, mostly followed by “Sex”, and “Income”. “Age” seems to be important for the classification of Widow(er)s. The confusion matrix for the focal tree, however, reveals that less than half of all Widow(er)s are classified as such (bottom row in Fig.4b). She also discovers that the rule definitions are often not based on whole numbers, such as “Age > 27.5”.

6 SENSITIVITY ANALYSIS OF TRADE-OFFS

Confidence in model selection is a multi-faceted topic. The visualizations described in the previous section provide no direct support for investigating how changes of the parameters involved in training,

post-processing, and evaluation would affect the trees. This section describes extensions to the tree generation process and the visualization which enable an effective sensitivity analysis of parameter variations.

6.1 Generating tree families for effective comparison

Stochastic parameter sampling as described in Sec. 4 efficiently generates a diverse set of alternatives to choose from. However, these samples are usually too diverse to support a focused sensitivity analysis. As a solution, we extend the stochastic generation process by a controlled variation of one or more user-specified parameters, which are subsequently referred to as *variation parameters*. In contrast to other parameters, variation parameters are varied in a full-factorial manner and define a tree candidate for every possible combination of values. For each stochastic sample of the other parameters (Sec. 4), the controlled variation thus defines a *family* of trees. All members of one family are referred to as *sibling trees*. They only differ by the values of one or more variation parameters. For illustration, consider the variation of one parameter in the guiding example:

For her report, Jane prefers rules based on simple integer numbers, e.g. “Age > 28” rather than “Age > 27.5”. She wonders if even multiples of 10 are sufficiently accurate. Thus, she varies the post-processing parameter “Round to significant digits” in three steps: {“no rounding”, “max. 2 significant digits”, and “max. 1 significant digit”}. As a result, 3 variations are created for each of the 300 stochastic samples, which differ by the performed rounding. The new number of candidates is 900, comprising 300 families of 3 trees each.

This two-step generation process ensures the existence of unbiased alternatives, and enables an effective assessment how a single tree, or the candidate set as a whole changes under controlled variations.

6.2 Sensitivity visualization

By default, the visualizations do not treat siblings differently from other possible candidates. As a result, one common Pareto-Front is computed, and shown in the quantitative and qualitative views.

TreePOD supports filtering the candidate set by variation parameters. In the candidate summary panel (Sec. 5.1), all values for each variation parameter are listed using labeled dot markers (see Fig. 1b). Clicking on a dot marker filters the set of visible tree candidates to those of the respective value. An additional marker labeled “any” does not filter on that parameter. Filters for multiple variation parameters are combined by a logical “AND”. We refer to the vector of all current variation parameter values as the *variation focus*. Changing the variation focus updates the set of tree candidates which also updates the Pareto front. The corresponding sibling of the previous focal tree becomes the new focal tree, which also updates the detail visualizations.

For a sensitivity analysis regarding a specific variation parameter, the user may click on its name in the summary panel (e.g., “Round to significant digits” in Figure 1). The scatter plot then supports comparing the impact of parameter changes at three levels of locality.

1) Point-wise sensitivity of the focal tree. As the most local level, the scatter plot displays the siblings of the current focal tree as colored points. Inspired by previous work on sensitivity analysis [4], ordinal variations are connected by lines and encoded using different levels of luminance in the order of variation. For example, the turquoise points in Fig. 5f show how the focal tree changes for increasing maximal tree depths. For variation parameters without inherent order such as the pruning method, all siblings are connected to the focal tree. In this case, hue is used to discriminate the values. Our implementation attempts to use different hue sets for encoding data classes and variation values. This avoids color scheme overlaps if the numbers of classes and compared parameter values are low, which is a frequent case.

2) Point-wise sensitivity of Pareto-optimal trees. As a less local level, point-wise sensitivities can be shown for all currently visible Pareto-optimal trees. This enables to investigate how the sensitivity changes along the Pareto front. For example, Fig. 5d shows that evaluating trees for validation data leads to a stronger accuracy loss for complex trees than for simple ones.

3) Sensitivity analysis of the Pareto front. As the most global level of sensitivity visualization, the Pareto front itself is shown for each variation step. Each front is computed individually based on the

candidates for the corresponding value of the investigated variation parameter. This enables a direct comparison of achievable trade-offs. In Fig. 1g, for example, the turquoise fronts indicate how the trade-off between accuracy and size changes for various rounding thresholds. The color scheme is the same as for point-wise sensitivity encoding.

6.3 Application to sub-tasks of model selection

The process of model selection comprises a number of sub-tasks which can be addressed by TreePOD. We identified four groups of tasks.

1) Sensitivity-aware selection of tree generation parameters This group of tasks refers to studying the global effect of changing tree generation parameters. The focus of interest is typically on the achievable model characteristics and not on individual trees. Therefore, visualizing the entire front is typically the most suitable level of locality in this case. Typical goals include refining parameter ranges for the stochastic variation or assessing their stability for increased confidence. Specific examples for this group of tasks are:

Assessing the benefits of feature inclusion: Using features with high explanatory power is essential for a good fit, but some features may be expensive to obtain. Sometimes, these costs can be quantified, e.g., expensive medical tests [22]. Other times, they are subjective, such as side-effects of medical tests [45]. The latter are often only vaguely known and harder to compare across features. To support both types of costs, TreePOD enables a qualitative comparison of feature inclusion by varying whether a user-specified subset of the features is included. As an example, Fig. 5a shows the achievable Pareto fronts when including *Income-related* features in explaining Marital Status, or not. A reason to omit them could be a generally high number of missing values, when collecting such data from surveys.

Assessing accuracy loss due to decision border rounding: Rounding numerical decision thresholds in a post-processing step increases a tree’s usefulness in human-oriented application contexts [15]. However, this typically decreases accuracy. Varying number rounding parameters, e.g., to n significant digits, supports the user in deciding how much accuracy should be sacrificed (see Fig. 5b).

Further examples refer to the variation of generation strategies, such as the feature selection criterion or the pruning method. For both parameters, several methods exist but no single one is considered generally superior [11, 30]. Visualizing the variation of Pareto fronts helps to understand the effect of different methods for the given dataset.

2) Assessment of model stability From a statistical point of view, a weakness of decision trees refers to their high variance compared to other model types [14]. Slight changes in the training data may lead to substantially different model definitions. TreePOD supports an assessment of model stability by controlled variation of training data subsets. In this case, siblings refer to trees trained for the same parameters, but based on different data. When using meaningful data categories as subsets, encoding the Pareto fronts allows to identify categories for which classification is easier than for others. For example, the scatter plot in Fig. 5c shows that Marital Status is harder to predict for some ethnicities than for others.

3) Sensitivity of accuracy to changed evaluation data Comparing model accuracy across different validation data subsets is a common approach for assessing generalizability to new data [14]. To enable such assessments, TreePOD supports a user-defined variation of the evaluation data subset analogous to the variation of generation parameters. In this case, siblings represent evaluations of the same tree for different data subsets. Showing these siblings for individual trees conveys how accuracy changes for different subsets, which supports the selection of robust models. Particular examples include:

Comparing training and validation data: Comparing tree evaluations for different training and validation data subsets provides guidance along the bias-variance trade-off. Fig. 5d, for example, shows a steadily increasing training accuracy, while the accuracy for validation data decreases for deeper trees due to over-fitting [14]. This provides effective guidance for selecting an adequate model complexity.

Comparing accuracy for data categories: Using meaningful data categories as evaluation subsets allows to identify a potential bias of the models, e.g. towards the most prevalent categories in the training

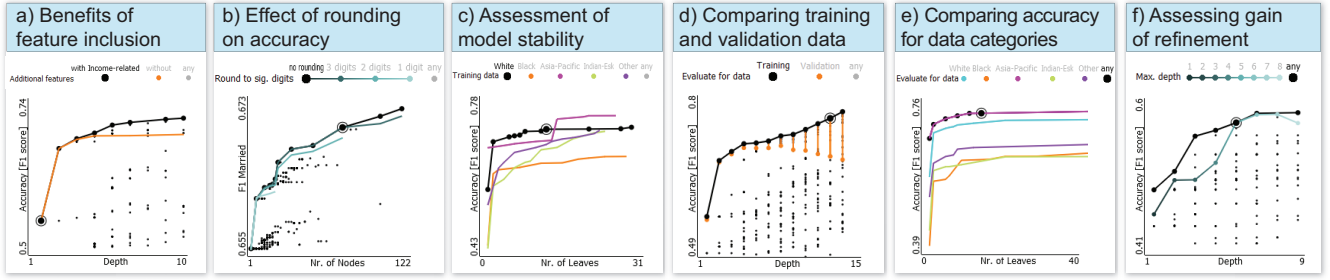


Figure 5. Various applications of a systematic variation of parameters in the generation, post-processing, and evaluation of decision trees. The sensitivity is shown for different levels of locality: the entire Pareto front (a, b, c, e), all Pareto optimal tree candidates (d), and a single tree (f).

data. Fig. 5e, for example, illustrates a variation of the evaluation data for different ethnic groups. The largest ethnic group of data records in the training data refers to “White” persons and also obtains more accurate classification than most others.

4) **Building confidence in a selected tree** TreePOD supports studying variations of a single tree to obtain confidence in its superiority. The point-wise sensitivity encoding is suitable for this purpose.

Assessing gain of refinement: By varying the termination criteria of tree construction (e.g. max. depth, or min. leaf size), TreePOD supports visualizing the benefits incurred by every split level. Reflecting the step-wise nature of the greedy construction process, the resulting line graph visualizes the construction history, to provide guidance for selecting an adequate depth. In Fig. 5f, for example, the variation of the maximal tree depth shows how the focal tree is not Pareto-optimal at first, but becomes part of the front after five refinement levels. Adding a split level increases the accuracy of the tree further, while 3 more levels yield a significant decrease for the validation data.

7 WORKFLOW INTEGRATION

Building decision trees involves multiple steps [46]. The previous sections focused on the description of TreePOD for analyzing and choosing among an existing set of candidate trees. This section outlines the integration of TreePOD in a workflow for building decision trees. The subsequent steps are roughly ordered by their sequence in a typical workflow. However, our implementation does not enforce a particular order and permits most of them at any time.

Selecting training and validation data: Selecting plausible input data is typically a first step. In our implementation, users may interactively brush multivariate views of the data such as scatter plots, parallel coordinates, and time series plots to define data subsets for training and validation (see Sec. 8). Interactive data selection is useful, e.g., to exclude artifacts such as outliers based on domain knowledge. Alternatively, the system automatically defines disjunctive data sets for training and validation by random sampling of the input data.

Definition of initial tree candidates: On demand, TreePOD allows adjusting the variation strategy per generation parameter, i.e., fixed, stochastically sampled, or subject to a controlled variation (see Fig. 6). As the parameters have default values for sampling, users may also simply press a “Train” button to start without specifying parameters.

Global stochastic refinement of tree candidates: Initially, 300 stochastic variations are generated by default. Users may adjust this number depending on, e.g., the size of the training data and the number of features. At any time, users may then press a button titled “Show me more” to generate additional stochastic samples. For each of them, the same controlled variations are applied as for the initial set of trees. The set of Pareto-optimal trees will be re-evaluated for this new set, updating all views. This type of global augmentation of tree candidates is useful if the initial sampling turns out to be too sparse overall.

Local stochastic refinement of tree candidates: For a more focused, result-oriented refinement, users can create variants of the selected focal tree. Pressing a button titled “Show me more like this” will create new samples by stochastically varying the generation parameters such that they are similar to those of the focal tree, e.g., lying within narrow intervals for quantitative parameters. Repeating this for different Pareto optimal candidates allows steering the refinement of the front, and ensuring that interesting regions obtain enough

samples. Alternatively, the user may inspect the particular parameter values for generating the focal tree. Users can then vary specific parameters while keeping all others fixed. For example, this enables to explicitly trigger the creation of additional hierarchy levels for a tree.

Extending the controlled variation: Users may specify or extend controlled variations of parameters at any time, e.g., if they identify interesting aspects for sensitivity analysis only after an initial inspection of the candidate trees. Each update of variation parameters is applied globally to all trees. This may generate new members of tree families or modify existing ones, e.g., if the controlled variation affects parameters which have previously been sampled stochastically.

Subjective validation of classification results: Clicking on any node of the focal tree as well as on rows and columns of the confusion matrix highlights the corresponding subset of training and validation data in the linked multivariate views. This supports a subjective validation of the classification results in the context of the actual data. In particular, this step may reveal if misclassifications are evenly distributed over the data or accumulate for, e.g., specific periods in case of time-dependent data or particular regions in case of spatial data. Sometimes, such findings may indicate structural breaks or insufficient quality for subsets of the data. Users can decide to exclude such subsets and re-run the generation for all models.

Extending the feature set: Detecting data subsets with many misclassifications may also inform domain experts about potentially missing features or may suggest the derivation of new features based on existing ones (e.g., decision boundaries defined by the interaction of multiple features). Derived features may, for example, be created in external computing environments and imported afterwards, e.g., from CSV files. Users may then either re-run the training for all tree candidates, or add the extended features as additional controlled variation.

Generation of sub-trees: It is sometimes helpful to focus the generation process on a particular sub-tree while considering other parts of the tree as given, e.g., if certain subsets of the data are more complex to model than others (Sec. 9 illustrates an example). In this case, users can specify a particular node of the focal tree as temporary root. This generates a set of candidates for this sub-tree using the same approaches for stochastic sampling and controlled variation as for the entire trees. Only these candidates are considered in this type of sub-tree mode. By default, only the data corresponding to the temporary root is considered for computation and visualization, and the result metrics refer to the sub-trees only. However, the structural tree still shows the position of the focal sub-tree within the entire tree as context (see Fig. 6). Upon leaving the sub-tree mode, the user may either add the focal sub-tree or all Pareto-optimal sub-trees as variants of the initiating focal tree to the overall set of tree candidates.

Local pruning of the focal tree: As the counterpart to growing sub-trees, users may also manually prune all nodes below a selected node of the focal tree. In contrast to automated pruning which is performed for all tree candidates, this type of local pruning is only applicable to the focal tree. The pruned tree is added to the set of candidates as a variation of the initiating focal tree.

8 IMPLEMENTATION

TreePOD has been implemented as a part of *Visplore*, a system for visual exploration of multivariate datasets. *Visplore* provides multiple linked views such as scatter plots, time series plots, and views for data

categorization. Data subsets defined by brushing these views can be used in TreePOD as described in Sec. 7. The system is implemented in C++ and uses OpenGL for rendering. A multi-threading architecture [34] is used to maintain interactivity during computations.

For the identification of decision trees, we integrated the CART implementation of the open source library OpenCV [36]. Post-processing operations such as rounding are implemented on top of the tree definitions produced by OpenCV. In most cases, OpenCV was fast enough to generate large numbers of trees in a few seconds. Specifically, generating 300 trees for a data set of 32541 data records and 12 features took on average 5 seconds on a Desktop PC with Intel i7-2600k CPU at 3.4 Ghz and 16GB RAM. From a technical point of view, the ability to generate large numbers of trees rapidly is a key prerequisite for our approach and specifically the interactive workflow.

9 EVALUATION

For evaluating TreePOD and the described workflow, we collaborated with four domain experts working for a transmission system operator and two experts from an IT service provider in the energy market. All of them have been active in this domain for multiple years. They are confronted with classification problems on a regular basis, e.g., for predicting market situations or for building prediction models of time series data. Nevertheless, all of them characterize themselves as having little background in statistical learning and very limited expertise with decision tree algorithms in particular. They used to address classification problems based on insights from static diagrams, intuition, and trial-and-error using common statistics software.

The evaluation took place in three workshops. In a first workshop, we introduced them for one hour to TreePOD by illustrating it based on three energy-related classification problems which were familiar to them from previous projects. They were allowed to ask questions at any time. Based on what they saw, the experts decided on a real-world classification problem as case study for a next workshop.

In this second workshop about one month later, we addressed that particular model selection problem (Sec. 9.1) after a brief recap of TreePOD. We strictly followed their instructions, but operated the software prototype ourselves. Two main reasons were limited time of the experts for familiarizing with all features, and the goal to keep them focused on aspects of the process rather than the implementation. Conducting the described case study took approximately one hour.

In a third workshop four months later, two of the experts used a deployed version of TreePOD to address a different model selection problem (Sec. 9.2.). This time, the experts controlled the system themselves, while we observed their actions and their workflow.

After each workshop, we asked the experts for their feedback using the rose-bud-thorn method [24] for another hour (Sec. 9.3).

9.1 Case study: prediction of imminent power shortages

The key task of power grid operators is to balance demand and supply of electricity. Volatile power sources such as wind farms, or fluctuations of energy prices may lead to spontaneous shortages or abundances in networks. Once such *critical situations* are in progress, they are expensive to fix. Recognizing their imminence in advance for early intervention can thus reduce financial costs significantly.

In a joint analysis session using TreePOD, domain experts identified decision trees predicting imminent critical situations. The target variable is a categorical time series with two classes “*critical in 15min*”, and “*ok in 15min*”, observed over 1 month ($\approx 260,000$ records). Features comprise: (1) the DELTA between power supply and demand, (2) the used proportion of a limited RESERVE of balance energy, (3) various transformations of DELTA and RESERVE such as sliding averages over the past 10min (e.g. DELTA_10), first derivatives that express the TENDENCY of change, (4) 39 POWERPLANT production time series, and (6) categories such as MINUTE and HOUR.

For illustration, Fig. 6a shows examples of imminent critical situations, where RESERVE_10 is at its limit. The purpose of the model is to alert human decision makers rather than to replace them. In addition to high accuracy, having a small set of interpretable rules is thus considered highly important by the experts.

The experts initially select the first and second half of the observed time period as training and validation data. For generating an initial set of tree candidates, the experts stochastically vary the used termination criterion and the subset of input features to obtain 100 samples (see Fig. 6b). As variation parameter, the degree of rounding is varied in 4 steps. This results in $100 \times 4 = 400$ candidates.

The experts set accuracy and the number of nodes as objectives in the scatter plot. All tree maps of Pareto-optimal candidates show large, pure blue regions (Fig. 6c). Inspecting detail views reveals that critical situations are hardly ever imminent when $|\text{RESERVE}_{10}|$ is below 76% of its limit (Fig. 6d). This is the first split of all Pareto optimal candidates. While this matches the expectation of the experts, the particular threshold value is relevant information for them. Classifying the remaining data, however, is more complex as shown by the noise at the margins of the increasingly complex tree maps. In order to focus the further analysis on explaining this remaining variance, the experts enter the sub-tree generation mode for the $|\text{RESERVE}_{10}| \geq 76\%$ node. This creates a separate batch of 400 sub-tree candidates.

The visualizations of the Pareto-front now show 11 Pareto-optimal sub-tree candidates (Fig. 6e,f). In the scatter plot, the colored Pareto fronts for the varied degrees of rounding show that enforcing 3 or 2 significant digits does not incur a significant accuracy loss for smaller trees, while rounding to 1 digit does (Fig. 6e). After inspecting the trees in detail, the experts decide for 2 significant digits.

Browsing the Pareto-optimal candidates reveals that the feature TENDENCY_RESERVE is used for the first split by most sub-trees. This makes sense for the experts, as this feature indicates an increase (positive values) or decline (negative) of available balance energy.

By inspecting the Pareto front in the scatter plot, the experts soon decide for a sub-tree with two splits and an accuracy of approximately 0.73 (Fig. 6e). While the next simple candidate with a single split is much less accurate, significant gains of accuracy conversely require a much larger number of splits which contradicts the requirement for simplicity. The experts inspect further details for this selected focal tree (Fig. 6f,g). They are surprised that the second split by MINUTE has a threshold of 52, which they wish to investigate further. For this purpose, we configured an additional view of our system beyond TreePOD for the experts. Specifically, stacked bars show the proportion of critical situations per minute within the hour cycle. A click on the MINUTE-based split node in TreePOD updates the stacked bars to show only the corresponding data (Fig. 6h). This visualization confirms the adequacy of the split and also indicates a similarly blue region at the beginning of each hour. Based on this cyclic pattern, the experts hypothesize that the temporal proximity to the full hour might be an even more suitable feature than MINUTE. A composite brush for $(\text{MINUTE} > 52 \text{ OR } \text{MINUTE} < 5)$ enables to express HOUR CHANGE as a new binary input feature for TreePOD.

The experts specify an additional controlled variation regarding the inclusion of this feature. The point-wise sensitivity of the focal tree confirms an accuracy gain of approximately 2% for the corresponding sibling. This sibling also belongs to the updated set of Pareto optimal candidates and thus becomes the new focal tree (Fig.6i).

The experts are already very satisfied with this tree. As a final check, they want to validate its generalizability. A controlled variation of the evaluation data confirms the tree’s accuracy for both training and validation data due to its relative simplicity (Fig.6j). More complex tree candidates are much less accurate for the validation data.

At the end of our joint session, the experts were very confident of having selected the most appropriate tree for their purpose. As a next step, they plan to test the performance in operation for a few weeks and eventually update the tree using TreePOD based on recent data.

9.2 Evaluation workshop with users of TreePOD

The third workshop took place four months later. After a brief recap, two of the experts controlled the system themselves for approximately one hour each, in individual sessions. The goal was to identify reasons for short-term changes of power production schedules, denoted as a categorical time series REDISPATCH (yes/no) over three months (2521 records). Features include 23 numerical time series represent-

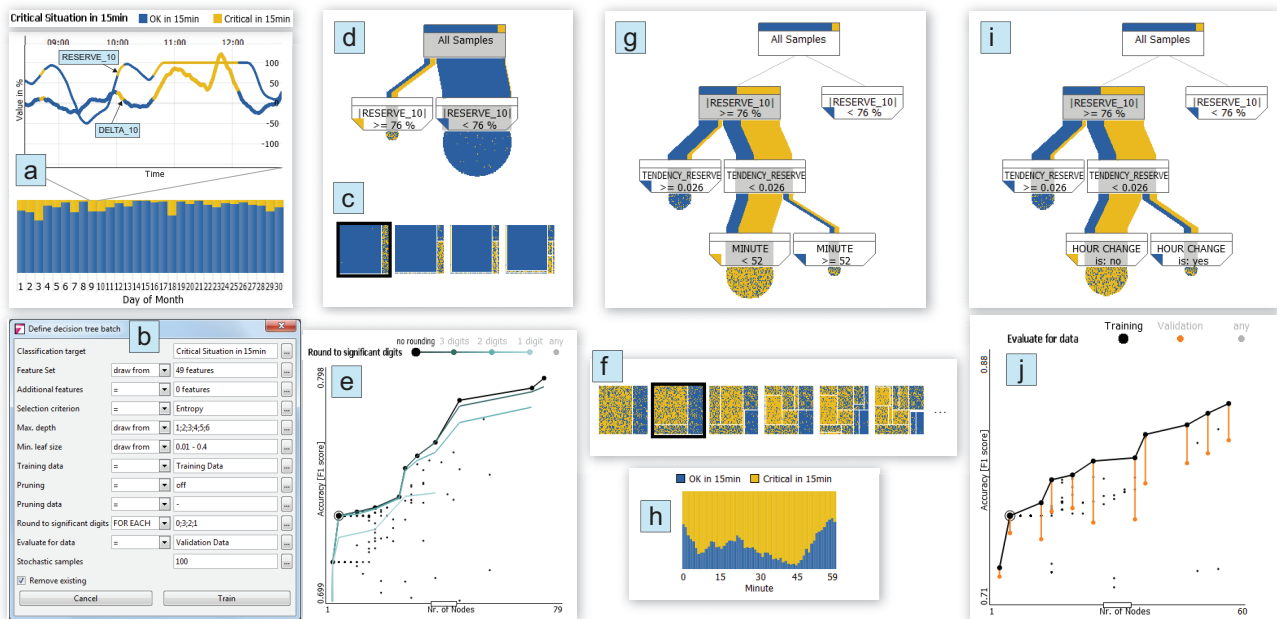


Figure 6. Predicting critical situations in power grid operation. Based on pre-classified data (a), varying decision tree generation parameters (b) obtains an ordered list of Pareto optimal model candidates (c). Inspecting the structure suggests an important first split (d). The Pareto fronts for different degrees of threshold roundness recommend a rounding degree of 2 (e) and a sub-tree candidate (f, g). The distribution of classification accuracy suggests adding the proximity to full hours as feature (h). The resulting tree (i) is better and generalizes for the validation data (j).

ing conditions of the network and the market, as well as temporal categories. This section describes how TreePOD was used by the experts. Screenshots of their insights can be found in the supplemental material. Feedback of the users is part of Sec. 9.3.

For an initial definition of tree candidates, the first user studied the dialog’s options in depth first. She then started with sampling only the termination criteria, but provided all features to every tree as fixed assignment. The only difference between the resulting trees was their degree of refinement, allowing her to assess the benefits of splits. Surprised by the use of feature EXCHANGE_1 for the first split, she investigated alternative first splits by varying the features, while allowing just one split (max depth = 2). Browsing these trees revealed that no single split allowed splitting off a significant number of redispatch cases, and that the selection of EXCHANGE_1 as the first-split feature was justified. She then resorted to the default settings, and created a new batch of 100 trees based on sampling the features and termination criteria. Surprised by the high variation among the candidates, she repeatedly used the “Show me more like this” button to create more samples near the Pareto-front. She then spent some time browsing the fronts. A linked time series view highlighting periods classified as “REDISPATCH: yes” by the focal tree allowed her to compare the recognized redispatches across trees. Watching this view while browsing, she identified trees explaining the previously unexplained redispatch cases. This was a new way of exploration we had not tried before. She finally concluded that the redispatch periods during the first two months can be classified well using trees of moderate depth (≤ 4), which she considered useful for reporting. Trees that also explain the periods of the last month, however, require significantly more nodes.

The second user defined an initial batch of 500 candidates by sampling the features and termination criteria. Browsing the Pareto-optimal trees enabled him a quick identification of important features, as well as a preferable tree depth (max 4-5) for reporting. Like the first user, he was curious about alternative explanations without the dominant feature EXCHANGE_1. Thus, he extended the candidates by controlled variation of omitting vs. providing this feature to the trees. He discovered that a related feature NET_1 is often selected as a substitute, resulting in trees with comparable accuracy. He then used the same linked time series view as the first user while browsing the trees. He hypothesized that the cause for redispatch periods might have changed after the second month. Thus, he decided to split the data sets based on this possible structural break, and trained trees for

each part individually. He discovered that trees for the third month did not use EXCHANGE_1, but rather four other features, confirming his hypothesis. Finally, controlled variation of border rounding showed him that rounding to 3 or even 2 significant digits incurs little accuracy loss for most trees, which he appreciated for his report.

In conclusion, both experts were satisfied with the explanations they found, and considered them useful for their reports.

9.3 Qualitative Feedback

The six domain experts stressed the importance of building classification models as part of their jobs. Some models need to be updated frequently due to rapid changes in the energy sector. Consequently, the time they can spend on tuning single models is limited (G3). Moreover, they believe that many domain experts in their field lack a deep statistical knowledge (G2). For all six experts, model accuracy and complexity are typically the most important aspects. Other requirements such as feature acquisition costs and model plausibility need to be considered as well, but are often hard to quantify. Thus, they appreciated that the controlled variation allowed them to compare discrete sets of model variants without the need for quantification.

The reaction of the experts to TreePOD was very positive overall. All of them praised the possibility of getting a fast overview of possible model characteristics as a huge step forward in comparison to their current practice (G1). In particular, all experts considered the knowledge about the variability of model characteristics and achievability of model objectives as significant gain of confidence (G4). The result-driven approach was embraced as very understandable. The detail visualizations of the model were considered crucial both for understanding the approach as such and for supporting a qualitative model assessment. In general, all experts claimed to have understood TreePOD within the first workshop to a degree which enabled them to think about applications to own classification problems. We specifically asked them if they consider the controlled variation as beneficial without deep algorithmic knowledge. Four experts answered that important variation options do not require such knowledge in their opinion, e.g., the set of input features or rounding levels. Two of the domain experts also considered the variation of other generation parameters as helpful for non-experts in statistics to develop an intuition of their impact.

When asked about specific visualizations, five experts considered the tree maps as important intermediate level of complexity between the abstract scatter plot and the detailed structural visualization. They

considered their linear order as an intuitive guidance through the candidates. However, all experts agreed that the scatter plot is crucial as an overview and for conveying the shape of the Pareto front, e.g., for an efficient perception of jumps and sparsely sampled regions.

As a shortcoming, two experts questioned the restriction to binary trees, i.e., each intermediate node having two children. Despite advantages of binary trees from a statistical point of view [14], they considered more general trees as easier to understand and to communicate, e.g., when subsequent splits refer to the same feature.

The experts who used TreePOD themselves found the default sampling parameters combined with the “Show me more like this” button highly enabling for users without statistical background. However, they considered the number of 20 added samples with every press of this button inadequately small. One expert considered a time-based specification a better alternative, e.g., sampling for 1-2 seconds. One expert suggested adding dedicated buttons to trigger important variations more easily, e.g., “create rounded variations”, or “omit feature”. When defining filters on result metrics, one expert suggested drawing the achievable Pareto front for the filtered trees as context. Concerning the bubble encoding of leaf nodes (see Fig. 4a), the users found purity better conveyed by the stacked bars and bands between nodes. However, one user said their correspondence to the tree maps helped to understand the latter visualization, which was unfamiliar at first.

The other experts also contributed numerous ideas for further extensions. One expert stressed that upper hierarchy levels should be definable from the outside in order to represent given (political) rules and classification schemes. Another expert requested a sensitivity analysis for decision thresholds of particular nodes. As a very interesting idea, one expert suggested using TreePOD to explain user-defined data subsets. For example, after brushing an anomalous period of energy production in a time series view, TreePOD could explain this period by other time series such as meteorological conditions.

10 DISCUSSION AND FUTURE WORK

TreePOD fosters a shift in the strategy for tuning the generation parameters of decision trees. Fully automated tree generation often results in a cumbersome trial-and-error parameter search [32]. Most previous work for cooperative decision tree construction [3, 23, 46] follow a local-to-global strategy for investigating the parameters [40]. These approaches can be classified as white-box integration of visualization and mining [5]. In contrast, TreePOD can be considered a black-box type of integration. A key advantage is to hide details of the generation process from users unless on explicit request (G2). Moreover, TreePOD encourages a global-to-local search strategy which starts with an overview of possible characteristics for reducing the risk of missing the global optimum (G1). TreePOD still supports a cooperative creation, but on a global scale rather than by focusing on a single tree. Specifically, controlled variations are applied to the entire set of candidates which enables a comparison of the effect across trees for higher user confidence (G4). However, this concept does not exclude local refinements of selected trees if explicitly requested by users (see Sec. 7).

TreePOD closely follows the Visual Analytics Mantra [17]: *To analyze first*, TreePOD generates a comprehensive set of decision candidates and computes quality metrics for them. TreePOD *shows the important* by focusing the selection on Pareto-optimal tree candidates. Users may *zoom and filter* by quality metrics. Adding tree candidates enables to *analyze further* for inspecting sensitivities regarding controlled variations of the tree generation parameters as well as for refining the sampling towards desirable tree properties. Additional views provide *details on demand* for a selected tree.

An important design decision of TreePOD is to restrict the number of Pareto objectives to two. This limitation has several significant advantages for keeping the approach understandable by users. For visualization, the simple representation as poly-line permits an intuitive comparison of variations of the entire front. For interaction, the linear order of tree candidates along the trade-off enables an intuitive switch from one tree to the next more accurate or more simple Pareto-optimal tree. For guidance in general, the set of Pareto optimal tree candidates is typically much smaller for two objectives than for

three or more objectives, which avoids overwhelming the user with too many alternatives (G3). Moreover, feedback by domain experts suggests that the trade-off between accuracy and complexity is typically the most important consideration, even if additional objectives such as feature acquisition cost exist. Additional objectives can be considered by filtering trees with undesirable values as a common approach to address multi-criteria decision problems [44]. Nevertheless, experimenting with visualization approaches for higher-dimensional multi-criteria decision making [33] is relevant as future work.

Regarding other scalability aspects, the use of hue restricts the number of target classes to approximately ten for perceptual reasons [47]. Even more so, as color is also used for encoding the variation. We also experimented with showing variations of multiple parameters simultaneously, but rejected this feature due to generating too complex visualizations in many cases. On the other hand, the visual complexity of TreePOD does not depend on the size and dimensionality of the training or validation data. As a practical limit of the data size, however, the approach strongly benefits from short training times of trees in order to generate a sufficiently dense sampling overall and of the Pareto front in particular. The quantitative overview scales well for large numbers of trees, considering that the most relevant information is the location and shape of the Pareto front. Conversely, a sparse sampling will in general obtain a very inaccurate approximation of the real Pareto front. While local refinements of the sampling help to mitigate this problem (Sec. 7), integrating advanced approaches for constructing the Pareto front [9, 50] are an important aspect of future work.

As a next step, we plan to conduct a long-term study based on deploying TreePOD to target users from multiple application domains. Moreover, we intend to extend the approach in order to further utilize information contained in the generated set of candidate trees. For example, analyzing the frequency and the context in which particular features are selected could provide useful information about their importance. Finally, we believe that core concepts of TreePOD are transferable to other types of models. Model selection is typically a multi-criteria problem. In addition to accuracy, objectives regarding, e.g., comprehensibility and feature acquisition cost apply to many types of models [12], e.g., regression polynomials. We thus plan to evaluate in how far the concepts of TreePOD regarding sampling, guidance, and variation also support the selection process for other types of models by replacing decision tree-specific result metrics and visualizations.

11 CONCLUSION

This paper described TreePOD, a new approach for sensitivity-aware selection of decision trees in the presence of multiple objectives. Besides accuracy, especially the need for comprehensible models is increasing [19]. To address this need, TreePOD fosters a global-to-local strategy for model selection in order to guide also non-experts in statistical modeling towards a confident selection of suitable trees.

Based on TreePOD, we described a holistic workflow for decision tree selection which combines aspects from white-box and black-box integration of visualization and data mining [5]. A case study conducted in pair-analysis with domain experts illustrated the ability of TreePOD to solve a relevant problem in the energy sector, and confirmed that non-experts in statistics were able to efficiently identify a suitable decision tree with high confidence. TreePOD is applicable to classification problems independent of the application domain. As one possible direction of future work, we believe that TreePOD is conceptually transferable to other types of models for increasing the efficiency and confidence in the selection process.

ACKNOWLEDGMENTS

This research is funded by the COMET K1 program – Competence Centers for Excellent Technologies (854174) by BMVIT, BMWWF, Styria, Styrian Business Promotion Agency – SFG and Vienna Business Agency. This work has also been supported by the Austrian Funding Agency (FFG) within the scope of the K-project DEXHELPP (843550). The COMET Programme is managed by FFG. Thanks go to all collaborators from the energy sector, and to C. Arbesser, O. Rafelsberger, S. Pajer, and Torsten Möller’s group for valuable comments.

REFERENCES

- [1] S. Afzal, R. Maciejewski, and D. Ebert. Visual analytics decision support environment for epidemic modeling and response evaluation. In *IEEE Conf. on Visual Analytics in Science and Technology (VAST)*, pages 191–200, 2011.
- [2] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel. Visual classification: An interactive approach to decision tree construction. In *Proceedings of the Fifth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, KDD '99, pages 392–396, NY, USA, 1999. ACM.
- [3] M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *Proceedings of the Sixth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, KDD '00, pages 179–188, NY, USA, 2000. ACM.
- [4] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. In *Computer Graphics Forum*, volume 30, pages 911–920, 2011.
- [5] E. Bertini and D. Lalanne. Investigating and reflecting on the integration of automatic data analysis and visualization in knowledge discovery. *ACM SIGKDD Explorations Newsletter*, 11(2):9–18, 2010.
- [6] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. CRC Press, New York, 1999.
- [7] S. Bruckner and T. Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1467–1475, 2010.
- [8] W. Castillo-Rojas, C. Vargas, and C. M. Villegas. Interactive visualization of association rules model using SOM. In *Proceedings of the XV International Conference on Human Computer Interaction*, page 104, 2014.
- [9] M. Czajkowski and M. Kretowski. A multi-objective evolutionary approach to pareto optimal model trees. a preliminary study. In *Theory and Practice of Natural Computing*, pages 85–96, 2016.
- [10] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.
- [11] F. Esposito, D. Malerba, G. Semeraro, and J. Kay. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):476–491, May 1997.
- [12] M. Gleicher. A framework for considering comprehensibility in modeling. *Big Data*, 4(2):75–88, 2016.
- [13] J. Han and N. Cercone. Interactive construction of decision trees. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 575–580, 2001.
- [14] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Second Edition*. Springer New York Inc., 2009.
- [15] J. Jung, C. Concannon, R. Shroff, S. Goel, and D. G. Goldstein. Simple rules for complex decisions. *CoRR*, arXiv:1702.04690, 2017.
- [16] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, pages 119–127, 1980.
- [17] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In S. Simoff, M. Böhlen, and A. Mazaika, editors, *Visual Data Mining*, pages 76–90. Springer, 2008.
- [18] M. M. Köksalan, J. Wallenius, and S. Zions. *Multiple criteria decision making from early history to the 21st century*. World Scientific, 2011.
- [19] H. Lakkaraju, S. H. Bach, and J. Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684, 2016.
- [20] P. Lévy. *Pixelization Paradigm: Visual Information Expert Workshop, VIEW 2006, Paris, April 2006, Revised Selected Papers*. Image Processing, Computer Vision, Pattern Recognition, and Graphics. Springer, 2007.
- [21] M. Lichman. UCI machine learning repository, 2013.
- [22] C. X. Ling, V. S. Sheng, and Q. Yang. Test strategies for cost-sensitive decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1055–1067, 2006.
- [23] Y. Liu and G. Salvendy. Design and evaluation of visualization support to facilitate decision trees classification. *International Journal of Human-Computer Studies*, 65(2):95–110, 2007.
- [24] LUMA Institute. Vision Statement: A Taxonomy of Innovation, 2014.
- [25] K. Matković, D. Gračanin, M. Jelović, and H. Hauser. Interactive visual steering—rapid visual prototyping of a common rail injection system. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1699–1706, 2008.
- [26] C. J. Meneses and G. G. Grinstein. Visualization for enhancing the data mining process. In *Aerospace/Defense Sensing, Simulation, and Controls*, pages 126–137, 2001.
- [27] T. Mühlbacher, H. Piringer, S. Gratzl, M. Sedlmair, and M. Streit. Opening the black box: Strategies for increased user involvement in existing algorithm implementations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1643–1652, Dec 2014.
- [28] E. Müller, I. Assent, R. Krieger, T. Jansen, and T. Seidl. Morpheus: interactive exploration of subspace clustering. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1089–1092, 2008.
- [29] T. Munzner. *Visualization analysis and design*. CRC Press, 2014.
- [30] S. K. Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2(4):345–389, 1998.
- [31] K.-M. Osei-Bryson. Evaluation of decision trees: a multi-criteria approach. *Computers and Operations Research*, 31(11):1933 – 1945, 2004.
- [32] L. Padua, H. Schulze, K. Matković, and C. Delrieux. Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections. *Computers & Graphics*, 41:99 – 113, 2014.
- [33] S. Pajer, M. Streit, T. Torsney-Weir, F. Spechtenhauser, T. Möller, and H. Piringer. Weightlifter: Visual weight space exploration for multi-criteria decision making. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):611–620, 2017.
- [34] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1113–1120, Nov. 2009.
- [35] K. Potter, A. Wilson, P.-T. Bremer, D. Williams, C. Doutriaux, V. Pascucci, and C. R. Johnson. Ensemble-vis: A framework for the statistical visualization of ensemble data. In *IEEE Conference on Data Mining Workshops*, pages 233–240, 2009.
- [36] K. Pulli, A. Baksheev, K. Korniyakov, and V. Eruhmov. Real-time computer vision with opencv. *Comm. of the ACM*, 55(6):61–69, 2012.
- [37] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [38] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of artificial intelligence research*, 4:77–90, 1996.
- [39] J. C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In *Proc. of the Fifth Int. Conf. on Coordinated and Multiple Views in Exploratory Visualization*, pages 61–71, 2007.
- [40] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2161–2170, 2014.
- [41] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG)*, 11(1):92–99, 1992.
- [42] J. Talbot, S. Lin, and P. Hanrahan. An extension of wilkinson's algorithm for positioning tick labels on axes. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2010.
- [43] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.
- [44] T. Torsney-Weir, M. Sedlmair, and T. Möller. Visualization for decision making under uncertainty. In *Workshop on Visualization for Decision Making under Uncertainty (VDMU)*, 2015.
- [45] P. D. Turney. Types of cost in inductive concept learning. *CoRR*, cs.LG/0212034, 2002.
- [46] S. van den Elzen and J. J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2011)*, pages 151–160, 2011.
- [47] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., 2004.
- [48] M. Ware, E. Frank, G. Holmes, M. Hall, and I. H. Witten. Interactive machine learning: letting users build classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.
- [49] H. Wickham, D. Cook, and H. Hofmann. Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(4):203–225, 2015.
- [50] H. Zhao. A multi-objective genetic programming approach to developing pareto optimal decision trees. *Decision Support Systems*, 43(3), 2007.
- [51] X. Zhu and I. Davidson. *Knowledge discovery and data mining: challenges and realities*. Premier reference source. Information Science Reference, 2007.

PAPER

C

A Partition-Based Framework for Building and Validating Regression Models

Published in *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1962-1971, 2013 [MP13]

A Partition-Based Framework for Building and Validating Regression Models

Thomas Mühlbacher and Harald Piringer

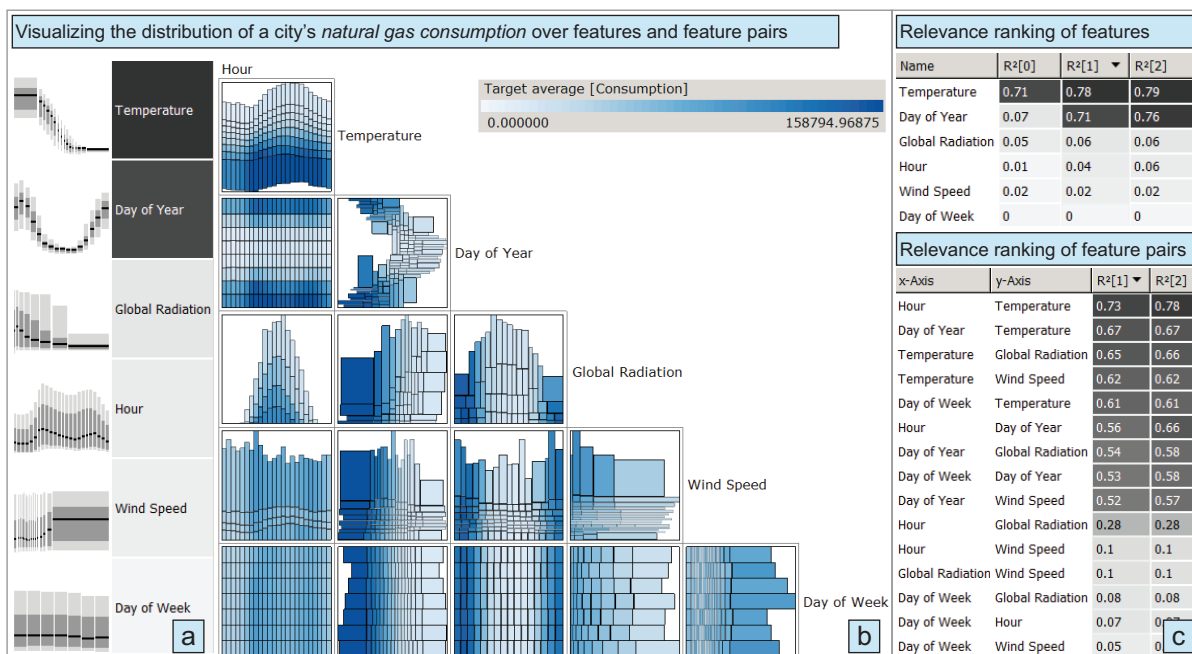


Fig. 1. Analyzing relationships using our framework: The conditional distribution of the dependent variable *natural gas consumption* is visualized over partitioned input features (a) and feature pairs (b), which are ranked by measures quantifying their relevance (c).

Abstract—Regression models play a key role in many application domains for analyzing or predicting a quantitative dependent variable based on one or more independent variables. Automated approaches for building regression models are typically limited with respect to incorporating domain knowledge in the process of selecting input variables (also known as feature subset selection). Other limitations include the identification of local structures, transformations, and interactions between variables. The contribution of this paper is a framework for building regression models addressing these limitations. The framework combines a qualitative analysis of relationship structures by visualization and a quantification of relevance for ranking any number of features and pairs of features which may be categorical or continuous. A central aspect is the local approximation of the conditional target distribution by partitioning 1D and 2D feature domains into disjoint regions. This enables a visual investigation of local patterns and largely avoids structural assumptions for the quantitative ranking. We describe how the framework supports different tasks in model building (e.g., validation and comparison), and we present an interactive workflow for feature subset selection. A real-world case study illustrates the step-wise identification of a five-dimensional model for natural gas consumption. We also report feedback from domain experts after two months of deployment in the energy sector, indicating a significant effort reduction for building and improving regression models.

Index Terms—Regression, model building, visual knowledge discovery, feature selection, data partitioning, guided visualization

1 INTRODUCTION

Regression analysis is a statistical technique for modeling a quantitative dependent variable Y as a function of one or more continuous or categorical independent variables X_1 to X_n . Common applications of regression models include prediction and sensitivity analysis of Y with respect to changes of independent variables. The field of sta-

tistical learning has developed many types of regression models and techniques supporting the process of model selection [21]. This process comprises identifying suitable values for model-specific parameters as well as selecting a minimal descriptive subset of independent variables, also known as *feature subset selection* [19] (we use the term *feature* as a synonym for *independent variable* in this paper). Benefits of having a minimal number of features include an improved model interpretability, reduced training times, and a reduced probability of overfitting while still providing an accurate fit [21].

In general, the trade-off between model complexity and accuracy explains one challenge in building regression models. Another challenge arises from the inability of incorporating domain knowledge into common automatic feature selection techniques (e.g., step-wise regression [13]). As different techniques may yield different results and often reflect aspects of the training data rather than domain knowledge,

• Thomas Mühlbacher is with the VRVis Research Center. E-Mail: tm@vrvis.at.

• Harald Piringer is with the VRVis Research Center. E-Mail: hp@vrvis.at.

Manuscript received 31 March 2013; accepted 1 August 2013; posted online 13 October 2013; mailed on 4 October 2013.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

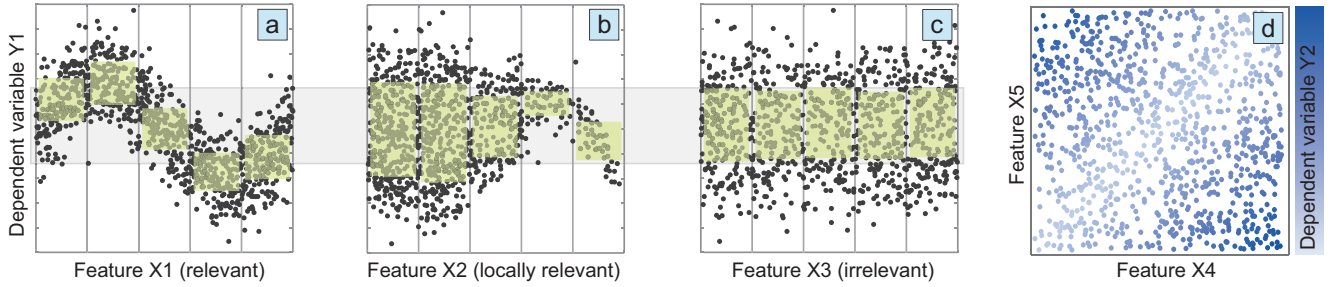


Fig. 2. Synthetic examples motivating goals of our framework. (a - c) Local variations of the conditional distribution of a dependent variable Y_1 explain X_1 as relevant due to a non-monotonic relationship with Y_1 , X_2 as locally relevant, and X_3 as irrelevant. Green rectangles indicate local dispersion and gray rectangles show global dispersion of Y_1 as measured by interquartile ranges. (d) Another dependent variable Y_2 is explained by the interaction of two features X_4 and X_5 .

“automated variable selection procedures are no substitute for careful thought” [1]. Additionally, many types of regression models imply structural assumptions (e.g., linear relationships). Knowledge about complex or local relationships (see Fig. 2a and b) as well as about interactions of variables (see Fig. 2d) is thus crucial for selecting an appropriate model type and for identifying suitable transformations of variables such as the logarithm, polynomial basis expansions (e.g., $X_2 = X_1^2$) or binary operations (e.g., $X_3 = X_1 \cdot X_2$). According to recent studies of Kandel et al. [24], feature selection and transformation are two of the most time-consuming challenges in data analysis.

This paper proposes an interactive framework for building regression models addressing these challenges. The approach combines a visualization of relationships between features and a quantitative target and a quantification of these relationships for ranking them by relevance. Using derived quantities like residuals as target supports different tasks of model building including feature subset selection, model validation, and model comparison. A central goal is to enable the identification of complex relationships (e.g., having discontinuities or local extrema) and local relationships (i.e., features explaining the target across a part of their domain, see Fig. 2b). To achieve this goal, a key idea is *partitioning* the feature space into disjoint regions for visualization and for quantification, providing an adjustable level of detail between a point-wise and a global analysis [29]. The framework supports inspecting individual features as well as *pairs of features* in order to enable the discovery of arbitrary bivariate interactions (see Fig. 1).

The application background motivating this work is the need for accurate prediction models in the energy sector. Most figures of this paper and an exemplary case study (Sec. 5.1) refer to predicting the consumption of natural gas in a large city. In this domain, a precise knowledge of the combined effects of meteorological and other factors on the consumption is crucial for minimizing costs and guaranteeing supply. Operating on generic continuous or categorical data, however, the proposed framework is not limited to any domain but addresses very general issues of regression analysis and knowledge discovery. Specifically, the contributions of this paper include:

- techniques for ranking variables and pairs of variables by their usefulness in predicting a quantitative target.
- a design space of partition-based visualizations showing local structures in the target distribution over one or two variables.
- applications of the framework for model validation and comparison, and an interactive workflow for feature selection.
- an evaluation of the framework based on a case study of a real-world modeling task and user feedback after two months of deployment in the energy sector.

2 RELATED WORK

Interactive pattern discovery and model building are key issues of Visual Analytics. Examples include clustering [30], classification [47], and learning distance functions [8]. This paper focuses on *regression-related tasks*, such as feature selection and model validation.

Regression has traditionally been a key issue in statistics, resulting in a variety of model types [21] as well as methods supporting model

selection [19], model comparison [27], and model validation [43]. Numerous measures have been proposed for quantifying relationships, many of them being limited to certain classes such as linear or monotonic relationships (e.g., Pearson correlation). As a more general indicator, the Maximum Information Coefficient (MIC) measures the mutual information of two features based on partitioning them at multiple resolutions [37]. Similar to our approach, the partitioning of MIC largely avoids structural assumptions, but we do not require a categorization of the dependent variable. More importantly, quantifying a relationship by a single value incurs a loss of information which may hide important structural aspects, e.g., due to data quality issues. For this reason, a comparison of multiple measures is advisable [1].

The Rank-by-Feature Framework (RbFF) [39] has been proposed as an interactive approach to support a comparison of statistical measures in combination with a visualization of qualitative aspects. The ability to handle univariate and bivariate measures and the good scalability for high-dimensional data motivated us to adopt the layout of the RbFF for our framework. However, the RbFF was neither designed to support regression-related tasks in general, nor the detection of relationships to a quantitative target in particular. The same is true for other techniques supporting an exploration of high-dimensional data by ranking visualizations based on screen-space metrics [50], class consistency measures [41], and the interestingness of point clouds [44].

A variety of approaches addresses the identification of multi-dimensional relationships in a more general sense. Besides common multivariate visualization techniques like scatterplot matrices [11] and parallel coordinates [23], some approaches explicitly denote a quantitative dependent variable. Guo et al. [18] support the discovery of multivariate trends. An interactive visualization of the model parameter space enables to detect multiple trends but is limited to linear models. Barlowe et al. [3] display distributions of partial derivatives for an identification of multi-dimensional relationships. The authors describe an interactive workflow for model construction, dimension reduction, and knowledge discovery. However, the interpretation of the visualizations may require significant training and it remains unclear in how far distributions of partial derivatives convey complex local structures. Other approaches support an exploration of relationships based on visualizing high-dimensional scalar functions by showing topological structures [16] or projections based on slicing [48, 45]. While useful for understanding an existing model, most tasks related to model building are not directly supported by such visualizations.

While some approaches address sensitivity analysis [17, 9], providing dedicated support for regression-related tasks has received little attention in Visual Analytics so far. Friendly uses shaded mosaic displays [15] to visualize averaged model residuals or target values across combinations of categorical dimensions. Described as a static diagram, this approach does not address aspects of high-dimensional data such as ranking and iterative feature selection. Moreover, handling continuous variables is not discussed. Berger et al. [6] use regression models for a continuous exploration of sampled parameter spaces, but do not cover model building. HyperMoVal [32] addresses the validation of regression models by relating validation data to function graphs of models based on slicing. However, this point-wise level of detail is

inappropriate to provide an overview over local structures.

Partition-based visualization techniques address this shortcoming by providing an intermediate level of detail. Converting continuous data to a frequency-based representation is often referred to as binning [40]. The goal is reducing complexity and ensuring the scalability for many data samples while preserving local structures to some degree. Variable binned scatterplots adapt bin size to the characteristics of the data for visualizing large data without overlapping [20]. Slingsby et al. [42] explored the effects of alternative layouts in space-filling hierarchical displays to show multiple aspects of large multivariate datasets. We provide a discussion of different layouts for partition-based visualizations of 1D and 2D domains in the context of regression.

Using partitioning for iterative feature subset selection, the work by May et al. [29] is most similar to ours. Mutual information measures between a target and partitioned features are visualized individually for each partition to show the local relevance while global aggregates rank features by relevance. Operating on a categorical target, their approach also supports classification while the required categorization of continuous targets introduces a problematic loss of detail for regression. In contrast, our framework does not categorize the target. This enables the visualization of local distributions as required for many tasks in regression. Moreover, our framework supports pairs of features as needed for detecting interactions between features.

3 A PARTITION-BASED FRAMEWORK FOR REGRESSION

This section introduces our framework for regression-related tasks. The approach is to support an exploration of relationships between a feature space X of continuous or categorical independent variables X_1 to X_n and a quantitative target T . As shown in Fig. 1, the main layout elements of our framework comprise tables of measures quantifying the relevance for individual features (1D) and pairs of features (2D) with respect to T as well as corresponding small-multiple visualizations conveying structural details of relationships. These visualizations include a list of plots (1D) and a half-diagonal matrix of plots showing all pair-wise combinations of features (2D). Ordering a table by a measure also ranks the corresponding small-multiple visualization as a guidance to potentially relevant plots (inferring an ordering for the matrix is discussed in previous work [31]).

The basis of visualization and ranking is the fact that relationships between a feature X_i or a pair of features X_i, X_j (henceforth abbreviated as $X_i[, X_j]$) and T manifest in local variations of the conditional distribution $P(T|X_i[, X_j])$ (see Fig. 2). Expressing the local mean values of the conditional distribution as a function is the fundamental concept of regression [21]. The key idea of our framework is approximating $P(T|X_i[, X_j])$ by *partitioning* the one- or two-dimensional domains into disjoint regions. Inspired by May et al. [29], the rationale is to provide an adjustable and computationally efficient level of intermediate detail between a point-wise and a global analysis.

The subsequent sections describe different aspects of partition-based exploration of relationships: Section 3.1 discusses general considerations and approaches to partitioning $X_i[, X_j]$. Section 3.2 describes partition-based visualizations that approximate the conditional distribution of T . Section 3.3 discusses a partition-based quantification of relevance. In addition to exploring relationships between X and a user-selected dependent variable Y (i.e., $T = Y$), Section 3.4 describes the application of our framework to common tasks in statistical modeling by using various derived quantities as T . Details on how to perform the partitioning, the visualization, the ranking, and the application are to a large degree independent of each other and can be extended separately, which is the motivation for us to refer to our approach as a framework. Section 3.5 then extends this framework to support an interactive workflow for feature subset selection.

3.1 Partitioning $X_i[, X_j]$

This section discusses general aspects of partitioning $X_i[, X_j]$ which are the basis for partition-based visualization and ranking in subsequent sections. In computer science, subdivision is a key concept to reduce a complex problem to a set of more simple ones. In the context of multidimensional data, examples of hierarchical subdivision include

search algorithms [12] and image processing [38]. In statistics, tree-based methods in general [21] and regression trees in particular have received substantial attention in literature due to their ability to flexibly capture relationships of complex structure [7, 14].

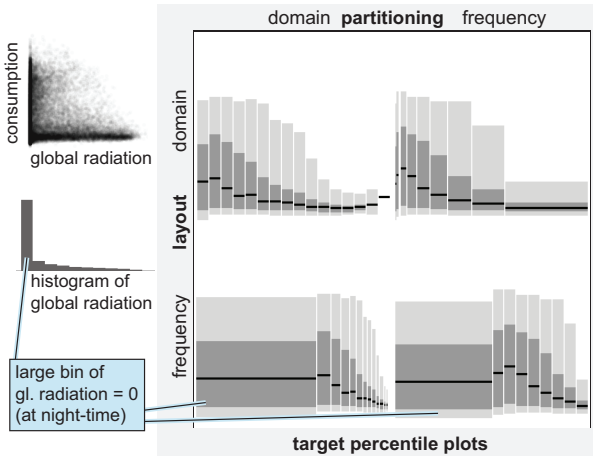
Our approach to approximate $P(T|X_i[, X_j])$ is inspired by regression trees in that an adaptation to complex structures is based on considering disjoint regions of $X_i[, X_j]$ separately from each other. However, we have different goals and constraints than most approaches to building regression trees. Rather than building an accurate regression tree for prediction, the goal of our approach is to locally approximate $P(T|X_i[, X_j])$ for a potentially large number of features. Due to this goal, an individual partitioning is required for each $X_i[, X_j]$, as opposed to applying the same partitioning to all features [34]. The result of partitioning $X_i[, X_j]$ is a set of disjoint regions where any data sample is contained in one region. For one-dimensional partitioning, each region is described by either a category if X_i is categorical or an interval if X_i is continuous. For two-dimensional partitioning, these restrictions independently apply to X_i and X_j , i.e., a region of two continuous features is an axis-aligned rectangle. Besides simplicity, the main reason for these restrictions is to enable a flexible visualization (see Sec. 3.2).

We identified three requirements for partitioning $X_i[, X_j]$: 1) *General applicability*: Assumptions about the distribution of $X_i[, X_j]$ should be avoided. 2) *Fast computation*: In the sense of Visual Analytics, the ultimate goal is to provide an interactive framework enabling workflows which tightly couple user-centric and computation-centric steps (see Sec. 3.5). Significant delays should thus be avoided when users change T , X or partition-specific parameters. Therefore, partitioning all $X_i[, X_j]$ should be feasible within at most a few seconds also in case of a large number of features for 1D and especially 2D analysis. 3) *Adjustability*: The degree of detail should be adjustable intuitively. This implies that regions should have a similar size in some sense in order to make regions comparable for a given distribution of data.

Concerning adjustability, the size of a region can be interpreted in different ways, i.e., as the size in the *domain* of $X_i[, X_j]$, or as the size with respect to the *number of data samples*. As a consequence, our framework supports two different approaches for partitioning $X_i[, X_j]$. **Domain-uniform partitioning.** This approach subdivides each continuous feature X_i into N intervals of equal domain size between the minimum and the maximum of X_i . The parameter N thus adjusts the degree of detail of the partitioning. For categorical features, the categorization is taken as subdivision. For feature pairs, the regions are the Cartesian product of the individual subdivisions of X_i and X_j . Domain-uniform partitioning has linear effort and is very fast. However, the distribution of data samples within $X_i[, X_j]$ is ignored. While this may be desirable, it is generally a problem in the presence of outliers and non-uniform distributions. Specifically, many resulting regions may be empty or contain a statistically insignificant number of samples.

Frequency-uniform partitioning. The goal of this approach is to define regions containing an identical (or at least similar) number of data samples, i.e., having a same relative frequency. Inspired by Kd-trees [5], the key concept is based on a binary hierarchical subdivision of continuous features by recursively splitting the data at the median of the respective subset of samples. In order to be also applicable to ordinal data, our consideration is that data samples having identical values in $X_i[, X_j]$ must be assigned to the same region. In this case, we shift the splitting location into the direction that generates more equally-sized subsets. For nominal data, the categorization is taken as the subdivision even for differently sized categories. For feature pairs, the subdivisions of X_i and X_j are interleaved, starting with the feature where the median is closer to the center of the domain. In case of a categorical feature X_i and a continuous X_j , the approach splits X_j separately for each category of X_i , i.e., the subdivision of X_i is done first. The recursion stops if either (1) the entire subset of data samples has identical values in $X_i[, X_j]$ or (2) a split would create at least one region having less than a user-defined minimal significance S_{min} of data samples, or (3) the recursion of any dimension has reached a maximal depth D_{max} . The reason for criterion 3 is to enforce a comparable degree of detail for any feature X_i in different pair-wise combinations X_i, X_j and X_i, X_k which is largely independent of X_j and X_k .

1D consumption percentiles (y-axis) / global radiation (x-axis)



2D avg. consumption (color) / day (x-axis) and temperature (y-axis)

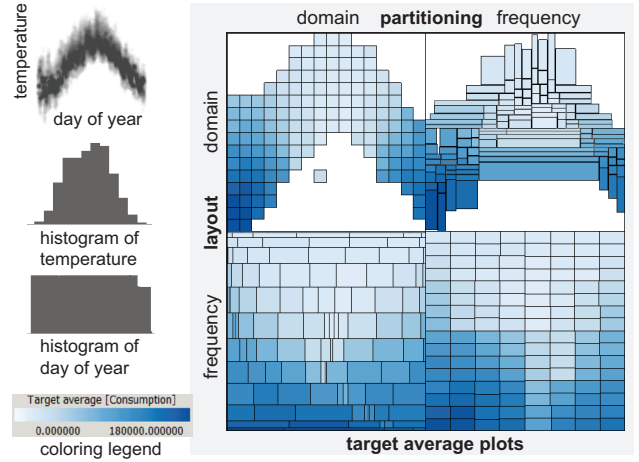


Fig. 3. Our design space of partition-based visualizations of relationships. While domain-preserving layouts are more intuitive to interpret, frequency-preserving layouts compensate for non-uniform distributions of X_i, X_j .

Without criterion 3, X_j being categorical could lead to a much more fine-grained subdivision of X_i than achieved for X_k being continuous. In general, D_{max} is the key parameter for adjusting the degree of detail while S_{min} ensures the significance for subsequent processing independently of the number of data samples.

An alternative to domain-uniform and frequency-uniform partitioning could be to maximize homogeneity of a region with respect to the structure of $P(T|X_i, X_j)$, as done for building regression trees [7]. However, finding optimal positions for splitting involves more computational effort, contradicting our requirement of fast computation. Moreover, changing T in the course of a workflow also requires a complete re-computation of the partitioning, which is not the case for domain- and frequency-uniform partitioning. For these reasons, our implementation of the framework currently does not support partitioning approaches that depend on the structure of $P(T|X_i, X_j)$. Conceptually, however, supporting these approaches would be compatible with the visualization and ranking mechanisms described below, provided that the shape of the resulting regions complies with the requirements stated above.

3.2 Partition-Based Visualization of Relationships

As motivated above, the key idea of our framework is to support an analysis of local variations of the conditional distribution $P(T|X_i, X_j)$ by partitioning X_i, X_j into disjoint regions. This section discusses considerations regarding the representation of this partitioning for visualization. As opposed to quantitative relevance measures (see Sec. 3.3), the goal of the visualization is to convey qualitative aspects of relationships such as location, shape, and significance of structures. In addition to considerations regarding the partitioning itself as discussed in Sec. 3.1, we identified two central design issues regarding partition-based visualizations of $P(T|X_i, X_j)$: How to layout regions within a plot, and how to visually represent $P(T|X_i, X_j)$.

3.2.1 Layout

As for partitioning, the size of each region R_k can either be interpreted as the covered part of the domain X_i, X_j or as the number of contained samples, i.e., the relative frequency of R_k . Our framework consequently discriminates two options for using the visual attribute *space* in order to assign a size and a location to each R_k . As will be discussed below, these layout options affect the X-axis for 1D domains and both axes for 2D domains (see Fig. 3).

Domain-preserving layout. Space is used to linearly represent the domain X_i, X_j between the minimal and maximal values of data samples in X_i, X_j . As for traditional function plots, extents of structures in X_i, X_j are thus directly perceptible.

Frequency-preserving layout. Space is used to represent the relative frequency of each region, i.e., the X-axis in the 1D case or the

entire plot in the 2D case represent 100% of the data. This layout thus generates a space-filling visualization as discussed extensively in the literature [4]. In 2D, the layout depends on how the data has been partitioned. For frequency-uniform partitioning, we directly represent the hierarchical structure of the subdivision, i.e., at each hierarchy level, the split of the respective axis is proportional to the frequency of the hierarchy nodes. For domain-uniform partitioning, we first subdivide the visual space in proportion to the feature being distributed more uniformly, and then to the other one (compare to Mosaic plots [15]). The benefit of a frequency-preserving layout is the optimal utilization of visual space and the direct perception of the significance of regions. The main drawback is a difficult interpretation regarding the extents and relative positions of regions in X_i, X_j .

In our framework, options for partitioning X_i, X_j and for layout can be chosen independently from each other. This defines a design space of partition-based visualizations where each combination has different advantages and disadvantages (see Fig. 3). In general, a suitable partitioning for visualization depends on the distribution of data samples. Less uniform distributions typically increase the necessity of distortion by frequency-uniform partitioning in order to guarantee a significant degree of detail for dense areas. To ensure flexibility, the partitioning granularity is controlled by the user. As a commonly used choice, we set the default number of splits per dimension to $\sqrt[4]{n}$ for domain-uniform mode, with n being the number of samples. For frequency-uniform mode, we use $D_{max} = 4$ and $S_{min} = 10$ as default subdivision limits. A suitable layout depends on the task. In context of model building, for example, detecting transformations benefits from a domain-preserving layout, while assessing the significance of local structures requires a frequency-preserving layout. We briefly discuss each combination individually:

Domain-uniform partitioning / domain-preserving layout. In our experience, this combination is the easiest to interpret. While particularly useful if large parts of X_i, X_j are uniformly distributed, entirely disregarding the frequency of regions introduces a visual bias for non-uniform distributions and makes it very sensitive to outliers.

Frequency-uniform partitioning / domain-preserving layout. This combination may be a suitable compromise to avoid distortion for non-uniform distributions. It is less sensitive to outliers which are included in outer regions. As a non-intuitive aspect, however, the different size of regions may falsely suggest a different significance and makes very dense regions difficult to perceive.

Domain-uniform partitioning / frequency-preserving layout. This combination is suitable if domain-uniform partitioning is required for application-specific reasons, but the significance must be visualized due to a non-uniform distribution of X_i, X_j . However, the partitioning may provide an insufficient resolution for dense regions.

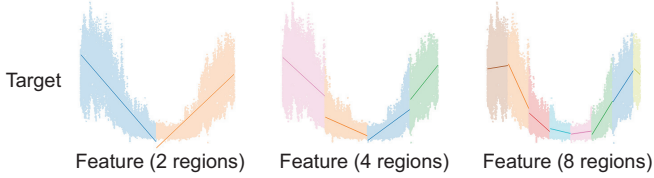


Fig. 4. The goodness-of-fit varies with the number of recursive subdivisions performed by a piece-wise linear ranking model Q_{X_i} .

Frequency-uniform partitioning / frequency-preserving layout.

This is the most effective combination to compensate for non-uniform distributions and outliers. A sufficient degree of detail is provided also for very dense regions. The layout ensures a sufficient size for perceiving the result at the cost of introducing a potentially significant distortion regarding the location of regions in $X_i[X_j]$.

3.2.2 Representation

After assigning a size and location to each region R_k , a key design issue concerns the visualization of the distribution $P(T|X_i[X_j])$. We distinguish between visualizing features and pairs of features (Fig. 3).

Visualization of $P(T|X_i)$. While the X-axis is used to represent the domain or the relative frequencies, the Y-axis depicts $P(T|X_i)$. Many options have been proposed in literature to visualize univariate distributions, e.g., variants of box plots [46, 35] and color-based histograms [28]. Very similar to box plots, our approach displays the median (black line), the quartiles (dark gray) and the 0.05 and 0.95 percentiles (light gray). As the main benefit, visualizing the median along multiple regions resembles familiar function graphs and the local dispersion is directly readable. The main drawback concerns the inability to adequately visualize multi-modal distributions.

Visualization of $P(T|X_i[X_j])$. In this case, the layout defines both axes and the visual proportions of each region may vary significantly, making a direct representation of $P(T|X_i[X_j])$ difficult. In order to limit the visual complexity, our current implementation visualizes a single distribution measure at a time by color, i.e., the average, the median, the variance, or the interquartile range. Depending on the task, the user may choose between a linear and a diverging transfer function (see Sec. 3.4) and may adjust its scaling. In future work, we intend to experiment with techniques for displaying multiple aspects of $P(T|X_i[X_j])$ at the same time, e.g., using saliency to display variance.

3.3 Partition-Based Relevance Ranking of Features

While the visualization of relationships provides qualitative information, many applications also require quantitative measures. In particular, a purely visual inspection of a high-dimensional feature space X is impractical especially for a pair-wise analysis. This section thus discusses methods for ranking $X_i[X_j]$ by quantitative measures that express the relevance for $P(T|X_i[X_j])$. In statistics, a common approach to automated feature selection is based on fitting a regression model for each candidate and ranking respective goodness-of-fit measures (also known as wrapper approach to feature ranking [27]). We adapt this approach by building a separate model $Q_{X_i[X_j]}$ for each $X_i[X_j]$ in a way that flexibly adapts to the structure of $P(T|X_i[X_j])$. As discussed in Sec. 3.1, regression trees comply with this requirement [7, 21] and are used as the model type of $Q_{X_i[X_j]}$. More specifically, we build piece-wise linear regression trees in order to exploit local linearity [36]. The hierarchical subdivision of $Q_{X_i[X_j]}$ (i.e., the tree) is based on frequency-uniform partitioning in order to enable an adaptation to non-uniform distributions. Conceptually, however, piece-wise linear models in our framework may be based on any subdivision approach, including domain-uniform partitioning or hierarchical subdivision approaches seeking optimal splits (see Sec. 3.1). The partitioning can be chosen independently for the visualization and the ranking, as they address different goals and face different constraints.

In automated approaches to model building, feature ranking is often used to incrementally refine an existing model M by adding or removing features (known as forward- or backward step-wise selection) [21]. This typically involves fitting variants of M that differ by the added or removed feature. In contrast, our ranking quantifies the relevance of

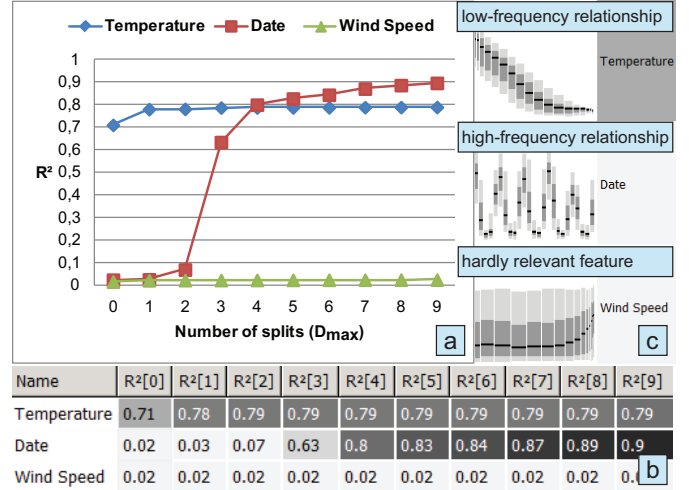


Fig. 5. The effect of increasing the complexity of Q_{X_i} on the measure R^2 for three features. (a, b) Goodness-of-fit curves as common in statistics are indicated by grayscales in our framework. They show the complexity of Q_{X_i} required to capture relationships of different frequencies (c).

$X_i[X_j]$ for $P(T|X_i[X_j])$ without making assumptions about the source of T (see Sec. 3.4). If used for interactively building a model M (Sec. 3.5), the models $Q_{X_i[X_j]}$ are independent from M with respect to the model type and complexity. Being used for an approximation of relevance rather than for prediction, $Q_{X_i[X_j]}$ also has a different purpose. For this reason, shortcomings of our type of regression trees are less problematic in our case, including discontinuities and a sub-optimal choice of split-points by frequency-uniform partitioning.

After fitting $Q_{X_i[X_j]}$, the quantification of relevance is based on the goodness-of-fit measure R^2 which is well-known and can be computed with linear effort [1]. Conceptually, integrating additional measures into our framework is straightforward (e.g., correlation measures).

As a general issue of statistical learning, model selection faces a trade-off between maximizing accuracy and minimizing model complexity, also known as the bias – variance trade-off [21]. In our case, the ability of $Q_{X_i[X_j]}$ to adapt to high-frequency structures depends on the number of splits which is determined by the parameter D_{max} as introduced in Sec. 3.1 (see Fig. 4). While a coarse subdivision is less prone to noise, the detection of complex structures may require a fine-grained subdivision. An appropriate model complexity thus depends on $P(T|X_i[X_j])$ and on domain knowledge about the features. In statistics, a common approach to analyze the effect of increasing model complexities is by plotting them against error metrics as curves (see Fig. 5a). Motivated by this approach, we compute a sequence $Seq\{Q_{X_i[X_j]}\}$ of models $Q_{X_i[X_j]}$ for each $X_i[X_j]$ for increasing values of D_{max} , and we compute R^2 measures for all variants of $Q_{X_i[X_j]}$.

As shown in Fig. 5, detecting high-frequency relationships requires more splits while the number of splits has hardly any effect on low-frequency relationships and irrelevant features. This holds as long as each leaf contains a significant number of samples, as ensured by the parameter S_{min} of frequency-preserving partitioning. For this reason, the ability to detect complex structures depends on the overall number of data samples, which is true in general for statistical learning [21].

The result of the quantification is shown as a table where columns represent increasing complexities of $Q_{X_i[X_j]}$ and rows correspond to the features or pairs of features $X_i[X_j]$ (see Fig. 5c). Each row thus represents a goodness-of-fit curve which is visually indicated by the background color of cells (see Fig. 5b). Vertically, each column can be considered a cut through the curves that can be used for ordering the table and for ranking the coordinated small-multiple display.

3.4 Applying the Framework to Model Building Tasks

The previous sections focused on task-independent concepts for ranking and visualizing relationships between features and a general quan-

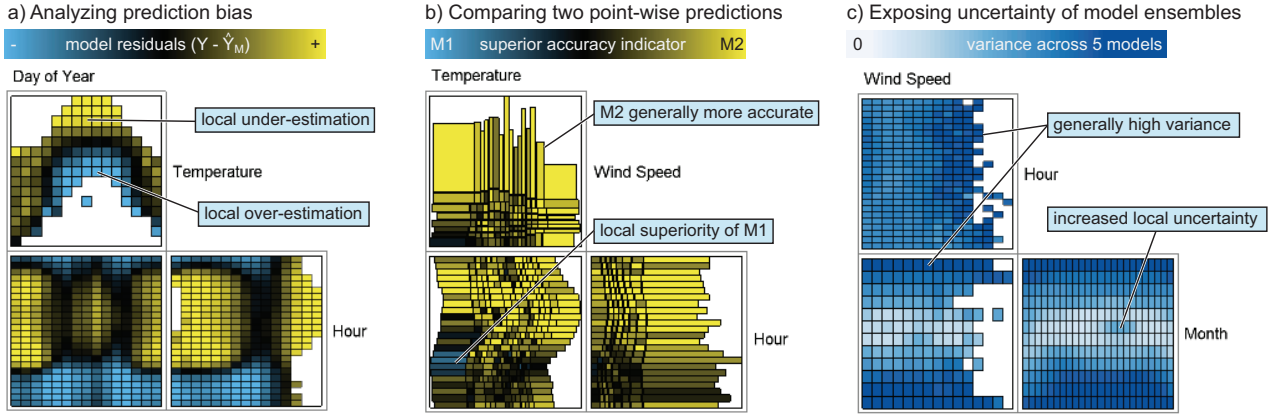


Fig. 6. Derived quantities as target T support different tasks in building regression models. (a) Residuals show the local prediction bias of a model, i.e., a tendency towards over- or under-estimation. (b) The difference of residual magnitudes indicates local superiority for a pair of models. (c) The point-wise variance of predictions by multiple models represents their local uncertainty.

titative target T . This section describes the application of the framework to common tasks in statistical modeling. The key idea is using different derived quantities as T . Henceforth, Y denotes actual observations of a dependent variable and \hat{Y}_M denotes corresponding predictions of Y by a model M . We identified the following set of tasks:

- **Identification of explaining features** ($T = Y$). Relating feature candidates to actual observations of Y helps in determining the features or pair-wise combinations of features having the strongest explanatory power (see Fig. 1). The direct visualization of $P(Y|X_i, X_j)$ resembles 1D and 2D function plots which typically makes the interpretation straightforward for domain experts. However, dominating relationships tend to obscure less distinct relationships for ranking and visualization (e.g., the effect of *Temperature* is dominating in Fig. 1).
- **Analysis of prediction bias** ($T = Y - \hat{Y}_M$). Visualizing the residuals of M reveals areas of over- or underestimation, i.e., the local bias of M . An appropriate scaling of T should be symmetric around the neutral value 0. In 2D, we use a diverging transfer function as suggested for this purpose [49] (see Fig. 6a). The prediction bias provides important information for detecting effects currently not captured by M . This includes relevant features being not yet part of M , in which case the prediction bias supports incremental feature selection (see Sec. 3.5). Another application is detecting an insufficient model complexity. For instance, modeling a non-linear effect of X_i by a linear term will show distinct areas of over- and underestimation in plots of X_i . In general, consulting the shape and size of areas comprising visually similar regions may facilitate identifying suitable transformations of features for model building. Conversely, small and incoherent areas often indicate noise rather than real effects.
- **Assessment of prediction accuracy** ($T = |Y - \hat{Y}_M|$). Visualizing the distribution of residual magnitudes of M reveals local differences in the prediction quality, exposing badly fitted areas.
- **Comparison of two models** ($T = |Y - \hat{Y}_{M1}| - |Y - \hat{Y}_{M2}|$). Visualizing the point-wise difference of residual magnitudes of the models $M1$ and $M2$ provides an overview of local model superiority (see Fig. 6b). The sign of the regional average of T indicates which model tends to be locally better (negative for $M1$, positive for $M2$), while the magnitude indicates by how much. The scaling of T is symmetric around 0, suggesting a diverging transfer function. Typical applications include model selection and the identification of composite models. In this case, ranking supports the selection of useful classifiers and the visualization may suggest decision boundaries.
- **Exposing uncertainty of model ensembles** ($T = \text{Var}(\hat{Y}_{M1} \dots \hat{Y}_{Mn})$). In this case, T is the point-wise variance of predictions of Y by the models $M1$ to Mn . In other

words, for the k^{th} record of the dataset, the n predictions \hat{y}_{kM1} to \hat{y}_{kMn} are aggregated by their variance or other measures of dispersion. Sources of model ensembles include different training data sets, variation of model-specific parameters, and different types of prediction models. A common application of ensemble data is analyzing the uncertainty of a prediction [22]. Our framework supports the identification of areas in 1D or 2D feature sub-spaces causing uncertainty (see Fig. 6c).

It should be noted that the tasks involving models operate solely on point-wise predictions of these models. They neither make assumptions about M , nor is access to an evaluable representation of M required. This makes the framework applicable to the validation and comparison of any type of quantitative prediction from any source. In the context of renewable energy, assessing and comparing forecasts of meteorological quantities from different providers is of great practical importance (e.g., day-ahead forecasts of temperature at a specific location). In this case, the prediction is based on physical rather than statistical models. Analysts in the energy sector do not have access to such models themselves, but still, the framework has successfully been applied for assessment and (composite) selection of providers.

3.5 Interactive Feature Subset Selection

This section describes extensions to the framework supporting an interactive workflow for feature selection (Sec. 5.1 illustrates an example). The principle of the workflow is based on forward selection of features in step-wise regression [1]. The key idea is to iteratively add features and transformations thereof to a model predicting a dependent variable Y . Each iteration seeks to reduce the remaining variance while ensuring that the selection is reasonable according to the domain knowledge of the user. In contrast to previous sections, this workflow requires the ability to create an evaluable regression model M for any number of features by fitting M to existing training data. A prerequisite of the workflow is thus the availability of training data D_T . In order to avoid overfitting, we also support the discrimination of separate validation data D_V for visualization, goodness-of-fit quantification and ranking. Both D_T and D_V must contain known values of Y .

We distinguish between two stages: During **initial model identification**, M does not yet exist and the framework shows the actual observations (i.e., $T = Y$). The goal of this stage is to verify the existence of useful features, potentially inferring a particular regression model *type* from the structure of relationships, and building an initial model M_1 based on a relevant feature or pair of features. The subsequent **model refinement** stage analyzes the local bias of a current version M_i of the model (i.e., $T = Y - \hat{Y}_{M_i}$). The goal of this stage is to identify relevant additional (transformations of) features for fitting M_{i+1} by extending the independent variables of M_i and continuing with model refinement, or to quit the workflow.

Our framework supports both stages, e.g., comparing different measures for ranking (pairs of) features with respect to T and partitioning

the data for visualization depending on the distribution of samples. Features can be added to M_i by clicking on their visual representation. This triggers the fitting of M_{i+1} which is set as the current model variant after completion, updating the ranking and visualization to consider the residuals of M_{i+1} . As a desirable effect, including a feature in M_{i+1} reduces the explanatory power of redundantly correlated features which are ranked lower in the next iteration as well. During model refinement, a list called *Quantitative Model Overview* (QMO) displays the root-mean-square-error (RMSE) and optionally also the global bias (i.e., the average of $Y - \hat{Y}_{M_i}$) for all variants of M . The QMO thus quantifies the gained accuracy for each iteration. Being computed on D_V , increasing model complexities may cause increasing values of the RMSE, which is a typical stopping criterion [21].

Additional feature candidates can be added to the investigation at any time, as well as transformations of features. An example offered by our implementation is a user-defined categorization of continuous values. This can facilitate the modeling of differently structured areas by fitting separate models for different parts of the data (i.e., building treed models, see Sec. 5.1). Other examples include bivariate feature transformations like multiplication in order to model interactions, as well as simple transformations like squaring and taking the logarithm. However, the interactive specification of transformations is a topic in its own right and details are beyond the scope of this paper.

There are several options for extending the workflow. First, visualizing M_i as a high-dimensional function during model refinement provides additional means for validation. Our implementation of the framework offers an interactive visualization based on hyper-slices [32] for this purpose (see Sec. 4). Second, multivariate visualizations like *parallel coordinates* help to relate the distribution of residuals across multiple variants of M . Third, it may often be reasonable to return to previous variants of M and to try out and compare different choices of features, e.g., if the QMO shows only modest gains of accuracy. Our implementation preserves previous model variants and supports back-ward steps. However, providing an adequate visual support for hierarchical branching of models is up to future work.

A limitation of assessing single $X_i[X_j]$ for step-wise model refinement is that useful higher-dimensional interactions of individually weak features might not get noticed. In contrast to best-subset selection methods (e.g. see Hastie [21]), manual step-wise selection is not guaranteed to produce feature subsets yielding a minimal RMSE, especially in the context of high-dimensional data ($|X| \gg 10$). However, a model with the minimal RMSE is not necessarily the best choice in a given application context. Additional reasons for choosing a step-wise approach are a superior run-time performance, comprehensibility and straightforward incorporation of expert knowledge. While identifying two-dimensional interactions is supported directly, a detection of higher-dimensional interactions is left for future work (see Section 6).

An application by real users (Sec. 5) has shown that this workflow supports two tasks. First, it supports interactive feature selection for building interpretable regression models. Conceptually, the workflow is applicable to any type of regression model. However, training times of at most several seconds are beneficial for smooth working. As the second task, the workflow supports the detection of more subtle relationships which are otherwise masked by more dominating effects. In this case, the model itself is of less interest, as it is rather used to subtract dominating effects from the data, exposing more subtle ones.

4 SYSTEM INTEGRATION AND IMPLEMENTATION

Our framework has been implemented as part of Visplore, a system for visual exploration and model building. Additional views of Visplore like histograms, scatterplots, and parallel coordinates support a flexible analysis of multivariate data by linked ad-hoc selections and derived data columns. In context of model building, they enable an interactive specification of training and validation data for ensuring an appropriate data quality (e.g., by removing outliers). Regression models can be identified and managed by the user. Supported types of models currently include generalized linear models, support vector regression based on the library LIBSVM [10], and piece-wise linear regression trees. Internally, a common interface for fitting and evalu-

ation enables an integration of additional model types. An implementation of HyperMoVal [32] supports a detailed point-wise validation of identified regression models (see Fig. 7i). All parts of Visplore implement a multi-threading architecture [33] to maintain interactivity regardless of the data size and the effort of involved computations. In case of the proposed framework, multi-threading is used for computing the relevance measures and the visualization. Intermediate results such as subsets of plots or ranking measures are displayed as soon as they become available in order to minimize delays. All parts are written in C++ and use OpenGL for rendering.

Regarding the performance of frequency-uniform partitioning, storing the order of values for each feature as a re-usable index enables an efficient implementation also for analyzing feature pairs. Specifically, computing the indices of 35 continuous features and 42869 data samples took 0.03 seconds in our implementation (recorded on an Intel i7-2600k CPU @ 3,4 Ghz). Computing the partitioning with D_{max} set to 10 and S_{min} set to 8 took another 0.19 seconds for the 35 features (1D) and 3.30 seconds for all 630 feature pairs (2D). Regarding the performance of ranking, computing the measures took additionally 0.38 seconds in 1D and 11.8 seconds in 2D. As a computationally cheaper yet less accurate alternative to fitting a linear model per region, fitting a constant model (i.e., the median value of each region) only took 0.15 seconds in 1D and 4.44 seconds in 2D. In general, computing percentiles of the distribution $P(T, X_i[X_j])$ as also required for visualization benefits from storing the order of T as an index, enabling linear effort and re-usage across all $X_i[X_j]$.

5 EVALUATION

For evaluating our framework, Sec. 5.1 demonstrates a case study of interactive feature selection in the energy sector. Sec. 5.2 then reports user feedback by 11 analysts after two months of deployment.

5.1 Case Study: Modeling Natural Gas Consumption

This section demonstrates our framework by building a regression model predicting the natural gas consumption of a large city as the dependent variable Y . Based on real data, this case study has been conducted by an analyst in the energy sector to investigate the influence of meteorological and calendric aspects as the independent variables X . This represents a direct application of the workflow described in Sec. 3.5. The data comprise hourly measurements for approximately five years (42869 samples) which are split into three years of training data D_T and two years of validation data D_V (annually interleaved).

For initial model identification, the 1D overview shows the conditional distribution of the consumption for each feature, i.e. $T = Y$ (Fig. 7a). Ranking the features by relevance immediately identifies *Temperature* and *Day of Year* as having a dominant effect on the target. Comparing their measures shows a slightly higher relevance of *Temperature* for coarse subdivisions while the relevance of *Day of Year* increases with the level of detail and exceeds *Temperature* for $D_{max} = 5$ (Fig. 7b). Knowing that the data only comprises 5 years, the analyst considers *Temperature* as the more useful feature for an initial model M_1 . Since the visualization suggests a non-linear relationship with at least one point of inflection, M_1 is fitted based on D_T as a third degree polynomial, i.e., a linear model including squared and cubic basis expansions. The Quantitative Model Overview shows an RMSE of 24853 units for D_V (Fig. 7c), confirming the information gain by M_1 as compared to the standard deviation of Y (52812 units).

Building M_1 updates the 1D overview for an analysis of its residuals for D_V in order to identify effects explaining the remaining variance, i.e., $T = Y - \hat{Y}_{M_1}$ (Fig. 7d). *Temperature* now ranks much lower as well as *Day of Year*, whose effect is partly captured by M_1 due to correlation with *Temperature*. In contrast, the ranking now identifies *Hour* as most relevant for the target. The visualization of the conditional distribution shows a consumption profile as a function having multiple local extrema (e.g., a distinct rise to a morning peak). This complex structure precludes simple low-degree polynomial basis expansions as before. Instead, the analyst categorizes *Hour* into *morning* [0am,6am), *day* [6am,8pm) and *evening* [8pm-0am) in order to build M_2 as a treed linear model. For each identified category of *Hour*, M_2 thus comprises a separate function including linear, squared, and cubic

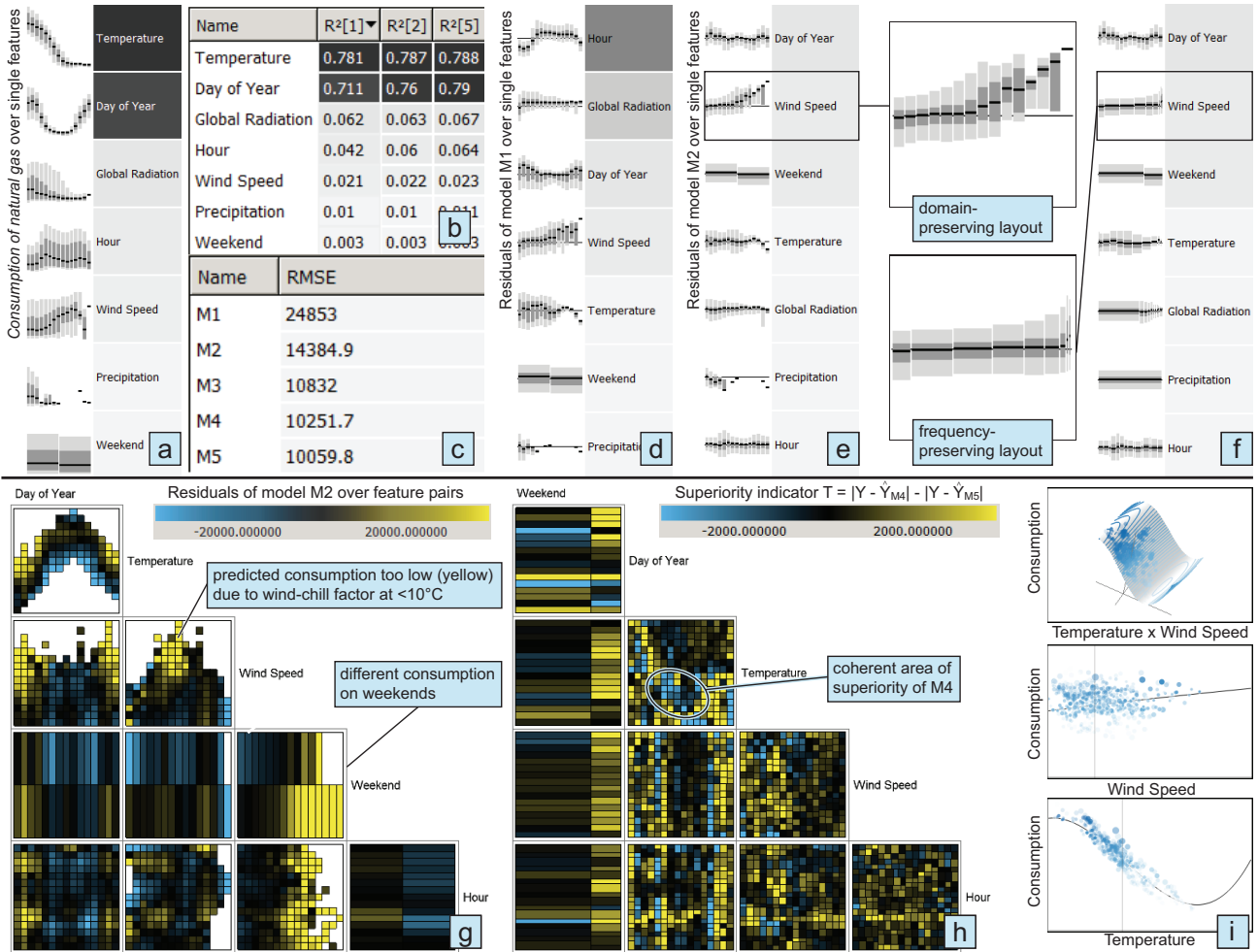


Fig. 7. A case study for model building. (a, b) Ranked overviews suggest *Temperature* as most relevant for predicting the target *Natural Gas Consumption* by a model *M1*. (c, d) Analyzing the local prediction bias suggests *Hour* as additional feature for reducing the error measure RMSE. (e, f) Frequency-preserving layout reveals the insignificance of a trend caused by a non-uniform distribution of *Wind Speed*. (g) Analyzing the local prediction bias for feature pairs reveals multiple interactions that inform further model refinements (c). (h) Comparing two model variants enables an assessment of local model superiority. (i) Additional views support an application of the final model for sensitivity analysis.

terms for *Temperature* as well as linear and squared terms for *Hour*. This enables a substantial reduction of the RMSE to 14384 units.

Another update of the 1D overview to analyze the residuals of *M2* ($T = Y - \hat{Y}_{M2}$) shows that the effect of *Hour* is captured well (Fig. 7e). Being correlated with *Hour*, the relevance of the feature *Global Radiation* is also reduced while *Day of Year*, *Wind Speed*, and a classification of days into weekends and working days lead the ranking. The visualization of *Wind Speed* suggests a strong effect which seemingly contradicts its ranking below *Day of Year*. However, switching the layout to frequency-preserving reveals the low significance of high wind speeds due to the sparsity of the data (Fig. 7f). Since no single feature seems to explain the remaining variance well, the analyst now turns to inspecting pair-wise interactions of features in the 2D overview.

Considering the average local prediction bias ($T = Y - \hat{Y}_{M2}$) for visualization and ranking in fact suggests useful pair-wise interactions of *Day of Year*, *Temperature* and *Wind Speed* (Fig. 7g shows the matrix for the five top-ranking features). The top-ranking pair reveals that the effect of *Temperature* significantly depends on the time of the year. Another plot shows a substantial underestimation for high wind speeds at low temperatures. The analyst hypothesizes that the reason might be a meteorological effect known as "wind-chill factor". While previous 1D overviews indicated a general tendency of increased consumption at high wind speed, the analysis of interactions enables a more comprehensive understanding of the influence of *Wind Speed*. Furthermore, 1D and 2D views suggest a general overestimation of

the consumption on weekends, e.g., due to the different consumption by industry. Capturing these effects by refining *M2* enables a further reduction of the RMSE for D_V (Fig. 7c): *M3* extends *M2* by adding cubic, squared and linear terms for *Day of Year* and refines the regression tree by a discrimination of *summer* (April to Sept.) and *winter* (remaining months). *M4* further refines the tree based on *Weekend*. Finally, *M5* extends *M4* by adding linear, squared, and cubic terms for *Wind Speed* plus interactions of the form $A \cdot B$, $A^2 \cdot B$ and $A \cdot B^2$ between *Wind Speed* and *Temperature* to account for the wind-chill factor.

Compared to *M4*, however, the significant additional complexity of *M5* only reflects in a modest reduction of the RMSE. In order to validate the superiority of *M5*, assigning the difference of residual magnitudes as target of the 2D overview enables a local comparison of *M4* and *M5* ($T = |Y - \hat{Y}_{M4}| - |Y - \hat{Y}_{M5}|$). In order to compensate for non-uniform distributions of features like *Temperature* and *Wind Speed*, the analyst applies the frequency-based partitioning and the frequency-preserving layout (Fig. 7h). While the dominance of yellow tones confirms the superiority of *M5* for large parts of the domain, the visualization also indicates areas where *M4* is superior. The analyst is surprised that considering *Wind Speed* increased the prediction accuracy especially for weekends while a coherent blue area in the combination of *Day of Year* and *Temperature* indicates a negative effect for certain temperatures especially during spring and summer. In general, however, the analyst is satisfied with *M5* as the final result of the workflow. An implementation of HyperMoVal [32] as an addi-

tional view of the system enables a detailed follow-up analysis of *M5*, e.g. regarding a sensitivity analysis of natural gas consumption and a model-based detection of outlying data samples (Fig. 7i).

5.2 User Feedback

Our framework has been deployed to 11 experts of two companies in the energy sector, i.e., an IT-service provider and a national power grid operator. The growing share of renewable energy and the advent of smart grids increasingly necessitate accurate prediction for risk management in this field. The experts have been dealing with prediction models for years and use MARS [14] as the prevailing model type. They have been using our framework on a daily basis for two months. While operational models are still built using external software, the experts employ our framework for the identification of useful features, interactions, and transformations of features as well as for the validation and comparison of identified (MARS-)models.

Before using our framework, these tasks were based on the inspection of data tables, static graphics, and correlation coefficients in tools like Excel and Matlab. They reported that generating, validating and comparing models was intransparent and required extensive trial-and-error. Establishing and validating hypotheses for new data or new models required approximately *the work of one day*.

According to the experts, our framework enables them to obtain the same insights within *half an hour*. A formerly empirical process of knowledge acquisition has been turned into a systematic one, saving substantial amounts of time. They consider the involved visualization as intuitive and fast to interpret and also suitable for a presentation to decision-makers and other stake holders. One expert stated that the process of communicating findings and arguing model deficiencies to end customers in the energy sector has been sped up from hours or even days to minutes using our visualizations.

Technologically, one analyst claimed that our ranking mechanism is more helpful in analyzing relationships than previously used correlation metrics, as it unveils non-linear structures of arbitrary shape. The 1D- and 2D-visualizations are consulted at a ratio of around 30:70 percent during the analysis, as interactions of two or more features generally play a very important role. The analysts generally prefer domain-uniform partitioning and -layout for their superior interpretability, but they usually employ the frequency-preserving approaches to check the significance of unexpected findings. In conclusion, the interviewed domain experts envision a high relevance of our framework for the energy sector. Their key suggestion for future work concerned a direct integration of the model type MARS in our framework.

6 DISCUSSION AND FUTURE WORK

As the key idea of Visual Analytics, our framework tightly integrates visualization, computation, and interaction at three levels. First, quantitative measures based on regression trees rank visualizations by relevance. Second, visualizing derived quantities supports diverse tasks in model building. Third, tightly coupling model visualization with model training enables an efficient loop of incremental discovery, refinement, and validation. Our framework thus supports all elements of the Visual Analytics Process as described by Keim et al. [26].

Furthermore, our framework addresses all six high-level tasks of visualization-based knowledge discovery as defined by Amar and Stasko [2]: 1) It *exposes uncertainty* of single models by showing the local variance of their residuals and of model ensembles by visualizing their point-wise variance. 2) It *concretizes relationships* by depicting and quantifying the conditional distribution of targets over domains of features and pairs of features. 3) It supports to *formulate cause and effect* by explicitly distinguishing between dependent and independent variables and expressing their relationship as regression model for investigation. 4) It directly addresses the *determination of domain parameters* by the workflow for step-wise feature selection. 5) It enables a *multivariate explanation* by considering pair-wise interactions between features as well as via the identification of multi-dimensional regression models. 6) It *confirms hypotheses* which are formulated as target dimensions or prediction models by visualizing the local structure of their conditional distribution.

Regarding scalability, a key benefit of partitioning is to avoid clutter for any number of data samples. The goal to enable interactive workflows restricts the computational complexity of methods for partitioning and ranking, which informed several design decisions as discussed in previous sections. The achieved performance supports tens of thousands of data samples and dozens of features even for a pair-wise analysis (see the measurements in Sec. 4) and can further be increased by using piece-wise constant rather than linear regression trees for ranking. In fact, sparse data is much more limiting the detection of significant relationships than large data which is a general problem of statistical learning [21]. Due to ranking features by relevance, the framework scales well for an individual inspection of truly high-dimensional data (i.e., hundreds of dimensions). A pair-wise analysis is inherently more challenging due to a quadratic growth of combinations. However, ranking also supports this case and enables to show only the most relevant part of the matrix.

Operating on generic categorical and continuous data, the approach is generally applicable to regression tasks in any domain. While the examples and the evaluation in this paper refer to the energy sector, preliminary tests also indicated a direct applicability to regression tasks in engineering, process optimization, and clinical trial analysis.

We see many directions for future work. 1) Partition-based ranking is conceptually also applicable to higher-order interactions but faces challenges regarding the exponential growth of combinations and the visualization. We intend to address these aspects for triples of features involving volume visualization for representation. 2) We intend to design and evaluate concepts to simultaneously visualize bias and variance of distributions in 2D. 3) While the current workflow supports a rather linear process for model building, we intend to design concepts for addressing a hierarchical process, i.e., supporting multiple model variants as refinements of a common base model. 4) The identification of feature transformations is currently solely based on the interpretation of the visualization by the user. An automated suggestion of suitable transformations could be an important help. 5) As suggested by the experts evaluating our approach, we intend to integrate additional types of regression models (e.g., MARS [14]) or even support a direct integration with statistics software such as R [25]. 6) While explicitly designed for regression, we intend to investigate an adaptation of the framework for classification.

7 CONCLUSION

This paper proposed a partition-based framework to support multiple tasks related to building regression models. As a key benefit, the framework provides a global overview over local relationships of any structure for features and pairs of features. We described a model-based method for quantifying relationships that provides guidance by ranking relationships for an efficient investigation of high-dimensional feature spaces. Both ranking and visualization flexibly adapt to non-uniform distributions as well as categorical features, and are computationally sufficiently inexpensive to scale for large and high-dimensional data. We discussed the application to a variety of tasks in building and validating regression models. A workflow for interactive model building enables a seamless integration of domain knowledge in the selection of features and transformations, and it supports a discovery of subtle relationships by compensating for dominant effects using regression. A real-world case study illustrated the application for building a complex model, and feedback by analysts in the energy sector suggested a significant effort reduction for model building. Motivated by these results, we believe that our framework will have a positive impact on regression in many fields.


ACKNOWLEDGMENTS

This work has been supported by the Austrian Funding Agency (FFG) within the scope of the COMET K1 program. Thanks go to all project participants of Hakom and Austrian Power Grid AG, and to E. Gröller, M. Buchetics, S. Pajer, and J. Kehrer for valuable comments.

REFERENCES

- [1] A. Agresti and B. Finlay. *Statistical Methods for the Social Sciences*. Pearson, fourth edition, 2007.
- [2] R. A. Amar and J. T. Stasko. A knowledge task-based framework for design and evaluation of information visualizations. In *Proc. IEEE Symp. on Information Visualization 2004 (InfoVis 2004)*, pages 143–150, 2004.
- [3] S. Barlowe, T. Zhang, Y. Liu, J. Yang, and D. J. Jacobs. Multivariate visual explanation for high dimensional datasets. In *Proc. of the 3rd IEEE Symp. on Visual Analytics Science and Technology (VAST 2008)*, pages 147–154, 2008.
- [4] T. Baudel and B. Broeskema. Capturing the design space of sequential space-filling layouts. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2593–2602, Dec. 2012.
- [5] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.
- [6] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty-aware exploration of continuous parameter spaces using multivariate prediction. *Comput. Graph. Forum*, 30(3):911–920, 2011.
- [7] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth and Brooks, 1984.
- [8] E. T. Brown, J. Liu, C. E. Brodley, and R. Chang. Dis-function: Learning distance functions interactively. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2012)*, pages 83–92, 2012.
- [9] Y.-H. Chan, C. Correa, and K.-L. Ma. Flow-based scatterplots for sensitivity analysis. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2010)*, pages 43–50, 2010.
- [10] C. Chang and C. Lin. LIB-SVM. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, Last visited 2013-03-04.
- [11] S. Cleveland and M. E. McGill, editors. *Dynamic Graphics for Statistics*. Wadsworth and Brooks/Cole, 1988.
- [12] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.
- [13] M. A. Effroymsen. Multiple regression analysis. In A. Ralston and H. S. Wilf, editors, *Mathematical Models for Digital Computers*, pages 191–203. 1960.
- [14] J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.
- [15] M. Friendly. Extending mosaic displays: Marginal, conditional, and partial views of categorical data. *Journal of Computational and Graphical Statistics*, 8:373–395, 1999.
- [16] S. Gerber, P. Bremer, V. Pascucci, and R. Whitaker. Visual Exploration of High Dimensional Scalar Functions. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1271–1280, 2010.
- [17] Z. Guo, M. Ward, E. Rundensteiner, and C. Ruiz. Pointwise local pattern exploration for sensitivity analysis. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2011)*, pages 131–140, 2011.
- [18] Z. Guo, M. O. Ward, and E. A. Rundensteiner. Model space visualization for multivariate linear trend discovery. In *Proc. of the 4th IEEE Symp. on Visual Analytics Science and Technology (VAST 2009)*, pages 75–82, 2009.
- [19] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, Mar. 2003.
- [20] M. C. Hao, U. Dayal, R. Sharma, D. Keim, and H. Janetzko. Variable Binned Scatter Plots. *Information Visualization*, 9(3):194 – 203, 2010.
- [21] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Second Edition*. Springer New York Inc., 2009.
- [22] J. C. Helton. Uncertainty and sensitivity analysis for models of complex systems. In F. Graziani, editor, *Computational Methods in Transport: Verification and Validation, Vol. 62*, pages 207–228. Springer, 2008.
- [23] A. Inselberg and B. Dimsdale. Parallel coordinates for visualizing multi-dimensional geometry. In *Computer Graphics 1987 (Proc. of CG International '87)*, pages 25–44, 1987.
- [24] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Enterprise data analysis and visualization: An interview study. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2917–2926, 2012.
- [25] J. Kehrler, R. N. Boubela, P. Filzmoser, and H. Piringer. A generic model for the integration of interactive visualization and statistical computing using R. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2012)*, pages 233–234, 2012.
- [26] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler. Visual analytics: Scope and challenges. In S. J. Simoff, M. H. Böhlen, and A. Mazeika, editors, *Visual Data Mining*, pages 76–90. Springer, 2008.
- [27] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1):273–324, 1997.
- [28] R. Kosara, F. Bendix, and H. Hauser. Time histograms for large, time-dependent data. In *Proc. of the 6th Joint IEEE TCVG - EUROGRAPHICS Symp. on Visualization (VisSym 2004)*, pages 45–54, 2004.
- [29] T. May, A. Bannach, J. Davey, T. Ruppert, and J. Kohlhammer. Guiding feature subset selection with an interactive visualization. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2011)*, pages 111–120, 2011.
- [30] E. J. Nam, Y. Han, K. Mueller, A. Zelenyuk, and D. Imre. Clustersculptor: A visual analytics tool for high-dimensional data. In *Proc. of the 2nd IEEE Symp. on Visual Analytics Science and Technology (VAST 2007)*, pages 75–82, 2007.
- [31] H. Piringer, W. Berger, and H. Hauser. Quantifying and comparing features in high-dimensional datasets. In *Proc. of the 6th International Conf. on Coordinated & Multiple Views in Exploratory Visualization (CMV2008)*, pages 240–245, 2008.
- [32] H. Piringer, W. Berger, and J. Krasser. Hypermovall: Interactive visual validation of regression models for real-time simulation. *Comput. Graph. Forum*, 29(3):983–992, 2010.
- [33] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A multi-threading architecture to support interactive visual exploration. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1113–1120, Nov. 2009.
- [34] M. A. Pitt, W. Kim, D. J. Navarro, and J. I. Myung. Global model analysis by parameter space partitioning. *Psychological Review*, 113:57–83, 2006.
- [35] K. Potter, J. Kniss, R. F. Riesenfeld, and C. R. Johnson. Visualizing summary statistics and uncertainty. *Comput. Graph. Forum*, 29(3):823–832, 2010.
- [36] J. R. Quinlan. Learning with continuous classes. In *Proc. of the 5th Australian Joint Conf. on Artificial Intelligence*, pages 343–348. World Scientific, 1992.
- [37] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *Science*, 334(6062):1518–1524, 2011.
- [38] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187–260, 1984.
- [39] J. Seo and B. Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *Proc. IEEE Symp. on Information Visualization 2004 (InfoVis 2004)*, pages 65–72, 2004.
- [40] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.
- [41] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Comput. Graph. Forum*, 28(3):831–838, 2009.
- [42] A. Slingsby, J. Dykes, and J. Wood. Configuring Hierarchical Layouts to Address Research Questions. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):977–984, 2009.
- [43] R. Snee. Validation of regression models: Methods and examples. *Technometrics*, 19(4):415–428, November 1977.
- [44] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. Magnor, and D. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proc. of the 4th IEEE Symp. on Visual Analytics Science and Technology (VAST 2009)*, pages 59–66, 2009.
- [45] T. Torsney-Weir, A. Saad, T. Möller, H.-C. Hege, B. Weber, and J.-M. Verbavatz. Tuner: Principled parameter finding for image segmentation algorithms using visual response surface exploration. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):1892–1901, 2011.
- [46] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [47] S. van den Elzen and J. J. van Wijk. Baobabview: Interactive construction and analysis of decision trees. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2011)*, pages 151–160, 2011.
- [48] J. van Wijk and R. van Liere. HyperSlice: Visualization of Scalar Functions of Many Variables. In *Proc. of the 4th Conf. on Visualization*, pages 119–125, 1993.
- [49] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., 2004.
- [50] L. Wilkinson, A. Anand, and R. Grossman. Graph-Theoretic Scagnostics. In *Proc. IEEE Symp. on Information Visualization 2005 (InfoVis 2005)*, pages 21–28, 2005.

PAPER

D 

Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System

Published in *Proceedings of IEEE Vis Conference 2014, Practitioner Experience Track (poster paper)*, 2014, [MP14]

Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System

Thomas Mühlbacher*
VRVis Research Center

Harald Piringer†
VRVis Research Center

ABSTRACT

This poster presents lessons learned from deploying the visual analytics system *Visplore* as an extension of a time series management software in the energy sector. *Visplore* addresses a variety of tasks in analysis and statistical modeling. Without guidance, however, our experience showed that new users often have difficulties to find effective setups for their tasks. In this poster, we describe task-tailored dashboards with restricted flexibility as one approach to improve the adoption by target users. Based on a use case of correlation analysis, we illustrate how dashboards allow users to address tasks without extensive training of the visualization software. We demonstrate how integrating *Visplore* dashboards with a time series management tool enables to offer a visual frontend for analysis and data selection. We report preliminary experience feedback, and we discuss challenges and opportunities of dashboards compared to software with unrestricted flexibility.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—User-Centered Design

1 BACKGROUND

Statistical modeling for forecasting involves extensive preprocessing and analysis of time series. To this end, our project partner HAKOM Solutions distributes a software for time series management to companies in the energy sector. This *time series manager* (TSM) supports the integration of multiple data sources and operations such as the resampling of time rasters, but offers very limited visualization capabilities.

As an extension to the TSM addressing this limitation, HAKOM distributes *Visplore*, our system for visual exploration and statistical modeling. *Visplore* supports a broad range of *views*, including scatter plots, line graphs or parallel coordinates, as well as specialized visualizations for model building and validation [2, 4]. The user may flexibly create, parameterize and layout any number of views. All views are linked by ad-hoc selections and derived data columns, and implement a multi-threading architecture to enhance large data scalability [5]. In context of the cooperation with HAKOM, the goal is to establish *Visplore* as a flexible visualization and analysis frontend for time series-based modeling tasks in the energy sector, that can be deployed along with the TSM.

As a first attempt, we integrated the full, unrestricted version of *Visplore* with HAKOM’s TSM. Time series from a database could be imported into an empty workbench, and analyzed using *Visplore*’s entire feature palette. This flexibility matched the requirements of expert users very well, as also shown by the evaluations of published *Visplore* views and workflows (e.g., [2]). However, our experience showed that it often asked too much of new users who had difficulties to find effective view configurations for a given task. Inspired by the general trend towards guidance in visual analysis, as well as the success of dashboards in Tableau [6] and other

business intelligence solutions, we decided to investigate whether task-specific dashboards with restricted flexibility would improve the adoption by our target users.

2 TASK-TAILORED DASHBOARDS

Task-tailored dashboards are predefined configurations of parameterized views and interaction components like selections, that address one particular, well-defined task. For dashboards, the functionality of the system is deliberately restricted. In contrast to the unrestricted version, no additional views can be opened, and only a limited set of visual parameters can be controlled to minimize the complexity.

2.1 Identifying Tasks and Appropriate Dashboards

In cooperation with HAKOM and end customers in the energy sector, we first identified questions and tasks that might benefit from dashboards. The result included tasks such as data quality profiling, correlation analysis and the discovery of patterns like seasonal effects or anomalies, as well as model-related tasks such as identifying suitable training data, comparing the accuracy of prediction models, and analyzing the uncertainty of prediction ensembles. In an iterative process of suggestion and feedback, we then created *Visplore* dashboards addressing the identified tasks.

2.2 Example: A Dashboard for Correlation Analysis

Figure 1 shows a dashboard for analyzing pairwise correlations between a target variable (“GAS CONSUMPTION”) and explanatory variables. Calendar heat maps provide an overview of the target variable’s distribution over time with respect to hours, months, and days of the week. A scatter plot matrix shows pairwise correlations and Pearson correlation coefficients, as well as a detail plot for a selected pair of variables [3]. Selecting a subset of time intervals in the calendar highlights the corresponding data in the correlation view, and compares the correlation based on all data points vs. the selected subset (see Fig. 1b). This allows an interactive discovery of variations in the explanatory power of variables, e.g., due to daily or seasonal effects. Please refer to the supplementary video for a thorough demonstration of this dashboard, as well as another dashboard addressing the detection of data quality issues [1].

2.3 Integration with the TSM: a Visual Frontend

Our previous experiences showed the importance of a seamless integration into existing workflows as a prerequisite for the adoption of visual analytics software. This has two implications: first, *Visplore* dashboards must be easily accessible from the TSM. Second, dashboards should have explicit results that can be accessed for downstream processing. Concerning the first goal, a list of appropriate dashboards is assembled dynamically in the TSM for a user-defined set of time series. Choosing from the list sends the data to *Visplore* and loads the respective configuration of views. Concerning the second goal, discovered patterns can be communicated back to the database as selections of data subsets at any time. This information is exported to the clipboard and can be pasted into the database using a spreadsheet frontend offered by the TSM. The dashboards thus address a full workflow of visual knowledge discovery and communication of findings (see Fig. 1). Use cases include labeling of identified clusters, masking outliers or measurement errors, or selecting training data for external modeling.

*e-mail: tm@vrvis.at

†e-mail: hp@vrvis.at

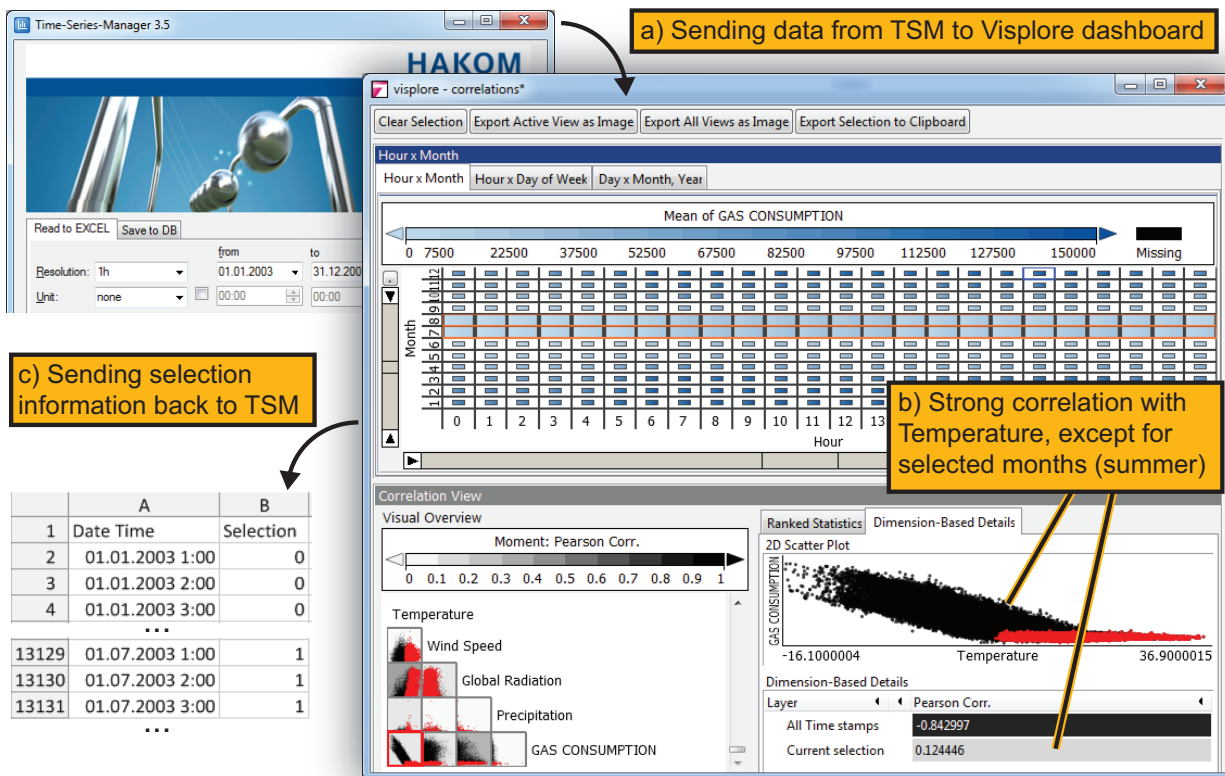


Figure 1: Analyzing the correlation of natural gas consumption and meteorological factors. Time series are sent to Visplore from the HAKOM time series manager (a). *Temperature* exhibits a strong Pearson correlation with *GAS CONSUMPTION*, but selecting the summer months in the calendar indicates a substantially weaker correlation for this season (b). Exporting the selection as a new time series makes this information available in the database for follow-up tasks (c).

3 PRELIMINARY EXPERIENCE FEEDBACK

Nine collaboratively identified dashboards for model-related tasks have been distributed by HAKOM for five months, and were demonstrated to potential customers in the energy sector. As a general observation, despite their restrictions, dashboards were considered more convincing and more easily applicable by new users than the unrestricted version of Visplore. This confirms that task-tailored dashboards are an effective way of conveying the value of interactive visualization. Customers agreed that collaboratively identified dashboards provide an efficient way of harnessing both the domain expert's experience with a task, and the visualization expert's experience with the software and visualization as such.

HAKOM is confident that dashboards are easier to market as an extension of the TSM than the unrestricted version of Visplore, while at the same time raising the interest of expert users for more. They plan to offer a set of dashboards as a demo version along with every deployment of their product TSM by the end of 2014.

4 DISCUSSION: CHALLENGES AND OPPORTUNITIES

A key challenge in dashboard design refers to identifying the ideal arrangement and number of views per dashboard [6]. Adding views may widen the scope of addressable tasks at the cost of increased complexity and reduced screen space per view.

Another challenge is keeping the number of control elements low while maintaining applicability to varying data characteristics such as different numbers, lengths, scales and distributions of time series. Automatic data-specific adjustment can mitigate this to a certain degree, but can typically not take the semantics of data into account. For example, adjusting the visualized data range to exclude outliers may produce a better overview in the presence of measurement errors, but will be counterproductive if outliers are meaningful. For such reasons, we found that easily accessible controls for adjust-

ing displayed ranges and filters are crucial, while automation can provide intelligent defaults. In general, we consider the process of removing controls as an opportunity for identifying usability deficiencies also in the unrestricted version.

As future work, we would like to evaluate the effect of dashboards on the long-term adoption of Visplore, and their potential to gain customers for the unrestricted version of the software.

ACKNOWLEDGEMENTS

This work has been supported by the Austrian Funding Agency (FFG) within the scope of the COMET K1 program. We wish to thank our project partner HAKOM for valuable discussions.

REFERENCES

- [1] Supplementary video illustrating two dashboards: <http://download.vrvis.at/va/posters/vast2014/video.avi>.
- [2] T. Mühlbacher and H. Piringer. A Partition-Based Framework for Building and Validating Regression Models. *IEEE Trans. on Visualization and Computer Graphics (VAST '13)*, 19(12):1962–1971, 2013.
- [3] H. Piringer, W. Berger, and H. Hauser. Quantifying and Comparing Features in High-Dimensional Datasets. In *Proc. of the 6th International Conf. on Coordinated & Multiple Views in Exploratory Visualization (CMV2008)*, pages 240–245, 2008.
- [4] H. Piringer, W. Berger, and J. Krasser. HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation. *Computer Graphics Forum (EuroVis '10)*, 29(3):983–992, 2010.
- [5] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):1113–1120, 2009.
- [6] Tableau Software. Visual Analysis Best Practices: A Guidebook. <http://www.tableausoftware.com/learn/whitepapers/tableau-visual-guidebook>, Last visited 2014-06-27, 2014.

PAPER

E

Statistical Forecasting in the Energy Sector: Task Analysis and Lessons Learned from Deploying a Dashboard Solution

Published in *Proceedings of IEEE Vis Conference 2015, Practitioner Experience Track, Short Paper*, 2015, [MAP15]

Statistical Forecasting in the Energy Sector: Task Analysis and Lessons Learned from Deploying a Dashboard Solution

Thomas Mühlbacher*
VRVis Research Center

Clemens Arbesser†
VRVis Research Center

Harald Piringer‡
VRVis Research Center

ABSTRACT

Statistical forecasting of time series in the energy sector comprises many tasks that benefit from a tight involvement of domain experts. Examples include the assessment of data quality, the selection of descriptive time series as model inputs, or the identification of structural breaks. As a prerequisite for designing visualizations to support these tasks, this paper presents a detailed analysis of forecasting tasks in the energy sector. As a second contribution, it presents lessons learned from the development and the commercialization of visualization dashboards addressing the identified tasks.

Index Terms: I.6.9.f [Simulation, Modeling and Visualization]: Visualization—Visualization Systems and Software

1 INTRODUCTION

Statistical forecasting of time series plays a key role for many tasks in the energy sector. Examples include the prediction of energy production and demand to guarantee supply, or the forecasting of energy prices for a cost-efficient allocation of resources. In the past decades, the energy market has undergone substantial organizational changes, such as the liberalization of markets, the growing share of renewable sources or the advent of smart grids. As a result of growing decentralization, the numbers of available time series like sensor measurements or forecast targets have increased substantially. At the same time, increased competition forces market players to react to changes quickly, and to update forecast models continuously in order to outperform others. Thus, domain experts are constantly faced with questions like: Which parts of the regularly acquired data are appropriate for the training and validation of models? Is the quality of forecast models sufficient? How can the accuracy of forecasts be improved? Which model variant should be used, and when?

Interactive visualization is a powerful tool to address such questions effectively, as demonstrated in the energy sector and beyond [4, 7, 12]. At the VRVis Research Center, we have developed *Visplore*, a system for visual exploration and statistical analysis. *Visplore* comprises a large set of views, including bar charts, line charts or scatter plots, as well as dedicated views for model validation [7] and model comparison [4]. Any number of views can be interactively created and parameterized. All views are linked by selections of data subsets and data attributes, and support a multi-threading architecture to enhance large data scalability [8]. In cooperation with HAKOM Solutions, an IT-service provider who distributes a platform for time series management and forecasting to more than 40 customers in the energy sector, our goal is to establish *Visplore* as a complementary visualization frontend for established forecasting tools and workflows.

As described in previous work [5], the approach is to define *task-tailored dashboards*, i.e., predefined configurations of parameter-

ized views that address one particular analysis task. For a seamless integration into existing workflows, dashboards are accessible directly from existing tools such as the HAKOM Time Series Manager (TSM) or forecasting software. A key finding of our previous work [5] was that pre-configured dashboards are considered as easier to market than the full version of *Visplore*. The flexibility of *Visplore* caters to expert users but often asked too much of new users, who had difficulties to assemble effective configurations for their tasks. The focus of the discussion in previous work were challenges and opportunities of dashboard design as opposed to deploying the full, unrestricted construction kit directly [5]. In contrast, the contribution of this paper can be summarized as follows:

- A task analysis of statistical forecasting in the energy sector;
- A discussion of dashboard system implementation aspects;
- Lessons learned for dashboard commercialization.

2 TASK ANALYSIS: FORECASTING IN THE ENERGY SECTOR

In cooperation with HAKOM, we compiled a structured list of recurring tasks and questions related to forecasting in the energy sector. In addition to deepening our understanding of the forecasting process as such, the goal was to identify potential gaps in prevalent workflows that might benefit from task-tailored dashboards. A first key insight of this process was that forecasting is typically regarded as a cycle of two distinct phases in the energy sector.

In the **model identification phase**, the goal is to obtain a satisfactory prediction model with respect to measures like accuracy, stability, or the cost of regularly acquiring input time series such as weather forecasts. This phase can benefit significantly from a human in the loop, since understanding structures, relationships and trade-offs as conveyed by visualization is crucial for an efficient identification and selection of models [4]. The phase comprises several tasks as described below, and is illustrated in Figure 1a.

The **operational phase** refers to the application of an established model to regularly acquired data for forecasting. It also includes the maintenance of the model by incorporating new training data and performing regular validation. In practice, the operational phase is often automated to a large degree, and human analysts are only involved to the extent of monitoring data and prediction quality. However, when the prediction quality of an operational model deteriorates over time, e.g., due to changes of initial assumptions, the model identification phase is returned to for a refinement of the model. Tasks of the operational phase are described below, and illustrated in Figure 1b.

A second result of the task analysis concerned the flow of insights and data between tasks, i.e., which outputs of a task are inputs to another (see Fig. 1). This information was important for integrating dashboards into existing workflows, as described in Sec. 3.

The structured characterization of tasks was a guiding roadmap in the development of our task-tailored dashboard solution (Section 3). Furthermore, we used it to structure marketing material such as videos and flyers, as well as a presentation at a user conference for 20 companies in the energy sector. Feedback by these companies suggested that it was understood well, and allowed them to effectively match the benefits of the respective dashboards to needs and gaps in their current workflows.

*e-mail: tm@vrvis.at

†e-mail: arbesser@vrvis.at

‡e-mail: hp@vrvis.at

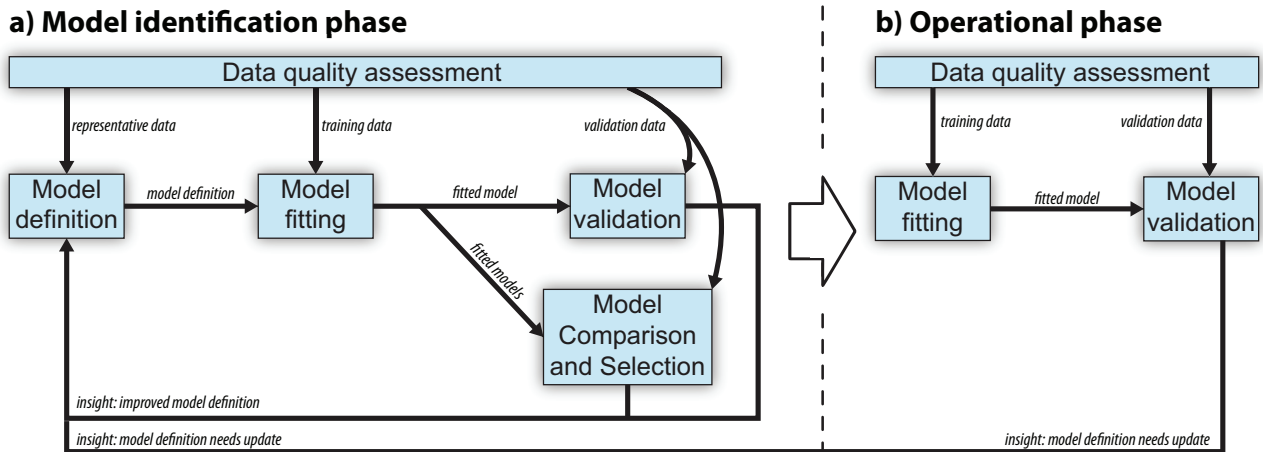


Figure 1: Statistical forecasting tasks in the energy sector are performed in two distinct phases. The goal of the model identification phase (a) is obtaining a satisfactory model. In the operational phase (b), the model is used for forecasting, and regularly adapted based on new data.

In the following, we provide a detailed description of tasks, their interplay, and their typical extent in the two phases.

Task 1: Data quality assessment refers to the identification of data problems such as missing data, constraint violations or anomalies. Aside from understanding and communicating such problems, the goal is to exclude time series or time periods with insufficient quality for downstream tasks like the training and validation of models, while avoiding a selection bias, i.e., maintaining a representative sample. Data quality assessment is a recurring task in both phases of the forecasting process, however, typically to a different extent: while model identification benefits from a thorough understanding of the frequency or causes of anomalies, users monitoring the regular fitting of an operational model typically prefer a concise summary.

Task 2: Model definition refers to defining the “interface” and structure of a model, i.e., the model type and complexity, input time series and transformations or interactions thereof, qualitative situations to be treated separately, as well as all model parameters that are *defined* rather than estimated from training data. For example, a forecast model for natural gas consumption could be defined as a MARS model [1] based on temperature, time of day and consumption values of previous hours and days, that will fit separate coefficients for holidays and for workdays. Model definition is inherently a task of the model identification phase, but is typically returned to when the definition of an operational model becomes invalid over time. Sub-tasks to support an informed model definition include:

Assessment of data descriptiveness. This task refers to understanding whether time series or interactions thereof are suitable predictors of a target time series, and under which circumstances. The goal is to obtain a subset of descriptive model input time series, and possibly also a subset of suitable time periods if descriptiveness varies over time.

Identification of structural breaks refers to understanding sudden shifts in the structure of time series, daily profiles, distributions, correlations or dependencies that should be modeled separately. An example of a structural break is shown in Fig. 2, where the daily load profile of a power line differs significantly for parts of the year. The goal is to characterize boundaries between structures to support an informed model parameterization and composition.

Task 3: Model fitting refers to the statistical estimation of coefficients according to a model definition, driven by concrete instances of training data. In the operational phase, fitting is typically performed frequently in regular intervals to keep the model up to date.

Task 4: Model validation. This task refers to validating predictions of a model against validation data of a reference time se-

ries. The primary goal is to obtain insights about accuracy, bias and stability of the model that help refining the model definition. Validation is performed in both phases, but typically more extensively during the identification of the initial model definition, as described below. Sub-tasks include:

Assessment of accuracy refers to investigating the magnitude of the prediction error with respect to a reference time series. The main goal is to understand whether the accuracy is sufficient, and if not, to characterize periods and situations of insufficient performance for refining the model definition. A typical approach is to inspect key performance indicators such as the root-mean-square error or mean absolute percentage error on multiple aggregation levels, e.g., globally, or for different weather situations, and to compare the prediction and reference time series in detail for periods of bad performance. As a secondary insight of this task, inadequate validation data (e.g., outliers) also manifests as large deviations, which relates this task to data quality assessment.

Analysis of systematic bias. The goal of this sub-task is to detect and characterize systematic over- or underestimations of a reference time series, e.g., at certain times of the day, weather situations or prices. The prediction bias provides information for detecting effects currently not captured by the model, e.g., due to omitted input time series, insufficient model complexity, or the omission of structural breaks [4]. Thus, this sub-task is typically a part of the model identification phase, and not regularly performed for every re-fitting of an operational model.

Assessment of stability refers to investigating the development of model performance across different validation data sets, e.g., sensor measurements of different days. A volatile prediction quality may indicate an insufficiently descriptive model definition, or an over-fitted model [2]. Single outliers in an otherwise stable development might point to exceptional validation data that should be investigated in further detail. Stability monitoring is typically performed in the operational phase as a sanity check of regularly updated fits.

Task 5: Model comparison and selection. This task refers to the comparison of multiple candidate models for the same prediction target. A typical approach is to compare key performance indicators based on deviations from a reference time series on multiple aggregation levels. This supports the goal of identifying the best-fitting model globally as well as for different situations individually, e.g., different times of day, or different market or weather situations. Furthermore, the comparison of model superiority across different data subsets supports the identification of decision boundaries for building composite models. Thus, this task is typically part of the model identification phase.

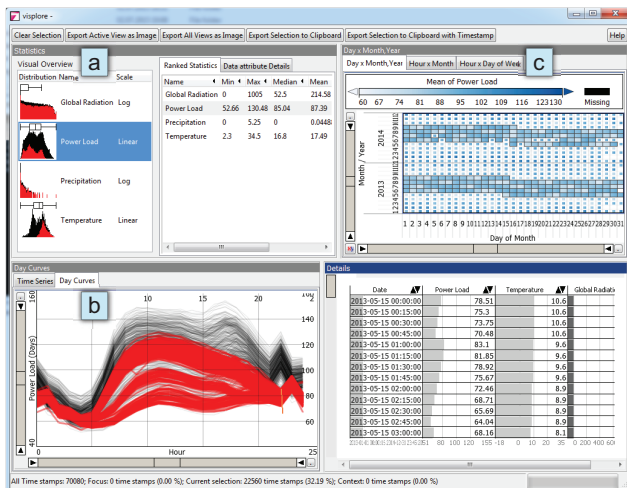


Figure 2: A dashboard for assessing time series plausibility. After selecting the time series “Power Load” in the statistical overview (a), its detailed inspection as overlaid day curves reveals two distinct daily profiles (b). Selecting one of the two clusters reveals the pattern as a distinct part of the year between May and September (c).

3 A DASHBOARD SOLUTION FOR FORECASTING

This section describes our approach of supporting forecast-related tasks using predefined visualization dashboards. First, we describe the process of designing dashboards based on our software Visplore, and discuss an exemplary dashboard supporting the assessment of time series plausibility (Sec. 3.1). Afterwards, we discuss selected implementation issues including an application layer API for customizing dashboard behaviour outside the Visplore implementation, as well as the integration of dashboards in arbitrary host applications (Sec. 3.2)

3.1 Dashboard Design in Visplore

Visplore is a software tool for exploratory analysis of multivariate data in linked views. It is developed at the VRVis Research Center and has been used in various fields since 2004. Supported types of views include time series graphs and table views, as well as scatterplots, parallel coordinates, and other well-known visualizations of multivariate data. Furthermore, Visplore provides task-specific visualizations as described in previous work, e.g., an adapted version of the rank-by-feature framework [9, 6] as well as dedicated views for the validation of prediction models [7, 4]. All views of Visplore support the concept of linking and brushing of data records, e.g., time stamps of time series. The user may define a selection by creating query components such as 1D intervals (e.g., in histograms), sets of categories (e.g., months in a calendar), and many others via brushing, causing all other views to update immediately.

Any number of views can be parameterized and arranged in a desired layout using drag-and-drop. Dashboard design in Visplore can thus be seen as visual programming based on existing system parts. Dashboards as shown in Figure 2 can be assembled within a few minutes. The resulting configurations, i.e., views and their layout can be stored in an xml-based format. Using this format, the assignment of time series to views is based on *tags* instead of actual time series names. For example, a rank-by-feature view can be configured to rank all time series with a tag “model inputs” by their correlation with a time series tagged as “target”. This allows applying dashboards to new data, while the actual data is supplied in the form of tagged time series to the dashboard, e.g., by a host application (see Sec. 3.2).

In an iterative process of suggestion and feedback with HAKOM, we created Visplore dashboards supporting the tasks identified in Section 2. As one example, Figure 2 shows a dash-

board supporting the assessment of time series plausibility, addressing the tasks *data quality assessment* and *identification of structural breaks*. In contrast to the unrestricted version of Visplore, no additional views can be opened, and only a limited set of visual parameters can be controlled. This minimizes the complexity and allows focusing on a particular, well-defined task.

As a recurring design principle, we arrange views in dashboards ordered by their aggregation granularity from top to bottom and from left to right [11]. In the example, a statistical overview in the top left corner shows the value distributions of time series as histograms and computes descriptive statistics for the currently selected time stamps (Fig. 2a). Selecting a time series in this overview by a click immediately shows the respective time series in the linked views for a detailed investigation, e.g., as line graph, or, as shown in the figure, as overlaid curves by day (Fig. 2b). We arrange alternative views of similar information granularity (e.g., time series vs. day curves) as *tabbed* views in the same dashboard position, if they do not rely on being visible at the same time. In the example, investigating day curves of the time series “Power Load” reveals two distinct daily profiles: days with a sudden positive local peak at 22:00h, and days with a slightly lower peak at 23:00h. Selecting one of the clusters using a line intersection brush highlights the selected days in red color. A linked calendar that encodes the selection of days using the size of heatmap cells reveals this cluster as a distinct part of the year between May and September (Fig. 2c). The discovery of this structural break can be used in the definition of a composite forecast model for the power load, with a decision boundary based on the day of the year. A button for exporting the current selection information to the clipboard as an indicator time series (see previous work [5]) allows making the selected data subset available in host applications, e.g., a forecasting engine.

3.2 Selected Implementation Issues

Visplore was initially designed as a standalone application rather than a construction kit for task-tailored dashboards. In recent years, the design of dashboards and their integration in existing tools and workflows has become a primary use-case, for the energy sector and beyond. As the flexibility of each dashboard is deliberately limited regarding aspects like the parameterization of views, multiple dashboard-relevant features had to be implemented. Examples include hover-triggered control elements for adjusting value ranges, filters and visual parameters directly from within views, or showing time series automatically in other views when clicked on their name, e.g., for details on demand (see Sec. 3.1). However, many dashboards called for interactions and workflows that were too task-specific to warrant becoming core features like the examples above. An example is the application of a particular transformation to a time series before showing it in other views, e.g., normalized, or sorted by value as duration curve. Another example is to initially focus views on a particular aspect of the data that is meaningful to a customer, e.g., a particular category or time frame.

To avoid an explosion of highly customized feature implementations in Visplore, we realized an application layer in the form of a Visplore API, with interfaces to languages such as Python¹. The current state of a dashboard such as data values, selections or view parameters can be queried and modified via calls to this API. By specifying a startup script for a dashboard in Python, customer-tailored behaviour can be realized without changing the implementation of Visplore. Typical examples include setting data-driven or customer-specific defaults, registering callbacks for state changes such as the selection of time series or time stamps, or adding new GUI elements such as buttons based on the GUI toolkit PyGTK².

As an important requirement for adoption, a seamless integration of dashboards into existing tools and workflows is essential [5]. On

¹<https://www.python.org>

²www.pygtk.org

a usability level, this implies that dashboards are easily accessible, e.g., from appropriate places in a host application. On a technical level, this involves the transfer of time series data to Visplore dashboards, as well as the transfer of explicit findings and results from dashboards back to the host application.

In the energy sector, the first tool that integrated Visplore was the HAKOM Time Series Manager (TSM). In an early version, we implemented a dedicated importer for a data format provided by TSM on our side. However, as additional host applications became relevant, we decided to define a client-agnostic data exchange protocol enabling communication with arbitrary host applications. Via TCP network instructions, the protocol defines the exchange of a multivariate data table along with optional meta information for rows, columns or values in the JSON³ format. As a special case, this allows Visplore to receive a set of time series with tags as required for our dashboards (see Sec. 3.1). A single description document of the protocol enables host application developers to transform data into our format, while our side does not need to know of the host's data model.

Regarding the communication of dashboard results, multivariate tables can be sent in the same manner as JSON via TCP. An example is sending an indicator time series encoding the currently selected time stamps as dichotomous information, i.e. "selected", vs. "not selected". In case host applications do not provide a TCP listener for accepting dashboard results, we also support an alternative export to the clipboard of the operating system. Supporting a user-triggered *paste from clipboard* is typically a more lightweight extension of host applications.

4 LESSONS LEARNED FOR COMMERCIALIZATION

A generic exploration system like Visplore matches the requirements of expert users very well, but is not easy to market as such [5]. In contrast, task-tailored dashboards address well-defined questions and the benefits are communicable more precisely. They typically require less training time, and are more straightforward to document and to test for developers than the full construction kit.

Regarding the scope of dashboard solutions, we experienced that dashboard individualization, i.e., tailoring to specific tasks of specific customers, is just as important as offering an off-the-shelf selection of dashboards. A standardized suite of dashboards enables efficient marketing and support, and is typically a good starting point for conveying the benefits of visualization based on a didactic story. However, many customers immediately asked for additional dashboards or adaptations of existing ones. Aside from a preference of familiar diagrams and terminologies, a main reason for individualization are the varying requirements of different user groups [10]. Expert analysts, for example, appreciate flexible dashboards with multiple linked views, drill-down possibilities, or the export of results for downstream analyses. Users whose task is the regular monitoring or reporting of indicators will prefer a concise summary over complex visualizations, and appreciate a quick export of graphics. In this respect, the configuration possibilities of Visplore (Sec. 3.1) are an invaluable asset that enables delivering everything from simple monitoring dashboards to flexible exploration systems. This is instrumental to offering "dashboard design as a service", which we expect to be a promising strategy complementing off-the-shelf solutions.

Regarding cooperation with industry partners, we have experienced various benefits of different collaboration forms. Direct cooperation with end customers is extremely helpful in understanding the needs and tasks of real users. On the other hand, the cooperation with HAKOM as a generic IT service provider in the energy sector enables multiplier effects on several levels. Due to their overview of the market, they are aware of visualization opportunities in prevalent workflows, and to abstract tasks to a level that can be addressed

efficiently by dashboards. Regarding marketing, their knowledge of the domain is helpful for translating the academic value of visualization to a business value that decision makers in the energy sector understand [3]. Finally, the integration of Visplore in their Time Series Manager enables HAKOM to market dashboards to their existing customer base as an extension of their software.

A recurring challenge in deploying a visualization system is to identify, reach and convince the person responsible for introducing visualization to an organization [3]. It is not guaranteed that the person making this decision is present in the initial demonstration meeting, and one typically does not get many chances. To be less dependent on the internal communication of conveyed benefits, we have found demonstration videos a very effective form of presentation that can be passed on internally. This increases the chances of reaching all key actors including decision makers as well as actual users, who will not only hear the exact intended wordings, but actually see the visualization system in action.

Visualization is not yet as prevalent in the context of energy forecasting as in other fields like Business Intelligence. A possible reason could be that many tasks that benefit from human insights have only become pressing in the more recent past, e.g., identifying superior models for market behaviour than other competitors, or an accurate forecasting of weather-dependent renewable power production. As future work, we would like to investigate the long term adoption of dashboards in the energy sector, as we believe the full potential for visualization in the field has not yet been explored.

ACKNOWLEDGEMENTS

This work has been supported by the Austrian Funding Agency (FFG) within the scope of the COMET K1 program. We wish to thank our project partner HAKOM, as well as Stephan Pajer and Oliver Rafelsberger for valuable discussions.

REFERENCES

- [1] J. H. Friedman. Multivariate Adaptive Regression Splines. *The Annals of Statistics*, 19(1):1–67, 1991.
- [2] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning, Second Edition*. Springer New York Inc., 2009.
- [3] S. Maier, T. May, and J. Kohlhammer. Selling Visualization: From Research Benefit to Business Value. *Proc. of the IEEE VIS Workshop on Business Visualization (BusinessVis)*, Paris, France, 2014.
- [4] T. Mühlbacher and H. Piringer. A Partition-Based Framework for Building and Validating Regression Models. *IEEE Trans. on Vis. and Comp. Graphics*, 19(12):1962–1971, 2013.
- [5] T. Mühlbacher and H. Piringer. Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System. In *Proc. of IEEE Vis 2014, Practitioner Experiences Track*, 2014.
- [6] H. Piringer, W. Berger, and H. Hauser. Quantifying and Comparing Features in High-Dimensional Datasets. In *Proc. of the 6th International Conf. on Coordinated & Multiple Views in Exploratory Visualization (CMV2008)*, pages 240–245, 2008.
- [7] H. Piringer, W. Berger, and J. Krasser. HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation. *Computer Graphics Forum*, 29(3):983–992, 2010.
- [8] H. Piringer, C. Tominski, P. Muigg, and W. Berger. A Multi-Threading Architecture to Support Interactive Visual Exploration. 15(6):1113–1120, Nov. 2009.
- [9] J. Seo and B. Shneiderman. A Rank-by-Feature Framework for Interactive Exploration of Multidimensional Data. *Information Visualization*, 4(2):pages 96–113, 2005.
- [10] M. Streit, H. Schulz, A. Lex, D. Schmalstieg, and H. Schumann. Model-Driven Design for the Visual Analysis of Heterogeneous Data. *IEEE Trans. on Vis. and Comp. Graphics*, 18(6):998–1010, 2012.
- [11] Tableau Software. Visual Analysis Best Practices: A Guidebook. <http://www.tableausoftware.com/learn/whitepapers/tableau-visual-guidebook>. Last visited 2015-08-07, 2014.
- [12] K. Zhao, M. O. Ward, E. A. Rundensteiner, and H. N. Higgins. Lo-Vis: Local Pattern Visualization for Model Refinement. *Computer Graphics Forum*, 33(3):331–340, 2014.

³<http://json.org>

Bibliography

- [AAR⁺16] Gennady Andrienko, Natalia Andrienko, Alexander Ryumkin, Valery Ryumkin, Gennady Kravchenko, Evgeny Tyabaev, Dmitry Khloptsov, and Svetlana Trofimova. Exploration and Refinement of Regression Tree Models with Interactive Maps and Spatial Data Transformations. *International Journal of Cartography*, 2(1):59–76, 2016.
- [ACH15] Genevera I Allen, Frederick Campbell, and Yue Hu. Comments on “Visualizing Statistical Models”: Visualizing Modern Statistical Methods for Big Data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(4):226–228, 2015.
- [ACKK14] Saleema Amershi, Maya Cakmak, William Bradley Knox, and Todd Kulesza. Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4):105–120, 2014.
- [AEEK99] Mihael Ankerst, Christian Elsen, Martin Ester, and Hans-Peter Kriegel. Visual Classification: An Interactive Approach to Decision Tree Construction. In *Proceedings of the Fifth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, KDD ’99, pages 392–396, NY, USA, 1999. ACM.
- [AEK00] Mihael Ankerst, Martin Ester, and Hans-Peter Kriegel. Towards an Effective Cooperation of the User and the Computer for Classification. In *Proceedings of the Sixth ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, KDD ’00, pages 179–188, NY, USA, 2000. ACM.
- [Ans73] Francis John Anscombe. Graphs in Statistical Analysis. *The American Statistician*, 27(1):17–21, 1973.
- [Ash06] Mark H. Ashcraft. *Cognition*. New Jersey: Pearson Education, 2006.
- [AW12] Zafar Ahmed and Chris Weaver. An Adaptive Parameter Space-Filling Algorithm for Highly Interactive Cluster Exploration. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2012)*, page 13–22. IEEE, 2012.

- [BAF⁺13] Markus Bögl, Wolfgang Aigner, Peter Filzmoser, Tim Lammarsch, Silvia Miksch, and Alexander Rind. Visual Analytics for Model Selection in Time Series Analysis. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2237–2246, 2013.
- [BB13] Patrick Breheny and Woodrow Burchett. Visualization of Regression Models using visreg. *R package*, pages 1–15, 2013.
- [BCD⁺09] Michael R Berthold, Nicolas Cebron, Fabian Dill, Thomas R Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. KNIME-The Konstanz Information Miner: Version 2.0 and Beyond. *ACM SIGKDD Explorations Newsletter*, 11(1):26–31, 2009.
- [BDW08] Sugato Basu, Ian Davidson, and Kiri Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. CRC Press, 2008.
- [Ber16] Scott Berinato. *Good Charts: The HBR Guide to Making Smarter, More Persuasive Data Visualizations*. Harvard Business Review Press, 2016.
- [BL10] Enrico Bertini and Denis Lalanne. Investigating and Reflecting on the Integration of Automatic Data Analysis and Visualization in Knowledge Discovery. *ACM SIGKDD Explorations Newsletter*, 11(2):9–18, 2010.
- [Bla02] Alan F Blackwell. First Steps in Programming: A Rationale for Attention Investment Models. In *Proc. of the IEEE 2002 Symposia on Human Centric Computing Languages and Environments*. IEEE, 2002.
- [BLBC12] Eli T. Brown, Jingjing Liu, Carla E. Brodley, and Remco Chang. Dis-Function: Learning Distance Functions Interactively. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2012)*, pages 83–92, 2012.
- [BNH14] Lauren Bradel, Chris North, and Leanna House. Multi-Model Semantic Interaction for Text Analytics. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2014)*, pages 163–172. IEEE, 2014.
- [BNTM16] Matthew Brehmer, Jocelyn Ng, Kevin Tate, and Tamara Munzner. Matches, Mismatches, and Methods: Multiple-View Workflows for Energy Portfolio Analysis. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):449–458, 2016.
- [BPBO15] Pierrick Bruneau, Philippe Pinheiro, Bertjan Broeksema, and Benoît Otjacques. Cluster Sculptor, an Interactive Visual Clustering System. *Neuro-computing*, 150:627–644, 2015.

- [BSL⁺08] Andreas Buja, Deborah F Swayne, Michael L Littman, Nathaniel Dean, Heike Hofmann, and Lisha Chen. Data Visualization with Multidimensional Scaling. *Journal of Computational and Graphical Statistics*, 17(2):444–472, 2008.
- [BZS⁺16] Sriram Karthik Badam, Jieqiong Zhao, Shivalik Sen, Niklas Elmqvist, and David Ebert. Timefork: Interactive Prediction of Time Series. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 5409–5420. ACM, 2016.
- [CCH01] Doina Caragea, Dianne Cook, and Vasant G Honavar. Gaining Insights into Support Vector Machine Pattern Classifiers using Projection-Based Tour Methods. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 251–256. ACM, 2001.
- [CCM03] David Cohn, Rich Caruana, and Andrew McCallum. Semi-Supervised Clustering with User Feedback. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, 4(1):17–32, 2003.
- [CENS06] Rich Caruana, Mohamed Elhawary, Nam Nguyen, and Casey Smith. Meta Clustering. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 107–118. IEEE, 2006.
- [CGM⁺17] Davide Ceneda, Theresia Gschwandtner, Thorsten May, Silvia Miksch, Hans-Jorg Schulz, Marc Streit, and Christian Tominski. Characterizing Guidance in Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics*, (1):111–120, 2017.
- [CLKP10] Jaegul Choo, Hanseung Lee, Jaeyeon Kihm, and Haesun Park. iVisClassifier: An Interactive Visual Analytics System for Classification Based on Supervised Dimension Reduction. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2010)*, pages 27–34. IEEE, 2010.
- [CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.
- [Com17] Computer Hope. Command line vs. GUI. <https://www.computerhope.com/issues/ch000619.htm>, last visited 2018-07-25, 2017.
- [CRM91] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The Information Visualizer, an Information Workspace. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, page 181–186. ACM, 1991.

- [Das17] Aritra Dasgupta. Towards Understanding Familiarity Related Cognitive Biases in Visualization Design and Usage. In *DECISIVE Workshop, IEEE VIS 2017*, 10 2017.
- [dat13] Dataiku - Collaborative Data Science Platform. Founded in 2013. <https://www.dataiku.com/>, last visited 21.08.2018, 2013.
- [DE98] Alan Dix and Geoffrey Ellis. Starting Simple: Adding Value to Static Visualisation through Simple Interaction. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 124–134. ACM, 1998.
- [DKM06] Tim Dwyer, Yehuda Koren, and Kim Marriott. IPSep-CoLa: An Incremental Procedure for Separation Constraint Layout of Graphs. *IEEE Trans. on Visualization and Computer Graphics*, 12(5):821–828, 2006.
- [DLW⁺17] Aritra Dasgupta, Joon-Yong Lee, Ryan Wilson, Robert Lafrance, Nick Cramer, Kristin Cook, and Samuel Payne. Familiarity vs Trust: A Comparative Study of Domain Scientists’ Trust in Visual Analytics and Conventional Analysis Methods. *IEEE Trans. on Visualization and Computer Graphics*, (1):1–1, 2017.
- [DMF07] Marie Desjardins, James MacGlashan, and Julia Ferraioli. Interactive Visual Clustering. In *Proceedings of the 12th International Conference on Intelligent User Interfaces*, pages 361–364. ACM, 2007.
- [DP12] Thomas H. Davenport and D.J. Patil. Data Scientist: The Sexiest Job of the 21st Century. Technical report, Harvard Business Review, <https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century>, last visited 01.06.2018, October 2012.
- [EFN11] Alex Endert, Patrick Fiaux, and Chris North. Unifying the Sensemaking Loop with Semantic Interaction. In *IEEE Workshop on Interactive Visual Text Analytics for Decision Making at VisWeek 2011*, 2011.
- [EHM⁺11] Alex Endert, Chao Han, Dipayan Maiti, Leanna House, Scotland Leman, and Chris North. Observation-Level Interaction with Statistical Models for Visual Analytics. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2011)*, pages 121–130. IEEE, October 2011.
- [ERT⁺17] Alex Endert, William Ribarsky, Cagatay Turkay, William Wong, Ian Nabney, Ignacio Díaz Blanco, and Fabrice Rossi. The State of the Art in Integrating Machine Learning into Visual Analytics. In *Computer Graphics Forum*, volume 36, pages 458–486. Wiley Online Library, 2017.
- [Fek13] Jean-Daniel Fekete. Visual Analytics Infrastructures: From Data Management to Exploration. *IEEE Computer*, 46(7):22–29, 2013.

- [Fis36] Ronald Aylmer Fisher. The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(7):179–188, 1936.
- [FOJ03] Jerry Alan Fails and Dan R Olsen Jr. Interactive Machine Learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, pages 39–45. ACM, 2003.
- [FPDm12] Danyel Fisher, Igor Popov, Steven Drucker, and m.c. schraefel. Trust Me, I’m Partially Right: Incremental Visualization Lets Analysts Explore Large Datasets Faster. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’12)*, page 1673–1682. ACM, 2012.
- [FPSS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. *AI magazine*, 17(3):37, 1996.
- [Fri99] Michael Friendly. Extending Mosaic Displays: Marginal, Conditional, and Partial Views of Categorical Data. *Journal of Computational and Graphical Statistics*, 8:373–395, 1999.
- [FWSN08] Jean-Daniel Fekete, Jarke J. Wijk, John T. Stasko, and Chris North. Information Visualization. Chapter: The Value of Information Visualization. pages 1–18. Springer-Verlag, Berlin, Heidelberg, 2008.
- [Gal86] Francis Galton. Regression Towards Mediocrity in Hereditary Stature. *The Journal of the Anthropological Institute of Great Britain and Ireland*, 15:246–263, 1886.
- [GAW⁺11] Michael Gleicher, Danielle Albers, Rick Walker, Ilir Jusufi, Charles D Hansen, and Jonathan C Roberts. Visual Comparison for Information Visualization. *Information Visualization*, 10(4):289–309, 2011.
- [Gle13] Michael Gleicher. Explainers: Expert Explorations with Crafted Projections. *IEEE Trans. on Visualization and Computer Graphics*, (12):2042–2051, 2013.
- [Gor98] Neil Gordon. Colour Blindness. *Public Health*, 112(2):81 – 84, 1998.
- [Gul95] Ranjay Gulati. Does Familiarity Breed Trust? The Implications of Repeated Ties for Contractual Choice in Alliances. *Academy of Management Journal*, 38(1):85–112, 1995.
- [GW09] David Gotz and Zhen Wen. Behavior-Driven Visualization Recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, pages 315–324. ACM, 2009.

- [GWR09] Zhenyu Guo, Matthew O. Ward, and Elke A. Rundensteiner. Model Space Visualization for Multivariate Linear Trend Discovery. In *Proc. of the 4th IEEE Symp. on Visual Analytics Science and Technology (VAST 2009)*, pages 75–82, 2009.
- [GZ09] David Gotz and Michelle X Zhou. Characterizing Users’ Visual Analytic Activity for Insight Provenance. *Information Visualization*, 8(1):42–55, 2009.
- [HBC⁺16] Nicolaus Henke, Jacques Bughin, Michael Chui, James Manyika, Tamim Saleh, Bill Wiseman, and Guru Sethupathy. The Age of Analytics: Competing in a Data-Driven World. Technical report, McKinsey Global Institute, December 2016.
- [HE12] Christopher Healey and James Enns. Attention and Visual Memory in Visualization and Computer Graphics. *IEEE Trans. on Visualization and Computer Graphics*, 18(7):1170–1188, July 2012.
- [Hea99] Marti Hearst. User Interfaces and Visualization. *Modern Information Retrieval*, pages 257–323, 1999.
- [HJM⁺11] Ming C Hao, Halldór Janetzko, Sebastian Mittelstädt, Water Hill, Umeshwar Dayal, Daniel A Keim, Manish Marwah, and Ratnesh K Sharma. A Visual Analytics Approach for Peak-Preserving Prediction of Large Seasonal Time Series. In *Computer Graphics Forum*, volume 30, pages 691–700. Wiley Online Library, 2011.
- [HMSA08] Jeffrey Heer, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical Histories for Visualization: Supporting Analysis, Communication, and Evaluation. *IEEE Trans. on Visualization and Computer Graphics*, 14(6), 2008.
- [HNH⁺12] Benjamin Höferlin, Rudolf Netzel, Markus Höferlin, Daniel Weiskopf, and Gunther Heidemann. Inter-Active Learning of Ad-Hoc Classifiers for Video Visual Analytics. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2012)*, pages 23–32. IEEE, 2012.
- [HOG⁺12] M. Shahriar Hossain, Praveen Kumar Reddy Ojili, Cindy Grimm, Rolf Mueller, Layne T. Watson, and Naren Ramakrishnan. Scatter/Gather Clustering: Flexibly Incorporating User Feedback to Steer Clustering Results. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2829–2838, 2012.
- [HTF09] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Second Edition*. Springer New York Inc., 2009.

- [IMI⁺10] Stephen Ingram, Tamara Munzner, Veronika Irvine, Melanie Tory, Steven Bergner, and Torsten Möller. DimStiller: Workflows for Dimensional Analysis and Reduction. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2010)*, pages 3–10, 2010.
- [JJ09] Sara Johansson and Jimmy Johansson. Interactive Dimensionality Reduction through User-Defined Combinations of Quality Metrics. *IEEE Trans. on Visualization and Computer Graphics*, 15(6):993–1000, 2009.
- [JWK14] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. Modeling User Preferences in Recommender Systems: A Classification Framework for Explicit and Implicit User Feedback. *ACM Trans. on Interactive Intelligent Systems (TiiS)*, 4(2):8, 2014.
- [JZF⁺09] Dong Hyun Jeong, Caroline Ziemkiewicz, Brian Fisher, William Ribarsky, and Remco Chang. iPCA: An Interactive System for PCA-Based Visual Analytics. In *Computer Graphics Forum*, volume 28, pages 767–774. Wiley Online Library, 2009.
- [KAF⁺08] Daniel A. Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual Analytics : Definition, Process, and Challenges. In A. Kerren, editor, *Information Visualization*, pages 154–175. Springer, Berlin, 2008.
- [Kei01] Daniel A. Keim. Datenvisualisierung und Data Mining. In *Vortragsfolien zu : 9. Fachtagung Datenbanksysteme in Büro, Technik und Wissenschaft (BTW ' 01), Deutsche Informatik Akademie (DIA), Oldenburg, Germany, 2001*, 2001.
- [Kei02] Daniel A Keim. Information Visualization and Visual Data Mining. *IEEE Trans. on Visualization and Computer Graphics*, (1):1–8, 2002.
- [KK08] Nimit Kumar and Krishna Kumamuru. Semisupervised Clustering with Metric Learning using Relative Comparisons. *IEEE Trans. on Knowledge and Data Engineering*, 20(4):496–503, 2008.
- [KKEM10] Daniel A Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann, editors. *Mastering The Information Age - Solving Problems with Visual Analytics*. Eurographics, 2010.
- [KLG⁺16] Paul Klemm, Kai Lawonn, Sylvia Glaßer, Uli Niemann, Katrin Hegenscheid, Henry Völzke, and Bernhard Preim. 3D Regression Heat Map Analysis of Population Study Data. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):81–90, 2016.
- [KLTH10] Ashish Kapoor, Bongshin Lee, Desney Tan, and Eric Horvitz. Interactive Optimization for Steering Machine Classification. In *Proceedings of the*

- SIGCHI Conference on Human Factors in Computing Systems*, pages 1343–1352. ACM, 2010.
- [KMRV15] Daniel Keim, Tamara Munzner, Fabrice Rossi, and Michael Verleysen. Bridging Information Visualization with Machine Learning. Technical report, Dagstuhl Seminar 15101, Dagstuhl Rep. 5 (3), 2015.
- [KPB14] Josua Krause, Adam Perer, and Enrico Bertini. INFUSE: Interactive Feature Selection for Predictive Modeling of High Dimensional Data. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1614–1623, 2014.
- [LB17] Po-Ming Law and Rahul C Basole. Designing Breadth-Oriented Data Exploration for Mitigating Cognitive Biases. In *DECISIVE: Workshop on Dealing with Cognitive Biases in Visualizations*, 2017.
- [LCM⁺17] Junhua Lu, Wei Chen, Yuxin Ma, Junming Ke, Zongzhuang Li, Fan Zhang, and Ross Maciejewski. Recent Progress and Trends in Predictive Visual Analytics. *Frontiers of Computer Science*, 11(2):192–207, 2017.
- [LGH⁺17] Yafeng Lu, Rolando Garcia, Brett Hansen, Michael Gleicher, and Ross Maciejewski. The State-of-the-Art in Predictive Visual Analytics. In *Computer Graphics Forum*, volume 36, pages 539–562. Wiley Online Library, 2017.
- [LKC⁺12] Hanseung Lee, Jaeyeon Kihm, Jaegul Choo, John Stasko, and Haesun Park. iVisClustering: An Interactive Visual Document Clustering via Topic Modeling. In *Computer Graphics Forum*, volume 31, pages 1155–1164. Wiley Online Library, 2012.
- [LKT⁺14] Yafeng Lu, Robert Krüger, Dennis Thom, Feng Wang, Steffen Koch, Thomas Ertl, and Ross Maciejewski. Integrating Predictive Analytics and Social Media. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2014)*, pages 193–202. IEEE, 2014.
- [LMS⁺12] Martin Luboschik, Carsten Maus, Hans-Jörg Schulz, Heidrun Schumann, and Adelinde Uhrmacher. Heterogeneity-Based Guidance for Exploring Multiscale Data in Systems Biology. In *2012 IEEE Symposium on Biological Data Visualization (BioVis)*, pages 33–40. IEEE, 2012.
- [LVdMdS12] Joshua Lewis, Laurens Van der Maaten, and Virginia de Sa. A Behavioral Investigation of Dimensionality Reduction. In *Proceedings of 34th Annual Meeting of the Cognitive Science Society*, volume 34, 2012.
- [LWBP14] Shusen Liu, Bei Wang, Peer-Timo Bremer, and Valerio Pascucci. Distortion-Guided Structure-Driven Interactive Exploration of High-Dimensional Data. In *Computer Graphics Forum*, volume 33, pages 101–110. Wiley Online Library, 2014.

- [MAP15] Thomas Mühlbacher, Clemens Arbesser, and Harald Piringer. Statistical Forecasting in the Energy Sector: Task Analysis and Lessons Learned from Deploying a Dashboard Solution. In *Proceedings of IEEE VIS 2015 (Visualization in Practice Track, Short Paper)*, 2015.
- [Mar17] Bernard Marr. 3 Massive Big Data Problems Everyone Should Know About. Technical report, Forbes, 06 2017.
- [MBD⁺11] Thorsten May, Andreas Bannach, James Davey, Tobias Ruppert, and Jörn Kohlhammer. Guiding Feature Subset Selection with an Interactive Visualization. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2011)*, pages 111–120, 2011.
- [MCB⁺11] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big Data: The Next Frontier for Innovation, Competition, and Productivity. Technical report, McKinsey Global Institute, June 2011.
- [ML14] Vladimir Molchanov and Lars Linsen. Interactive Design of Multidimensional Data Projection Layout. In *EuroVis 2014 - Short Papers*. The Eurographics Association, 2014.
- [MLMP18] Thomas Mühlbacher, Lorenz Linhardt, Torsten Möller, and Harald Piringer. TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):174–183, Jan 2018.
- [Mor15] Lisa Morgan. Universities Expand Curriculum To Meet Data Scientist Demand. Technical report, Information Week, <https://www.informationweek.com/big-data/big-data-analytics/universities-expand-curriculum-to-meet-data-scientist-demand/d/d-id/1323503>, last visited 01.06.2018, December 2015.
- [MP77] James T Milord and Raymond P Perry. A Methodological Study of Overload. *The Journal of General Psychology*, 97(1):131–137, 1977.
- [MP13] Thomas Mühlbacher and Harald Piringer. A Partition-Based Framework for Building and Validating Regression Models. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):1962–1971, December 2013.
- [MP14] Thomas Mühlbacher and Harald Piringer. Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System. In *Proceedings of IEEE Vis 2014 Posters (Visualization in Practice Track)*, 2014.
- [MPG⁺14] Thomas Mühlbacher, Harald Piringer, Samuel Gratzl, Michael Sedlmair, and Marc Streit. Opening the Black Box: Strategies for Increased User Involvement in Existing Algorithm Implementations. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1643–1652, December 2014.

- [MPR18] Sina Mohseni, Alyssa Pena, and Eric D Ragan. ProvThreads: Analytic Provenance Visualization and Segmentation. In *Proceedings of IEEE Vis 2017 Poster Sessions*, 2018.
- [MS11] Prem Melville and Vikas Sindhwani. Recommender Systems. In *Encyclopedia of Machine Learning*, pages 829–838. Springer, 2011.
- [MSDK12] Thorsten May, Martin Steiger, James Davey, and Jörn Kohlhammer. Using Signposts for Navigation in Large Graphs. In *Computer Graphics Forum*, volume 31, pages 985–994. Wiley Online Library, 2012.
- [Mun14] Tamara Munzner. *Visualization Analysis and Design*. AK Peters/CRC Press, 2014.
- [NCE⁺11] Chris North, Remco Chang, Alex Endert, Wenwen Dou, Richard May, Bill Pike, and Glenn Fink. Analytic Provenance: Process + Interaction + Insight. In *CHI’11 Extended Abstracts on Human Factors in Computing Systems*, pages 33–36. ACM, 2011.
- [Nic98] Raymond S Nickerson. Confirmation Bias: A Ubiquitous Phenomenon in Many Guises. *Review of general psychology*, 2(2):175, 1998.
- [Nis17] Kan Nishida. The Third Wave: Democratization of Data Science/Algorithms. <https://blog.exploratory.io/data-science-by-you-dawn-of-third-wave-e89f2999d994>, last visited 31.07.2018, 2017.
- [Nor13] Don Norman. *The Design of Everyday Things: Revised and Expanded Edition*. Constellation, 2013.
- [NVH⁺16] Alexander Nussbaumer, Katrien Verbert, Eva-Catherine Hillemann, Michael A Bedek, and Dietrich Albert. A Framework for Cognitive Bias Detection and Feedback in a Visual Analytics Environment. In *2016 European Intelligence and Security Informatics Conference (EISIC)*, pages 148–151. IEEE, 2016.
- [OK98] Douglas W Oard and Jinmook Kim. Implicit Feedback for Recommender Systems. In *Proceedings of the AAAI Workshop on Recommender Systems*, volume 83. WoUongong, 1998.
- [PBK10] Harald Piringer, Wolfgang Berger, and Jürgen Krasser. HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation. *Computer Graphics Forum*, 29(3):983–992, 2010.
- [Pea01] Karl Pearson. LIII. On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

- [PLvdM⁺17] Nicola Pezzotti, Boudewijn PF Lelieveldt, Laurens van der Maaten, Thomas Höllt, Elmar Eisemann, and Anna Vilanova. Approximated and User Steerable tSNE for Progressive Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics*, 23(7):1739–1752, 2017.
- [PPL10] Kai Puolamaki, Panagiotis Papapetrou, and Jefrey Lijffijt. Visually Controllable Data Mining Methods. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 409–417. IEEE, 2010.
- [PSMD14] Luciana Padua, Hendrik Schulze, Krešimir Matković, and Claudio Delrieux. Interactive Exploration of Parameter Space in Data Mining: Comprehending the Predictive Quality of Large Decision Tree Collections. *Computers & Graphics*, 41:99 – 113, 2014.
- [PSTW⁺17] Stephan Pajer, Marc Streit, Thomas Torsney-Weir, Florian Spechtenhauser, Torsten Möller, and Harald Piringer. WeightLifter: Visual Weight Space Exploration for Multi-Criteria Decision Making. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):611–620, 2017.
- [PTH13] Reid Porter, James Theiler, and Don Hush. Interactive Machine Learning in Data Exploitation. *Computing in Science & Engineering*, 15(5):12–20, 2013.
- [Ran71] William M Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.
- [RL15] Bastian Rieck and Heike Leitte. Persistent Homology for the Evaluation of Dimensionality Reduction Schemes. In *Computer Graphics Forum*, volume 34, pages 431–440. Wiley Online Library, 2015.
- [Rob07] Jonathan C Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization, 2007. CMV’07*, pages 61–71. IEEE, 2007.
- [ROC97] Ronald A. Rensink, Kevin J. O’Regan, and James J. Clark. To See or Not to See: The Need for Attention to Perceive Changes in Scenes. *Psychological Science*, 8(5):368–373, 1997.
- [RPN⁺08] Salvatore Rinzivillo, Dino Pedreschi, Mirco Nanni, Fosca Giannotti, Natalia Andrienko, and Gennady Andrienko. Visually Driven Analysis of Movement Data by Progressive Clustering. *Information Visualization*, 7(3-4):225–239, 2008.

- [SASS16] Hans-Jörg Schulz, Marco Angelini, Giuseppe Santucci, and Heidrun Schumann. An Enhanced Visualization Process Model for Incremental Visualization. *IEEE Trans. on Visualization and Computer Graphics*, 22(7):1830–1842, 2016.
- [SBvLK08] Tobias Schreck, Jürgen Bernard, Tatiana von Landesberger, and Jörn Kohlhammer. Visual Cluster Analysis of Trajectory Data with Interactive Kohonen Maps. In *Proc. of the 3rd IEEE Symp. on Visual Analytics Science and Technology (VAST 2008)*, pages 3–10. IEEE, 2008.
- [Set12] Burr Settles. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.
- [SHB⁺14] Michael Sedlmair, Christoph Heinzl, Stefan Bruckner, Harald Piringer, and Torsten Möller. Visual Parameter Space Analysis: A Conceptual Framework. *IEEE Trans. on Visualization and Computer Graphics*, 20:2161–2170, 2014.
- [Shn83] Ben Shneiderman. Direct Manipulation: A Step Beyond Programming Languages. *IEEE Computer*, 16(8):57–69, August 1983.
- [Shn96] Ben Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proc. of the 1996 IEEE Symposium on Visual Languages*, pages 336–343, 1996.
- [SIH00] Frank M Shipman III and Haowei Hsieh. Navigable History: A Reader’s View of Writer’s Time. *New Review of Hypermedia and Multimedia*, 6(1):147–167, 2000.
- [SJS⁺17] Bruno Schneider, Dominik Jäckle, Florian Stoffel, Alexandra Diehl, Johannes Fuchs, and Daniel Keim. Visual Integration of Data and Model Space in Ensemble Learning. *Symposium on Visualization in Data Science (VDS) at IEEE VIS 2017*, 2017.
- [SLBC03] Deborah F Swayne, Duncan Temple Lang, Andreas Buja, and Dianne Cook. GGobi: Evolving from XGobi into an Extensible Framework for Interactive Data Visualization. *Computational Statistics & Data Analysis*, 43(4):423–444, 2003.
- [SMM12] Michael Sedlmair, Miriah Meyer, and Tamara Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Trans. on Visualization and Computer Graphics*, (12):2431–2440, 2012.
- [SPG14] Charles D Stolper, Adam Perer, and David Gotz. Progressive Visual Analytics: User-Driven Visual Exploration of In-Progress Analytics. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1653–1662, 2014.

- [SS02] Jinwook Seo and Ben Shneiderman. Interactively Exploring Hierarchical Clustering Results [Gene Identification]. *IEEE Computer*, 35(7):80–86, 2002.
- [SSFJK14] Chad A Steed, J Edward Swan, Patrick J Fitzpatrick, and TJ Jankun-Kelly. A Visual Analytics Approach for Correlation, Classification, and Regression Analysis. In *Innovative Approaches of Data Visualization and Visual Analytics*, pages 25–45. IGI Global, 2014.
- [SSMT13] Hans-Jörg Schulz, Marc Streit, Thorsten May, and Christian Tominski. Towards a Characterization of Guidance in Visualization. In *Poster at IEEE Conference on Information Visualization (InfoVis)*, 2013.
- [SSP⁺15] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Sean McLoone, and Alessandro Beghi. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Trans. on Industrial Informatics*, 11(3):812–820, June 2015.
- [SSS⁺14] Dominik Sacha, Andreas Stoffel, Florian Stoffel, Bum Chul Kwon, Geoffrey Ellis, and Daniel A Keim. Knowledge Generation Model for Visual Analytics. *IEEE Trans. on Visualization and Computer Graphics*, 20(12):1604–1613, 2014.
- [SSS⁺16] Arman Shehabi, Sarah Smith, Dale Sartor, Magnus Herrlin, Richard Brown, Jonathan Koomey, Eric Masanet, Nathaniel Horner, Inês Azevedo, and William Lintner. United States Data Center Energy Usage Report. Technical report, 06 2016.
- [SSZ⁺17] Dominik Sacha, Michael Sedlmair, Leishi Zhang, John A. Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C. North, and Daniel A. Keim. What you see is what you can change: Human-Centered Machine Learning by Interactive Visualization. *Neurocomputing*, 2017.
- [Sti81] Stephen M. Stigler. Gauss and the Invention of Least Squares. *The Annals of Statistics*, 9(3):465–474, 1981.
- [SZ88] William Samuelson and Richard Zeckhauser. Status Quo Bias in Decision Making. *Journal of Risk and Uncertainty*, 1(1):7–59, 1988.
- [SZS⁺17] Dominik Sacha, Leishi Zhang, Michael Sedlmair, John A Lee, Jaakko Peltonen, Daniel Weiskopf, Stephen C North, and Daniel A Keim. Visual Interaction with Dimensionality Reduction: A Structured Literature Analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):241–250, 2017.
- [TC05] James J Thomas and Kristin A Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. IEEE, 2005.

- [TGRM14] Vernon Turner, John F Gantz, David Reinsel, and Stephen Minton. The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things. *IDC Analyze the Future*, 16, 2014.
- [TK73] Amos Tversky and Daniel Kahneman. Availability: A Heuristic for Judging Frequency and Probability. *Cognitive Psychology*, 5(2):207–232, 1973.
- [TK74] Amos Tversky and Daniel Kahneman. Judgment under Uncertainty: Heuristics and Biases. *Science*, 185(4157):1124–1131, 1974.
- [TK06] Leila Takayama and Eser Kandogan. Trust as an Underlying Factor of System Administrator Interface Choice. In *CHI’06 Extended Abstracts on Human Factors in Computing Systems*, pages 1391–1396. ACM, 2006.
- [TKBH17] Cagatay Turkay, Erdem Kaya, Selim Balcisoy, and Helwig Hauser. Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. *IEEE Trans. on Visualization and Computer Graphics*, 23(1):131–140, 2017.
- [TLKT09] Justin Talbot, Bongshin Lee, Ashish Kapoor, and Desney S Tan. EnsembleMatrix: Interactive Visualization to Support Machine Learning with Multiple Classifiers. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1283–1292. ACM, 2009.
- [TLS⁺14] Cagatay Turkay, Alexander Lex, Marc Streit, Hanspeter Pfister, and Helwig Hauser. Characterizing Cancer Subtypes using Dual Analysis in Caleydo StratomeX. *IEEE Computer Graphics and Applications*, 34(2):38–47, 2014.
- [TM05] Fan-Yin Tzeng and Kwan-Liu Ma. Opening the Black Box - Data Driven Visualization of Neural Networks. In *Proc. IEEE Visualization 2005*, pages 383–390. IEEE, 2005.
- [TPRH11a] Cagatay Turkay, Július Parulek, Nathalie Reuter, and Helwig Hauser. Integrating Cluster Formation and Cluster Evaluation in Interactive Visual Analysis. In *Proceedings of the 27th Spring Conference on Computer Graphics*, pages 77–86. ACM, 2011.
- [TPRH11b] Cagatay Turkay, Július Parulek, Nathalie Reuter, and Helwig Hauser. Interactive Visual Analysis of Temporal Cluster Structures. In *Computer Graphics Forum*, volume 30, pages 711–720. Wiley Online Library, 2011.
- [vdEvW11] Stef van den Elzen and Jarke J. van Wijk. BaobabView: Interactive Construction and Analysis of Decision Trees. In *Proc. of the IEEE Conf. on Visual Analytics Science and Technology (VAST 2011)*, pages 151–160, 2011.

- [VT03] Bruce Vanstone and Clarence Tan. A Survey of the Application of Soft Computing to Investment and Financial Trading. In *Proceedings of the Australian and New Zealand Intelligent Information Systems Conference*, pages 211–216, 12 2003.
- [vWvL93] Jarke van Wijk and Robert van Liere. HyperSlice: Visualization of Scalar Functions of Many Variables. In *Proc. of the 4th Conf. on Visualization*, pages 119–125, 1993.
- [War04] Colin Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.
- [WBP⁺17] Emily Wall, Leslie Blaha, Celeste Lyn Paul, Kristin Cook, and Alex Endert. Four Perspectives on Human Bias in Visual Analytics. In *DECISIVE: Workshop on Dealing with Cognitive Biases in Visualizations*, 2017.
- [WBWK00] Michelle Q Wang Baldonado, Allison Woodruff, and Allan Kuchinsky. Guidelines for Using Multiple Views in Information Visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM, 2000.
- [WCH15] Hadley Wickham, Dianne Cook, and Heike Hofmann. Visualizing Statistical Models: Removing the Blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(4):203–225, 2015.
- [WFH⁺01] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. Interactive Machine Learning: Letting Users Build Classifiers. *International Journal of Human-Computer Studies*, 55(3):281–292, 2001.
- [WHA07] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.
- [Wic18] Hadley Wickham. You can’t do data science in a GUI. <https://speakerdeck.com/hadley/you-cant-do-data-science-in-a-gui?slide=20>, last visited 2018-07-25, 2018.
- [Wil06] Leland Wilkinson. *The Grammar of Graphics*. Springer Science & Business Media, 2006.
- [WM04] Matt Williams and Tamara Munzner. Steerable, Progressive Multidimensional Scaling. In *IEEE Symp. on Information Visualization, 2004. INFOVIS 2004*, pages 57–64, 2004.
- [WMA⁺16] Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Trans. on Visualization and Computer Graphics*, 22(1):649–658, 2016.

- [WSW⁺18] Kanit Wongsuphasawat, Daniel Smilkov, James Wexler, Jimbo Wilson, Dandelion Mané, Doug Fritz, Dilip Krishnan, Fernanda B Viégas, and Martin Wattenberg. Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow. *IEEE Trans. on Visualization and Computer Graphics*, 24(1):1–12, 2018.
- [YKSJ07] Ji Soo Yi, Youn ah Kang, John T. Stasko, and Julie A. Jacko. Toward a Deeper Understanding of the Role of Interaction in Information Visualization. *IEEE Trans. on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.
- [ZBB⁺13] Xiang Zhao, Emery R Boose, Yuriy Brun, Barbara Staudt Lerner, and Leon J Osterweil. Supporting Undo and Redo in Scientific Data Analysis. *Evaluation*, 1:C2, 2013.
- [ZF14] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.
- [Zhu06] Xiaojin Zhu. Semi-Supervised Learning Literature Survey. *Computer Science, University of Wisconsin-Madison*, 2(3):4, 2006.
- [Zlo77] Moshe M. Zloof. Query-by-Example: A Data Base Language. *IBM systems Journal*, 16(4):324–343, 1977.
- [ZWRH14] Kaiyu Zhao, Matthew O Ward, Elke A Rundensteiner, and Huong N Higgins. LoVis: Local Pattern Visualization for Model Refinement. In *Computer Graphics Forum*, volume 33, pages 331–340. Wiley Online Library, 2014.

Thomas Mühlbacher

Visual Analytics Researcher



thomasmuehlbacher1987@gmail.com - Tel: +43 / 650 561 55 00 - Donau-City-Straße 12/1/83, 1220 Vienna

Born: July 8, 1987 in Korneuburg, AT - Austrian citizen - Married

Highlights of Qualifications

Competences

Visual Analytics	●●●●●
Information Visualization	●●●●●
Data Science and Analysis	●●●●●
Machine Learning	●●●●○
Analytical Software Architectures	●●●●●
Software Engineering	●●●●○
Human-Computer Interaction	●●●●○
Computer Graphics, Rendering	●●●●●
Video / Audio Production	●●●●○
Webdesign	●●●●○

Programming Skills

C, C++	●●●●●
Java, C#	●●●●○
Matlab, R, Python	●●●●○
HTML, PHP, SQL	●●●●○
CSS, JavaScript	●●●●○

Languages

German (native)	●●●●●
English (business fluent)	●●●●●
French (school knowledge)	●●○○○

Other skills

- Confident speaker and teacher
- Broad general knowledge
- Driving license

Education

2013 - present	Ongoing PhD studies in Computer Science Vienna University of Technology
2010 - 2013	Master's degree studies "Visual Computing", Vienna University of Technology. Graduated with honours, average grade: 1.0
2006 - 2010	Bachelor's degree studies in Media Informatics, Vienna University of Technology, Graduated with honours
2006	Obligatory Military Service of Austria Finished as guard commander
1997 - 2005	Secondary School "Sigmund Freud Gymnasium", 1020 Vienna Graduated with honours.
2003 - 2004	Software Developer C/C++ course. WIFI institute of Vienna

Work Experience

2012 - present	Researcher at VRVis Forschungs-GmbH, Visual Analysis Group, led by Dr. Harald Piringer. Donau-City-Straße 11, 1220 Vienna Tasks include research, software development and engineering, data analysis, supervision of students
2010 - 2011	Tutor at the Institute of Computer Graphics and Algorithms, Vienna University of Technology, for courses Computer Graphics I, InfoVis, and Visualization I and II. (part-time job, 9h per week)
2005 - 2009	Summer Internships (1 month per year) at WienIT, IT-service provider of the Viennese municipal utility company
2004	Summer Internship (1 month) at Bank Austria Creditanstalt

Awards and Prizes

2013	Best Paper Award at the IEEE VAST Conference with Harald Piringer, for the paper: "A Partition-Based Framework for Building and Validating Regression Models"
2008 - 2011	4x Performance Scholarship (Leistungsstipendium) awarded by the Vienna University of Technology for exceptional grades

Memberships / Volunteering

2014 - present	Peer-reviewer for IEEE VAST, InfoVis, EuroVis, Pacific Vis and IEEE TVCG.
2010 - 2013	Computer Graphics Club @Vienna University of Technology
2012	Student Volunteer at the 2012 EuroVis Conference
2007 - 2017	Co-founder, co-organizer and guitar player in the Metal band "Harmanic"
2017 - 2018	Guitar player in the Metal band "Project: Epigone"

List of Publications

Peer-Reviewed Journal Publications as First Author

1. MÜHLBACHER T. and PIRINGER H.:
"A Partition-Based Framework for Building and Validating Regression Models".
IEEE Transactions on Visualization and Computer Graphics, 19(12): 1962-1971, 2013. **Best paper award.**
2. 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 Transactions on Visualization and Computer Graphics, 20(12): 1643-1652, 2014
3. MÜHLBACHER T., LINHARDT L., MÖLLER T., and PIRINGER, H.:
"TreePOD: Sensitivity-Aware Selection of Pareto-Optimal Decision Trees"
IEEE Transactions on Visualization and Computer Graphics, 24(1): 174-183, 2018

Peer-Reviewed Journal Publications as Co-Author

4. LUKSCH C., TOBLER R.F., MÜHLBACHER T., SCHWÄRZLER M., and WIMMER, M.:
"Real-Time Rendering of Glossy Materials with Regular Sampling".
The Visual Computer, 30 (6-8), 717-727, 2014
5. ARBESSER C., SPECHTENHAUSER F., MÜHLBACHER T., and PIRINGER, H.:
"Visplause: Visual Data Quality Assessment of Many Time Series Using Plausibility Checks."
IEEE Transactions on Visualization and Computer Graphics, 23(1), 641-650, 2016

Conference / Workshop Publications

6. MÜHLBACHER T. and PIRINGER H.:
"Task-tailored Dashboards: Lessons Learned from Deploying a Visual Analytics System".
IEEE VIS Conference, Practitioner Experience Track, Poster, 2014
7. MÜHLBACHER T., ARBESSER C., and PIRINGER, H.:
"Statistical Forecasting in the Energy Sector: Task Analysis and Lessons Learned from Deploying a Dashboard Solution"
IEEE VIS Conference, Visualization in Practice Track, Short Paper with Talk, 2015
8. ARBESSER C., MÜHLBACHER T., KOMORNYIK S., and PIRINGER H.:
"Visual Analytics for Domain Experts: Challenges and Lessons Learned".
Proceedings of the Second International Symposium on Virtual Reality and Visual Computing, p.1-6, 2017

Theses

9. MÜHLBACHER T.,
"Real-Time Rendering of Measured Materials"
Master Thesis at the Vienna University of Technology, supervised by Dr. Michael Wimmer, 2012
10. ARBESSER C., and MÜHLBACHER T.,
"Interactive Lighting and Material Design"
Bachelor Thesis at the Vienna University of Technology, supervised by Dr. Paul Guerrero, 2009